



Πανεπιστήμιο Δυτικής Μακεδονίας

Τμήμα Μηχανικών Πληροφορικής & Τηλεπικοινωνιών

**ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ ΔΥΝΑΜΙΚΟΥ
ΙΣΤΟΧΩΡΟΥ ΕΞΕΤΑΣΗΣ
ΦΟΙΤΗΤΩΝ ΓΙΑ ΤΗΝ ΥΠΟΣΤΗΡΙΞΗ ΤΗΣ
ΑΣΥΓΧΡΟΝΗΣ
ΤΗΛΕΚΠΑΙΔΕΥΣΗΣ**

Βασιλούδης Παντελής

Επιβλέπων: Δασυγένης Μηνάς

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής
Υπολογιστών

Ιούνιος 2015, Κοζάνη

Περιεχόμενα

Περιεχόμενα	3
Κατάλογος σχημάτων	7
Κατάλογος πινάκων	8
Κατάλογος κωδίκων	8
Δήλωση Πνευματικών δικαιωμάτων	11
Ευχαριστίες	13
Περίληψη	15
Abstract	17
Διάρθρωση κειμένου	19
Κεφάλαιο 1	21
Εισαγωγή	21
1.1 Διαδικτυακή εξέταση φοιτητών	21
1.1.1 Σκοπός και πλεονεκτήματα	22
1.1.2 Δυναμικός ιστόχωρος	22
1.1.3 Αρχιτεκτονική δυναμικού ιστοχώρου	24
1.2 Εξυπηρετητής δικτύου	25
1.2.1 Apache HTTP Server	26
1.3 XHTML	26
1.4 CSS	28
1.4.1 Πλεονεκτήματα CSS έναντι μορφοποίησης μέσω HTML	28
1.5 PHP	29
1.5.1 Πλεονεκτήματα της PHP	30
1.5.2 Χρησιμότητα και πλεονεκτήματα της PHP 5	31
1.5.3 JavaScript	32
1.5.4 Δυναμικές ιστοσελίδες με PHP & MySQL	33
1.6 Βάση Δεδομένων	33
1.6.1 Σύστημα διαχείρισης βάσεων δεδομένων	34
1.7 MySQL	35
1.7.1 Πλεονεκτήματα της MySQL	36
1.7.2 PhpMyAdmin	37
1.8 Η επέκταση PDO και τα πλεονεκτήματά της	37
Κεφάλαιο 2	39
Ανάλυση και σχεδίαση	39
2.1 Ανάλυση του ιστοχώρου	39
2.2 Ρόλοι χρηστών του ιστοχώρου	40
2.3 Απαιτήσεις του ιστοχώρου	42
2.4 Περιπτώσεις χρήσης	44

2.4.1	Είσοδος στο σύστημα	44
2.4.2	Προσθήκη νέας ερώτησης	45
2.4.3	Προσθήκη νέου διαγωνίσματος	48
2.5	Βάση δεδομένων	52
2.5.1	Departments	56
2.5.2	Professors	57
2.5.3	Subjects	57
2.5.4	Categories	59
2.5.5	Topics	60
2.5.6	Blobtypes	61
2.5.7	Questions	62
2.5.8	Tests	66
2.5.9	Topicpercentage	70
2.5.10	Tokens	71
2.5.11	Testsequences	72
2.5.12	Qualityresults	76
2.5.13	Questionfeedback	77
Κεφάλαιο 3		80
Υλοποίηση του ιστοχώρου		80
3.1	Οδηγίες εγκατάστασης	80
3.2	Αρχική σελίδα	82
3.2.1	Index.php και login_check.php	83
3.2.2	Τεχνικές ασφάλειας	90
3.3	Σελίδες διαχείρισης – ονοματολογία	92
3.3.1	Admin.php και admin_session_check.php	93
3.3.2	Admin_sidebar.html και get_numof_ftext_mesges.php	96
3.2.3	Aprofessor.php και add_professor.php	101
3.2.4	Aquestion.php, aquestion_upload.php και download_csv.php	106
3.3	Σελίδες φοιτητών	112
3.3.1	Steste.php και add_answers.php	112
3.3.2	Timer_sidebar.html και get_time.php	125
Κεφάλαιο 4		130
Παρουσίαση του ιστοχώρου		130
4.1	Λειτουργίες διαχειριστή	130
4.1.1	Σύνδεση και αποσύνδεση διαχειριστή	130
4.1.2	Διαχείριση αντικειμένων	132
4.1.3	Διαχείριση τμημάτων	132
4.1.4	Διαχείριση καθηγητών	133
4.1.5	Διαχείριση μαθημάτων	134

4.1.6	Διαχείριση κατηγοριών	135
4.1.7	Διαχείριση ενοτήτων	135
4.1.8	Διαχείριση ερωτήσεων	136
4.1.9	Διαχείριση διαγωνισμάτων	138
4.1.10	Διαχείριση σειράς κωδικών	141
4.1.11	Διαχείριση αποτελεσμάτων	142
4.1.12	Επισκόπηση γραφημάτων	146
4.2	Λειτουργίες χρήστη	147
4.2.1	Ελεύθερη εξέταση	148
4.2.2	Σύνδεση και έναρξη νέας εξέτασης	149
4.2.3	Εξέταση με χρήση token	150
4.2.4	Βαθμοί διαγωνισμάτων	151
Κεφάλαιο 5		153
Επίλογος		153
5.1	Σύνοψη και συμπεράσματα	153
5.2	Μελλοντικές επεκτάσεις	155
5.2.1	Υποστήριξη πολλαπλών σωστών απαντήσεων	155
5.2.2	Δημιουργία εφαρμογής για Android	156
Βιβλιογραφία		157

Κατάλογος σχημάτων

Σχήμα 1. 1: Δυναμική ιστοσελίδα με PHP και MySQL	33
Σχήμα 1. 2: Επικοινωνία client – server – database	38
Σχήμα 1. 3: Προσχέδιο βάσης δεδομένων	54
Σχήμα 1. 4: Διάγραμμα οντοτήτων-συσχετίσεων	55
Σχήμα 1. 5: Αποτυχία σύνδεσης	131
Σχήμα 1. 6: Αποσύνδεση χρήστη	132
Σχήμα 1. 7: Βασικό μενού	132
Σχήμα 1. 8: Αποτυχία διαγραφής τμήματος	133
Σχήμα 1. 9: Προσθήκη νέου μαθήματος	135
Σχήμα 1. 10: Πίνακας διαθέσιμων ενοτήτων	136
Σχήμα 1. 11: Προσθήκη αρχείου CSV με λανθασμένα στοιχεία	138
Σχήμα 1. 12: Μετρήσεις ερωτήσεων	139
Σχήμα 1. 13: Επιτυχημένη επεξεργασία ερώτησης	140
Σχήμα 1. 14: Προεπισκόπηση διαγωνίσματος	141
Σχήμα 1. 15: Έλεγχος σειράς κωδικών	142
Σχήμα 1. 16: Επισκόπηση απαντήσεων εξέτασης	144
Σχήμα 1. 17: Ενεργός έλεγχος εξέτασης	145
Σχήμα 1. 18: Βαθμολόγηση απαντήσεων ελεύθερου κειμένου	145
Σχήμα 1. 19: Πίνακας διαθέσιμων μηνυμάτων	146
Σχήμα 1. 20: Γράφημα διαγωνίσματος	147
Σχήμα 1. 21: Ελεύθερη εξέταση	149
Σχήμα 1. 22: Ολοκλήρωση εξέτασης	150
Σχήμα 1. 23: Βαθμοί διαγωνισμάτων	152

Κατάλογος πινάκων

Πίνακας 1. 1: Πίνακας Departments	56
Πίνακας 1. 2: Πίνακας Professors	57
Πίνακας 1. 3: Πίνακας Subjects	58
Πίνακας 1. 4: Πίνακας Categories	59
Πίνακας 1. 5: Πίνακας Topics	60
Πίνακας 1. 6: Πίνακας Blobtypes	61
Πίνακας 1. 7: Πίνακας Questions	64
Πίνακας 1. 8: Πίνακας Tests	67
Πίνακας 1. 9: Πίνακας Topicpercentage	70
Πίνακας 1. 10: Πίνακας Tokens	71
Πίνακας 1. 11: Πίνακας Testsequences	73
Πίνακας 1. 12: Πίνακας Qualityresults	76
Πίνακας 1. 13: Πίνακας Questionfeedback	78

Κατάλογος κωδίκων

Κώδικας 1. 1: Index.php – αρχικοποίηση	83
Κώδικας 1. 2: Index.php – έλεγχος cookies	85
Κώδικας 1. 3: Index.php – δημιουργία φόρμας	86
Κώδικας 1. 4: Login_check.php – έλεγχος εισόδου	87
Κώδικας 1. 5: Connect.php	88
Κώδικας 1. 6: Login_check.php – δημιουργία cookies	89
Κώδικας 1. 7: Παράδειγμα χρήσης PDO	92
Κώδικας 1. 8: Admin.php – έλεγχος τιμών sessions	93
Κώδικας 1. 9: Admin_header.php	94
Κώδικας 1. 10: Logout.php	95
Κώδικας 1. 11: Admin.php – κυρίως σώμα	96
Κώδικας 1. 12: Admin_sidebar.html – διάκριση αντικειμένων διαχείρισης	97
Κώδικας 1. 13: Admin_sidebar.html – εμφάνιση αριθμού μηνυμάτων και απαντήσεων ελεύθερου κειμένου	98
Κώδικας 1. 14: Get_numof_ftext_mesges.php – αναζήτηση αποτελεσμάτων με αβαθμολόγητες απαντήσεις	98

Κώδικας 1. 15: Get_numof_ftext_mesges.php – καταμέτρηση αβαθμολόγητων απαντήσεων _____	99
Κώδικας 1. 16: Get_numof_ftext_mesges.php – εμφάνιση αποτελεσμάτων _____	100
Κώδικας 1. 17: Aprofessor.php – εμφάνιση μηνυμάτων _____	101
Κώδικας 1. 18: Aprofessor.php – αυτόματη εμφάνιση στοιχείων ____	103
Κώδικας 1. 19: Get_full_name.php _____	104
Κώδικας 1. 20: Add_professor.php – έλεγχος κενής εισόδου _____	105
Κώδικας 1. 21: Add_professor.php – έλεγχοι αποδεκτών στοιχείων _	106
Κώδικας 1. 22: Aquestion.php – φόρμα προσθήκης πολλαπλών ερωτήσεων _____	106
Κώδικας 1. 23: Aquestion_upload.php – άνοιγμα εισόδου _____	108
Κώδικας 1. 24: Aquestion_upload.php – μήνυμα αποτυχημένων εγγραφών _____	109
Κώδικας 1. 25: Aquestion.php – εμφάνιση προτύπου CSV _____	110
Κώδικας 1. 26: Download_csv.php – δημιουργία προτύπου CSV ____	111
Κώδικας 1. 27: Steste.php – υπολογισμός ερωτήσεων ανά ενότητα _	115
Κώδικας 1. 28: Steste.php – Επιλογή τυχαίων ερωτήσεων ανά ενότητα _____	116
Κώδικας 1. 29: Steste.php – ανακάτεμα επιλογών ερώτησης ____	118
Κώδικας 1. 30: Steste.php – κουμπιά επιλογής λειτουργίας _____	119
Κώδικας 1. 31: Add_answers.php – ανάλυση απάντησης πολλαπλής επιλογής _____	121
Κώδικας 1. 32: Steste.php – επιλογή εναπομείναντα ερωτήσεων με τυχαία σειρά _____	122
Κώδικας 1. 33: Add_answers.php – περίπτωση υπαρχόντων παραλειπόμενων ερωτήσεων _____	124
Κώδικας 1. 34: Steste.php – επιλογή παρελειπόμενων ερωτήσεων με τυχαία σειρά _____	124
Κώδικας 1. 35: Timer_sidebar.html – κάλεσμα αρχείου get_time.php και εμφάνιση αποτελεσμάτων _____	125
Κώδικας 1. 36: Timer_sidebar.html – μείωση και τέλος χρόνου _____	127
Κώδικας 1. 37: Get_time.php – εύρεση χρόνου λήξης _____	128
Κώδικας 1. 38: Get_time.php – υπολογισμός εναπομείναντα χρόνου	129

Δήλωση Πνευματικών δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και μόνο.

Copyright (C) Βασιλούδης Παντελής & Δασυγένης Μηνάς, 2015, Κοζάνη

Ευχαριστίες

Θέλω να ευχαριστήσω την οικογένεια μου για τη στήριξη που μου έδειξε όλα τα χρόνια των σπουδών μου, αλλά ιδιαίτερα κατά την εκπόνηση της πτυχιακής μου, καθώς και τον καθηγητή μου, Δασυγένη Μηνά, για την άμεση βοήθεια, την στήριξή του και την υπομονή που έδειξε κατά τη δημιουργία της παρούσας πτυχιακής εργασίας.

Επίσης θα ήθελα να ευχαριστήσω τους στενούς μου φίλους και συμφοιτητές για το κουράγιο που μου έδωσαν μέχρι να ολοκληρώσω αυτό το δύσκολο έργο.

Βασιλούδης Παντελής

Περίληψη

Στην εποχή της παγκοσμιοποίησης και του διαδικτύου, τα Πανεπιστημιακά Ιδρύματα πρέπει να αναδιαμορφώσουν τα προγράμματα σπουδών, ώστε να συμπεριλάβουν τεχνολογίες ασύγχρονης τηλεκπαίδευσης, προκειμένου να παραμείνουν ανταγωνιστικά και να προσελκύουν τους καλύτερους φοιτητές και διδάσκοντες. Μια τεχνολογία που πρέπει να προστεθεί στη φαρέτρα των προγραμμάτων υποδομής, αποτελεί η τηλε-εξέταση. Στην παρούσα διπλωματική, αναπτύχθηκε ένας πλήρης, ευέλικτος και ασφαλής ιστότοπος για τη διενέργεια τηλε-εξετάσεων.

Συγκεκριμένα στην παρούσα διπλωματική εργασία χρησιμοποιήθηκαν προγράμματα ελεύθερου λογισμικού για να κατασκευαστεί ένας δυναμικός ιστότοπος για την αυτοματοποιημένη εξέταση φοιτητών στα πλαίσια υποστήριξης της ασύγχρονης τηλεκπαίδευσης, υποστηριζόμενος από μια βάση δεδομένων RDBMS. Η γλώσσα ανάπτυξης της εφαρμογής είναι η δυναμική γλώσσα προγραμματισμού εφαρμογών διαδικτύου PHP και έγινε χρήση της επέκτασης PDO για την ασφαλή σύνδεση με τη βάση δεδομένων MySQL.

Αρχικά γίνεται μια εισαγωγή στο θέμα της εργασίας όπου παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την κατασκευή του ιστότοπου. Τα επόμενα βήματα είναι η παρουσίαση της αρχιτεκτονικής του δυναμικού ιστοχώρου εξέτασης φοιτητών και ο σχεδιασμός της βάσης δεδομένων. Στη συνέχεια πραγματοποιείται η υλοποίηση του συστήματος και μετέπειτα περιγράφονται όλες οι λειτουργίες του. Τέλος, γίνεται αναφορά στα συμπεράσματα που προέκυψαν και στις μελλοντικές επεκτάσεις του συστήματος.

Λέξεις κλειδιά: ιστότοπος, τηλεκπαίδευση, εξέταση φοιτητών, ελεύθερο λογισμικό, PHP, PDO, MySQL

Abstract

In the era of globalization and the Internet, the Universities must re-shape curricula, to include asynchronous e-learning technologies in order to remain competitive and attract the best students and teachers. A technology that should be added to the arsenal of infrastructure programs is tele-examination. In this thesis, a complete, flexible and secure website to perform online tests was developed.

Specifically in this thesis, open source software was used to construct a dynamic website for automated student examination in order to support asynchronous e-learning, supported by an RDBMS database. The application development language is the dynamic programming language for web applications, PHP, and the PDO extension is used for a safely connection to the MySQL database.

Initially, there is a brief introduction to the subject of the project in which the technologies used to build the website are presented. The next step is the presentation of the architecture of the dynamic student examination website and the design of the database. Then the system implementation takes place, and afterwards all of it's functions are described. Finally, a reference is made to the conclusions and the system's future extensions.

Keywords: website, e-learning, student examination, open source software, PHP, PDO, MySQL

Διάρθρωση κειμένου

Κεφάλαιο 1

Γίνεται μια εισαγωγή στο θέμα της διπλωματικής. Περιγράφεται η διαδικτυακή εξέταση φοιτητών και όλες οι τεχνολογίες που βοήθησαν να αναπτυχθεί η πλατφόρμα.

Κεφάλαιο 2

Το κεφάλαιο αυτό παρέχει πληροφορίες για τον σχεδιασμό του ιστότοπου, της βάσης δεδομένων και την μεταξύ τους επικοινωνία. Περιγράφονται οι απαιτήσεις για τον σχεδιασμό της βάσης και δίνεται λεπτομερής περιγραφή των οντοτήτων, δεδομένων και των συσχετίσεων που έχουν δημιουργηθεί.

Κεφάλαιο 3

Σε αυτό το κεφάλαιο γίνεται η υλοποίηση της διαδικτυακής πλατφόρμας εξέτασης φοιτητών. Αναλύεται ο προγραμματισμός των σελίδων της και οι τεχνικές που χρησιμοποιήθηκαν.

Κεφάλαιο 4

Στο κεφάλαιο αυτό παρουσιάζονται οι λειτουργίες του δυναμικού ιστοχώρου που αναπτύχθηκε και οι δυνατότητες που προσφέρει στον διαχειριστή και τον χρήστη.

Κεφάλαιο 5

Στο 5^ο και τελευταίο κεφάλαιο γίνεται η σύνοψη της εργασίας. Αναφέρονται τα συμπεράσματα που προέκυψαν από τη δημιουργία της διαδικτυακής πλατφόρμας, καθώς και οι μελλοντικές επεκτάσεις της.

Κεφάλαιο 1

Εισαγωγή

Σε αυτό το κεφάλαιο γίνεται η εισαγωγή στη διαδικτυακή πλατφόρμα εξέτασης φοιτητών. Περιγράφεται ο τρόπος δημιουργίας της και οι τεχνολογίες που χρησιμοποιούνται σε αυτή. Δηλαδή, η λειτουργία της γλώσσας προγραμματισμού εφαρμογών διαδικτύου PHP, η βάση δεδομένων MySQL και η μεταξύ τους σύνδεση με την επέκταση PDO. Έτσι, στο επόμενο κεφάλαιο ακολουθεί η ανάλυση και ο σχεδιασμός του ιστότοπου και της βάσης δεδομένων. Ας εξηγήσουμε λοιπόν αρχικά τι είναι το σύστημα που αναπτύχθηκε.

1.1 Διαδικτυακή εξέταση φοιτητών

Η διαδικτυακή εξέταση φοιτητών είναι η δυνατότητα που δίνεται στους φοιτητές των διάφορων πανεπιστημιακών ιδρυμάτων να εξεταστούν στα μαθήματά τους ηλεκτρονικά, δηλαδή κάνοντας χρήση ενός ηλεκτρονικού υπολογιστή και του Διαδικτύου. Παλιότερα, η εξέταση των φοιτητών γινόταν μόνο χειρόγραφα και με την παρουσία του διδάσκοντα και των φοιτητών στην τάξη την ίδια ώρα. Καθώς όμως τα τμήματα εξέλιξαν τους τρόπους διδασκαλίας τους και πρόσθεσαν την τηλεεκπαίδευση, μια από τις πρωτοποριακές εφαρμογές που αναπτύχθηκαν με τη χρήση της ασύγχρονης επικοινωνίας μεταξύ των χρηστών είναι η πλατφόρμα ηλεκτρονικής εξέτασης.

Πρόκειται για ένα σύστημα διαχείρισης εξέτασης που παρέχει ερωτήσεις τύπου πολλαπλής επιλογής κατά κύριο λόγο, αλλά και ελεύθερου κειμένου. Το σύστημα αυτό έχει τη δυνατότητα να αποθηκεύει όλες τις απαντήσεις των ερωτήσεων και να παρέχει στο χρήστη-φοιτητή πληροφορίες σχετικά με την επίδοσή του και το χρόνο στον οποίο ολοκλήρωσε το τεστ. Είναι δε ασύγχρονο, διότι κάθε φοιτητής μπορεί να ξεκινήσει μια εξέταση σε διαφορετικό χρόνο από τους υπόλοιπους, έχοντας όμως στη διάθεσή του το ίδιο χρονικό περιθώριο για την υποβολή του διαγωνίσματος. Αυτό που το καθιστά πρωτοποριακό είναι η δυνατότητα ανώνυμης εξέτασης μέσω της έκδοσης ενός ειδικού ανώνυμου αναγνωριστικού (token) που μοιράζεται στους φοιτητές, αλλά και η δυνατότητα προβολής

γραφημάτων που αφορούν τα αποτελέσματα. Ουσιαστικά δημιουργήθηκε για την βελτιστοποίηση της εκμάθησης των φοιτητών και την αποδοτικότερη αξιολόγησή τους. Για να έχει νόημα όλο αυτό, πρέπει να καλύπτει κάποιο σκοπό και πλεονεκτήματα, που θα αναλύσουμε αμέσως τώρα.

1.1.1 Σκοπός και πλεονεκτήματα

Ο σκοπός, δηλαδή, της διαδικτυακής εξέτασης φοιτητών είναι να αξιολογηθούν όλες οι μαθητικές ικανότητες, ώστε να βελτιστοποιηθεί η διδασκαλία των καθηγητών. Τα οφέλη είναι πολλαπλά και αφορούν τόσο τους φοιτητές όσο και το διδάσκοντα. Αφ' ενός οι φοιτητές αφιερώνουν περισσότερο χρόνο στη μελέτη με αποτέλεσμα την καλύτερη αφομοίωση της ύλης. Αφ' ετέρου, η εφαρμογή του συστήματος των ερωτήσεων πολλαπλής επιλογής επιτρέπει την ευρύτερη κάλυψη της διδακτέας ύλης του μαθήματος και παρέχει πολύτιμα συμπεράσματα στο διδάσκοντα. Η κατάλληλη επεξεργασία της επιμέρους βαθμοδότησης των ερωτήσεων, αναδεικνύει τις αδυναμίες στην κατανόηση των διδακτικών ενοτήτων. Ο διδάσκων προσαρμόζει τις διαλέξεις, δίνοντας έμφαση στα σημεία στα οποία η εξέταση δείχνει χαμηλά ποσοστά επιτυχίας. Όλα αυτά γίνονται εφικτά μέσω ενός δυναμικού ιστοχώρου.

1.1.2 Δυναμικός ιστόχωρος

Ιστόχωρο ονομάζουμε ένα σύνολο από ενότητες διαδικτυακών σελίδων που συνήθως συστεγάζονται στον ίδιο διακομιστή-εξυπηρετητή (server). Οι σελίδες ενός ιστοτόπου εμφανίζονται κάτω από το ίδιο όνομα χώρου (domain) π.χ. iexams.uowm.gr. Οι ιστοσελίδες αλληλοσυνδέονται και μπορεί ο χρήστης να μεταβεί από τη μία στην άλλη κάνοντας «κλικ», επιλέγοντας δηλαδή συνδέσμους που υπάρχουν στο κείμενο ή στις φωτογραφίες της ιστοσελίδας. Οι σύνδεσμοι προς άλλες σελίδες εμφανίζονται συνήθως υπογραμμισμένοι και με μπλε χρώμα για να είναι γρήγορα ξεκάθαρο στον επισκέπτη ότι πρόκειται για σύνδεσμο προς άλλη ιστοσελίδα, χωρίς όμως πάντα να είναι αυτό απαραίτητο.

Δυναμικές ιστοσελίδες [1] είναι αυτές που προσφέρουν αλληλεπιδραστικά στοιχεία ανάμεσα στον χρήστη και την ίδια την σελίδα. Πίσω από τους δυναμικούς ιστότοπους υπάρχουν βάσεις

δεδομένων που συνεχώς αποθηκεύουν και επεξεργάζονται πληροφορίες. Οι βάσεις δεδομένων και η ικανότητα αλληλεπίδρασης και επικοινωνίας μεταξύ διακομιστή και βάσεων δεδομένων είναι βασικό και απαραίτητο στοιχείο για την ανάπτυξη μιας δυναμικής ιστοσελίδας. Για την δημιουργία ενός δυναμικού ιστοχώρου χρειάζονται πολλές γραμμές κώδικα προγραμματισμού. Στην πραγματικότητα παρόλο που με μια πρώτη ματιά οι δυναμικές ιστοσελίδες μοιάζουν σε εμφάνιση με τις στατικές, υπάρχει τεράστια διάφορα στην κατασκευή τους και τον τρόπο λειτουργίας τους.

Ουσιαστικά, μια δυναμική ιστοσελίδα είναι μια εφαρμογή, ένα πρόγραμμα που κατασκευάζει αυτό που βλέπει ο επισκέπτης της ιστοσελίδας τη στιγμή της επίσκεψης. Δηλαδή η εμφάνιση και οι πληροφορίες που επιστρέφει η δυναμική ιστοσελίδα στον επισκέπτη της είναι συνδυασμός πληροφοριών που δομούνται και τακτοποιούνται σύμφωνα με τον αλγόριθμο που έχει δημιουργήσει ο προγραμματιστής για την δυναμική σελίδα. Αυτές οι πληροφορίες αντλούνται είτε από κάποια βάση δεδομένων, είτε από κάποια άλλη ιστοσελίδα, είτε από τον χρήστη-επισκέπτη της ιστοσελίδας, είτε από άλλες πηγές. Όλα αυτά γίνονται την στιγμή της επίσκεψης στη σελίδα. Συνεπώς διαφορετικοί επισκέπτες, διαφορετικές χρονικές στιγμές, διαφορετικά συμβάντα προκαλούν την εμφάνιση διαφορετικών αποτελεσμάτων στην ιστοσελίδα. Η χρήση των βάσεων δεδομένων, προσθέτει ένα από τα μεγάλα πλεονεκτήματα των ιστοσελίδων αυτών. Επιτρέπει την εύκολη προσθαφαίρεση και αλλαγή περιεχομένου, καθώς δεν απαιτείται να επεξεργάζεται κανείς κάθε φορά το πηγαίο αρχείο της ιστοσελίδας, αλλά απλά να διαχειρίζεται έμμεσα το περιεχόμενο στην βάση δεδομένων (συνήθως μέσω φορμών) και οι υπόλοιπες διαδικασίες γίνονται αυτοματοποιημένα από τον "μηχανισμό" της ιστοσελίδας. Έτσι μπορεί να υπάρχει μια συνεχής ανανέωση από κάποιο διαχειριστή που δεν έχει γνώσεις προγραμματισμού. Για την εύκολη διαχείριση του περιεχομένου σε μία δυναμική ιστοσελίδα, υπάρχει πρόσθετα στην ιστοσελίδα ένας εύχρηστος μηχανισμός CMS (Content Management System), μέσω του οποίου γίνεται προσθαφαίρεση περιεχομένου. Ουσιαστικά, ο δημιουργός του ιστότοπου γράφει ένα μέρος κώδικα με τον οποίο δημιουργείται ένα φιλικό περιβάλλον για αλλαγές στις βάσεις δεδομένων χωρίς να απαιτείται η γνώση προγραμματισμού. Φυσικά, το περιβάλλον αυτό, η "περιοχή διαχείρισης" της ιστοσελίδας, προστατεύεται με κωδικό πρόσβασης και δεν μπορούν να εισέλθουν σε αυτή οι απλοί επισκέπτες και γενικότερα όσοι δεν έχουν δικαιώματα

διαχειριστή. Όλα αυτά δεν θα ήταν δυνατό να συμβούν, αν δεν υπήρχαν οι κατάλληλες τεχνολογίες για να υποστηρίξουν το εγχείρημα.

1.1.3 Αρχιτεκτονική δυναμικού ιστοχώρου

Για την βασική λειτουργία μιας τέτοιας πλατφόρμας, απαιτείται από μεριάς υλικού ένας διακομιστής- εξυπηρετητής ιστοσελίδων (web server) στον οποίο φιλοξενείται ο ιστότοπος. Στην παρούσα εργασία χρησιμοποιήθηκε ένας διακομιστής που δουλεύει σε λειτουργικό σύστημα FreeBSD. Κατά κύρια βάση όμως ο ιστότοπος αναπτύχθηκε και δοκιμάστηκε τοπικά σε υπολογιστή με Windows XP μέσω του πακέτου προγραμμάτων ελεύθερου λογισμικού XAMPP (έκδοση 1.7.3), το οποίο περιέχει τον εξυπηρετητή ιστοσελίδων http Apache, τη βάση δεδομένων MySQL, μέσω του εργαλείου διαχείρισης PhpMyAdmin και ένα διερμηνέα για κώδικα γραμμένο σε γλώσσες προγραμματισμού PHP και Perl. Όμως δε υπάρχει λόγος να σταθεί κανείς στο λειτουργικό σύστημα γιατί όπως θα φανεί στη συνέχεια αυτό δεν έχει κάποιο ρόλο και δε διαφοροποιεί σε τίποτα την συγγραφή του κώδικα, την κατασκευή και λειτουργία ενός ιστοχώρου που έχει κατασκευαστεί με PHP και MySQL και η οποία τρέχει σε διακομιστή βασιζόμενο στον Apache.

Από μεριάς λογισμικού ακόμη, είναι απαραίτητος ένας φυλλομετρητής ιστοσελίδων ή περιηγητής Ιστού (Web browser) [2]. Αυτός είναι ένα λογισμικό που επιτρέπει στον χρήστη του να προβάλλει, και να αλληλεπιδρά με, κείμενα, εικόνες, βίντεο, μουσική, παιχνίδια και άλλες πληροφορίες συνήθως αναρτημένες σε μια ιστοσελίδα ενός ιστότοπου στον Παγκόσμιο Ιστό ή σε ένα τοπικό δίκτυο. Το κείμενο και οι εικόνες σε μια ιστοσελίδα μπορεί να περιέχουν υπερσυνδέσμους προς άλλες ιστοσελίδες του ίδιου ή διαφορετικού ιστότοπου. Ο περιηγητής Ιστού επιτρέπει στον χρήστη την γρήγορη και εύκολη πρόσβαση σε πληροφορίες που βρίσκονται σε διάφορες ιστοσελίδες και ιστότοπους εναλλάσσοντας τις ιστοσελίδες μέσω των υπερσυνδέσμων. Οι φυλλομετρητές χρησιμοποιούν τη γλώσσα μορφοποίησης HTML για την προβολή των ιστοσελίδων, για αυτό η εμφάνιση μιας ιστοσελίδας μπορεί να διαφέρει ανάλογα με τον περιηγητή Ιστού. Για να είναι λειτουργικός ο ιστότοπος, ο τρόπος υλοποίησης του πρέπει να είναι τέτοιος ώστε να υποστηρίζεται από οποιοδήποτε περιηγητή Ιστού. Κατά την ανάπτυξη της παρούσας εργασίας έγινε χρήση του Chrome (έκδοση 39.0), αλλά ο ιστόχωρος

ηλεκτρονικής εξέτασης δοκιμάστηκε και στον Internet Explorer (έκδοση 8.0) και τον Mozilla Firefox (έκδοση 27.0.1).

Με τον διακομιστή και τον φυλλομετρητή ιστοσελίδων έτοιμους για χρήση, το επόμενο βήμα είναι η δημιουργία του ιστότοπου. Το γράψιμο του κώδικα των ιστοσελίδων έγινε με τη βοήθεια του Notepad++. Πρόκειται για τον πιο γνωστό επεξεργαστή κειμένου (text editor). Είναι το αγαπημένο εργαλείο συγγραφής κώδικα των προγραμματιστών, καθώς αποτελεί ένα ελαφρύ πρόγραμμα το οποίο όμως έχει πάρα πολλές δυνατότητες. Η γλώσσα σήμανσης που χρησιμοποιήθηκε για την ανάπτυξη του ιστοχώρου είναι η XHTML 1.0 Transitional και για τον έλεγχο εμφάνισης των εγγράφων του έγινε χρήση των φύλλων στυλ CSS. Όπως ειπώθηκε παραπάνω, η πλατφόρμα διαδικτυακής εξέτασης φοιτητών αναπτύχθηκε στη γλώσσα προγραμματισμού PHP (έκδοση 5.3.1) με τη βοήθεια της βάσης δεδομένων MySQL (έκδοση 5.1.41) και την μεταξύ τους σύνδεση με την επέκταση PDO. Παρακάτω θα εξηγήσουμε τι σημαίνει και πως λειτουργεί το καθένα απ' αυτά, καθώς και γιατί χρησιμοποιήθηκε η κάθε έκδοση.

1.2 Εξυπηρετητής δικτύου

Η διαφορετικά διακομιστής είναι το λογισμικό που τρέχει σε έναν διαδικτυακό κόμβο και επιτρέπει σε άλλους υπολογιστές να αποκτούν αντίγραφα των ιστοσελίδων που είναι αποθηκευμένες σε αυτόν. Πολύ δημοφιλείς εξυπηρετητές δικτύου είναι ο Apache και ο Microsoft Internet Information Server .

Οι διακομιστές είναι ηλεκτρονικοί υπολογιστές στο διαδίκτυο που φιλοξενούν ιστοσελίδες, προσφέροντας σελίδες όταν ζητηθούν από κάποιον επισκέπτη. Κάθε διακομιστής έχει μια μοναδική διεύθυνση IP (Internet Protocol) ώστε οι άλλοι υπολογιστές στο διαδίκτυο να μπορούν να επικοινωνούν μαζί του. Όποιος θέλει να ανεβάσει την ιστοσελίδα του στο διαδίκτυο πρέπει να νοικιάσει ένα χώρο σε κάποιο διακομιστή ο οποίος στη συνέχεια θα φιλοξενεί τη σελίδα. Ο εξυπηρετητής δίνει μια μοναδική διεύθυνση σε κάθε ιστοσελίδα που φιλοξενεί. Αν κάποιος συνδεθεί στο διαδίκτυο, ο υπολογιστής του παίρνει μια μοναδική IP διεύθυνση (από τον πάροχο του). Αυτή η διεύθυνση χαρακτηρίζει πλέον τον συγκεκριμένο υπολογιστή στο δίκτυο. Όταν λοιπόν από αυτόν τον υπολογιστή ζητηθεί μια ιστοσελίδα ο περιηγητής Ιστού στέλνει μια αίτηση στη συγκεκριμένη ιστοσελίδα με την IP διεύθυνση του υπολογιστή που έκανε την αίτηση. Για να φτάσει η

αίτηση στο προορισμό της περνάει από ενδιάμεσους σταθμούς. Αφού φτάσει η αίτηση στον προορισμό ο εξυπηρετητής που φιλοξενεί την ιστοσελίδα την επιστρέφει σε μορφή HTML με προορισμό την IP του αποστολέα της αίτησης. Η σελίδα φτάνει στον περιηγητή που την ζήτησε (περνώντας από ενδιάμεσους σταθμούς ξανά) και αυτός την εμφανίζει σε γραφική μορφή στην οθόνη.

1.2.1 Apache HTTP Server

Ο Apache [3] είναι ένας HTTP server ανοικτού κώδικα που μπορεί να εγκατασταθεί σε περιβάλλοντα διάφορων σύγχρονων λειτουργικών συστημάτων. Αναπτύχθηκε και συντηρείται από μια ομάδα εθελοντών που ήθελαν να υλοποιήσουν έναν εύρωστο κώδικα για διακομιστή δικτύου που να μην είναι εμπορικός αλλά να υποστηρίζει πολλά χαρακτηριστικά. Όπως έχει αποδειχθεί είναι ο πιο γρήγορος, σταθερός, ασφαλής και υποστηρίζει τα περισσότερα χαρακτηριστικά από κάθε άλλο διακομιστή δικτύου. Ενδεικτικά λειτουργικά συστήματα στα οποία μπορεί να εγκατασταθεί είναι διάφορες εκδόσεις των Windows, το Linux, το Unix, και το Mac OS X. Ο Apache είναι εγκατεστημένος στο 80% των διακομιστών παγκοσμίως (πάνω από 6 εκατομμύρια). Επιπλέον, είναι παραμετροποιήσιμος σε μεγάλο βαθμό. Βέβαια η σωστή εγκατάσταση του και η παραμετροποίηση του δεν είναι μια εύκολη διαδικασία. Έτσι ένας σχετικά αρχάριος χρήστης υπολογιστών πολύ δύσκολα θα καταφέρει μόνος του να στήσει έναν διακομιστή. Ακόμα και μετά την αρχική εγκατάσταση του, απαιτείται επέμβαση σε κάποια αρχεία ώστε να ρυθμιστούν κάποιες παράμετροι για τη σωστή, σταθερή και αποδοτική λειτουργία του. Βεβαίως σε καμιά περίπτωση δε σημαίνει ότι πρέπει κάποιος να αποθαρρύνεται στη χρήση του Apache λόγω απαιτήσεων στην εγκατάσταση. Εξάλλου οποιοδήποτε λογισμικό για στήσιμο ενός διακομιστή δεν είναι υπόθεση ενός νέου και αρχάριου χρήστη υπολογιστών.

1.3 XHTML

Η XHTML (EXtensible HyperText Markup Language) [4] όπως ορίζει το όνομά της είναι μια εκτεταμένη γλώσσα σήμανσης υπερκειμένου. Ως γλώσσα σήμανσης (markup language) ορίζεται ένα σύνολο εντολών που επιτρέπουν τη προσθήκη πληροφορίας στο περιεχόμενο μιας πηγής

δεδομένων. Το περιεχόμενο μπορεί να είναι κείμενο, εικόνα ή οποιαδήποτε άλλη μορφή ηλεκτρονικής πληροφορίας. Συγκεκριμένα η XHTML είναι μια πιο αυστηρή και πιο καθαρή έκδοση της HTML (HyperText Markup Language) και μοιάζει πολύ με την HTML 4.01, η οποία είναι η τελευταία αναθεώρηση του προτύπου HTML το οποίο διαχειρίζεται ο συνεταιρισμός W3 (World Wide Web Consortium). Ο ίδιος οργανισμός ανέπτυξε και διαχειρίζεται το πρότυπο της XHTML. Η τελευταία αναθεώρηση της XHTML είναι η XHTML 1.0 και ελευθερώθηκε στην αγορά στις 26 Ιανουαρίου 2000. Σήμερα όλοι οι περιηγητές Ιστού είναι συμβατοί και μπορούν να "διαβάσουν" XHTML κώδικα, ενώ όλο και περισσότερες ιστοσελίδες χρησιμοποιούν το πρότυπο της XHTML αντί αυτό της HTML. Η XHTML είναι επί της ουσίας συνδυασμός της γλώσσας σήμανσης HTML και της μεταγλώσσας XML. Πιο συγκεκριμένα είναι μια παραλλαγή της HTML γραμμένη με κανόνες σύνταξης της XML, δηλαδή ο κώδικας ενός XHTML αρχείου μπορεί να περιέχει σχεδόν όλες τις ετικέτες που περιέχει και η HTML, με διαφορά την αυστηρότερη σύνταξη του κώδικα. Η XML σχεδιάστηκε για να περιγράφει δεδομένα, ενώ η HTML σχεδιάστηκε για να εμφανίζει δεδομένα. Η W3C ανέπτυξε την XHTML συνδυάζοντας την αυστηρότητα της XML με τις δυνατότητες της HTML. Η XHTML λόγω της αυστηρότητας στην σύνταξή της, μπορεί να "τρέξει" χωρίς να χρειάζεται μεγάλη επεξεργαστική ισχύ από τα υπολογιστικά συστήματα. Μερικοί συντακτικοί κανόνες της XHTML [5] είναι:

- Τα έγγραφα πρέπει να είναι ορθά διατυπωμένα.
- Τα στοιχεία πρέπει να είναι σωστά φωλιασμένα το ένα μέσα στο άλλο.
- Για την XML τα πεζά και κεφαλαία γράμματα είναι διαφοροποιημένα.
- Οι ετικέτες λήξης είναι υποχρεωτικές.
- Τα κενά στοιχεία δηλώνονται με μια ανάποδη πλάγια κάθετο.
- Οι τιμές των ιδιοτήτων πρέπει να κλείνονται σε εισαγωγικά.

Επειδή η XHTML είναι μια εφαρμογή της XML, τα φύλλα φυλλομετρία είναι υποχρεωτικά για να προσδιοριστεί η παρουσίαση της πληροφορίας ενός XHTML εγγράφου.

Η χρήση της παραλλαγής XHTML 1.0 Transitional έγινε για να επωφεληθούμε από τα χαρακτηριστικά XHTML, συμπεριλαμβανομένων του CSS, αλλά με μικρές προσαρμογές στη σήμανση προς όφελος όσων

προβάλουν σελίδες με παλαιότερα προγράμματα περιήγησης που δεν μπορούν να "καταλάβουν" κώδικα CSS. Αυτό σημαίνει ότι τα "παρουσιαστικά" στοιχεία και τα χαρακτηριστικά που έχουν καταργηθεί επιτρέπονται στη σήμανση.

1.4 CSS

Η CSS (Cascading Style Sheets-Διαδοχικά Φύλλα Στυλ) [6] είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στιλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη. Με την χρήση CSS μπορούμε να ορίζουμε χρώματα και μεγέθη οργανωμένα σε στυλ και έπειτα να εφαρμόζουμε τα στυλ αυτά στα στοιχεία των σελίδων του ιστοχώρου μας. Με αυτόν τον τρόπο, κάθε φορά που αλλάζουμε το χρώμα ενός στυλ, αλλάζει το χρώμα όλων των στοιχείων που έχουν αναφορά στο στυλ αυτό. Έτσι αν έχουμε ορίσει ένα στυλ για το κύριο μενού του ιστοχώρου, τότε θα χρειάζεται να αλλάξουμε το χρώμα του στυλ αυτού και αυτόματα θα εφαρμοστεί σε όλες τις σελίδες.

1.4.1 Πλεονεκτήματα CSS έναντι μορφοποίησης μέσω HTML

Εκτός από την ευκολία στην διαχείριση ενός ιστότοπου, ένα άλλο σημαντικό πλεονέκτημα της χρήσης CSS στις σελίδες είναι ο "καθαρότερος" κώδικας, χωρίς πολλές ιδιότητες στις ετικέτες οι οποίες τον κάνουν δυσανάγνωστο. Επιπλέον κάνει γρηγορότερη την πλοήγηση καθώς το αρχείο, μέσα στο οποίο ορίζονται τα στυλ, "διαβάζεται" από τον φυλλομετρητή ιστοσελίδων μόνο μια φορά και έπειτα αποθηκεύεται στην μνήμη cache, μειώνοντας έτσι το μέγεθος της πληροφορίας που λαμβάνεται από τους φυλλομετρητές.

Τα πλεονεκτήματα της χρήσης CSS έναντι της χρήσης ιδιοτήτων HTML είναι τα εξής:

- Πολύ μεγαλύτερη ευελιξία. Το CSS κατέστησε εφικτές μορφοποιήσεις οι οποίες ήταν αδύνατες ή πολύ δύσκολες με την κλασική HTML.
- Ευκολότερη συντήρηση των ιστοσελίδων. Η εμφάνιση ενός ολόκληρου ιστοχώρου μπορεί να ελέγχεται από ένα μόνο εξωτερικό αρχείο CSS. Έτσι, κάθε αλλαγή στο στυλ της ιστοσελίδας μπορεί να γίνεται με μία μοναδική αλλαγή σε αυτό το αρχείο, αντί για την επεξεργασία πολλών σημείων σε κάθε σελίδα που υπάρχει στον ιστότοπο.
- Μικρότερο μέγεθος αρχείου, δεδομένου ότι ο κάθε κανόνας μορφοποίησης γράφεται μόνο μια φορά και όχι σε κάθε σημείο που εφαρμόζεται.
- Καλύτερο SEO (Search Engine Optimization-Βελτιστοποίηση Μηχανών Αναζήτησης). Οι μηχανές αναζήτησης δεν "μπερδεύονται" ανάμεσα σε περιεχόμενο και τη μορφοποίησή του, αλλά έχουν πρόσβαση στο περιεχόμενο σκέτο, οπότε είναι πολύ ευκολότερο να το καταγράψουν και να το αρχειοθετήσουν (indexing).

1.5 PHP

Η PHP [7] είναι μια γλώσσα προγραμματισμού για τη δημιουργία ιστοσελίδων με δυναμικό περιεχόμενο, δηλαδή περιεχόμενο που αλλάζει αυτόματα ανάλογα με τα στοιχεία του θεατή της. Τα αρχικά αντιπροσωπεύουν το "PHP: Hypertext Preprocessor". Η PHP χρησιμοποιείται όχι για την διακόσμηση μιας ιστοσελίδας αλλά για τον χειρισμό των λειτουργιών και εργασιών που διεκπεραιώνει. Συνεπώς, ο κώδικας που γράφεται για μια ιστοσελίδα σε γλώσσα PHP δεν γίνεται άμεσα αντιληπτός αλλά μετά από την επέμβαση του θεατή στην ιστοσελίδα. Μια σελίδα PHP περνά από επεξεργασία από ένα συμβατό διακομιστή του Παγκόσμιου Ιστού (π.χ. Apache), ώστε να παραχθεί σε πραγματικό χρόνο το τελικό περιεχόμενο, που θα σταλεί στο πρόγραμμα περιήγησης των επισκεπτών σε μορφή κώδικα HTML.

Αυτό που ξεχωρίζει την PHP από μια γλώσσα όπως η JavaScript [8], η οποία εκτελείται στην πλευρά του χρήστη (client-side), είναι ότι ο κώδικάς της εκτελείται στον διακομιστή. Ίσως το δυνατότερο και πιο σημαντικό χαρακτηριστικό της PHP είναι η υποστήριξη που παρέχει σε μια ευρεία γκάμα από βάσεις δεδομένων. Έτσι, το να δημιουργήσουμε μια ιστοσελίδα που να παρέχει υποστήριξη σε βάσεις δεδομένων είναι απίστευτα απλό. Η σύνταξη της PHP μοιάζει πολύ με τη σύνταξη της γλώσσας προγραμματισμού Perl και της C. Συνήθως χρησιμοποιείται με εξυπηρετητή Apache σε διάφορα λειτουργικά συστήματα. Παράλληλα υποστηρίζει και την διεπαφή Internet Server Application Programming Interface (ISAPI) καθιστώντας τη δυνατή να χρησιμοποιηθεί και σε Microsoft Windows με εξυπηρετητή Microsoft Internet Information Server.

Η γλώσσα PHP είναι ανοικτό-ελεύθερο λογισμικό και μπορεί να εγκατασταθεί σχεδόν σε όλα τα λειτουργικά συστήματα όπως Windows, Linux, Mac OS X, Risc OS κλπ αλλά υποστηρίζεται και από τους περισσότερους εξυπηρετητές ιστοσελίδων όπως ο Apache, που χρησιμοποιήθηκε στη παρούσα εργασία, ή ο IIS. Η PHP μπορεί να λειτουργήσει είτε ως εγκατεστημένη μονάδα (module) στον εξυπηρετητή ιστοσελίδων είτε μέσω ενός επεξεργαστή CGI (Common Gateway Interface) σεναρίων. Μπορεί να χρησιμοποιηθεί για εκτέλεση σεναρίων (scripts) από την πλευρά του απομακρυσμένου εξυπηρετητή ιστοσελίδων όπως γίνεται και με τα σενάρια CGI. Επίσης η PHP μπορεί να χρησιμοποιηθεί για είσοδο/έξοδο δεδομένων από τον χρήστη.

1.5.1 Πλεονεκτήματα της PHP

Ποιά είναι τα πλεονεκτήματα που έκαναν την PHP τόσο διάσημη και χρησιμοποιούμενη;

- Ταχύτητα.
- Σταθερότητα.
- Ασφάλεια.
- Απλότητα.

Παρακάτω περιγράφεται συνοπτικά το καθένα από αυτά.

Όχι μόνο η PHP παρουσιάζει μεγάλη ταχύτητα εκτέλεσης, αλλά απαιτεί και από τον υπολογιστή πολύ λιγότερους πόρους σε σχέση με τις άλλες scripting γλώσσες. Έτσι λοιπόν μιλάμε για μία γλώσσα πολύ γρήγορη στην εκτέλεσή της που απαιτεί πολύ λίγους πόρους και η οποία μπορεί και συνεργάζεται άψογα με άλλα προγράμματα και

λειτουργικά συστήματα. Επιπλέον η PHP αποτελεί ένα κάλυμμα μεταξύ πολλών system calls πράγμα που την κάνει πάρα πολύ γρήγορη.

Δεν θα ήταν καλό για μία scripting γλώσσα το να είναι γρήγορη αλλά το σύστημα να καταρρέει κάθε μερικές εκατοντάδες σελίδες. Καμία εφαρμογή δεν είναι αλάνθαστη, αλλά με μία τόσο μεγάλη κοινωνία αναλυτών της PHP κανένα λάθος δεν θα μπορούσε να αντέξει στο χρόνο. Η PHP χρησιμοποιώντας ένα δικό της σύστημα διαχείρισης πόρων και σε συνδυασμό με το ότι διαθέτει έναν εξεζητημένο τρόπο διαχείρισης των μεταβλητών αποτελεί ένα πολύ δυνατό σύστημα.

Το σύστημα θα πρέπει να προστατεύεται από κακόβουλους hackers και επιθέσεις από χρήστες είτε μιλάμε για προγραμματιστές είτε για άτομα που απλώς "σερφάρουν". Η PHP παρέχει πολλά επίπεδα ασφάλειας τα οποία μπορούν να ενεργοποιηθούν κατά περίπτωση.

Εξαιτίας της απλότητάς της η PHP δίνει τη δυνατότητα σε νέους προγραμματιστές να φτάσουν σε επιθυμητό επίπεδο παραγωγικότητας πάρα πολύ γρήγορα. Όσον αφορά προγραμματιστές με γνώσεις C ή Javascript αυτοί φτάνουν σε επιθυμητό επίπεδο σχεδόν αμέσως. Επίσης, λόγω του συστήματος υπομονάδων που διαθέτει μας δίνει πολύ εύκολα την δυνατότητα να χρησιμοποιήσουμε πολύ απλά όποιες βιβλιοθήκες εμείς θέλουμε από διάφορες περιοχές.

1.5.2 Χρησιμότητα και πλεονεκτήματα της PHP 5

Η PHP 5 είναι η πιο πρόσφατη έκδοση της PHP και είναι ενισχυμένη να κάνει την γλώσσα πιο ισχυρή, ασφαλή, πιο γρήγορη και πιο εύκολη στη συγγραφή. Υπάρχουν πολλά πλεονεκτήματα σε σύγκριση με τη νεότερη έκδοση. Η PHP 5 διαθέτει εντυπωσιακά χαρακτηριστικά που βελτιώνουν τη διατηρησιμότητα των εφαρμογών. Αυτά τα χαρακτηριστικά περιλαμβάνουν το νέο μοντέλο αντικειμένων που υποστηρίζει λειτουργίες όπως αφηρημένες τάξεις, διασυνδέσεις, και ενθυλάκωση. Με αυτά τα χαρακτηριστικά, είναι ευκολότερο να εφαρμοστεί η αρχιτεκτονική λογισμικού σε μια εφαρμογή PHP. Για να είμαστε ακριβείς, γίνεται πιο εύκολο να γραφτεί σωστά δομημένος, υψηλής ποιότητας κώδικας. Η βάση του κώδικα είναι καθαρότερη, πιο εύκολο να διαβαστεί, και κατά συνέπεια, πιο εύκολο να διατηρηθεί. Η PHP 5 είναι σίγουρα πιο γρήγορη από ό,τι PHP 4. Η τελευταία έκδοση βασίζεται στη μηχανή Zend 2, τον πυρήνα της γλώσσας PHP. Ο χειρισμός αντικειμένων έχει ξαναγραφεί εξολοκλήρου. Έτσι, ο αντικειμενοστραφής κώδικας ιδίως εκτελείται γρηγορότερα στη PHP 5. Τα βασικά πλεονεκτήματα της PHP 5 [9] είναι τα παρακάτω:

- Ισχυρή υποστήριξη για τον αντικειμενοστραφή προγραμματισμό.
- Μια εξολοκλήρου ξαναγραμμένη επέκταση MySQL.
- Μια πρώτης τάξεως υλοποίηση SOAP (Simple Object Access protocol).
- Καθαρός έλεγχος σφαλμάτων με εξαιρέσεις.
- Σουίτα διαλειτουργικών εργαλείων XML.

Συνήθως η PHP χρησιμοποιείται συνδυασμένη με μια άλλη γλώσσα προγραμματισμού, την JavaScript.

1.5.3 JavaScript

Η JavaScript [10] είναι διερμηνευμένη γλώσσα προγραμματισμού για ηλεκτρονικούς υπολογιστές. Αρχικά αποτέλεσε μέρος της υλοποίησης των φυλλομετρητών Ιστού, ώστε τα σενάρια από την πλευρά του πελάτη (client-side scripts) να μπορούν να επικοινωνούν με τον χρήστη, να ανταλλάσσουν δεδομένα ασύγχρονα και να αλλάζουν δυναμικά το περιεχόμενο του εγγράφου που εμφανίζεται.

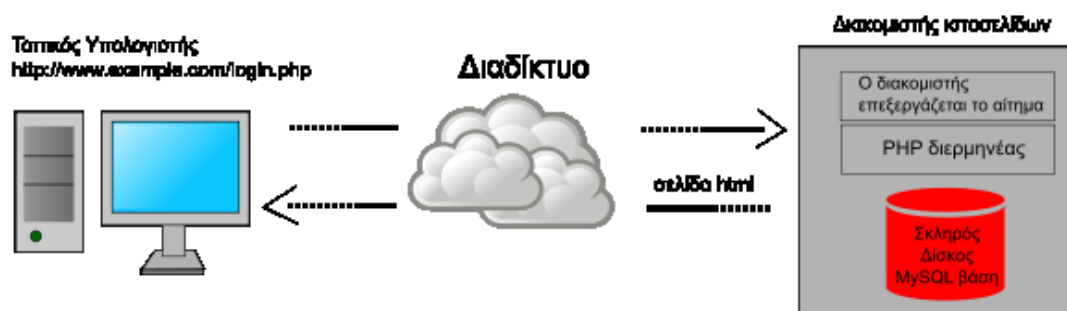
Επί της ουσίας, είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστραφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Ο κώδικας JavaScript συνήθως ενσωματώνεται στον HTML κώδικα [11]. Με αυτή μπορούμε να εκτελέσουμε κάποια πράγματα όταν συμβαίνει ένα γεγονός, για παράδειγμα όταν ο χρήστης πατήσει με το ποντίκι πάνω σε ένα html στοιχείο ,να εκτελείται κάποιο script και λαμβάνουμε τα αντίστοιχα αποτελέσματα. Η javascript μπορεί να διαβάσει και να αλλάξει τα περιεχόμενα ενός html στοιχείου. Για παράδειγμα, η παρακάτω JavaScript έκφραση: `document.write("<h1>" + name + "</h1>")` γράφει το περιεχόμενο της μεταβλητής name μέσα σε μια HTML σελίδα.

Ας εξηγήσουμε τώρα πως επικοινωνεί η PHP με μια βάση δεδομένων MySQL ώστε να δημιουργήσουν δυναμικό περιεχόμενο.

1.5.4 Δυναμικές ιστοσελίδες με PHP & MySQL

Οι πρώτες δυναμικές ιστοσελίδες [12] ξεκίνησαν χρησιμοποιώντας σενάρια εντολών σε γλώσσα προγραμματισμού C ή Perl τα οποία καλούνταν από την ιστοσελίδα μέσω σεναρίων εντολών CGI. Ο σχεδιασμός δυναμικών ιστοσελίδων με τη μορφή αυτή δεν ήταν εύκολη υπόθεση και για αυτό το λόγο αναπτύχθηκαν τεχνολογίες ανάπτυξης ιστοσελίδων όπως για παράδειγμα η εμπορική τεχνολογία ASP.NET (της Microsoft) ή τεχνολογίες ανοικτού κώδικα όπως της γλώσσας PHP & MySQL. Η ιδέα είναι ότι ο διαχειριστής της ιστοσελίδας χρησιμοποιεί σενάρια εντολών σε γλώσσα PHP τα οποία αποθηκεύουν το περιεχόμενο της ιστοσελίδας σε μια βάση δεδομένων MySQL. Ο χρήστης όταν συνδέεται στον απομακρυσμένο διακομιστή ιστοσελίδων καλεί ένα σενάριο εντολών php το οποίο χρησιμοποιώντας δεδομένα από τον σκληρό δίσκο (συνήθως πολυμεσικά αρχεία - εικόνες - ήχους) και την βάση MySQL δημιουργεί την σελίδα δυναμικά. Στις δυναμικές ιστοσελίδες αν ο διαχειριστής θέλει να αλλάξει την γενική γραφική παρουσίαση, απλά τροποποιεί το κώδικα php και όλη η ιστοσελίδα αποκτά διαφορετική εμφάνιση (σε αντίθεση με τις στατικές ιστοσελίδες όπου πρέπει να αλλάξουν όλες οι τοπικά αποθηκευμένες σελίδες html). Προτού εξηγήσουμε τι είναι η MySQL πρέπει να γίνει έστω μια μικρή αναφορά στις βάσεις δεδομένων.



Σχήμα 1. 1: Δυναμική ιστοσελίδα με PHP και MySQL

1.6 Βάση Δεδομένων

Μια Βάση Δεδομένων (Database) [13] είναι ένας οργανωμένος τρόπος αποθήκευσης πληροφοριών και πρόσβασης σε αυτές. Μια βάση δεδομένων είναι κάτι παραπάνω από μια απλή συλλογή αποθηκευμένων στοιχείων. Βάση δεδομένων είναι ένας και μοναδικός χώρος αποθήκευσης των δεδομένων ο οποίος ορίζεται μία φορά και μπορεί να χρησιμοποιηθεί ταυτόχρονα από πολλούς τελικούς χρήστες. Έτσι αντί να έχουμε ξεχωριστά αρχεία με πλεονάζουσα πληροφορία, η βάση αποτελεί πλέον μία διαμοιρασμένη συλλογική πηγή. Μία βάση δεδομένων κρατάει εκτός από τα στοιχεία κάποιου οργανισμού και μία περιγραφή των στοιχείων αυτών.

Η προσέγγιση αυτή των βάσεων δεδομένων, να διατηρούνται δηλαδή ξεχωριστά τα στοιχεία της βάσης από τα διάφορα προγράμματα εφαρμογής, μοιάζει πολύ με την προσέγγιση των σύγχρονων γλωσσών προγραμματισμού όπου για κάθε αντικείμενο παρέχεται ένα εσωτερικός ορισμός, αλλά και ένας ξεχωριστός εξωτερικός ορισμός. Έτσι οι τελικοί χρήστες ενός αντικειμένου γνωρίζουν μόνο τον εξωτερικό ορισμό του και έχουν άγνοια για το πώς έχει οριστεί αυτό το αντικείμενο και πως αυτό λειτουργεί. Έτσι με αυτή την προσέγγιση μπορούμε να αλλάξουμε τον εσωτερικό ορισμό ενός αντικειμένου χωρίς να επηρεαστεί ο τελικός χρήστης εφόσον ο εξωτερικός ορισμός του αντικειμένου παραμένει ο ίδιος. Με το ίδιο σκεπτικό η βάση δεδομένων ξεχωρίζει την δομή των δεδομένων από τα προγράμματα εφαρμογών και την αποθηκεύει στη βάση. Εάν νέες δομές δεδομένων προστεθούν ή παλιές αλλαχθούν τα προγράμματα εφαρμογών δεν επηρεάζονται (παραμένουν τα ίδια) εφόσον οι δομές που αλλάχθηκαν δεν επηρεάζουν άμεσα τα προγράμματα εφαρμογών.

1.6.1 Σύστημα διαχείρισης βάσεων δεδομένων

Ένα πρόγραμμα που διαχειρίζεται βάσεις δεδομένων αποκαλείται Σύστημα Διαχείρισης Βάσεων Δεδομένων (DBMS, Database Management System) [14] και με την βοήθειά του μπορούμε να αποθηκεύσουμε, προσθέσουμε, τροποποιήσουμε, εμφανίσουμε ή και διαγράψουμε τα αποθηκευμένα δεδομένα. Συγκεκριμένα η MySQL που χρησιμοποιήθηκε στην παρούσα εργασία είναι RDBMS (Relational Database Management System), δηλαδή σχεσιακή βάση δεδομένων, αυτή που περιέχει περισσότερες από μια οντότητες και ταυτόχρονα υπάρχει κάποια/ες συσχετίσεις ανάμεσα στις οντότητες της.

Η RDBMS είναι η βάση για την SQL, και για όλα τα σύγχρονα συστήματα βάσεων δεδομένων όπως το MS SQL Server, IBM DB2,

Oracle, MySQL και Microsoft Access. Τα δεδομένα στο RDBMS αποθηκεύονται σε αντικείμενα της βάσης δεδομένων που ονομάζονται πίνακες. Ένας πίνακας είναι μια συλλογή από σχετικές καταχωρήσεις δεδομένων και αποτελείται από στήλες και γραμμές.

Η MySQL είναι ένα πολύ γρήγορο και δυνατό, σύστημα διαχείρισης βάσεων δεδομένων.

1.7 MySQL

Ο πηγαίος κώδικας της MySQL είναι διαθέσιμος με ελεύθερη άδεια χρήσης. Μια βάση δεδομένων επιτρέπει στον διαχειριστή ενός ιστοτόπου να αποθηκεύει, να αναζητά, να ταξινομεί και να ανακαλεί τα δεδομένα αποτελεσματικά. Ο διακομιστής MySQL ελέγχει την πρόσβαση στα δεδομένα, για να μπορούν να δουλεύουν πολλοί χρήστες ταυτόχρονα, για να παρέχει γρήγορη πρόσβαση και να διασφαλίζει ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση. Συνεπώς η MySQL είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών. Χρησιμοποιεί την SQL (Structured Query Language), την τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων, παγκόσμια. Η MySQL είναι διαθέσιμη από το 1996, αλλά η ιστορία της ξεκινά από το 1979.

Η MySQL δημιουργήθηκε από μια σουηδική εταιρεία MySQL AB, το 1995. Οι προγραμματιστές της πλατφόρμας ήταν ο Michael Widenius (Monty), David Axmark και Allan Larsson. Ο κύριος σκοπός ήταν να παρέχει αποτελεσματικές και αξιόπιστες επιλογές διαχείρισης δεδομένων στο σπίτι και σε επαγγελματίες χρήστες. Πάνω από μισή ντουζίνα άλφα και βήτα εκδόσεων της πλατφόρμας κυκλοφόρησαν μέχρι το 2000. Αυτές οι εκδόσεις ήταν συμβατές με σχεδόν όλες τις μεγάλες πλατφόρμες.

Η ονομασία MySQL περιέχει δύο στοιχεία [15]. Το My είναι το όνομα της κόρης του συνιδρυτή του συστήματος Monty Widenius (Μόντι Βιδένιους) και το SQL αναφέρεται στη γλώσσα SQL (Structured Query Language), μια γλώσσα υπολογιστών που σχεδιάστηκε ανεξάρτητα από την MySQL για τη διαχείριση δεδομένων, σε ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (MySQL, PostgreSQL, Oracle κλπ). Το πρότυπο της γλώσσας SQL αναπτύχθηκε ξεχωριστά από τις υλοποιήσεις συστημάτων διαχείρισης βάσεων δεδομένων (όπως της MySQL, της PostgreSQL, της Oracle κλπ). Θεωρητικά ένας κώδικας SQL θα πρέπει να τρέχει με τον ίδιο τρόπο και σε μια βάση MySQL όπως και σε μια βάση

PostgreSQL. Κάθε υλοποίηση ενός συστήματος διαχείρισης σχεσιακών βάσεων δεδομένων μπορεί να διαφέρει με κάποιο άλλο παρέχοντας επιπλέον δυνατότητες, βελτιωμένα χαρακτηριστικά.

1.7.1 Πλεονεκτήματα της MySQL

Η MySQL έχει πολλά πλεονεκτήματα [16], όπως χαμηλό κόστος, εύκολη διαμόρφωση και μάθηση και ο κώδικας προέλευσης είναι διαθέσιμος δωρεάν. Πρώτα από όλα η MySQL είναι ένα πακέτο βάσης το οποίο είναι ανοιχτού λογισμικού, δηλαδή διανέμεται δωρεάν και δεν χρειάζεται κάποια εταιρία οργανισμός ή κάποιος τελικός χρήστης να δαπανήσει κάποιο χρηματικό ποσό για να την αγοράσει. Μπορεί πολύ απλά να την λάβει μέσω του διαδικτύου και να την εγκαταστήσει στον υπολογιστή του. Ο παραπάνω παράγοντας είναι πολύ σημαντικός ειδικά όταν μιλάμε για μία εταιρία ή ένα γραφείο μικρού βεληνεκούς χωρίς μεγάλους χρηματικούς πόρους για τους οποίους όμως η δημιουργία μίας βάσης δεδομένων είναι απαραίτητη. Ένα άλλο πολύ σημαντικό πλεονέκτημα που έχει φέρει την MySQL ανάμεσα στα πιο χρησιμοποιούμενα πακέτα βάσης είναι ότι πρόκειται για ένα πακέτο το οποίο βρίσκεται στην αγορά πάρα πολλά χρόνια, με αποτέλεσμα να έχει δοκιμαστεί σε πάρα πολλές περιπτώσεις από μικρούς οργανισμούς μέχρι εταιρίες κολοσσούς. Η ύπαρξη του πακέτου αυτού τόσα χρόνια έχει ως αποτέλεσμα να έχει μία πάρα πολύ σταθερή βάση χωρίς προβλήματα, αφού τα περισσότερα έχουν δοκιμαστεί και λυθεί όλον αυτόν τον καιρό. Επιπλέον ένα άλλο σημαντικό πλεονέκτημα, το οποίο ίσως να μην είναι και τόσο γνωστό, είναι ότι μιλάμε για ένα αρκετά ελαφρύ πακέτο, ίσως το πιο ελαφρύ της αγοράς το οποίο έχει πολύ μικρές απαιτήσεις σε πόρους υπολογιστικής ισχύος από τον υπολογιστή στον οποίο θα στηθεί. Το συγκεκριμένο πακέτο στα πρώτα χρόνια της ζωής του προοριζόταν για UNIX σταθμούς εργασίας αλλά με την πάροδο των χρόνων και την εισαγωγή της Microsoft στην αγορά δικτύων και την ανάπτυξη των ανάλογων λειτουργικών συστημάτων για διακομιστές, έγινε διαθέσιμο και για την πλατφόρμα των Windows. Ίσως ο όμως σημαντικότερος παράγοντας της καθιέρωσης της MySQL να είναι οι πάρα πολύ μικροί χρόνοι απόκρισης του συγκεκριμένου πακέτου σε σύγκριση με άλλα πακέτα βάσεων δεδομένων της αγοράς.

Για τη διαχείριση της MySQL χρησιμοποιήθηκε το εργαλείο PhpMyAdmin.

1.7.2 PhpMyAdmin

Το PhpMyAdmin είναι ένα εργαλείο διαχείρισης της βάσης δεδομένων του συστήματος μέσω ενός περιηγητή Ιστού. Είναι λογισμικό ανοικτού κώδικα γραμμένο σε PHP, ανεξάρτητο λογισμικού (cross platform) και πολυγλωσσικό (62 γλώσσες). Μέσω της έξυπνης διεπαφής του (interface) επιτρέπει τη δημιουργία/ διαγραφή μιας βάσης, δημιουργία/ διαγραφή/ αλλαγή πινάκων – πεδίων, χειρισμό δικαιωμάτων, εξαγωγή δεδομένων σε διάφορες μορφές. Όσον αφορά την σύνδεση της PHP με τη MySQL στον κώδικα που αναπτύχθηκε, αυτή έγινε χάρη στην επέκταση PDO, τα χαρακτηριστικά της οποίας αναφέρονται παρακάτω.

1.8 Η επέκταση PDO και τα πλεονεκτήματά της

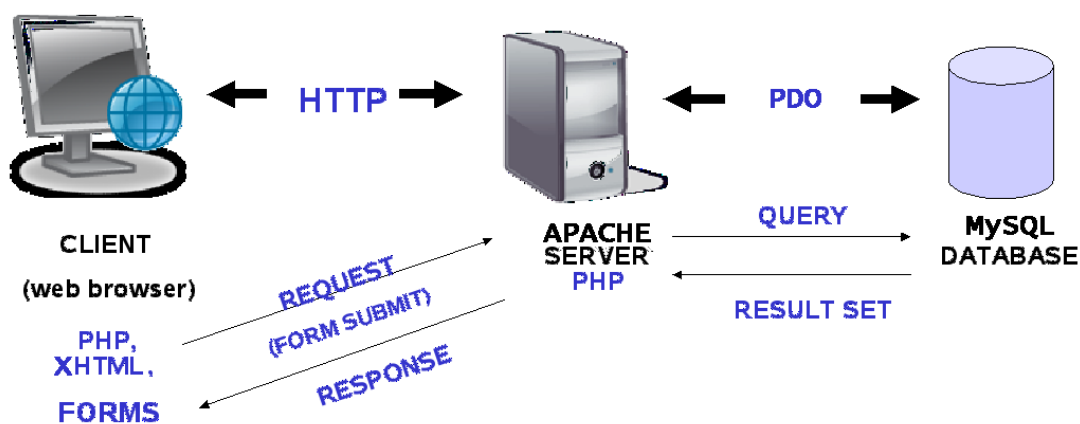
Η PDO (PHP Data Objects-Αντικείμενα Δεδομένων της PHP) είναι μια επέκταση της PHP για πρόσβαση σε σχεσιακές βάσεις δεδομένων (RDBMS). Επιτρέπει την πρόσβαση σε μια ποικιλία από διαφορετικές βάσεις δεδομένων ανάλογα με το εγκατεστημένο πρόγραμμα οδήγησης και παρέχει έναν εύκολο τρόπο για την αναζήτηση, την ανάκτηση των αποτελεσμάτων, την προετοιμασία προτάσεων SQL και την ασφάλεια των βάσεων δεδομένων στην πλατφόρμα που αναπτύχθηκε.

Η επιλογή της χρήσης PDO [17] έναντι της χρήσης απλών MySQL δηλώσεων έγινε κατά κύριο λόγο για θέματα ασφαλείας. Η PDO είναι ένα αφαιρετικό στρώμα βάσης δεδομένων που προστατεύει από την έγχυση/ ένεση SQL (SQL Injection), μια διαδεδομένη μέθοδο επίθεσης σε διαδικτυακές εφαρμογές που χρησιμοποιούν συστήματα βάσεων δεδομένων, κάνοντας χρήση έτοιμων δηλώσεων (prepared statements). Στα συστήματα διαχείρισης βάσεων δεδομένων, μια έτοιμη δήλωση ή παραμετροποιημένη δήλωση είναι μια προ-εκτελεσμένη δήλωση SQL που δέχεται μηδέν ή περισσότερες παραμέτρους με όνομα. Είναι ένα χαρακτηριστικό που χρησιμοποιείται για να εκτελεστούν οι ίδιες ή παρόμοιες δηλώσεις βάσης δεδομένων επανειλημμένα με υψηλή απόδοση. Συνήθως χρησιμοποιείται με δηλώσεις SQL, όπως τα ερωτήματα (queries). Η έτοιμη δήλωση έχει τη μορφή ενός προτύπου στο οποίο ορισμένες σταθερές τιμές αντικαθιστούνται κατά την διάρκεια κάθε εκτέλεσης

Σε σύγκριση με την άμεση εκτέλεση των δηλώσεων SQL, οι έτοιμες δηλώσεις έχουν δύο βασικά πλεονεκτήματα [18]:

- Μειώνουν την τον χρόνο ανάλυσης, καθώς η προετοιμασία για το ερώτημα γίνεται μόνο μία φορά (αν και η δήλωση εκτελείται πολλές φορές).
- Οι δεσμευμένοι παράμετροι (bound parameters) που χρησιμοποιούνται ελαχιστοποιούν το εύρος ζώνης για το διακομιστή, καθώς αποστέλλονται μόνο οι παράμετροι κάθε φορά, και όχι ολόκληρο το ερώτημα.
- Όπως ειπώθηκε οι έτοιμες δηλώσεις είναι πολύ χρήσιμες κατά των ενέσεων SQL, επειδή οι τιμές των παραμέτρων, οι οποίες μεταδίδονται αργότερα χρησιμοποιώντας ένα διαφορετικό πρωτόκολλο, δεν χρειάζεται να έχουν "διαφύγει" σωστά (escaped). Αν το αρχικό πρότυπο δήλωσης δεν προέρχεται από εξωτερική είσοδο, η ένεση SQL δεν μπορεί να συμβεί.

Ένα άλλο πλεονέκτημα της PDO είναι ότι αποτελεί μια αντικειμενοστραφή διεπαφή. Θα αναφερθούμε περαιτέρω στην ένεση SQL και την σωστή χρήση της PDO για την αποφυγή της στο κεφάλαιο 3.



Σχήμα 1. 2: Επικοινωνία client – server – database

Σε αυτό το σημείο τελείωσε το θεωρητικό υπόβαθρο της παρούσας διπλωματικής. Αφού έγινε η εισαγωγή στο θέμα της εργασίας, ώστε να μπορεί να γίνει κατανοητός ο ιστότοπος που αναπτύχθηκε, ακολουθεί η ανάλυση του συστήματος προτού υλοποιηθεί.

Κεφάλαιο 2

Ανάλυση και σχεδίαση

Έχοντας εφοδιαστεί με γνώσεις σχετικά με τις τεχνολογίες που θα χρησιμοποιηθούν για τη δημιουργία του ιστοτόπου, προχωράμε με τη σχεδίασή του. Σε αυτό το κεφάλαιο καταγράφονται οι ρόλοι των χρηστών και οι απαιτήσεις του συστήματος. Σαν συνέπεια όλων αυτών σχεδιάζεται και περιγράφεται αναλυτικά η βάση δεδομένων, ώστε να ακολουθήσει στο 3^ο κεφάλαιο η περιγραφή των λειτουργιών που προσφέρει το σύστημα.

2.1 Ανάλυση του ιστοχώρου

Ο ιστοχώρος αναπτύχθηκε για να εξυπηρετεί τις ανάγκες ενός τυπικού πανεπιστημίου, με πολλαπλά τμήματα, μαθήματα, διδάσκοντες και φοιτητές. Ως εκ τούτου, διαθέτει μια μεγάλη ευελιξία ως προς το είδος των τεστ που θα μπορούν να δημιουργηθούν. Συγκεκριμένα, υποστηρίζει ταυτόχρονα πολλαπλά μαθήματα και πολλές κατηγορίες εξέτασης για κάθε μάθημα, οι οποίες με τη σειρά τους μπορούν να περιέχουν έναν αριθμό από τεστ, στοιχεία τα οποία ρυθμίζονται από το διαχειριστή μέσω της εφαρμογής. Κάθε οντότητα που δημιουργείται σχετίζεται με την προηγούμενη. Για παράδειγμα, όταν ο καθηγητής προσθέτει ένα διαγώνισμα πρέπει να επιλέξει ένα απ' τα μαθήματα τα οποία έχει δημιουργήσει νωρίτερα. Επίσης, τα τεστ έχουν ερωτήσεις τύπου πολλαπλής επιλογής και ερωτήσεις που δέχονται απάντηση τύπου κειμένου. Οι ερωτήσεις πολλαπλής επιλογής βαθμολογούνται αυτόματα, ενώ οι ερωτήσεις ανοιχτού κειμένου βαθμολογούνται από

το διδάσκοντα, μέσω ειδικών δικτυακών σελίδων. Μια ερώτηση μπορεί να έχει και επισυναπτόμενο χωρίς περιορισμό. Αν υπάρχει επισυναπτόμενο, ο φοιτητής βλέπει την κατάλληλη ένδειξη (Μεταφόρτωση Επισυναπτόμενου) και αν πατήσει στο δεσμό, τότε αυτό κατεβαίνει στον υπολογιστή του. Ο διδάσκων έχει τη δυνατότητα να κάνει τροποποιήσεις, προσθήκες και διαγραφές των ερωτήσεων, και να ρυθμίζει τις θεματικές ενότητες εξέτασης. Μια ερώτηση μπορεί επίσης, να ρυθμιστεί ώστε να είναι προσωρινά μη διαθέσιμη. Υπάρχει και η δυνατότητα για μαζική εισαγωγή ερωτήσεων και απαντήσεων για ένα τεστ μέσω ενός αρχείου CSV.

Τα τεστ όλα έχουν ένα συγκεκριμένο χρονικό περιθώριο, ατομικό για κάθε φοιτητή (δηλαδή, ένας φοιτητής μπορεί να ξεκινήσει 10 λεπτά αργότερα από έναν άλλο και να έχει ξεχωριστό μετρητή από τον πρώτο φοιτητή). Ο διδάσκων μπορεί να ρυθμίζει τον διαθέσιμο υπολειπόμενο χρόνο λήξης για κάθε τεστ (να προσθέτει 5 λεπτά ή να κάνει λήξη χρόνου) και να παρακολουθεί σε πραγματικό χρόνο την εξέλιξη μιας συγκεκριμένης εξέτασης. Κάθε διαγώνισμα ρυθμίζεται ώστε να έχει ημερομηνία έναρξης και λήξης, δηλαδή το περιθώριο στο οποίο θα είναι διαθέσιμο. Μετά την υποβολή του τεστ από ένα φοιτητή, ο διδάσκων μπορεί να ενημερώνεται για το βαθμό του φοιτητή, παράλληλα με αναλυτικά στατιστικά στοιχεία που αφορούν τις απαντήσεις που δόθηκαν. Είναι ακόμη δυνατή η δημιουργία ενός PDF με τα αποτελέσματα αυτά.

Τα τεστ είναι διαθέσιμα στους φοιτητές, ύστερα από αυθεντικοποίηση με όνομα χρήστη και κωδικό, ή μέσω ειδικού ανώνυμου αναγνωριστικού (token) ή τελείως ανώνυμα. Τα tokens παράγονται από ειδική σελίδα του ιστοχώρου, όπου δημιουργούνται σε μορφή PDF για να τα εκτυπώσει ο διδάσκων και να τα μοιράσει στους φοιτητές. Μαζί με τα tokens επισυνάπτεται ένα QR code. Το QR code είναι μια εικόνα που παράγεται από αντίστοιχη βιβλιοθήκη και αντιστοιχεί σε μια διεύθυνση που μπορεί να ανοίξει άμεσα τη σελίδα ένας φοιτητής. Τέλος, δημιουργούνται τα κατάλληλα δυναμικά γραφήματα που θα ενημερώνουν το διδάσκοντα για τα αποτελέσματα της ενεργής εξέτασης ή προηγούμενων εξετάσεων.

2.2 Ρόλοι χρηστών του ιστοχώρου

Η βασική αρχή για να γίνει σωστός σχεδιασμός, είναι να βρεθούν οι χρήστες του συστήματος, που αλληλεπιδρούν με αυτό. Έτσι εντοπίστηκαν πέντε:

- Ο καθολικός διαχειριστής του συστήματος.
- Ο απλός διαχειριστής/ διδάσκοντας.
- Ο χρήστης/ φοιτητής με αυθεντικοποιημένη εγγραφή.
- Ο ανώνυμος χρήστης με χρήση token.
- Ο πλήρως μη αυθεντικοποιημένος/ ελεύθερος χρήστης.

Ο ρόλος του καθένα διαφορετικού χρήστη αναλύεται στις επόμενες παραγράφους και οι λεπτομέρειες σχετικά με αυτές, συμπληρώνονται κατά την ανάπτυξη του συστήματος.

Ο ρόλος του καθολικού διαχειριστή που αλληλεπιδρά με το σύστημα είναι πολυσύνθετος. Επικεντρώνεται στην προσθήκη και επίβλεψη των οντοτήτων, που έχουν να κάνουν με την αποθήκευση πληροφοριών για τους διδάσκοντες, τα μαθήματα, τις κατηγορίες, τις ερωτήσεις, τα διαγωνίσματα κτλ. Ο καθολικός διαχειριστής προσθέτει και ενημερώνει κυρίως τις πληροφορίες για τους διδάσκοντες και τις κατηγορίες των μαθημάτων, οι οποίες δεν είναι επεξεργάσιμες απ' τους απλούς διαχειριστές. Η ενημέρωση των πληροφοριών των καθηγητών απαιτείται να γίνεται περιοδικά ούτως ώστε το σύστημα να είναι πάντα ενήμερο για τους νέους καθηγητές κάθε ακαδημαϊκού έτους. Έτσι, προσφέρεται όσο το δυνατόν καλύτερη πληροφορία στον απλό διαχειριστή.

Αυτός με την σειρά του είναι υπεύθυνος για την προσθήκη, την ενημέρωση και τη διαγραφή των πληροφοριών που αφορούν τα μαθήματά του, τις ερωτήσεις και τα διαγωνίσματα αυτών. Μπορεί να δει τα αποτελέσματα των φοιτητών για τα τεστ των μαθημάτων του μόνο.

Ο κύριος ρόλος του συστήματος είναι αυτός του απλού χρήστη/ φοιτητή. Ουσιαστικά, είναι αυτός ο οποίος θα μπορεί μέσω του ιστοχώρου να δώσει μια εξέταση. Ο εγγεγραμμένος χρήστης πρέπει να εισάγει κατά την είσοδό του στο σύστημα το όνομα χρήστη και το συνθηματικό του. Μπορεί να έχει πρόσβαση έτσι στα τεστ που είναι διαθέσιμα εκείνη τη στιγμή αλλά και στους βαθμούς όλων των διαγωνισμάτων που έχει υποβάλει .

Ο ανώνυμος χρήστης είναι η περίπτωση του χρήστη που δεν έχει εγγραφεί στο σύστημα με συνθηματικό, αλλά αντίθετα έκανε χρήση

ενός κωδικού ανώνυμης περιήγησης στον ιστότοπο, δηλαδή ενός token, ή εισήρθε "σκανάροντας" ένα QR code από μια φορητή συσκευή. Αυτός έχει την δυνατότητα εξέτασης, αλλά δεν μπορεί να δει τον βαθμό όλων των τεστ, παραμόνο αυτού που υπέβαλε με το token που μόλις χρησιμοποίησε.

Τέλος ο ελεύθερος χρήστης είναι αυτός που έχει τις πιο περιορισμένες δυνατότητες. Μπορεί μόνο να δώσει εξετάσεις που είναι ελεύθερες για το κοινό και δεν έχει πρόσβαση στον βαθμό του.

2.3 Απαιτήσεις του ιστοχώρου

Στο σημείο αυτό πρέπει να σημειωθεί ότι οι χρήστες του συστήματος δε χρειάζεται να κάνουν αρχική εγγραφή, καθώς το σύστημα επικοινωνεί με την πλατφόρμα "Laboratory of Digital Systems and Computer Architecture" του τμήματος Μηχανικών Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Δυτικής Μακεδονίας, η οποία βρίσκεται υπό την επίβλεψη του διδάσκοντα Δασυγένη Μηνά. Από αυτήν τη πλατφόρμα δανείζεται τα στοιχεία των χρηστών της και τα χρησιμοποιεί έτοιμα και ανέπαφα.

Έχοντας αναλύσει λοιπόν τους ρόλους των χρηστών του ιστότοπου εξέτασης φοιτητών, μπορούν να εκφραστούν και οι απαιτήσεις που έχουν αυτοί από την εφαρμογή.

Έτσι ένας απλός διαχειριστής μπορεί:

- Να συνδεθεί στο σύστημα.
- Να διαχειρίζεται τα μαθήματα (προσθήκη, επεξεργασία, διαγραφή).
- Να διαχειρίζεται τις ενότητες των μαθημάτων.
- Να διαχειρίζεται τις ερωτήσεις (δυνατότητα προσθήκης πολλαπλών ερωτήσεων και προσθήκης επισυναπτόμενων).
- Να διαχειρίζεται τα διαγωνίσματα (δυνατότητα προσθήκης επισυναπτόμενων).
- Να προσθέτει, να διαγράφει, να εκτυπώνει σειρές από tokens και να ελέγχει την κατάσταση χρήσης τους.
- Να δει τα αποτελέσματα και τους βαθμούς των εξετάσεων των φοιτητών, τόσο σε πίνακες όσο και σε γραφήματα κατά την διάρκεια των εξετάσεων αλλά και αφότου ολοκληρωθούν.

- Να τροποποιήσει τον διαθέσιμο χρόνο υποβολής των ενεργών εξετάσεων, προσθέτοντας 5 λεπτά ή αφαιρώντας όλο τον χρόνο (λήξη εξέτασης).
- Να εκτυπώσει όλα τα αποτελέσματα μιας εξέτασης ή ενός συγκεκριμένου φοιτητή.

Ο καθολικός διαχειριστής κατά συνέπεια μπορεί να κάνει εξίσου όλα τα παραπάνω, καθώς και:

- Να διαχειρίζεται τα τμήματα του Πανεπιστημίου.
- Να διαχειρίζεται τους καθηγητές.
- Να διαχειρίζεται τις κατηγορίες των μαθημάτων.

Ένας εγγεγραμμένος χρήστης μπορεί:

- Να παίρνει μέρος σε μια εξέταση, με ερωτήσεις πολλαπλής επιλογής και ελεύθερου κειμένου.
- Να προσπερνάει ερωτήσεις, τις οποίες θα συναντά ξανά στο τέλος, αφού έχουν προηγηθεί όλες οι ερωτήσεις του διαγωνίσματος.
- Να στέλνει μηνύματα προς τον διδάσκοντα κατά τη διάρκεια μιας εξέτασης τα οποία αφορούν μία συγκεκριμένη ερώτηση.
- Να δει τον βαθμό του μόλις ολοκληρωθεί η εξέταση (αν έχει ρυθμιστεί να είναι ορατός από τον καθηγητή). Ο βαθμός ενδέχεται να μην είναι τελικός αν υπήρχαν ερωτήσεις ελεύθερου κειμένου.
- Να δει ξανά τις τελικές βαθμολογίες του για όλα τα διαγωνίσματα που έχει υποβάλει (αν ο βαθμός τους είναι ορατός).

Ένας χρήστης που συνδέθηκε στο σύστημα με token μπορεί:

- Να παίρνει μέρος σε μια εξέταση, με ερωτήσεις πολλαπλής επιλογής και ελεύθερου κειμένου.
- Να προσπερνάει ερωτήσεις, τις οποίες θα συναντά ξανά στο τέλος, αφού έχουν προηγηθεί όλες οι ερωτήσεις του διαγωνίσματος.
- Να στέλνει μηνύματα προς τον διδάσκοντα κατά τη διάρκεια μιας εξέτασης τα οποία αφορούν μία συγκεκριμένη ερώτηση.
- Να δει τον βαθμό του με το πέρας της εξέτασης.

- Να δει ξανά τον βαθμό του για οποιαδήποτε εξέταση έχει πραγματοποιήσει χρησιμοποιώντας token, εάν το έχει φυλάξει και το γνωρίζει.

Ενώ ο ελεύθερος χρήστης μπορεί μόνο:

- Να πραγματοποιήσει μια εξέταση που περιλαμβάνει μόνο ερωτήσεις πολλαπλής επιλογής.
- Να παραλείπει ερωτήσεις για τις οποίες δε γνωρίζει την απάντηση.
- Να δει τον τελικό βαθμό του, μόλις ολοκληρωθεί η εξέταση.

2.4 Περιπτώσεις χρήσης

Σε αυτή την ενότητα αναλύονται ορισμένες από τις περιπτώσεις χρήσης [19] του ιστοχώρου, καθώς αυτές είναι αρκετές και η εκτενής ανάλυση όλων θεωρείται υπερβολική.

2.4.1 Είσοδος στο σύστημα

Σύντομη περιγραφή

Ένας χειριστής μπορεί να εισέλθει στο σύστημα χρησιμοποιώντας τα στοιχεία λογαριασμού του. Αυτή είναι η περίπτωση χρήσης "Είσοδος στο σύστημα".

Χειριστές

Διαχειριστής, διδάσκοντας και εγγεγραμμένος χρήστης.

Βασική ροή

Αυτή η περίπτωση ξεκινά μόλις ο χρήστης μπει στην αρχική σελίδα του ιστότοπου και επιλέξει το "Είσοδος", από το πλαίσιο "Σύνδεση χρήστη".

- Το πλαίσιο αυτό περιέχει δύο πεδία που ζητούν το όνομα χρήστη και το συνθηματικό.
- Ο χρήστης εισάγει τα στοιχεία του και περιμένει την ανταπόκριση από το σύστημα.

- Το σύστημα αναγνωρίζει ως σωστές τις πληροφορίες που δόθηκαν από τον χρήστη και τον μεταφέρει στην εκάστοτε αρχική σελίδα ανάλογα με την ιδιότητά του (καθολικός διαχειριστής, διδάσκοντας, φοιτητής).

Εναλλακτική ροή 1 – ο χειριστής εισάγει λανθασμένη πληροφορία

Σε αυτή τη περίπτωση το σύστημα δε μπορεί να επικυρώσει τα στοιχεία του χειριστή, επειδή είναι λανθασμένα.

- Η ιστοσελίδα εμφανίζει προειδοποιητικό μήνυμα λάθους.
- Έπειτα ζητείται από τον χειριστή η επανεισαγωγή των στοιχείων.
- Το σύστημα ελέγχει τα δεδομένα που δέχτηκε. Αν είναι σωστά, τότε συνεχίζει με το βήμα 3 της βασικής ροής. Διαφορετικά επανέρχεται στην εναλλακτική ροή 1.

Κατάσταση εισόδου

Δεν υπάρχει.

Κατάσταση εξόδου

- Εισαγωγή του χρήστη στο σύστημα.
- Μη εισαγωγή λόγω εσφαλμένων δεδομένων και ως εκ τούτου παραμονή στην αρχική σελίδα σύνδεσης.

2.4.2 Προσθήκη νέας ερώτησης

Σύντομη περιγραφή

Αυτή η περίπτωση χρήσης συμβαίνει όταν ένας χειριστής προσθέτει μια μία τις ερωτήσεις μέσω της ειδικής φόρμας του ιστοχώρου.

Χειριστές

Καθολικός και απλός διαχειριστής.

Βασική ροή

Αυτή η περίπτωση ξεκινά μόλις ο διαχειριστής επιλέξει το "Προσθήκη ερωτήσεων" από το βασικό μενού του ιστοχώρου.

- Η ιστοσελίδα εμφανίζει μια φόρμα με 8 ή 18 πεδία, ανάλογα με το αν ο διαχειριστής επιλέξει να καταχωρήσει ερώτηση ελεύθερου κειμένου ή πολλαπλής επιλογής, από τα οποία τα 5 ή 10 αντίστοιχα είναι υποχρεωτικά.
- Ο διαχειριστής εισάγει τα δεδομένα που θέλει του και περιμένει την ανταπόκριση από το σύστημα.
- Το σύστημα αναγνωρίζει ως σωστές τις πληροφορίες που δόθηκαν και είτε προσθέτει τα στοιχεία ή μεταφέρει τον διαχειριστή στην προεπισκόπηση της ερώτησης, αναλόγως την επιλογή του.
- Αν ο διαχειριστής επιλέξει την προσθήκη, τότε η νέα ερώτηση καταχωρείται στη βάση δεδομένων.

Εναλλακτική ροή 1 – ο διαχειριστής παραλείπει ένα υποχρεωτικό πεδίο

- Εμφανίζεται προειδοποιητικό μήνυμα ότι δεν συμπληρώθηκαν όλα τα υποχρεωτικά πεδία.
- Ο διαχειριστής πηγαίνει πίσω στη φόρμα προσθήκης ερώτησης.
- Εισάγει από την αρχή τα ίδια δεδομένα, προσέχοντας να μην αφήσει κανένα υποχρεωτικό πεδίο κενό.
- Το σύστημα ελέγχει τα δεδομένα και αν είναι επαρκή συνεχίζει με το βήμα 4 της βασικής ροής, αλλιώς επανέρχεται στην εναλλακτική ροή 1.

Εναλλακτική ροή 2 – ο διαχειριστής επιλέγει μη έγκυρο επισυναπτόμενο

- Εμφανίζεται προειδοποιητικό μήνυμα ότι η μορφή του επισυναπτόμενου δεν υποστηρίζεται ή ότι είναι πολύ μεγάλο σε μέγεθος bytes.
- Ο διαχειριστής πηγαίνει πίσω στη φόρμα προσθήκης ερώτησης.
- Εισάγει από την αρχή τα ίδια δεδομένα, επιλέγοντας νέο επισυναπτόμενο ή κανένα.
- Το σύστημα ελέγχει τα δεδομένα και αν το επισυναπτόμενο είναι έγκυρο (και δεν έχει πέσει σε περίπτωση κάποιας άλλης

εναλλακτικής ροής) συνεχίζει με το βήμα 4 της βασικής ροής, αλλιώς επανέρχεται στην εναλλακτική ροή 2.

Εναλλακτική ροή 3 – η ερώτηση είναι ήδη καταχωρημένη

- Μόλις ο διαχειριστής επιλέξει την προσθήκη της ερώτησης, το σύστημα τον μεταφέρει αυτόματα στη σελίδα με την φόρμα, εμφανίζοντας το κατάλληλο μήνυμα λάθους.
- Ο διαχειριστής μπορεί να επιλέξει να εισάγει νέα δεδομένα, διαφορετικά από τα προηγούμενα.
- Το σύστημα ελέγχει τα δεδομένα και αν καλύπτουν τις προϋποθέσεις συνεχίζει με το βήμα 4 της βασικής ροής, αλλιώς επανέρχεται στην εναλλακτική ροή 3.

Εναλλακτική ροή 4 – δεν υπάρχουν διαθέσιμα μαθήματα ή ενότητες

- Η εισαγωγή ενός μαθήματος και μιας ενότητας είναι υποχρεωτική. Έτσι αν δεν προϋπάρχει καταχωρημένο κάποιο μάθημα ή ενότητα τότε το σύστημα εμφανίζει προειδοποιητικό μήνυμα στη θέση των δύο αυτών πεδίων.
- Αν ο διαχειριστής πατήσει "Προσθήκη" ή "Προεπισκόπηση", τότε το σύστημα εκτελεί την εναλλακτική ροή 1. Διαφορετικά ο διδάσκοντας προσθέτει αρχικά νέα μαθήματα και ενότητες και επιστρέφει στη βασική ροή.

Εναλλακτική ροή 5 – ένα από τα επισυναπτόμενα δε γίνεται να προστεθεί

- Αν ο τύπος του επισυναπτόμενου δεν υποστηρίζεται από το σύστημα ή το μέγεθός του υπερβαίνει το όριο του συστήματος (5Mb) ή υπάρχει ήδη στο σύστημα ένα αρχείο με ίδιο όνομα, τότε εμφανίζεται το ανάλογο μήνυμα σφάλματος.
- Ο διαχειριστής επιλέγει άλλο αρχείο ή αλλάζει το όνομα του αρχείου που θέλει να ανεβάσει και ξαναπροσπαθεί. Αν δεν υπάρχει πρόβλημα, το σύστημα συνεχίζει με το βήμα 4 της βασικής ροής.

Κατάσταση εισόδου

Πρέπει να υπάρχουν διαθέσιμα μαθήματα και ενότητες για την εισαγωγή μιας νέας ερώτησης.

Κατάσταση εξόδου

- Η ερώτηση καταχωρείται στη βάση δεδομένων.
- Η προσθήκη της ερώτησης δεν είναι δυνατή, επειδή ο διαχειριστής παρέλειψε την εισαγωγή δεδομένων ενός ή περισσότερων υποχρεωτικών πεδίων.
- Η προσθήκη της ερώτησης δεν είναι δυνατή, επειδή ο διαχειριστής εισήγαγε ένα επισυναπτόμενο που δεν έχει έγκυρη μορφή ή μέγεθος.
- Η προσθήκη της ερώτησης δεν είναι εφικτή, επειδή μια ολότητα έχει ήδη προστεθεί στο σύστημα.
- Υπάρχει πρόβλημα με την προσθήκη της ερώτησης, επειδή ένα από τα επισυναπτόμενα δεν τηρεί τις προϋποθέσεις.

2.4.3 Προσθήκη νέου διαγωνίσματος

Σύντομη περιγραφή

Η περίπτωση χρήσης "Προσθήκη νέου διαγωνίσματος" επιτρέπει σε έναν χειριστή να προσθέσει μια νέα εξέταση μέσω της ειδικής φόρμας του ιστοχώρου.

Χειριστές

Καθολικός και απλός διαχειριστής.

Βασική ροή

Αυτή η περίπτωση ξεκινά μόλις ο χειριστής επιλέξει το "Προσθήκη διαγωνισμάτων" από το βασικό μενού του ιστοχώρου.

- Η ιστοσελίδα εμφανίζει μια φόρμα με 16 πεδία, από τα οποία τα 11 είναι υποχρεωτικά.
- Ο χειριστής εισάγει τις πληροφορίες που αφορούν το διαγώνισμα και περιμένει την ανταπόκριση από το σύστημα.

- Το σύστημα αναγνωρίζει ως σωστές τις πληροφορίες που δόθηκαν και μεταφέρει τον χειριστή στην προεπισκόπηση του διαγωνίσματος που καταχώρησε.
- Αν ο χειριστής επιλέξει την προσθήκη, τότε η νέα εξέταση καταχωρείται στη βάση δεδομένων.

Εναλλακτική ροή 1 – ο διαχειριστής παραλείπει ένα υποχρεωτικό πεδίο

- Στη σελίδα προεπισκόπησης εμφανίζεται προειδοποιητικό μήνυμα ότι δεν συμπληρώθηκαν όλα τα υποχρεωτικά πεδία.
- Ο χειριστής πηγαίνει πίσω στη φόρμα προσθήκης διαγωνίσματος.
- Εισάγει από την αρχή τα ίδια δεδομένα, προσέχοντας να μην αφήσει κανένα υποχρεωτικό πεδίο κενό.
- Το σύστημα ελέγχει τα δεδομένα και αν είναι επαρκή, συνεχίζει με το βήμα 4 της βασικής ροής, αλλιώς επανέρχεται στην εναλλακτική ροή 1.

Εναλλακτική ροή 2 – ο διαχειριστής επιλέγει λανθασμένο αριθμό ερωτήσεων

- Αν ο επιλεγμένος αριθμός συνολικών ερωτήσεων είναι μεγαλύτερος από τον αριθμό των διαθέσιμων ερωτήσεων για το επιλεγμένο μάθημα ή την κατηγορία ή τα επιλεγμένα επίπεδα δυσκολίας ή τα ποσοστά ενοτήτων, τότε στη σελίδα προεπισκόπησης εμφανίζεται το σχετικό προειδοποιητικό μήνυμα.
- Ο χειριστής πηγαίνει πίσω στη φόρμα προσθήκης εξέτασης.
- Εισάγει από την αρχή τα δεδομένα, επιλέγοντας διαφορετικό αριθμό συνολικών ερωτήσεων ή διαφορετικό συνδυασμό ποσοστών ενοτήτων ή επιπέδων δυσκολίας.
- Το σύστημα ελέγχει τα δεδομένα και αν είναι σωστά συνεχίζει με το βήμα 4 της βασικής ροής, αλλιώς επανέρχεται στην εναλλακτική ροή 2.

Εναλλακτική ροή 3 – ο διαχειριστής επιλέγει λανθασμένα ποσοστά ενοτήτων

- Αν τα επιλεγμένα ποσοστά ενοτήτων δεν συμπληρώνουν άθροισμα 100%, τότε στη σελίδα προεπισκόπησης διαγωνίσματος εμφανίζεται το κατάλληλο προειδοποιητικό μήνυμα.
- Ο χειριστής πηγαίνει πίσω στη φόρμα προσθήκης εξέτασης.
- Εισάγει από την αρχή τα δεδομένα, επιλέγοντας διαφορετικά ποσοστά ενοτήτων.
- Αν αυτά έχουν άθροισμα 100%, το σύστημα συνεχίζει με το βήμα 4 της βασικής ροής, αλλιώς επανέρχεται στην εναλλακτική ροή 3.

Εναλλακτική ροή 4 – ο διαχειριστής επιλέγει λανθασμένη ημερομηνία λήξης

- Αν η επιλεγμένη ημερομηνία λήξης είναι παλιότερη από την ημερομηνία έναρξης, τότε το σύστημα εμφανίζει το κατάλληλο προειδοποιητικό μήνυμα στη σελίδα προεπισκόπησης.
- Ο χειριστής πηγαίνει πίσω στη φόρμα προσθήκης διαγωνίσματος.
- Εισάγει από την αρχή τα δεδομένα, επιλέγοντας διαφορετική ημερομηνία λήξης.
- Αν αυτή είναι νεότερη από την ημερομηνία έναρξης, το σύστημα συνεχίζει με το βήμα 4 της βασικής ροής, αλλιώς επανέρχεται στην εναλλακτική ροή 4.

Εναλλακτική ροή 5 – ο διαχειριστής επιλέγει λανθασμένο διαθέσιμο χρόνο εξέτασης

- Στη περίπτωση που ο χειριστής επιλέξει μηδενικό διαθέσιμο χρόνο υποβολής του διαγωνίσματος το σύστημα εμφανίζει το κατάλληλο προειδοποιητικό μήνυμα στη σελίδα προεπισκόπησης.
- Ο χειριστής πηγαίνει πίσω στη φόρμα προσθήκης διαγωνίσματος.
- Εισάγει από την αρχή τα δεδομένα, επιλέγοντας διαφορετικό διαθέσιμο χρόνο εξέτασης.
- Αν είναι σωστός, το σύστημα συνεχίζει με το βήμα 4 της βασικής ροής, αλλιώς επανέρχεται στην εναλλακτική ροή 5.

Εναλλακτική ροή 6 – το διαγώνισμα είναι ήδη καταχωρημένο

- Σε αυτή τη περίπτωση ο διαχειριστής εισάγει δεδομένα που συνδυαστικά υπάρχουν ήδη καταχωρημένα στο σύστημα. Αρχικά μεταφέρεται στη προεπισκόπηση του διαγωνίσματος.
- Μόλις επιλέξει την προσθήκη των δεδομένων, το σύστημα επανέρχεται στην αρχική φόρμα εμφανίζοντας μήνυμα λάθους.
- Ο διαχειριστής εισάγει εκ νέου διαφορετικά δεδομένα.
- Αν δεν είναι ήδη καταχωρημένα, το σύστημα συνεχίζει με το βήμα 4 της βασικής ροής, αλλιώς επανέρχεται στην εναλλακτική ροή 6.

Εναλλακτική ροή 7 – δεν υπάρχουν διαθέσιμα μαθήματα ή κατηγορίες

- Η εισαγωγή ενός μαθήματος και μιας κατηγορίας είναι υποχρεωτική όταν ο διαχειριστής προσθέτει ένα νέο διαγώνισμα. Επομένως αν δεν προϋπάρχει καταχωρημένο κάποιο μάθημα ή κατηγορία, το σύστημα εμφανίζει προειδοποιητικό μήνυμα στη θέση των δύο αυτών πεδίων.
- Αν ο διαχειριστής πατήσει συνέχεια τότε το σύστημα εκτελεί την εναλλακτική ροή 1. Διαφορετικά προσθέτει αρχικά νέα μαθήματα και κατηγορίες και επιστρέφει στη βασική ροή.

Κατάσταση εισόδου

Πρέπει να υπάρχουν διαθέσιμα μαθήματα και κατηγορίες για την εισαγωγή ενός νέου διαγωνίσματος.

Κατάσταση εξόδου

- Το διαγώνισμα καταχωρείται στη βάση δεδομένων.
- Η προεπισκόπηση του διαγωνίσματος εμφανίζει προειδοποιητικό μήνυμα, ότι ο διαχειριστής παρέλειψε την εισαγωγή δεδομένων ενός ή περισσότερων υποχρεωτικών πεδίων.
- Η προεπισκόπηση του διαγωνίσματος δεν είναι εφικτή, επειδή ο διαχειριστής εισήγαγε λανθασμένο αριθμό ερωτήσεων.
- Η προεπισκόπηση του διαγωνίσματος δεν είναι εφικτή, επειδή ο διαχειριστής εισήγαγε λανθασμένα ποσοστά ενοτήτων.

- Στην προεπισκόπηση του διαγωνίσματος εμφανίζεται μήνυμα λάθους, επειδή ο διαχειριστής εισήγαγε λανθασμένη ημερομηνία λήξης.
- Στην προεπισκόπηση του διαγωνίσματος εμφανίζεται μήνυμα λάθους, επειδή ο διαχειριστής εισήγαγε λανθασμένο διαθέσιμο χρόνο εξέτασης.
- Η προσθήκη διαγωνίσματος δεν είναι εφικτή, επειδή ο ίδιος συνδυασμός στοιχείων του διαγωνίσματος είναι ήδη καταχωρημένος στη βάση δεδομένων.

Αφού περιγράψαμε τις απαιτήσεις του συστήματος, τους ρόλους του διαχειριστή και του χρήστη και ορισμένες από τις περιπτώσεις χρήσης, ήρθε η ώρα να σχεδιάσουμε την βάση δεδομένων με την οποία θα συνδέεται ο ιστότοπος.

2.5 Βάση δεδομένων

Η δημιουργία της βάσης δεδομένων του ιστότοπου διαδικτυακής εξέτασης φοιτητών έγινε με στόχο να αποθηκεύονται με τον καλύτερο δυνατό τρόπο οι πληροφορίες σχετικά με τα διαγωνίσματα, τόσο για έναν φοιτητή που συλλέγει τις πληροφορίες και προσθέτει νέες, όσο και για έναν καθηγητή που τις προσθέτει και τις επεξεργάζεται. Ο σχεδιασμός της βάσης δεδομένων έγινε με γνώμονα τις απαιτήσεις του συστήματος που αναλύθηκαν σε προηγούμενο στάδιο και ήταν όσο πιο προσεγγμένος γίνεται, ώστε να μην υπάρχουν άσκοπα πεδία ή πεδία που επαναλαμβάνονται.

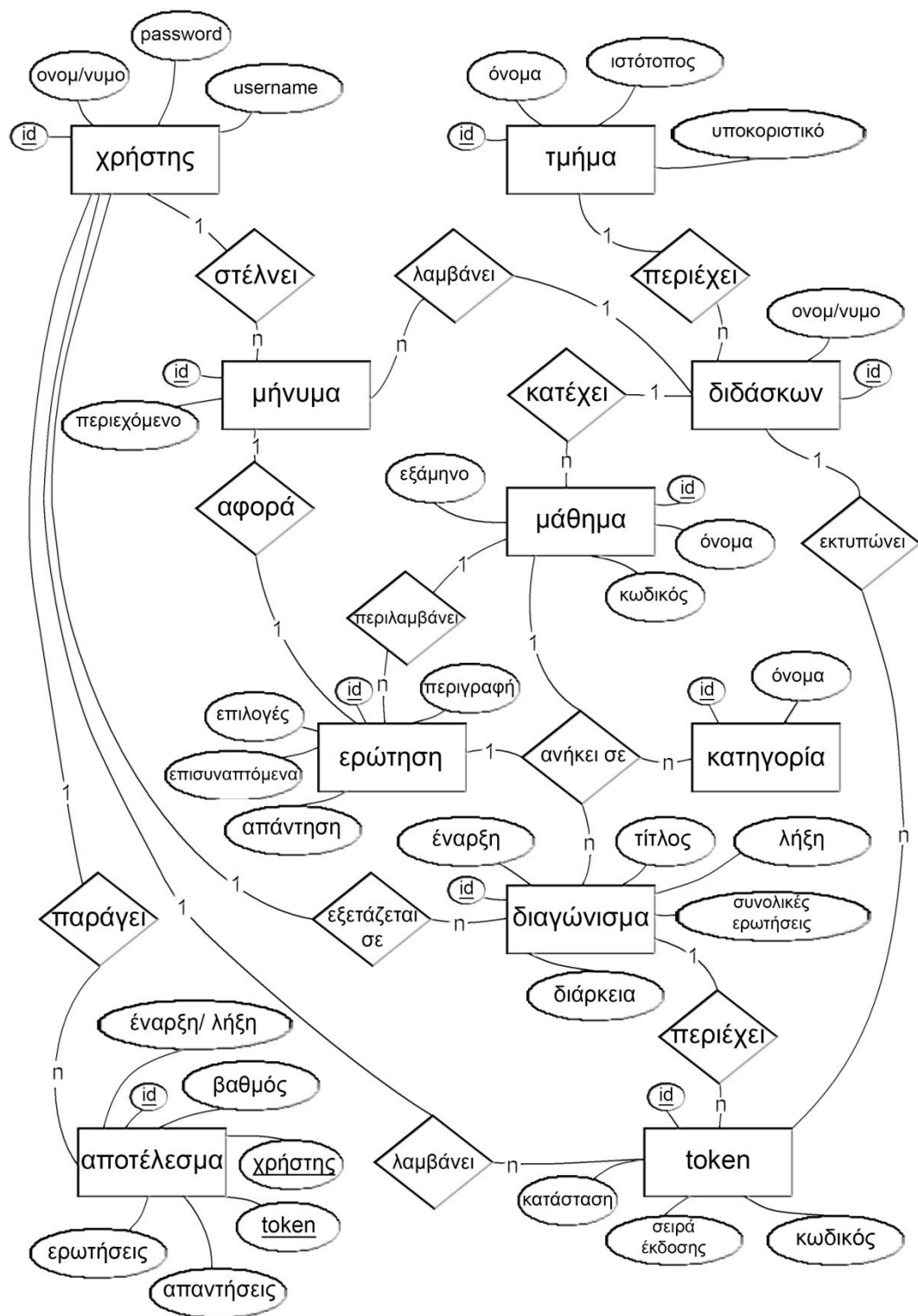
Αρχικά έγινε μια προσπάθεια να αποτυπωθεί σχηματικά η αλληλεπίδραση των δεδομένων του ιστοτόπου, με αποτέλεσμα το Σχήμα 1. 3. Αυτό το σχήμα παρουσιάζει δέκα οντότητες (χρήστες, διδάσκοντες, τμήματα, μαθήματα, κατηγορίες, ερωτήσεις, διαγωνίσματα, tokens, αποτελέσματα και μηνύματα). Έτσι ένας διδάσκοντας μπορεί να διαθέτει μαθήματα σε ένα τμήμα. Τα διαγωνίσματα δημιουργούνται από ερωτήσεις που ανήκουν σε ένα μάθημα και μια συγκεκριμένη κατηγορία. Με τη σειρά τους δημιουργούνται και τα tokens για ένα συγκεκριμένο διαγώνισμα. Τέλος οι χρήστες υποβάλλοντας ένα τεστ παράγουν τα αποτελέσματα και μπορούν να καταθέσουν ένα μήνυμα προς τους διδάσκοντες για μια

συγκεκριμένη ερώτηση, αν πιστεύουν ότι χρειάζεται περαιτέρω εξήγηση.

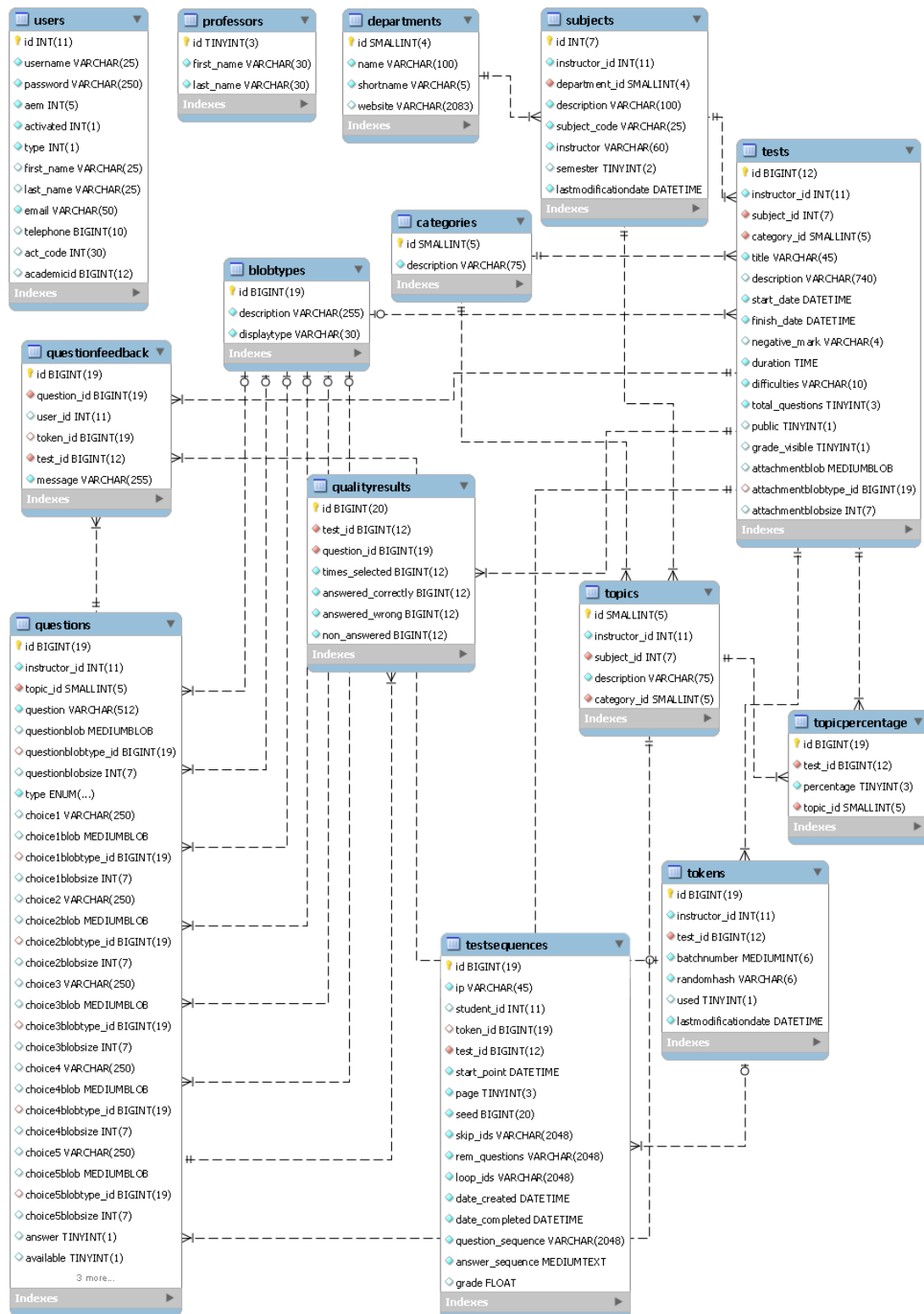
Φυσικά αυτό το σχήμα αναδιαμορφωνόταν διαρκώς κατά τη διάρκεια της ανάπτυξης του συστήματος, ανάλογα με τις ανάγκες. Καθώς εμπλουτιζόταν με νέα στοιχεία, η τελική δομή της βάσης δεδομένων που προέκυψε περιλαμβάνει τους παρακάτω πίνακες:

- Users
- Departments
- Professors
- Subjects
- Categories
- Topics
- Blootypes
- Questions
- Tests
- Topicpercentage
- Tokens
- Testsequences
- Qualityresults
- questionfeedback

Παρακάτω θα αναλύσουμε καθένα από τα πεδία των πινάκων αυτών. Υπενθυμίζεται βέβαια, ότι ο πρώτος πίνακας Users έχει αντιγραφεί αυτού καθ αυτού από την βάση δεδομένων της πλατφόρμας "Laboratory of Digital Systems and Computer Architecture", η οποία βρίσκεται υπό την επίβλεψη του διδάκτορα Δασυγένη Μηνά. Γι' αυτό τον λόγο δε θα αναλύσουμε τα πεδία του στη παρούσα εργασία, ωστόσο πρέπει να αναφερθεί ότι είναι τύπου MyISAM, ενώ οι υπόλοιποι πίνακες είναι τύπου InnoDB. Επομένως δεν είναι εφικτή η εξάρτηση του πρώτου πίνακα με τους υπόλοιπους με χρήση εξωτερικών κλειδιών.



Σχήμα 1. 3: Προσχέδιο βάσης δεδομένων



Σχήμα 1. 4: Διάγραμμα οντοτήτων-συσχετίσεων

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
Id	smallint(4)	όχι	κανένα
name	varchar(100)	όχι	κανένα
shortname	varchar(5)	όχι	κανένα
website	varchar(2083)	ναι	κενό

Πίνακας 1. 1: Πίνακας Departments

2.5.1 Departments

Στον πίνακα Departments αποθηκεύονται όλα τα τμήματα του Πανεπιστημίου. Για την αποδοτικότερη υλοποίηση του ιστότοπου, διαχειρίζονται από τον καθολικό διαχειριστή μόνο. Αυτό σημαίνει ότι δεν υπάρχει πεδίο αποθήκευσης του καθηγητή που προσθέτει κάθε νέο τμήμα.

id

Είναι το πρωτεύον κλειδί του πίνακα. Για κάθε νέα εγγραφή, δημιουργείται και ένα καινούργιο που είναι η τιμή του κλειδιού της προηγούμενης εγγραφής αυξημένη κατά ένα. Χρησιμοποιήθηκε ο τύπος δεδομένων smallint(4), ο οποίος υποστηρίζει έως 65535 εγγραφές.

name

Το όνομα που αποδίδεται στο τμήμα. Είναι κατά κανόνα αλφαριθμητικό, οπότε επιλέχτηκε ο τύπος δεδομένων varchar(100). Οι 100 χαρακτήρες είναι υπεραρκετοί για να καλύψουν το μέγεθος πολυσύνθετων ονομάτων.

shortname

Το υποκοριστικό του τμήματος. Χρησιμοποιήθηκε ο τύπος δεδομένων varchar(5).

website

Ο ιστότοπος του τμήματος, ο οποίος μπορεί να παραλειφθεί για κάποια

εγγραφή. Επιλέχτηκε ο τύπος δεδομένων varchar(2083), γιατί απαιτούνται το πολύ 2083 χαρακτήρες για τον υπερσύνδεσμο ενός ιστότοπου.

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
id	tinyint(3)	όχι	κανένα
first_name	varchar(30)	όχι	κανένα
last_name	varchar(30)	όχι	κανένα

Πίνακας 1. 2: Πίνακας Professors

2.5.2 Professors

Ο πίνακας Professors κρατάει αποθηκευμένα τα βασικά στοιχεία των διδασκόντων του Πανεπιστημίου.

id

Είναι το πρωτεύον κλειδί του πίνακα και ενημερώνεται αυτόματα για κάθε νέα εγγραφή. Επιλέχτηκε ο τύπος tinyint(3) που υποστηρίζει μέχρι και 255 διδάσκοντες.

first_name

Το όνομα του διδάσκοντα. Είναι κατά κανόνα αλφαριθμητικό, οπότε επιλέχτηκε ο τύπος δεδομένων varchar(30) που καλύπτει ένα τυπικό όνομα.

last_name

Το επίθετο του διδάσκοντα. Είναι επίσης αλφαριθμητικό, έτσι επιλέχτηκε παρομοίως ο τύπος δεδομένων varchar(30).

2.5.3 Subjects

Στον πίνακα Subjects αποθηκεύονται οι πληροφορίες για τα μαθήματα του κάθε τμήματος. Εδώ, σε αντίθεση με τον πίνακα Departments, αποθηκεύεται ο χρήστης που δημιουργεί μια νέα εγγραφή.

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
id	int(7)	όχι	κανένα
instructor_id	int(11)	όχι	κανένα
department_id	smallint(4)	όχι	κανένα
description	varchar(100)	όχι	κανένα
subject_code	varchar(25)	όχι	κανένα
instructor	varchar(60)	ναι	κενό
semester	tinyint(2)	όχι	κανένα
lastmodificationdate	datetime	όχι	κανένα

Πίνακας 1. 3: Πίνακας Subjects

id

Είναι το πρωτεύον κλειδί του πίνακα. Για κάθε νέα εγγραφή δημιουργείται και ένα καινούριο, που είναι η τιμή του κλειδιού της προηγούμενης εγγραφής αυξημένη κατά ένα. Έτσι χρησιμοποιήθηκε ο τύπος δεδομένων int(7).

instructor_id

Είναι το id του καθηγητή που προσθέτει την εγγραφή και είναι ίδιου τύπου με το πεδίο id του πίνακα Users, δηλαδή int(11).

department_id

Αποτελεί δευτερεύον κλειδί που συνδέεται με το πεδίο id του πίνακα Departments. Αποθηκεύει το id του τμήματος στο οποίο ανήκει το μάθημα. Χρησιμοποιήθηκε έτσι ο ίδιος τύπος με το πεδίο αυτό, smallint(4).

description

Το όνομα του τμήματος. Αποθηκεύεται σε μορφή varchar. Το μήκος του πεδίου ορίστηκε σε 100, υπεραρκετό για να καλύψει ένα πολυσύνθετο όνομα

subject_code

Αποθηκεύει τον κωδικό μαθήματος, όπως αυτός ορίζεται από το

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
Id	smallint(5)	όχι	κανένα
description	varchar(75)	όχι	κανένα

Πίνακας 1. 4: Πίνακας Categories

τμήμα. Το πεδίο είναι κατά κανόνα αλφαριθμητικό, όποτε επιλέγεται τύπος varchar με ένα ικανοποιητικό μέγεθος για όλες τις περιπτώσεις. Το μήκος 25 χαρακτήρων θεωρείται αρκετό.

instructor

Το ονοματεπώνυμο του καθηγητή που διδάσκει το μάθημα. Αποθηκεύεται σε μορφή varchar με μήκος πεδίου ίσο με 60.

semester

Το εξάμηνο στο οποίο διδάσκεται το μάθημα. Το πεδίο είναι κατά κανόνα αριθμητικό με πεδίο τιμών 1-10, όποτε επιλέγεται τύπος tinyint(2).

lastmodificationdate

Η ημερομηνία τελευταίας τροποποίησης του μαθήματος. Επιλέχτηκε ο τύπος datetime, ο οποίος είναι της μορφής "χρόνος – μήνας – μέρα ώρα : λεπτά : δευτερόλεπτα".

2.5.4 Categories

Ο πίνακας Categories χρησιμοποιείται για την αποθήκευση των κατηγοριών που προσθέτονται μόνο από τον καθολικό διαχειριστή και είναι ίδιες για όλα τα μαθήματα. Η κατηγορία ενός μαθήματος μπορεί να είναι "Εργαστήριο 1, Θεωρία 1" κ.τ.λ. και χαρακτηρίζεται από τα παρακάτω πεδία.

id

Είναι το πρωτεύον κλειδί του πίνακα. Χρησιμοποιήθηκε ο τύπος δεδομένων smallint με μήκος 5 που υποστηρίζει 65.535 εγγραφές.

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
id	smallint(5)	όχι	κανένα
instructor_id	int(11)	όχι	κανένα
subject_id	int(7)	όχι	κανένα
description	varchar(75)	όχι	κανένα
category_id	smallint(5)	όχι	κανένα

Πίνακας 1. 5: Πίνακας Topics

description

Το όνομα της κατηγορίας, το οποίο αποθηκεύεται σε μορφή varchar(75).

2.5.5 Topics

Ο πίνακας Topics χρησιμοποιείται για την αποθήκευση των ενοτήτων του κάθε μαθήματος, οι οποίες ανήκουν σε μια συγκεκριμένη κατηγορία.

id

Είναι το πρωτεύον κλειδί του πίνακα, το οποίο αυξάνεται αυτόματα για κάθε νέα εγγραφή. Χρησιμοποιήθηκε ο τύπος δεδομένων smallint(5) που υποστηρίζει μέχρι και την τιμή 65.535.

instructor_id

Κρατάει αποθηκευμένο το id του καθηγητή που προσθέτει την εγγραφή και είναι ίδιου τύπου με το πεδίο id του πίνακα Users, δηλαδή int(11).

subject_id

Αποτελεί δευτερεύον κλειδί που συνδέεται με το πρωτεύον κλειδί id του πίνακα Subjects. Έτσι ο τύπος που επιλέχτηκε είναι ο int(7).

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
Id	bigint(19)	όχι	κανένα
description	varchar(255)	όχι	κανένα
displaytype	varchar(30)	όχι	κανένα

Πίνακας 1. 6: Πίνακας Blobtypes

description

Η περιγραφή ή όνομα της ενότητας. Αποθηκεύεται σε μορφή varchar(75).

category_id

Αποθηκεύει το id της κατηγορίας στην οποία ανήκει η ενότητα. Επομένως είναι δευτερεύον κλειδί και έχει τύπο smallint(5).

2.5.6 Blobtypes

Ο πίνακας Blobtypes έχει αποθηκευμένα τα βασικά χαρακτηριστικά των επισυναπτόμενων σε μια ερώτηση ή διαγώνισμα.

id

Είναι το πρωτεύον κλειδί του πίνακα. Για νέο επισυναπτόμενο που εισάγεται, δημιουργείται και ένα μοναδικό id. Χρησιμοποιήθηκε ο τύπος δεδομένων bigint(19) που υποστηρίζει περισσότερες από δεκαοχτώ πεντάκις εκατομμύρια εγγραφές.

description

Αποθηκεύει το όνομα του επισυναπτόμενου και είναι τύπου varchar(255), καθώς είναι κατεξοχήν αλφαριθμητικό.

displaytype

Είναι ο τύπος του επισυναπτόμενου, δηλαδή μπορεί να παίρνει τιμές "image/jpeg, application/pdf" κ.τ.λ. Έχει την μορφή varchar(30).

2.5.7 Questions

Ο πίνακας Questions χρησιμοποιείται για την αποθήκευση των ερωτήσεων ενός μαθήματος. Η ερώτηση ενός μαθήματος μπορεί να χαρακτηριστεί από τα παρακάτω στοιχεία.

id

Είναι το πρωτεύον κλειδί του πίνακα. Για κάθε εγγραφή που εισάγεται, δημιουργείται και ένα καινούριο, που είναι η τιμή του κλειδιού της προηγούμενης εγγραφής, αυξημένη κατά ένα. Χρησιμοποιήθηκε ο τύπος δεδομένων bigint(19) για να υποστηρίξει τις ανάγκες μιας τυπικής σχολής με πολλές ερωτήσεις.

instructor_id

Περιέχει το id του διδάσκοντα που προσθέτει την εγγραφή και όπως ήταν αναμενόμενο είναι τύπου int(11) όπως και στον πίνακα Users.

topic_id

Είναι δευτερεύον κλειδί που συνδέεται με το id του πίνακα Topics. Αποθηκεύει την ενότητα στην οποία ανήκει η ερώτηση. Γι' αυτό ορίζεται ως smallint(5).

question

Σε αυτό το πεδίο βρίσκεται αποθηκευμένη η ερώτηση. Είναι κατά κανόνα αλφαριθμητικό, οπότε έχει τύπο varchar(512), χαρακτήρες αρκετοί για την περιγραφή μιας ερώτησης.

questionblob

Έχει αποθηκευμένο το επισυναπτόμενο της ερώτησης (αν υπάρχει) σε μορφή `mediumblob` που είναι κατεξοχήν `binary` και έχει μέγιστο μέγεθος 16 MB (megabytes).

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
id	bigint(19)	όχι	κανένα
instructor_id	int(11)	όχι	κανένα
topic_id	smallint(5)	όχι	κανένα
question	varchar(512)	όχι	κανένα
questionblob	mediumblob	ναι	κενό
questionblobtype_id	bigint(19)	ναι	κενό
questionblobsize	int(7)	ναι	κενό
type	enum('πολλαπλής επιλογής', 'ελεύθερο κείμενο')	όχι	κανένα
choice1	varchar(250)	ναι	κενό
choice1blob	mediumblob	ναι	κενό
choice1blobtype_id	bigint(19)	ναι	κενό
choice1blobsize	int(7)	ναι	κενό
choice2	varchar(250)	ναι	κενό
choice2blob	mediumblob	ναι	κενό
choice2blobtype_id	bigint(19)	ναι	κενό
choice2blobsize	int(7)	ναι	κενό
choice3	varchar(250)	ναι	κενό
choice3blob	mediumblob	ναι	κενό
choice3blobtype_id	bigint(19)	ναι	κενό
choice3blobsize	int(7)	ναι	κενό
choice4	varchar(250)	ναι	κενό
choice4blob	mediumblob	ναι	κενό
choice4blobtype_id	bigint(19)	ναι	κενό
choice4blobsize	int(7)	ναι	κενό
choice5	varchar(250)	ναι	κενό
choice5blob	mediumblob	ναι	κενό
choice5blobtype_id	bigint(19)	ναι	κενό
choice5blobsize	int(7)	ναι	κενό
answer	tinyint(1)	ναι	κενό
available	tinyint(1)	ναι	1
difficulty	enum('1','2','3','4','5')	όχι	κανένα
language	enum('GR','ENG')	ναι	GR
lastmodificationdate	datetime	όχι	κανένα

Πίνακας 1. 7: Πίνακας Questions

Questionblobtype_id

Έχει αποθηκευμένο το id του πίνακα Blobtypes και έτσι αποτελεί δευτερεύον κλειδί. Έχει τύπο bigint(19), όπως και το id του πίνακα Blobtypes.

questionblobsize

Περιέχει το μέγεθος του επισυναπτόμενου της ερώτησης σε Bytes και είναι τύπου int(7) που έχει μέγιστη τιμή πάνω από 4 δισεκατομμύρια.

type

Περιέχει τον τύπο της ερώτησης ο οποίος μπορεί να πάρει 2 τιμές. Είτε "πολλαπλής επιλογής" ή "ελεύθερο κείμενο". Η MySQL αποθηκεύει αυτού του είδους πεδία, που παίρνουν συγκεκριμένες τιμές, ως enum με πεδίο ορισμού αυτές τις τιμές.

choice 1 – choice 5

Αποθηκεύονται οι 5 επιλογές της ερώτησης. Η κάθε επιλογή είναι αλφαριθμητικό κείμενο html και επιπλέον μπορεί να περιέχει μια εικόνα με τη μορφή . Αποθηκεύεται σε τύπο varchar(250).

choice1blob – choice5blob

Έχει αποθηκευμένα το επισυναπτόμενα των επιλογών 1 έως 5 (αν υπάρχουν) σε μορφή mediumblob που έχει μέγιστο μέγεθος 16 MB (megabytes).

choice1blobtype_id – choice5blobtype_id

Αποθηκεύονται τα id των επισυναπτόμενων (αν υπάρχουν) των επιλογών 1 έως 5. Είναι δευτερεύοντα κλειδιά με τύπο bigint(19), όπως το πεδίο id του πίνακα Blobtypes.

choice1blobsize – choice5blobsize

Έχει αποθηκευμένα τα μεγέθη των επισυναπτόμενων των επιλογών 1 έως 5 και είναι τύπου int(7).

answer

Αποθηκεύεται ο αριθμός της σωστής απάντησης. Η σωστή απάντηση όπως θα εξηγήσουμε και αργότερα αποθηκεύεται πάντα στο πεδίο choice 1, οπότε το πεδίο answer έχει πάντα την τιμή 1.

available

Περιέχει μια "σημαία (flag)", δηλαδή μια τιμή μήκους 1^{ος} ψηφίου που δηλώνει αν η ερώτηση είναι διαθέσιμη για να περιληφθεί σε ένα τεστ ή αν είναι προσωρινά απενεργοποιημένη. Έτσι είναι τύπου tinyint(1) και όπως θα δούμε στη συνέχεια μπορεί να πάρει 2 τιμές, 0 ή 1.

difficulty

Είναι ένας αριθμός τύπου enum με πεδίο τιμών 1-5 που δηλώνει το επίπεδο δυσκολίας της ερώτησης.

language

Εδώ αποθηκεύεται η γλώσσα της ερώτησης. Αυτή μπορεί να πάρει μόνο δύο τιμές ανάλογα με το αν είναι στα ελληνικά ή τα αγγλικά. Ο τύπος της είναι enum('GR','ENG').

lastmodificationdate

Η ημερομηνία τελευταίας τροποποίησης της ερώτησης. Είναι αποθηκευμένη με τη μορφή datetime.

2.5.8 Tests

Στον πίνακα Tests αποθηκεύονται τα χαρακτηριστικά των διαγωνισμάτων ενός μαθήματος. Αυτά ακολουθούν παρακάτω.

id

Είναι το πρωτεύον κλειδί του πίνακα. Για κάθε νέο διαγώνισμα που εισάγεται, το κλειδί ενημερώνεται αυτόματα. Έχει τον τύπο bigint(12) που υποστηρίζει πάνω από δεκαοχτώ πεντάκις εκατομμύρια εγγραφές.

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
Id	bigint(12)	όχι	κανένα
instructor_id	int(11)	όχι	κανένα
subject_id	int(7)	όχι	κανένα
category_id	smallint(5)	όχι	κανένα
title	varchar(45)	όχι	κανένα
description	varchar(740)	ναι	κενό
start_date	datetime	όχι	κανένα
finish_date	datetime	όχι	κανένα
negative_mark	varchar(4)	ναι	κενό
duration	time	όχι	κανένα
difficulties	varchar(10)	όχι	κανένα
total_questions	tinyint(3)	όχι	κανένα
public	tinyint(1)	ναι	0
grade_visible	tinyint(1)	ναι	1
attachmentblob	mediumblob	ναι	κενό
attachmentblobtype_id	bigint(19)	ναι	κενό
attachmentblobsize	int(7)	ναι	κενό

Πίνακας 1. 8: Πίνακας Tests

instructor_id

Περιέχει το id του διαχειριστή-διδάσκοντα που προσθέτει την εξέταση και είναι τύπου int(11).

subject_id

Είναι δευτερεύον κλειδί που συνδέεται με το id του πίνακα Subjects. Αποθηκεύει το μάθημα στο οποίο ανήκει το διαγώνισμα. Γι' αυτό ορίζεται ως int(7).

category_id

Περιέχει το id της κατηγορίας (πίνακας Categories) στην οποία ανήκει το διαγώνισμα. Αποτελεί επίσης δευτερεύον κλειδί και είναι τύπου smallint(5).

title

Σε αυτό το πεδίο αποθηκεύεται ο τίτλος του διαγωνίσματος. Χρησιμοποιήθηκε ο τύπος varchar(45), δηλαδή 45 αλφαριθμητικοί χαρακτήρες.

description

Σε αυτό το πεδίο αποθηκεύεται μια σύντομη περιγραφή του διαγωνίσματος, η οποία είναι προαιρετική. Χρησιμοποιήθηκε ο τύπος varchar(740), ο οποίος καλύπτει μια σύντομη και κατανοητή περιγραφή.

start_date

Είναι η ημερομηνία έναρξης της διαθεσιμότητας του διαγωνίσματος και αποθηκεύεται ως τύπος datetime.

finish_date

Παρομοίως σε μορφή "χρόνος – μήνας – μέρα ώρα : λεπτά : δευτερόλεπτα" αποθηκεύεται και η ημερομηνία λήξης της διαθεσιμότητας του διαγωνίσματος. Έτσι επιλέχτηκε ο τύπος datetime.

negative_mark

Αποθηκεύεται η αρνητική βαθμολογία (αν υπάρχει) ανά ερώτηση. Αυτός είναι ένα ποσοστό επί τοις 100 και είναι πραγματικός αριθμός. Επειδή όμως σε έναν πραγματικό αριθμό μπορεί να χρησιμοποιηθεί και το κόμμα αλλά και η τελεία ως στοιχείο διαχωρισμού, ανάλογα με την γλώσσα γραφής, χρησιμοποιήθηκε ο τύπος varchar(4) αντί για float που δεν δέχεται το ελληνικό κόμμα.

duration

Αποτελεί τον διαθέσιμο χρόνο εξέτασης και ως εκ τούτου αποθηκεύεται με την μορφή "ώρα – λεπτά – δευτερόλεπτα". Επομένως επιλέχτηκε ο τύπος datetime.

difficulties

Εδώ αποθηκεύεται ένα string (ακολουθία χαρακτήρων) με τα επίπεδα δυσκολίας που μπορούν να έχουν οι ερωτήσεις του διαγωνίσματος. Καθώς τα επίπεδα είναι 5, μαζί με τον χαρακτήρα διαχωρισμού το πεδίο μπορεί να φτάσει τους 10 χαρακτήρες. Έτσι χρησιμοποιήθηκε ο τύπος varchar(10).

total_questions

Περιέχει τον αριθμό των συνολικών ερωτήσεων του διαγωνίσματος. Οι συνολικές ερωτήσεις μπορεί να είναι τριψήφιος αριθμός, οπότε επιλέχτηκε ο τύπος tinyint(3).

public

Έχει αποθηκευμένη μια "σημαία (flag)" με τιμές 0 ή 1 που δηλώνουν αν το διαγώνισμα είναι διαθέσιμο στους ελεύθερους χρήστες ή όχι αντίστοιχα. Είναι τύπου tinyint(1).

grade_visible

Έχει αποθηκευμένη μια σημαία, δηλαδή έναν αριθμό (1 ή 0) που δηλώνει αν ο τελικός βαθμός του τέστ θα είναι ορατός στον φοιτητή. Γι' αυτό έχει τύπο tinyint(1).

attachmentblob

Έχει αποθηκευμένο το επισυναπτόμενο του διαγωνίσματος (αν υπάρχει) σε μορφή mediumblob που είναι κατεξοχήν binary και έχει μέγιστο μέγεθος 16 MB (megabytes).

attachmentblobtype_id

Κρατάει αποθηκευμένο το id του πίνακα Blobtypes και έτσι αποτελεί δευτερεύον κλειδί, που δηλώνει τον τύπο του επισυναπτόμενου. Έχει τύπο bigint(19).

attachmentblobsize

Περιέχει το μέγεθος του επισυναπτόμενου του διαγωνίσματος σε

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
id	bigint(19)	όχι	κανένα
test_id	bigint(12)	όχι	κανένα
topic_id	smallint(5)	όχι	κανένα
percentage	tinyint(3)	όχι	κανένα

Πίνακας 1. 9: Πίνακας Topicpercentage

Bytes και είναι τύπου int(7) που έχει μέγιστη τιμή πάνω από 1 εκατομμύριο.

2.5.9 Topicpercentage

Αυτός ο πίνακας περιέχει αποθηκευμένα τα χαρακτηριστικά της μίξης ποσοστών ενοτήτων των ερωτήσεων ενός διαγωνίσματος. Για παράδειγμα ένα διαγώνισμα μπορεί να περιέχει ερωτήσεις 20% από την ενότητα 1 και 80% από την ενότητα 2. Αυτές οι πληροφορίες αποθηκεύονται εδώ.

id

Είναι το πρωτεύον κλειδί του πίνακα. Έχει τον τύπο bigint(19), διότι πρέπει να υποστηρίζει περισσότερες εγγραφές απ' ότι ο πίνακας Tests, αφού για κάθε διαγώνισμα μπορούν να υπάρξουν πολλαπλές εγγραφές.

test_id

Είναι δευτερεύον κλειδί και αποτελεί κοινό πεδίο με το id του πίνακα Tests. Δηλώνει το διαγώνισμα το οποίο περιέχει ένα συγκεκριμένο ποσοστό μιας ενότητας. Είναι τύπου bigint(12).

topic_id

Αποτελεί επίσης δευτερεύον κλειδί αφού συνδέεται με το id του πίνακα Topics. Δηλώνει την ενότητα στην οποία ανήκει το ποσοστό. Είναι τύπου smallint(5).

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
Id	bigint(19)	όχι	κανένα
instructor_id	int(11)	όχι	κανένα
test_id	bigint(12)	όχι	κανένα
batchnumber	mediumint(6)	όχι	κανένα
randomhash	varchar(6)	όχι	κανένα
Used	tinyint(1)	ναι	0
lastmodificationdate	datetime	όχι	κανένα

Πίνακας 1. 10: Πίνακας Tokens

percentage

Είναι το ποσοστό της ενότητας που δηλώνει το προηγούμενο πεδίο, εκφρασμένο επί τοις 100. Η μεγαλύτερη τιμή που μπορεί να δοθεί σε αυτό το πεδίο είναι 100, οπότε επιλέχτηκε ο τύπος tinyint(3).

2.5.10 Tokens

Οι κωδικοί ανώνυμης περιήγησης αποθηκεύονται στον Πίνακα Tokens. Τα χαρακτηριστικά τους αναλύονται παρακάτω.

id

Είναι το πρωτεύον κλειδί του πίνακα. Για κάθε εγγραφή που εισάγεται, δημιουργείται και ένα καινούριο id, που είναι η τιμή του κλειδιού της προηγούμενης εγγραφής, αυξημένη κατά ένα. Χρησιμοποιήθηκε ο τύπος δεδομένων bigint(19) για να υποστηρίξει τις ανάγκες μιας τυπικής σχολής με πολλούς κωδικούς.

instructor_id

Σε αυτό το πεδίο αποθηκεύεται το id του διδάσκοντα που προσθέτει το token και όπως ήταν αναμενόμενο είναι τύπου int(11) όπως και στον πίνακα Users.

test_id

Είναι δευτερεύον κλειδί που συνδέεται με το πεδίο id του πίνακα Tests. Αποθηκεύει το id του διαγωνίσματος στο οποίο ανήκει το κάθε token και είναι τύπου bigint(12), όπως και στον πίνακα Tests.

batchnumber

Εδώ αποθηκεύεται ο αριθμός σειράς ή αλλιώς ομάδας έκδοσης του token. Είναι αριθμητικό πεδίο τύπου mediumint(6), διότι αποτελείται από 6 τυχαία αριθμητικά ψηφία.

randomhash

Το randomhash που αποθηκεύεται σε αυτό το πεδίο είναι 6 ψηφία (αριθμοί και χαρακτήρες) και παράγεται με sha256(random()), από το οποίο κρατάμε τους 6 πρώτους χαρακτήρες. Γι' αυτό είναι τύπου varchar(6).

used

Εδώ αποθηκεύεται μια σημαία, η οποία αν είναι 1 τότε το token έχει χρησιμοποιηθεί, διαφορετικά αν είναι 0 σημαίνει ότι είναι αχρησιμοποίητο. Ορίζεται ως τύπου tinyint(1).

lastmodificationdate

Η ημερομηνία τελευταίας τροποποίησης (όπως θα δούμε παρακάτω τα tokens γίνεται μόνο να προστεθούν και να διαγραφούν) του token. Είναι αποθηκευμένη με τη μορφή datetime.

2.5.11 Testsequences

Σε αυτόν τον πίνακα κρατούνται αποθηκευμένα τα χαρακτηριστικά όλων των απαντήσεων για κάθε εξέταση, εκτός απ' αυτές που πραγματοποιούνται από ελεύθερους χρήστες.

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
id	bigint(19)	όχι	κανένα
ip	varchar(45)	όχι	κανένα
student_id	int(11)	ναι	κενό
token_id	bigint(19)	ναι	κενό
test_id	bigint(12)	όχι	κανένα
start_point	datetime	όχι	κανένα
page	tinyint(3)	όχι	κανένα
seed	bigint(20)	όχι	κανένα
skip_ids	varchar(2048)	όχι	κανένα
rem_questions	varchar(2048)	όχι	κανένα
loop_ids	varchar(2048)	όχι	κανένα
date_created	datetime	όχι	κανένα
date_completed	datetime	όχι	κανένα
question_sequence	varchar(2048)	όχι	κανένα
answer_sequence	mediumtext	όχι	κανένα
grade	float	ναι	0

Πίνακας 1. 11: Πίνακας Testsequences

id

Είναι το πρωτεύον κλειδί του πίνακα. Έχει τον τύπο bigint(19) για να υποστηρίξει πολλαπλές απαντήσεις για κάθε εξέταση.

ip

Αποτελεί του ip του χρήστη που υπέβαλε το διαγώνισμα. Είναι τύπου varchar(45), που είναι αρκετά μεγάλο για να υποστηρίξει διευθύνσεις IP του πρωτοκόλλου IPv6.

student_id

Σε αυτό το πεδίο αποθηκεύεται το id του χρήστη που υπέβαλε το διαγώνισμα, εάν δε χρησιμοποίησε token αλλά εισήλθε στο σύστημα με όνομα χρήστη και συνθηματικό. Είναι τύπου int(11) όπως και στον

πίνακα Users.

token_id

Αποτελεί δευτερεύον κλειδί καθώς συνδέεται με το id του πίνακα Tokens. Δηλώνει το token (αν έχει χρησιμοποιηθεί) με το οποίο εισήλθε ο χρήστης στην εξέταση. Είναι τύπου bigint(19).

test_id

Είναι επίσης δευτερεύον κλειδί και αποτελεί κοινό πεδίο με το id του πίνακα Tests. Δηλώνει το διαγώνισμα το οποίο υποβλήθηκε. Είναι τύπου bigint(12).

start_point

Είναι το σημείο εκκίνησης της εξέτασης. Επί της ουσίας, αποτελεί την ημερομηνία έναρξης της εξέτασης με την διαφορά ότι μπορεί να αλλάξει κατάλληλα, αν ο διδάσκοντας επιλέξει να προσθέσει λεπτά ή να λήξει τον διαθέσιμο χρόνο υποβολής. Χρησιμοποιείται δηλαδή για να εξυπηρετήσει τον σκοπό τροποποίησης του χρονικού περιθωρίου της εξέτασης. Έχει τύπο datetime.

page

Αυτό το πεδίο κρατάει τον αριθμό σελίδας στην οποία βρίσκεται ο χρήστης καθώς εξετάζεται και ουσιαστικά χρησιμοποιείται σε περίπτωση που ο χρήστης αναγκαστεί να αλλάξει υπολογιστή σε κάποιο σημείο της εξέτασης, ώστε να μπορέσει να συνεχίσει από το ίδιο.

seed

Είναι ένας τυχαίος αριθμός που, όπως θα δούμε στη συνέχεια, παράγεται κατά την εκτέλεση του κώδικα στη διάρκεια της εξέτασης και βοηθάει στο να παραχθεί μία συγκεκριμένη σειρά τυχαίων ερωτήσεων, που επαναλαμβάνεται καθώς ο χρήστης προχωράει σε επόμενες ερωτήσεις. Αυτό το πεδίο είναι και πάλι αναγκαίο μόνο σε περίπτωση που γίνει αλλαγή υπολογιστή, ώστε να μη χαθεί η ακολουθία των τυχαίων ερωτήσεων.

skip_ids

Με το ίδιο σκεπτικό, εδώ αποθηκεύονται τα id των ερωτήσεων που ο χρήστης έχει παραλείψει, και οι οποίες εμφανίζονται ξανά αφού έχουν εμφανιστεί όλες οι ερωτήσεις.

rem_questions

Εδώ αποθηκεύεται η ακολουθία των id των ερωτήσεων που απομένουν να εμφανιστούν στην εξέταση. Κάθε φορά που απαντάται μια ερώτηση, η λίστα μειώνεται κατά ένα id. Ενώ θα μπορούσε απλώς να κρατηθεί σε session, αυτή η τιμή χρησιμοποιείται και πάλι για τη περίπτωση που γίνει αλλαγή υπολογιστή.

loop_ids

Και πάλι για την ίδια περίπτωση αποθηκεύουμε τη λίστα των id των ερωτήσεων που ο χρήστης παρέλειψε, όπως ακριβώς στο πεδίο skip_ids με τη διαφορά ότι αν το τρέχον πεδίο δεν είναι κενό, ο εξεταζόμενος έχει ολοκληρώσει τον πρώτο βρόγχο ερωτήσεων και βρίσκεται σε επόμενο.

date_created

Η ημερομηνία κατά την οποία δημιουργήθηκε το συγκεκριμένο διαγώνισμα. (όπως αναφέρεται παρακάτω κάθε διαγώνισμα είναι μια τυχαία μίξη των ίδιων ερωτήσεων με διαφορετική σειρά). Είναι αποθηκευμένη με τη μορφή datetime.

date_completed

Εδώ αποθηκεύεται η ημερομηνία κατά την οποία υποβάλλεται το διαγώνισμα. Είναι κι' αυτή, όπως ήταν αναμενόμενο, τύπου datetime.

question_sequence

Είναι ένα string που περιέχει τα id των ερωτήσεων που επιλέχτηκαν, με τη σειρά που επιλέχτηκαν. Είναι τύπου varchar(2048) για να υποστηρίξει μεγάλα και πολλαπλά id μαζί με το διαχωριστικό τους που είναι το κόμμα.

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
Id	bigint(20)	όχι	κανένα
test_id	bigint(12)	όχι	κανένα
question_id	bigint(19)	όχι	κανένα
times_selected	bigint(12)	όχι	κανένα
answered_correctly	bigint(12)	όχι	κανένα
answered_wrong	bigint(12)	όχι	κανένα
non_answered	bigint(12)	όχι	κανένα

Πίνακας 1. 12: Πίνακας Qualityresults

answer_sequence

Εδώ αποθηκεύεται επίσης ένα string που περιέχει τα id των ερωτήσεων που απαντήθηκαν, με τη σειρά που απαντήθηκαν, μαζί με την εκάστοτε απάντηση (σύμφωνα με τη σειρά που εμφανίστηκαν οι επιλογές στον φοιτητή), την σωστή απάντηση και την θέση της απάντησης που δόθηκε στον πίνακα Questions. Τα στοιχεία αυτά διαχωρίζονται μεταξύ τους με τους χαρακτήρες ">" και "(")" αντίστοιχα. Είναι τύπου mediumtext ώστε να υποστηρίζει το εκτενές και πολύπλοκο string με στοιχείο διαχωρισμού το κόμμα.

grade

Σε αυτό το πεδίο είναι αποθηκευμένος ο τελικός βαθμός της εξέτασης. Επιλέχτηκε ο τύπος float επειδή είναι πραγματικός αριθμός.

2.5.12 Qualityresults

Αυτός ο πίνακας περιέχει στατιστικά στοιχεία που αφορούν τις ερωτήσεις και τα αποτελέσματα μιας εξέτασης. Με τη βοήθεια αυτών των στοιχείων παράγονται τα γραφήματα του συστήματος.

id

Είναι το πρωτεύον κλειδί του πίνακα. Για κάθε εγγραφή που εισάγεται, αυτό αυξάνεται αυτομάτως. Χρησιμοποιήθηκε ο τύπος δεδομένων bigint(20) που υποστηρίζει υπεραρκετές εξετάσεις.

test_id

Είναι δευτερεύον κλειδί που συνδέεται με το πεδίο id του πίνακα Tests. Αποθηκεύει το id του διαγωνίσματος στο οποίο ανήκει η ερώτηση της οποίας τα στατιστικά στοιχεία αποθηκεύουμε. Είναι τύπου bigint(12).

question_id

Σε αυτό το πεδίο αποθηκεύεται το id του της ερώτησης της οποίας τα στατιστικά στοιχεία προσθέτονται στον πίνακα. Είναι επομένως δευτερεύον κλειδί, τύπου bigint(19).

times_selected

Εδώ με τον τύπο bigint(12) αποθηκεύεται ένας αριθμός που δηλώνει πόσες φορές επιλέχτηκε η ερώτηση στο συγκεκριμένο διαγώνισμα. Επιλέχτηκε αυτός ο τύπος καθώς καλύπτει τεράστιους αριθμούς.

answered_correctly

Αριθμός που δηλώνει πόσες φορές απαντήθηκε σωστά η ερώτηση και είναι αποθηκευμένος με τη μορφή bigint(12).

answered_wrong

Το πεδίο αυτό περιέχει έναν αριθμό που δηλώνει πόσες φορές απαντήθηκε λάθος η ερώτηση στο συγκεκριμένο διαγώνισμα. Είναι κι αυτό τύπου bigint(12).

non_answered

Αντίστοιχα είναι ένας αριθμός που δείχνει πόσες φορές έμεινε αναπάντητη η ερώτηση και είναι τύπου bigint(12).

2.5.13 Questionfeedback

Σε αυτόν τον πίνακα αποθηκεύονται τα μηνύματα των φοιτητών που

Πεδίο	Τύπος	Κενό	Προκαθορισμένο
Id	bigint(19)	όχι	κανένα
test_id	bigint(12)	όχι	κανένα
question_id	bigint(19)	όχι	κανένα
user_id	int(11)	ναι	κενό
token_id	bigint(19)	ναι	κενό
message	varchar(255)	όχι	κανένα

Πίνακας 1. 13: Πίνακας Questionfeedback

αποστέλλονται κατά τη διάρκεια μιας εξέτασης και αφορούν μια ερώτηση του διαγωνίσματος (για παράδειγμα ότι δεν είναι καλά διατυπωμένη ή κάποια απορία πάνω στην ερώτηση).

id

Είναι το πρωτεύον κλειδί του πίνακα και αυξάνεται κατά ένα για κάθε νέα εγγραφή. Ο τύπος δεδομένων bigint(19) υποστηρίζει υπεραρκετά μηνύματα.

test_id

Είναι δευτερεύον κλειδί που συνδέεται με το πεδίο id του πίνακα Tests. Αποθηκεύει το id του διαγωνίσματος στο οποίο ανήκει η ερώτηση την οποία αφορά το μήνυμα. Είναι τύπου bigint(12).

question_id

Σε αυτό το πεδίο αποθηκεύεται το id του της ερώτησης την οποία αφορά το μήνυμα. Είναι δευτερεύον κλειδί, τύπου bigint(19).

user_id

Εδώ αποθηκεύεται το id του φοιτητή που έστειλε το μήνυμα, εάν δε χρησιμοποίησε token αλλά εισήλθε στο σύστημα με όνομα χρήστη και συνθηματικό, ή το id του διδάσκοντα που έστειλε ένα μήνυμα-απάντηση. Είναι τύπου int(11).

token_id

Αντίστοιχα αυτό το πεδίο περιέχει το id του token, εάν έχει χρησιμοποιηθεί. Είναι τύπου bigint(19) και αποτελεί δευτερεύον κλειδί.

message

Το πεδίο αυτό έχει αποθηκευμένο το ίδιο το μήνυμα που αποστέλλει ο φοιτητής προς τον διδάσκοντα. Είναι τύπου varchar(255) ώστε να υπάρχει το περιθώριο για αρκετές λέξεις.

Εδώ έχει ολοκληρωθεί η παρουσίαση της αρχιτεκτονικής του ιστοχώρου και έγινε η ανάλυση της βάσης δεδομένων. Ήρθε η ώρα λοιπόν να συνεχίσουμε με την υλοποίηση του ιστοτόπου, δηλαδή θα περιγράψουμε τα βασικότερα σημεία του κώδικα που γράφτηκε ώστε να δημιουργηθούν οι ιστοσελίδες του συστήματος.

Κεφάλαιο 3

Υλοποίηση του ιστοχώρου

Γνωρίζοντας πλέον τις απαιτήσεις του ιστόχωρου, στο τρέχον κεφάλαιο γίνεται η υλοποίησή του, παρουσιάζεται ο κώδικάς του και αναλύονται οι τεχνικές php, javascript και ajax που χρησιμοποιήθηκαν στα βασικότερα σημεία του. Θα αναφερθούν τα βασικότερα σημεία του προγραμματισμού των ιστοσελίδων και θα επεξηγηθεί ο σκοπός τους. Έτσι μετέπειτα θα ακολουθήσει η παρουσίαση του ιστοχώρου και των λειτουργιών που προσφέρει στον διαχειριστή και τον χρήστη. Αρχικά, δίνουμε τις οδηγίες εγκατάστασης του ιστοτόπου σε ένα νέο σύστημα.

3.1 Οδηγίες εγκατάστασης

Για να εγκατασταθεί και να λειτουργήσει ο ιστότοπος στον διακομιστή που θα τον "στεγάζει", είναι απαραίτητο ο τελευταίος να έχει εγκατεστημένο κάποιο πακέτο που περιέχει τον Apache Server, τη βάση δεδομένων MySQL και την PHP. Το πιο διαδεδομένο πακέτο αυτού του είδους είναι το XAMPP που είναι διαθέσιμο για τα λειτουργικά Windows, Linux, Solaris και Mac OS X. Το πακέτο αυτό διανέμεται δωρεάν στον ιστότοπο <https://www.apachefriends.org/download.html>. Είτε χρησιμοποιηθεί αυτό το πακέτο ή κάποιο άλλο, θα πρέπει να είναι σίγουρο ότι περιέχεται η PHP 5. Καλό είναι να εγκατασταθεί η έκδοση 1.7.3 (ή νεότερη) του XAMPP, όπως έγινε στη παρούσα περίπτωση. Οδηγίες εγκατάστασης του εκάστοτε πακέτου μπορούν να βρεθούν εύκολα μέσω οποιασδήποτε μηχανής αναζήτησης. Αυτό που θα πρέπει να προσεχθεί κατά την εγκατάστασή του όμως, είναι να συμπεριληφθούν σε αυτή τα χαρακτηριστικά Apache, MySQL και phpMyAdmin. Αφού ολοκληρωθεί η διαδικασία, πρέπει να εκκινήσουμε τις υπηρεσίες Apache και MySQL μέσω του πίνακα ελέγχου του XAMPP.

Επόμενο βήμα είναι η ρύθμιση της MySQL μέσω του εργαλείου phpMyAdmin. Μπορούμε να μεταφερθούμε στη σελίδα αυτού αν πατήσουμε το κουμπί "Admin" δίπλα απ' την υπηρεσία MySQL στον πίνακα ελέγχου του XAMPP. Εκεί θα δούμε το εργαλείο διαχείρισης των

βάσεων δεδομένων. Πρώτα απ' όλα, πρέπει να πάμε στη καρτέλα "προνόμια" και στην επεξεργασία αυτών να ορίσουμε το όνομα χρήστη και τον κωδικό πρόσβασης της MySQL. Το σύστημα εφόσον είναι τοπικό (δηλαδή το XAMPP εγκαταστάθηκε στον υπολογιστή που βρισκόμαστε και όλα τα αρχεία του ιστοτόπου πρόκειται να περαστούν σε αυτόν), είναι εξ ορισμού "localhost". Προσοχή: αυτός που εγκαθιστά τον ιστοτόπο πρέπει να θυμάται αυτά τα στοιχεία, διότι θα χρειαστούν αργότερα. Εν συνεχεία στην αρχική σελίδα της phpMyAdmin πηγαίνουμε στο πεδίο "Δημιουργία νέας βάσης" και πληκτρολογούμε το όνομα της βάσης δεδομένων, το οποίο στη περίπτωση μας είναι "uexam". Το πεδίο MySQL connection collation θα πρέπει να είναι επιλεγμένο ως "utf8_general_ci". Άλλο ένα βήμα το οποίο είναι απαραίτητο είναι να σιγουρευτούμε ότι η επέκταση PDO της PHP είναι εγκατεστημένη. Αυτό το κάνουμε βλέποντας το αρχείο php.ini που βρίσκεται στον φάκελο C:\xampp\php (σε Windows) και ελέγχοντας ότι υπάρχει η γραμμή "extension=php_pdo.dll". Αν όχι, τότε πρέπει να την προσθέσουμε.

Μετά απ' όλα αυτά είμαστε έτοιμοι να περάσουμε τα αρχεία μας στο σύστημα. Συγκεκριμένα, ο φάκελος που περιέχει όλα τα αρχεία (i-exams) θα πρέπει να τοποθετηθεί στον κατάλογο C:\xampp\htdocs. Αν ο υπολογιστής έχει λειτουργικό Linux, ο τρόπος διαφέρει λίγο. Σε αυτή την περίπτωση ανοίγουμε το τερματικό και εκτελούμε την γραμμή "mkdir ~/public_html_" για να δημιουργήσουμε τον φάκελο public_html και στη συνέχεια εκτελούμε "sudo ln -s ~/public_html /opt/lampp/htdocs/\$USER_" για να συνδέσουμε τον παραπάνω φάκελο με τον κατάλογο htdocs του XAMPP. Έπειτα μπορούμε να μεταφέρουμε όλα τα αρχεία του ιστοτόπου (χωρίς τον αρχικό φάκελο i-exams) στον φάκελο public_html. Αυτό δεν είναι αρκετό, καθώς πρέπει να δώσουμε τα κατάλληλα δικαιώματα στα αρχεία. Τα ελάχιστα είναι r (read) στο others για όλα τα απλά αρχεία και x (execute) στο others για όλους τους φακέλους. Επιπλέον οι φάκελοι grimg και uploads πρέπει να έχουν και r και w (write) στο others. Ο ιστοτόπος μας είναι πλέον προσβάσιμος μέσω του URL <http://localhost/i-exams/> αν βρισκόμαστε σε Windows ή <http://localhost/username/> αν το λειτουργικό σύστημα είναι Linux, όπου *username* είναι το επιθυμητό όνομα χρήστη του καταλόγου (το \$USER που χρησιμοποιήθηκε στην παραπάνω εντολή). Όμως δεν είναι έτοιμος για χρήση ακόμα.

Τελευταίο βήμα είναι η αρχικοποίηση του ιστοτόπου και της βάσης δεδομένων. Αυτό γίνεται αν μεταφερθούμε στο URL <http://localhost/i-exams/install>, μέσω του περιηγητή Ιστού που χρησιμοποιούμε. Αμέσως

θα δούμε στην οθόνη την σελίδα εγκατάστασης, όπου αρκεί να ακολουθήσουμε τις οδηγίες. Τα πεδία που φαίνονται θα πρέπει να συμπληρωθούν σύμφωνα με τα στοιχεία ρύθμισης της MySQL που δώσαμε προηγουμένως. Προσοχή: στα συστήματα Unix ενδέχεται να χρησιμοποιείται unix socket για την σύνδεση στο localhost. Το path του (διεύθυνση καταλόγου) μπορεί να βρεθεί απ' το αρχείο my.cnf. Σε αυτή τη περίπτωση πρέπει να επιλεχθεί το checkbox "MySQL Socket" και να συμπληρωθεί το αντίστοιχο πεδίο με το παραπάνω path, το οποίο θα έχει μορφή παρόμοια με "/var/run/mysqld/mysqld.sock". Πατώντας "Εγκατάσταση" δημιουργούνται οι πίνακες στη βάση δεδομένων. Έπειτα καλό είναι να διαγραφεί ο φάκελος C:\xampp\htdocs\i-exams\install\ για λόγους ασφαλείας. Τέλος, πρέπει να τροποποιήσουμε το αρχείο C:\xampp\htdocs\i-exams\ibase\connect.php, όπου \$dbhost = "σύστημα", \$dbase = "όνομα βάσης", \$dbuser = "όνομα χρήστη" και \$dbpass = "κωδικός πρόσβασης" σύμφωνα με τα στοιχεία που συμπληρώσαμε παραπάνω. Στη περίπτωση χρήσης unix socket η γραμμή "\$dbconn= new PDO("mysql:host=\$dbhost;" θα πρέπει να αντικατασταθεί με "\$dbconn= new PDO("mysql: unix_socket=\$dbhost;", όπου το \$dbhost θα έχει οριστεί ίσο με το path του unix socket.

3.2 Αρχική σελίδα

Η αρχική σελίδα του ιστοτόπου είναι κοινή για τον διαχειριστή και τον χρήστη. Όταν ένας πελάτης HTTP (γενικώς ένας περιηγητής ιστού) ζητά μια διεύθυνση URL που παραπέμπει σε μια δομή καταλόγου αντί για μία πραγματική ιστοσελίδα μέσα στον κατάλογο, ο εξυπηρετητής ιστού θα προσφέρει μια γενική σελίδα, η οποία συχνά αναφέρεται ως κύρια σελίδα ή σελίδα "δείκτης" (index).

Το παραδοσιακό όνομα για μια τέτοια σελίδα είναι index.html, αλλά οι πιο σύγχρονοι διακομιστές HTTP προσφέρουν μία ρυθμιζόμενη λίστα ονομάτων των αρχείων που ο διακομιστής μπορεί να χρησιμοποιήσει ως δείκτη. Εάν ο διακομιστής έχει ρυθμιστεί ώστε να υποστηρίζει εκτέλεση σεναρίων από τη πλευρά του διακομιστή (server-side scripting), η λίστα περιλαμβάνει συνήθως εγγραφές που επιτρέπουν δυναμικό περιεχόμενο να χρησιμοποιηθεί ως αρχική σελίδα (π.χ. index.php). Αυτή η περίπτωση ισχύει στη παρούσα εργασία, οπότε και η κύρια σελίδα του ιστοχώρου είναι η index.php. Παρακάτω θα αναλύσουμε αυτό το αρχείο, καθώς και το login_check.php στο οποίο γίνεται ο έλεγχος για το αν είναι συνδεδεμένος ή όχι ο χρήστης.

3.2.1 Index.php και login_check.php

```
1. <?php session_start(); ob_start();?>
2. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
3. <html xmlns="http://www.w3.org/1999/xhtml" lang="gr" xml:lang="gr">
4. <head>
5.
6.     <!-- Meta Data -->
7.     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8.     <meta name="author" content="Βασιλούδης Παντελής"/>
9.     <meta name="description" content="Διαδικτυακή Πλατφόρμα Εξέτασης
   Φοιτητών με την υποστήριξη απεριόριστων μαθημάτων, ερωτήσεων και τεστ,
   καθώς και την άμεση παραγωγή εγγράφων με τις απαντήσεις." />
10.    <meta name="keywords" content="Διαδικτυακό, Σύστημα,
   Πλατφόρμα, Εξέταση, απεριόριστα, ερωτήσεις, μαθήματα, απαντήσεις, λύσεις,
   άμεσα" />
11.
12.    <!-- Page Title -->
13.    <title>Διαδικτυακή Πλατφόρμα Εξέτασης Φοιτητών</title>
14.
15.    <!-- Link to Style External Sheet -->
16.    <link rel="stylesheet" type="text/css" href="css/style.css" me-
   dia="screen" />
17.
18. </head>
```

Κώδικας 1. 1: Index.php – αρχικοποίηση

Αρχικά στο αρχείο index.php ξεκινάει μια νέα συνεδρία (session). Μια συνεδρία είναι ένας τρόπος να αποθηκεύονται πληροφορίες (σε μεταβλητές) που θα χρησιμοποιηθούν σε πολλαπλές σελίδες. Σε αυτή τη περίπτωση θέλουμε να κρατάμε αποθηκευμένο το όνομα χρήστη και το συνθηματικό, ώστε να γίνεται ένας έλεγχος σε κάθε σελίδα για την ιδιότητα του συνδεδεμένου χρήστη και αναλόγως να παραμένει συνδεδεμένος στις σελίδες τις οποίες είναι εξουσιοδοτημένος να δει. Μια συνεδρία ξεκινά με τη λειτουργία session_start() και πάντα στην πρώτη γραμμή, πριν από οποιοδήποτε άλλο στοιχείο που τυπώνεται στη σελίδα. Οι μεταβλητές συνεδρίας ορίζονται με την παγκόσμια μεταβλητή της PHP: \$_SESSION. Αμέσως μετά είναι απαραίτητο να χρησιμοποιήσουμε την συνάρτηση ob_start(), διότι όπως φαίνεται αργότερα υπάρχει ένας βρόγχος ανακατεύθυνσης που οδηγεί στο αρχείο login_check, όπου ορίζονται τα cookies. Επί της ουσίας στέλνεται

μια κεφαλίδα (header) στον περιηγητή, η οποία οδηγεί σ' αυτό το αρχείο. Το πρόβλημα όμως που παρουσιάζεται είναι ότι τα cookies πρέπει πάντα να ορίζονται πριν σταλθεί οποιοδήποτε header στη σελίδα. Μια τεχνική για να αποφευχθεί αυτό είναι να χρησιμοποιήσουμε τη συνάρτηση `ob_start()`, η οποία δημιουργεί μια παύση κρατώντας στη μνήμη του διακομιστή οτιδήποτε δημιουργείται χωρίς να το εμφανίζει ή εκτελεί, μέχρις ότου να σταλθεί το header. Ουσιαστικά με αυτόν τον τρόπο αποθηκεύεται ο κώδικας του `login_check.php`, όπου δημιουργούνται τα cookies και έπειτα γίνεται πραγματικά η ανακατεύθυνση, οπότε και εκτελείται ο κώδικας.

Στη συνέχεια γίνεται η δήλωση τύπου εγγράφου, ή DOCTYPE. Είναι μια ετικέτα που συνδέει ένα συγκεκριμένο XML έγγραφο (για παράδειγμα, μια ιστοσελίδα) με έναν ορισμό τύπου εγγράφου (DTD) (για παράδειγμα, τον τυπικό ορισμό μιας συγκεκριμένης έκδοσης της HTML ή XHTML). Στον παρόν ιστότοπο χρησιμοποιήθηκε η XHTML 1.0 Transitional, όπως φαίνεται στη 2^η γραμμή (Κώδικας 1. 1). Η δήλωση αυτή πρέπει να είναι το πρώτο πράγμα που υπάρχει σε ένα κείμενο με HTML στοιχεία, πριν την ετικέτα `<html>` (και μετά την έναρξη του session).

Εν συνεχεία, δηλώνονται τα μεταδεδομένα. Αυτά είναι δεδομένα (πληροφορίες) σχετικά με άλλα δεδομένα. Μια ετικέτα `<meta>` παρέχει μεταδεδομένα σχετικά με το έγγραφο HTML (XHTML στην περίπτωση μας). Τα μεταδεδομένα δεν εμφανίζονται στην σελίδα, αλλά αναλύονται από τη μηχανή. Τα στοιχεία αυτά χρησιμοποιούνται συνήθως για να καθορίσουν την περιγραφή της σελίδας, λέξεις-κλειδιά, τον συντάκτη του εγγράφου, το σετ χαρακτήρων και άλλα μεταδεδομένα. Τα μεταδεδομένα μπορούν να χρησιμοποιηθούν από τα προγράμματα περιήγησης (πώς να εμφανίσουν το περιεχόμενο ή επαναφόρτωση της σελίδας), τις μηχανές αναζήτησης (λέξεις-κλειδιά) ή άλλες υπηρεσίες ιστού. Να σημειωθεί ότι το σετ χαρακτήρων πρέπει να είναι UTF-8 και σε συνδυασμό με αυτό η κωδικοποίηση κάθε αρχείου HTML, PHP και Javascript θα πρέπει να είναι UTF-8 χωρίς BOM, για να εμφανίζονται σωστά τα ελληνικά. Παρακάτω δηλώνεται ο τίτλος της σελίδας, ο οποίος φαίνεται στην καρτέλα του προγράμματος περιήγησης. Επίσης, υπάρχει ένας σύνδεσμος προς εξωτερικό αρχείο του συστήματος, τύπου φύλλων στυλ, με όνομα `style.css` που δίνει την εμφάνιση σε όλες τις σελίδες του συστήματος. Όλα αυτά τα στοιχεία περικλείονται μέσα στην ετικέτα `<head>`.

```

1. if(isset($_COOKIE['cookname']) && isset($_COOKIE['cookpass']) && !filter_input(INPUT_COOKIE, 'cookname', FILTER_SANITIZE_STRING) === false && !filter_input(INPUT_COOKIE, 'cookpass', FILTER_SANITIZE_STRING) === false)
2. {
3.     $_SESSION['username'] = $_COOKIE['cookname'];
4.     $_SESSION['password'] = $_COOKIE['cookpass'];
5.     //redirect to home page
6.     header("location: login_check.php");
7.     exit();

```

Κώδικας 1. 2: Index.php – έλεγχος cookies

Γίνεται ένας έλεγχος για το αν είναι ενεργά τα cookies "cookname" και "cookpass" τα οποία αποθηκεύουν το όνομα χρήστη και το συνθηματικό σε περίπτωση που ο χρήστης τσεκάρει το κουτάκι "Remember me", όπως θα δούμε λίγο παρακάτω. Τα cookies είναι μικρά αρχεία κειμένου τα οποία αποθηκεύονται στον φυλλομετρητή μας κατά την πλοήγησή μας στο διαδίκτυο. Σκοπός τους είναι να ειδοποιούν τον ιστότοπο που επισκέπτεται ο χρήστης, για την προηγούμενη δραστηριότητά του. Συνήθως περιγράφουν στοιχεία μας όπως όνομα χρήστη (username) και συνθηματικό πρόσβασης (password) με σκοπό κατά την επίσκεψή μας στον ίδιο ιστότοπο αργότερα, να μας "θυμάται" και να μην χρειάζεται να συνδεθούμε και πάλι. Αν λοιπόν τα συγκεκριμένα cookies είναι ενεργά, τότε αποθηκεύουμε την τιμή τους στα αντίστοιχα sessions. Ο λόγος που αποθηκεύουμε τις τιμές των cookies σε sessions είναι ότι τα τελευταία αποθηκεύονται στον διακομιστή, σε αντίθεση με τα πρώτα που αποθηκεύονται στη μηχανή του πελάτη, καθιστώντας τα ασφαλέστερα. Οι πραγματικές τιμές αποκρύπτονται από τον πελάτη, και μπορούμε να ελέγξουμε πότε τα δεδομένα θα λήξουν και δε θα ισχύουν πλέον. Αν ήταν όλα βασισμένα σε cookies, ένας χρήστης θα μπορούσε να τροποποιήσει τα δεδομένα των cookies του και στη συνέχεια να στείλει διάφορα αιτήματα στον ιστότοπο. Για περαιτέρω ασφάλεια, εκτός από τον έλεγχο που γίνεται για το αν είναι κενά τα cookies, περνάμε επίσης από ένα φίλτρο της PHP (filter_input) που θα εξηγήσουμε στην αμέσως επόμενη παράγραφο.

```

1. <form id="login" name="login" method="post" action="login_check.php">
2. <label for="username">Όνομα χρήστη (username):</label>
3. <input id="username" name="username" class="text" value=""/> <br></br>
4. <label for="password">Συνθηματικό (password):</label>
5. <input id="password" name="password" type="password" class="text" value=""/><br></br>
6. <input type="checkbox" name="remember"/> Remember Me
7. <div class="sep"></div>

```

```

8. <button type="submit" class="ok">Είσοδος</button>
9. <!-- get the value of error message that is given in login validation and return
it respectively -->
10. <?php
11.     if( isset($_SESSION['MSG_ARR']) && is_array($_SESSION['MSG_ARR'])
&& count($_SESSION['MSG_ARR']) >0 )
12.     {
13.         foreach($_SESSION['MSG_ARR'] as $msg)
14.         {
15.             if ($msg == 'Σφάλμα Σύνδεσης. Πρέπει να εισάγετε το Όνομα χρήστη
και το Συνθηματικό.')
16.             {
17.                 echo '<div class="n_warning">';
18.                 echo '<p>',$msg,</p>';
19.                 echo '</div>';
20.                 unset($_SESSION['MSG_ARR']);
21.             }
22.         }
23.     }
24.     ?>
25. </form>

```

Κώδικας 1. 3: Index.php – δημιουργία φόρμας

Εν συνεχεία όπως φαίνεται παραπάνω, δημιουργούμε την φόρμα όπου ο επισκέπτης του ιστοχώρου εισάγει τα στοιχεία του. Δίνουμε την δυνατότητα στον χρήστη να παραμείνει συνδεδεμένος ακόμα κι αν κλείσει τον φυλλομετρητή, μέσω ενός checkbox με όνομα "remember". Έπειτα, όπως θα δούμε αμέσως μετά, αν τα στοιχεία είναι εσφαλμένα τότε αποστέλλεται ένα μήνυμα σφάλματος από το login_check.php στο αρχείο index.php. Στο τελευταίο (Κώδικας 1. 3, γραμμή 11) ελέγχουμε αν είναι ενεργό το session που περιέχει το μήνυμα, το προβάλλουμε με την κατάλληλη μορφοποίηση και απενεργοποιούμε το session. Με άλλα λόγια αν τα στοιχεία δεν είναι σωστά, ο χρήστης ανακατευθύνεται και πάλι στην ίδια σελίδα index, όπου εμφανίζεται το μήνυμα.

Τα δεδομένα της φόρμας αποστέλλονται στο αρχείο login_check.php όπου γίνεται ο κατάλληλος έλεγχος.

```

1. //connect to database
2. include 'dbase/connect.php';
3.
4. //set the timezone to Athens
5. date_default_timezone_set('Europe/Athens');

```

```

6.
7. if (isset($_SESSION['username']) && isset($_SESSION['password']))
8. {
9.     $username = $_SESSION['username'];
10.    $password = $_SESSION['password'];
11. }
12. else
13. {
14.    //get the user input values from the index login form
15.    $username = filter_input(INPUT_POST, 'username', FIL-
    TER_SANITIZE_STRING);
16.    $password = filter_input(INPUT_POST, 'password', FIL-
    TER_SANITIZE_STRING);
17. }
18.
19. if($username == "" || $password == "")
20. {
21.    $msg_arr[] = 'Σφάλμα Σύνδεσης. Πρέπει να εισάγετε το Όνομα
    χρήστη και το Συνθηματικό.';
22.    $msgflag = true;
23. }

```

Κώδικας 1. 4: Login_check.php – έλεγχος εισόδου

Αρχικά γίνεται Include το αρχείο connect.php, στο οποίο ορίζονται οι παράμετροι για την σύνδεση με την MySQL και την βάση δεδομένων uexam. Οι παράμετροι αυτοί διαφέρουν ανάλογα με τον Server στον οποίο "στεγάζεται" ο ιστόχωρος και το username και password που χρησιμοποιούνται για την σύνδεση με τη MySQL. Ο τυπικός κώδικας στην περίπτωση που χρησιμοποιείται το XAMPP στον ίδιο υπολογιστή που βρίσκονται όλα τα αρχεία του συστήματος φαίνεται παρακάτω. Δημιουργούμε λοιπόν ένα νέο αντικείμενο PDO με τις παρακάτω παραμέτρους και το αποθηκεύουμε στη μεταβλητή \$dbconn. Αξιοσημείωτο είναι ότι θέτουμε το σετ χαρακτήρων ίσο με utf8 για να αποθηκεύονται σωστά τα ελληνικά γράμματα.

```

1. // Connection data (host, database, username, password)
2. $dbhost = 'localhost';
3. $dbbase = 'uexam';
4. $dbuser = 'root';
5. $dbpass = 'pass';
6.
7. try
8. {

```

```

9.     $dbconn= new PDO("mysql:host=$dbhost;dbname=$dbase", $dbuser,
$dbpass);
10.    $dbconn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
11.
12.    // Set UTF-8 encoding
13.    $dbconn->exec("set names utf8");
14.    }
15.
16.    catch (PDOException $e)
17.    {
18.        echo 'Connection to the database failed:' . $e->getMessage();
19.    }

```

Κώδικας 1. 5: Connect.php

Πίσω στο login_check.php, ελέγχονται λοιπόν τα sessions και σε περίπτωση που είναι ήδη ενεργά (πράγμα που σημαίνει ότι ο χρήστης συνδέθηκε προηγουμένως στη σελίδα επιλέγοντας την λειτουργία "Remember me" και έχοντας κλείσει τη σελίδα δίχως να αποσυνδεθεί, ανακατευθύνθηκε απευθείας στο αρχείο login_check.php την επόμενη φορά που την επισκέφτηκε) αποθηκεύονται οι τιμές τους στις ανάλογες μεταβλητές, όπως φαίνεται παραπάνω. Διαφορετικά διαβάζονται οι τιμές των δεδομένων που εισήγαγε ο χρήστης, αφού περάσουν από ένα φίλτρο που ελέγχει αν προέρχονται όντως από τη μέθοδο POST και αν είναι τύπου string. Αν είναι κενές, τότε αποθηκεύεται σε έναν πίνακα το μήνυμα σφάλματος και θέτουμε την αντίστοιχη σημαία ίση με true (αληθή).

```

1. else
2. {
3.    //select user login values
4.    $query = $dbconn->prepare("SELECT type, username, password FROM users WHERE BINARY username= :uname AND BINARY password= :passw");
5.    $query->bindParam(':uname', $username);
6.    $query->bindParam(':passw', $password);
7.    $query->execute();
8.    $rows = $query->fetch(PDO::FETCH_NUM);
9.    if($rows > 0)
10.    {
11.        //pass the login values to Session
12.        $_SESSION['username'] = $rows[1];
13.        $_SESSION['password'] = $rows[2];
14.        //get the login date and time

```

```

15.     $_SESSION['login_datetime'] = date('Y-m-d | H:i');
16.     //set the remember me cookie
17.     if(isset($_POST['remember']))
18.     {
19.         setcookie("cookname", $_SESSION['username'], time()+60*60*24,
"/");
20.         setcookie("cookpass", $_SESSION['password'], time()+60*60*24, "/");
21.     }
22.     if ($rows[0] == 1)
23.     {
24.         //go to admin home page
25.         header("location: admin.php");
26.         exit();
27.     }
28.     else
29.     {
30.         //get the login date and time
31.         header("location: student.php");
32.         exit();
33.     }
34.     }
35.     else
36.     {
37.         $msg_arr[] = 'Σφάλμα Σύνδεσης. Πρέπει να εισάγετε το Όνομα
χρήστη και το Συνθηματικό.';
38.         $msgflag = true;
39.     }
40.     }
41.
42.     if($msgflag)
43.     {
44.         $_SESSION['MSG_ARR'] = $msg_arr;
45.         session_write_close();
46.         $dbconn = null;
47.         //redirect to home page
48.         header("location: index.php");
49.         exit();
50.     }

```

Κώδικας 1. 6: Login_check.php – δημιουργία cookies

Διαφορετικά αναζητούνται στη βάση δεδομένων τα χαρακτηριστικά του χρήστη που αντιστοιχεί στα δεδομένα που εισήγαγε. Αν γίνει αντιστοίχιση και υπάρχει όντως κάποιος χρήστης με αυτά τα στοιχεία (username και password), τότε αποθηκεύονται τα χαρακτηριστικά του

σε μεταβλητές session. Αν επιπλέον είχε τσεκαριστεί το κουτάκι "Remember me", τότε τα sessions αυτά αποθηκεύονται σε cookies (τα οποία όπως είδαμε ελέγχονται στο index.php). Τα συγκεκριμένα cookies ορίζονται να έχουν διάρκεια ισχύος 1 μέρα (60 δευτερόλεπτα * 60 λεπτά * 24 ώρες), έκτος αν ο χρήστης αποσυνδεθεί, όπως θα δούμε στη συνέχεια. Έπειτα αν το πεδίο type του πίνακα Users στη βάση δεδομένων, το οποίο φανερώνει τον τύπο του χρήστη, είναι ίσο με 1, τότε ο χρήστης είναι διαχειριστής και ανακατευθύνεται στην κεντρική σελίδα του. Διαφορετικά είναι απλός χρήστης (δηλαδή φοιτητής) και ανακατευθύνεται στην δική του αρχική ιστοσελίδα.

Αν δε γίνει αντιστοίχιση με κάποιον υπαρκτό χρήστη, αυτό σημαίνει ότι τα στοιχεία που έδωσε ο επισκέπτης του ιστοτόπου ήταν λανθασμένα, οπότε και πάλι αποθηκεύεται ένα μήνυμα λάθους στον πίνακα msg_arr[] και ορίζεται η μεταβλητή msgflag ως true. Αν λοιπόν είναι ορισμένη αυτή η μεταβλητή, πράγμα που πρακτικά σημαίνει ότι έχει γίνει κάποιο λάθος κατά τη διαδικασία σύνδεσης (είτε δεν εισήχθηκαν δεδομένα ή ήταν εσφαλμένα), τότε αποθηκεύεται το ειδοποιητικό μήνυμα σε ένα session, τερματίζεται η σύνδεση με τη βάση δεδομένων και ο χρήστης ανακατευθύνεται πίσω στην αρχική σελίδα (index.php) όπου τυπώνεται το μήνυμα, με τον τρόπο που είδαμε παραπάνω.

3.2.2 Τεχνικές ασφάλειας

Όπως είδαμε στη προηγούμενη παράγραφο, σε ορισμένα σημεία του κώδικα χρησιμοποιήσαμε την συνάρτηση filter_input [20]. Αυτή παίρνει μια μεταβλητή διαβάζοντας τον τύπο της και την φιλτράρει με διάφορες μεθόδους "απολύμανσης" (sanitization). Το φιλτράρισμα μαζί με την επικύρωση δεδομένων, είναι τρόποι για να αποφύγουμε την αποδοχή επιβλαβών δεδομένων από τον χρήστη.

Όταν οι χρήστες αποστέλλουν τα δεδομένα, θα πρέπει να βεβαιωθούμε ότι έχουν δώσει κάτι που περιμέναμε. Για παράδειγμα, θα πρέπει να βεβαιωθούμε ότι τα δεδομένα ενός αριθμητικού πεδίου είναι όντως αριθμός ή ότι τα δεδομένα ενός αλφαριθμητικού πεδίου είναι τύπου string. Αυτός ο έλεγχος είναι κυρίως απαραίτητος κατά τη σύνδεση του χρήστη καθώς πρέπει να γίνεται πιστοποίηση ενός φυσικού προσώπου που δε μπορούμε ποτέ να εμπιστευτούμε. Όπως θα δούμε αργότερα σε αυτό το κεφάλαιο, χρειάζεται επίσης, να δοθεί προσοχή στα δεδομένα που καταχωρεί ο φοιτητής κατά την διάρκεια

της εξέτασης, ιδιαίτερα στις απαντήσεις ελεύθερου κειμένου και στα μηνύματα που αποστέλλει προς τον καθηγητή. Θα πρέπει να σιγουρευτούμε ότι δεν υπάρχει τίποτα απροσδόκητο σε αυτούς τους τομείς. Κυρίως, θα πρέπει να ελέγξουμε ότι τα πεδία που δεν πρέπει να έχουν οποιοδήποτε περιεχόμενο HTML δεν περιέχουν πραγματικά HTML. Αυτό το πετυχαίνουμε με το φίλτρο FILTER_SANITIZE_SPECIAL_CHARS που καθαρίζει το κείμενο που εισάγει ο χρήστης από τους ειδικούς χαρακτήρες της HTML.

Το να βεβαιωθούμε ότι έχουν υποβληθεί δεδομένα που είναι απαλλαγμένα από απροσδόκητο περιεχόμενο είναι μόνο το μισό κομμάτι της ασφάλειας. Θα πρέπει επίσης, να τα επικυρώσουμε, δηλαδή να σιγουρευτούμε ότι περιέχουν τιμές με τις οποίες μπορούμε πραγματικά να εργαστούμε. Για παράδειγμα αν αναμένουμε έναν αριθμό μεταξύ 1 και 10, θα πρέπει να ελέγξουμε αυτή την τιμή. Με το ίδιο σκεπτικό μια είσοδος ημερομηνίας θα πρέπει να βρίσκετε στην επιτρεπτή περιοχή τιμών, όπως είδαμε στις περιπτώσεις χρήσης του κεφαλαίου 2. Το ίδιο ισχύει και για όλους τους άλλους ελέγχους των φορμών. Υπάρχει upload αρχείου; Θα πρέπει να βεβαιωθούμε ότι έχει έναν αναμενόμενο τύπο και το κατάλληλο μέγεθος.

Μια ακόμη τεχνική ασφάλειας, η οποία αποτελεί την πιο αποτελεσματική μορφή "εξυγίανσης" για την αποτροπή της SQL injection είναι η παραμετροποίηση χρησιμοποιώντας PDO [21]. Χρησιμοποιώντας παραμετροποιημένα ερωτήματα, το ερώτημα είναι διαχωρισμένο από τα δεδομένα, έτσι ώστε να απομακρυνθεί η απειλή της SQL injection πρώτης τάξης. Η SQL injection είναι μια τεχνική όπου οι κακόβουλοι χρήστες μπορούν να κάνουν "ένεση" (injection) εντολές SQL σε μια πρόταση SQL, μέσω μιας εισόδου (input) της ιστοσελίδας. Οι εντολές SQL που έχουν προέλθει από ένεση μπορούν να αλλάξουν την δήλωση SQL και να θέσουν σε κίνδυνο την ασφάλεια της εφαρμογής. Όμως με τη σωστή εφαρμογή της PDO, χρησιμοποιώντας έτοιμες δηλώσεις (prepared statements), αντί για τα ίδια τα δεδομένα χρησιμοποιούνται σύμβολα "κράτησης θέσης" αυτών, οπότε δεν μπορούν να συμπεριληφθούν επιπλέον πληροφορίες στη δήλωση. Μια τυπική δήλωση SQL με χρήση PDO έχει την παρακάτω σύνταξη.

```
1. $current_datetime = date('Y-m-d H:i:s');
2.
3. $test = $_GET['tid'];
4.
5. //select the test only if it's still available
6. $query = $dbconn->prepare("SELECT * FROM tests WHERE id= :tid AND
start_date<=:cdate AND finish_date>=:cdate");
```

```
7. $query->bindParam(':tid', $test);
8. $query->bindParam(':cdate', $current_datetime);
9. $query->execute();
10. $rows = $query->fetch(PDO::FETCH_ASSOC);
```

Κώδικας 1. 7: Παράδειγμα χρήσης PDO

Αρχικά χρησιμοποιείται η μέθοδος `prepare`, η οποία ετοιμάζει την δήλωση για εκτέλεση και επιστρέφει ένα αντικείμενο της δήλωσης. Η τελευταία περιέχει σύμβολα "κράτησης θέσης" των μεταβλητών (placeholders), όπως είπαμε. Εν συνεχεία με την μέθοδο `bindParam` συνδέουμε τα σύμβολα αυτά με τις πραγματικές μεταβλητές και τελικά εκτελούμε την δήλωση.

Μετά από αυτή την μικρή παρένθεση, συνεχίζουμε με την επεξήγηση των βασικότερων σημείων του κώδικα των σελίδων διαχείρισης, οι οποίες αποτελούν το βασικότερο κομμάτι της εργασίας και αυτό που δημιουργήθηκε πρώτο.

3.3 Σελίδες διαχείρισης – ονοματολογία

Πριν αναλύσουμε τον κώδικα των αρχείων του τίτλου, είναι πρέπει να αναφέρουμε εν συντομία την ονοματολογία των αρχείων που δημιουργήσαμε. Κάθε σελίδα που προσφέρει και μια λειτουργία είναι γραμμένη σε ξεχωριστό αρχείο. Τα αρχεία που αφορούν τις σελίδες του διαχειριστή ξεκινάνε με τον χαρακτήρα "a" (admin). Οι σελίδες επισκόπησης των αντικειμένων τελειώνουν με ένα "v" (view). Οι σελίδες που αφορούν την επεξεργασία των αντικειμένων έχουν ένα "e" (edit) στο τέλος, ενώ αν είναι σελίδα προεπισκόπησης προηγούνται οι χαρακτήρες "pr" (preview). Όσες σελίδες χρησιμοποιούν Javascript για εμφάνιση ενός ή περισσότερων drop-down menu καλούν ένα αρχείο που ξεκινά με τη λέξη "display" (εμφάνισε). Με το ίδιο σκεπτικό όπου χρησιμοποιείτε Ajax για λήψη ή και εμφάνιση δεδομένων σε πραγματικό χρόνο (π.χ. κάθε ένα δευτερόλεπτο) καλείται ένα αρχείο που έχει μπροστά τη λέξη "get" (λάβε). Τα αρχεία που περιέχουν τον κώδικα που εκτελεί την προσθήκη ενός αντικειμένου ξεκινούν με "add" (πρόσθεσε), όσα εκτελούν διαγραφή "delete" (διέγραψε), ενώ για αφαίρεση στοιχείου (π.χ. επισυναπτόμενου) ξεκινούν με "remove" (αφαίρεσε). Παρομοίως όσα αρχεία ξεκινούν με τη λέξη "update" (ανανέωσε), αφορούν τον κώδικα που επεξεργάζεται και ανανεώνει ένα ήδη υπάρχον αντικείμενο. Παρακάτω ξεκινά η ανάλυση του κώδικα του

αρχείου admin.php, στο οποίο δημιουργείται η αρχική σελίδα του διαχειριστή. Έπειτα θα συνεχίσουμε με τα αρχεία που καλούνται απ' το πρώτο.

3.3.1 Admin.php και admin_session_check.php

Στη κεντρική σελίδα του διαχειριστή το πρώτο πράγμα που διαβάζει ο φυλλομετρητής είναι ένα εξωτερικό αρχείο, το admin_session_check.php. Αυτό το αρχείο είναι το πρώτο πράγμα που περιλαμβάνεται σε όλες τις σελίδες του διαχειριστή και εκεί γίνεται ο παρακάτω έλεγχος για το αν οι τιμές των sessions που αποθηκεύτηκαν στο αρχείο login_check.php αντιστοιχούν όντως σε διαχειριστή και όχι σε οποιονδήποτε άλλο χρήστη.

```
1. //select user login values
2. $query = $dbconn->prepare("SELECT type, aem, first_name, last_name, id
FROM users WHERE BINARY username= :uname AND BINARY password=
:passw");
3. $query->bindParam(':uname', $_SESSION['username']);
4. $query->bindParam(':passw', $_SESSION['password']);
5. $query->execute();
6. $rows = $query->fetch(PDO::FETCH_NUM);
7.
8. if ($rows[0] == 1)
9. {
10.     //pass the login values to Session
11.     $_SESSION['type'] = $rows[0];
12.     $_SESSION['aem'] = $rows[1];
13.     $_SESSION['first_name'] = $rows[2];
14.     $_SESSION['last_name'] = $rows[3];
15.     $_SESSION['user_id'] = $rows[4];
16.     $dbconn = null;
17. }
18. else
19. {
20.     //redirect to student home page
21.     header("location: student.php");
```

Κώδικας 1. 8: Admin.php – έλεγχος τιμών sessions

Διαβάζονται λοιπόν τα χαρακτηριστικά του χρήστη από την βάση δεδομένων σύμφωνα με τις τιμές του username και password που

έχουν τα sessions και αν ο τύπος έχει τιμή ίση με 1, τότε ο χρήστης παραμένει στη σελίδα του διαχειριστή και αποθηκεύονται σε session ο τύπος του, το AEM του, το ονοματεπώνυμό του και το id του. Αλλιώς (ο τύπος του είναι ίσος με 0) είναι ένας φοιτητής που προσπαθεί να προσπελάσει τη σελίδα, οπότε δε του επιτρέπεται η πρόσβαση και ανακατευθύνεται αμέσως στη δική του κεντρική σελίδα.

Έπειτα κάνουμε include(περιλαμβάνουμε) τον κώδικα που εμφανίζει την κεφαλίδα του ιστοτόπου, με το αρχείο admin_header.php, το οποίο περιλαμβάνεται σε όλες τις σελίδες του διαχειριστή. Στην κεφαλίδα αναγράφεται το ονοματεπώνυμο του χρήστη, καθώς και η ιδιότητά του (καθολικός ή απλός διαχειριστής/ καθηγητής), ανάλογα με την τιμή που έχει η σταθερά \$_SESSION['type'] που ορίστηκε προηγουμένως.

```
1. <div class="left">
2.     <p>Χρήστης:
3.     <strong>
4.     <!-- display the user name -->
5.     <?php
6.         echo $_SESSION['first_name']." ".$_SESSION['last_name'];
7.         if ($_SESSION['type']==1 && $_SESSION['aem']==0)
8.             { echo " (Καθολικός διαχειριστής); "; }
9.         else if ($_SESSION['type']==1 && $_SESSION['aem']!=0)
10.            { echo " (Καθηγητής); "; }
11.     ?>
12.     </strong>
13.     <!-- logout with warning message -->
14.     [ <a href="logout.php" title="Logout" onclick="return confirm('Τα
    δεδομένα τα οποία έχετε συμπληρώσει σε φόρμες θα χαθούν σε περίπτωση
    που ΔΕΝ επιβεβαιώσατε την προσθήκη τους. Είστε σίγουροι ότι θέλετε να
    αποσυνδεθείτε;')">Αποσύνδεση</a> ]
15.     </p>
16. </div>
17. <div class="right">
18.     <div class="align-right">
19.     <p>Τελευταία είσοδος:
20.     <strong>
21.     <!-- display the login date time -->
22.     <?php
23.         echo $_SESSION['login_datetime'];
24.     ?>
25.     </strong>
26.     </p>
```

Κώδικας 1. 9: Admin_header.php

Επίσης, δίπλα από αυτά τα στοιχεία δημιουργούμε ένα σύνδεσμο για την αποσύνδεση του χρήστη, η οποία συνοδεύεται από ένα μήνυμα από τον περιηγητή Ιστού, το οποίο προειδοποιεί τον χρήστη ότι σε περίπτωση που δεν ολοκλήρωσε την προσθήκη δεδομένων σε οποιαδήποτε από τις φόρμες του ιστοτόπου, τα δεδομένα θα χαθούν. Για να το καταφέρουμε αυτό, καλούμε μια συνάρτηση συμβάντος (όταν πατήσει ο χρήστης τον σύνδεσμο), την `return confirm`, και ο χρήστης οδηγείται στο αρχείο `logout.php`. Εκεί καταστρέφονται όλες οι μεταβλητές σε `sessions`, λήγουν τα `cookies` και γίνεται ανακατεύθυνση στην αρχική σελίδα, όπως φαίνεται παρακάτω.

```
1. <?php
2. session_start();
3. session_destroy();
4. if(isset($_COOKIE['cookname']) && isset($_COOKIE['cookpass'])){
5.     setcookie("cookname", "", time()-60*60*24, "/");
6.     setcookie("cookpass", "", time()-60*60*24, "/");
7. }
8.
9. header ("Location: index.php");
10.     exit();
11.     ?>
```

Κώδικας 1. 10: Logout.php

Έπειτα περιλαμβάνουμε το αρχείο που εμφανίζει το μενού στα αριστερά της σελίδας και το οποίο θα αναλύσουμε στην αμέσως επόμενη παράγραφο. Από την αρχή της παραγράφου μέχρι αυτό το σημείο, ο κώδικας είναι επαναλαμβανόμενος σε κάθε αρχείο που χρησιμοποιείται για τις σελίδες του διαχειριστή. Από κει και πέρα αυτό που αλλάζει είναι το περιεχόμενο που διαφέρει ανάλογα με την λειτουργία που προσφέρει η κάθε σελίδα. Βέβαια να μην ξεχάσουμε ότι πάντα πριν το κύριο σώμα (`<body>`) της σελίδας, περιλαμβάνεται το εξωτερικό αρχείο `style.css`, το οποίο βρίσκεται στον φάκελο του συστήματος με όνομα `"css"` και κρίναμε ότι δεν είναι απαραίτητο να το αναλύσουμε στην παρούσα διπλωματική, καθώς είναι αρκετά ευκολονόητο για κάποιον που έχει βασικές γνώσεις CSS.

Έπειτα στο βασικό μέρος (`div` με αναγνωριστικό `id='main'`) της σελίδας αυτής, αλλά και κάθε άλλης σελίδας που περιλαμβάνει μια

φόρμα ή γενικά μια λειτουργία, αναγράφεται μια σύντομη περιγραφή σχετικά με την λειτουργία που προσφέρει η σελίδα.

```
1. <div id="main">
2.
3. <div class="full_w">
4. <div class="h_title"><b>I-EXAMS</b> | Μενού Διαχείρισης</div>
5. <p align="justify">
6. Καλώς ήρθατε στις σελίδες <b>Διαχείρισης</b>.
7. <br/><br/>
8. Εδώ ο διαχειριστής μπορεί να προσθέσει και να τροποποιήσει τις
   πληροφορίες των μαθημάτων, των ενοτήτων, των ερωτήσεων και των
   διαγωνισμάτων. Μπορεί επιπλέον να ελέγξει τις απαντήσεις των μαθητών και
   τις βαθμολογίες ενός τεστ,
9. καθώς επίσης να ελέγξει πληροφορίες σχετικά με τις απαντήσεις τους κατά
   τη διάρκεια της εξέτασης. Ακόμα μπορεί να τροποποιήσει τον διαθέσιμο χρόνο
   εξέτασης για κάθε φοιτητή ξεχωριστά.
10.      Επίσης υπάρχει η δυνατότητα ανώνυμης εξέτασης με την έκδοση
   ειδικών κωδικών που παραδίδονται εκτυπωμένοι στους φοιτητές.
11.      <br/><br/>
12.      Μπορείτε να ξεκινήσετε με την προσθήκη των προαναφερθέντων
   στοιχείων χρησιμοποιώντας το μενού διαχείρισης στα αριστερά.
13.      </p>
14.      </div>
```

Κώδικας 1. 11: Admin.php – κυρίως σώμα

3.3.2 Admin_sidebar.html και get_numof_ftext_mesges.php

```
4 <!-- Link to jquery class if not already included-->
5 <script type="text/javascript">
6 if(typeof jQuery == 'undefined'){
7 document.write('<script          type="text/javascript"          src="js/jquery-
   1.3.2.js"></'+script>');
8 }
9 </script>
10
11 <div id="sidebar">
12 <?php
13      //if the user has global authority
14      if ($_SESSION['aem'] == 0)
15      {
16 ?>
```

```

17 <div class="box">
18 <div class="h_title">Διαχείριση των <b>τμημάτων</b></div>
19 <ul>
20 <li class="b2"><a class="icon add_page" href="adepartment.php">Προσθήκη
   τμημάτων</a></li>

```

Κώδικας 1. 12: Admin_sidebar.html – διάκριση αντικειμένων διαχείρισης

Στο αρχείο admin_sidebar.html ελέγχουμε αν έχει συμπεριληφθεί ήδη (στο <head> της σελίδας) η βιβλιοθήκη JQuery και αν όχι τότε την συμπεριλαμβάνουμε στο αρχείο. Η jquery είναι μια βιβλιοθήκη JavaScript σχεδιασμένη να απλοποιήσει την υλοποίηση σεναρίων (scripting) στη πλευρά του πελάτη (client-side) της HTML. Είναι ελεύθερο λογισμικό και σε αυτό το σημείο είναι αναγκαία διότι όπως θα δούμε αμέσως μετά χρησιμοποιούμε την συνάρτησή της, \$.ajax(), η οποία εκτελεί ένα ασύγχρονο αίτημα AJAX. Η τελευταία [ζ] είναι η τέχνη της ανταλλαγής δεδομένων με τον διακομιστή και η ενημέρωση ενός μέρους της ιστοσελίδας - χωρίς επαναφόρτωση ολόκληρης της σελίδας. Εδώ, όπως βλέπουμε στη παραπάνω εικόνα, αν η τιμή της μεταβλητής \$_SESSION['aem'] (η οποία ορίστηκε στο admin_session_check.php) είναι ίση με 0, τότε εμφανίζουμε τις επιπλέον επιλογές στο μενού για τον καθολικό διαχειριστή. Παρακάτω στο ίδιο αρχείο θέλουμε να εμφανίζουμε σε πραγματικό χρόνο, τον αριθμό των εκκρεμών απαντήσεων ελεύθερου κειμένου και των μηνυμάτων.

```

1. <li class="b1"><a class="icon photo" href="aexamv.php">Ενεργός έλεγχος
   εξέτασης</a></li>
2. <script type="text/javascript">
3.     $(document).ready(function(){
4.         setInterval(function(){
5.             $.ajax({
6.                 type:"GET",
7.                 url:"get_numof_ftext_mesges.php",
8.                 data:{action:"shownumber"},
9.                 success:function(data){
10.                    $("#messages_freertext").html(data);
11.                }
12.            });
13.        }, 1000);
14.    });
15. </script>
16. <div id="messages_freertext"></div>

```

Κώδικας 1. 13: Admin_sidebar. html – εμφάνιση αριθμού μηνυμάτων
και απαντήσεων ελεύθερου κειμένου

Αυτό γίνεται με τη βοήθεια της AJAX. Δημιουργούμε μια συνάρτηση που εκτελείται κάθε 1000 millisecond = 1 second (Κώδικας 1. 13, γραμμή 13). Η συνάρτηση καλεί ένα ασύγχρονο αίτημα AJAX το οποίο λαμβάνει δεδομένα (το output, με την κατάλληλη μορφοποίηση) που παράγονται στο αρχείο get_numof_ftext_mesges.php και τα εμφανίζει στο <div> με id= "messages_freetext" (Κώδικας 1. 13, γραμμές 10 και 16).

```
1. $grade_mark = '%#%';
2. $answer_null = "";
3.
4. //see if all the free text questions where unanswered or not
5. if ($_SESSION['aem'] == 0)
6. {
7.     $query0 = $dbconn->prepare("SELECT question_sequence, an-
      answer_sequence FROM testsequences WHERE answer_sequence NOT LIKE
      :grade_mark AND answer_sequence!= :empty");
8.     $query0->bindParam(':grade_mark', $grade_mark);
9.     $query0->bindParam(':empty', $answer_null);
10.    $query0->execute();
11.    $rows0 = $query0->fetchAll(PDO::FETCH_ASSOC);
12. }
13. else
14. {
15.    //if the admin is a teacher, show only his items
16.    $query0 = $dbconn->prepare("SELECT testse-
      quences.question_sequence, testsequences.answer_sequence FROM testse-
      quences INNER JOIN tests ON testsequences.test_id = tests.id WHERE testse-
      quences.answer_sequence NOT LIKE :grade_mark AND testse-
      quences.answer_sequence!= :empty AND tests.instructor_id= :uid");
17.    $query0->bindParam(':grade_mark', $grade_mark);
18.    $query0->bindParam(':empty', $answer_null);
19.    $query0->bindParam(':uid', $_SESSION['user_id']);
20.    $query0->execute();
21.    $rows0 = $query0->fetchAll(PDO::FETCH_ASSOC);
22. }
```

Κώδικας 1. 14: Get_numof_ftext_mesges.php – αναζήτηση
αποτελεσμάτων με αβαθμολόγητες απαντήσεις

Σε αυτό το αρχείο κάνουμε μια αναζήτηση στον πίνακα Testsequences της βάσης δεδομένων, όπου το πεδίο answer_sequence δεν είναι κενό και δεν περιέχει τον χαρακτήρα #. Όπως θα δούμε αρκετά παρακάτω, αυτός ο χαρακτήρας εισάγεται δίπλα στην απάντηση ελεύθερου κειμένου για να δηλώσει ότι έχει βαθμολογηθεί. Με αυτό τον τρόπο αναζητούμε τις απαντήσεις που δεν έχουν ακόμα βαθμολογηθεί και όπως βλέπουμε επαναλαμβάνουμε το ερώτημα SQL δύο φορές, μια για κάθε περίπτωση ρόλου χρήστη (καθολικός ή απλός διαχειριστής).

```
1. $answered_freertext = 0;
2. foreach ($rows0 as $questions)
3. {
4.     $questions_array = rtrim($questions['question_sequence'], ",");
5.     $questions_array = explode(",", $questions_array);
6.     $answer_array = rtrim($questions['answer_sequence'], "|");
7.     $answer_array = explode("|", $answer_array);
8.     foreach ($questions_array as $qid)
9.     {
10.        //get the type of the question
11.        $query01 = $dbconn->prepare("SELECT type FROM questions WHERE
id= :qid");
12.        $query01->bindParam(':qid', $qid);
13.        $query01->execute();
14.        $rows01 = $query01->fetch(PDO::FETCH_NUM);
15.        //count the free text answers
16.        if ($rows01[0] == 'ελεύθερο κείμενο')
17.        {
18.            foreach ($answer_array as $ans)
19.            {
20.                $qid_ans = explode(">", $ans, 2);
21.                $question_id = $qid_ans[0];
22.                //if the question's answer is found
23.                if ($question_id == $qid)
24.                {
25.                    $answered_freertext = $answered_freertext + 1;
26.                }
27.            }
28.        }
29.    }
30. }
```

Κώδικας 1. 15: Get_numof_ftext_mesges.php – καταμέτρηση
αβαθμολόγητων απαντήσεων

Αφού βρούμε τα αποτελέσματα, προχωράμε με την καταμέτρηση των απαντήσεων ελεύθερου κειμένου (που δεν έχουν απαντηθεί). Αυτός ο κώδικας θα γίνει πιο κατανοητός μόλις δείξουμε πως αποθηκεύονται στη βάση οι ερωτήσεις που επιλέχτηκαν στην εξέταση και οι απαντήσεις τους. Στην αρχή λοιπόν αφαιρούμε από τα string των ερωτήσεων και απαντήσεων τα διαχωριστικά τους (το κόμμα και η κάθετη γραμμή αντίστοιχα) και περνάμε κάθε τιμή σε έναν πίνακα (Κώδικας 1. 15, γραμμές 4-7). Έπειτα για κάθε id των ερωτήσεων της εξέτασης ελέγχουμε τον τύπο της ερώτησης. Αν αυτός είναι ελεύθερο κείμενο, τότε ψάχνουμε στον πίνακα των απαντήσεων να βρούμε μια αντιστοίχιση στο id της ερώτησης. Αν αυτή βρεθεί, σημαίνει ότι η συγκεκριμένη ερώτηση ελεύθερου κειμένου απαντήθηκε (αλλά δε βαθμολογήθηκε ακόμα), οπότε αυξάνουμε την μεταβλητή \$answered_freetext κατά 1. Με αυτό τον τρόπο βρίσκουμε πόσες συνολικά απαντήσεις ελ/ρου κειμένου υπάρχουν σε κάθε αποτέλεσμα και εμφανίζουμε αυτό τον αριθμό στο μενού στα αριστερά της σελίδας όπως φαίνεται στον κώδικα παρακάτω.

```

1. if ($answered_freetext == 0)
2. {
3.     echo      '<li      class="b1"><a      class="icon      report"
href="afreetextv.php">Απαντήσεις ελ/ρου κειμένου: 0</a></li>';
4. }
5. else
6. {
7.     echo      '<li      class="b1"><a      class="icon      report"
href="afreetextv.php">Απαντήσεις      ελ/ρου      κειμένου:      <font      col-
or="red">'. $answered_freetext. '</font></a></li>';
8. }

```

Κώδικας 1. 16: Get_numof_ftext_mesges.php – εμφάνιση αποτελεσμάτων

Αν υπάρχει έστω και μία αβαθμολόγητη ερώτηση, τότε εμφανίζεται με κόκκινα γράμματα ο συνολικός αριθμός. Με απλούστερο τρόπο βρίσκουμε και τα συνολικά μηνύματα που έχουν στείλει οι φοιτητές και τα προβάλλουμε κι αυτά κάθε 1 δευτερόλεπτο με τη βοήθεια της AJAX όπως είδαμε προηγουμένως.

Αφού ολοκληρώθηκε η αρχική σελίδα του διαχειριστή, θα προχωρήσουμε στις σελίδες διαχείρισης των αντικειμένων (καθηγητές, ερωτή-

σεις, διαγωνίσματα κ.τ.λ.). Συγκεκριμένα θα γίνει ανάλυση μόνο ορισμένων σελίδων που ο κώδικας τους είναι σημαντικός για να κατανοηθούν οι λειτουργίες του συστήματος. Τα αρχεία που θα παραλείψουμε περιέχουν κώδικα παρόμοιο με τα προηγούμενα, οπότε δεν είναι απαραίτητη η ανάλυσή τους.

3.2.3 Aprofessor.php και add_professor.php

Στο αρχείο `aprofessor.php` δημιουργούμε τη φόρμα προσθήκης ενός νέου καθηγητή. Επειδή κάθε αντικείμενο από δω και πέρα συνδέεται με τον καθηγητή που το προσθέτει, έτσι ώστε κάθε καθηγητής να μπορεί να δει μόνο τα δικά του, θα αποθηκεύουμε στη βάση δεδομένων το `id` του χρήστη που το πρόσθεσε (πίνακας `Users`) και έχει ονοματεπώνυμο (`first_name`, `last_name`) ίδιο με αυτό που υπάρχει στον πίνακα `Professors`. Γι' αυτό τον λόγο, ο καθηγητής που προστίθεται θα πρέπει να είναι υπαρκτή οντότητα του πίνακα `Users`. Έτσι αν τα πεδία `first_name` και `last_name` που εισάγει ο διαχειριστής δεν αντιστοιχούν σε υπαρκτό χρήστη, τότε παράγουμε ένα προειδοποιητικό μήνυμα, όπως φαίνεται παρακάτω στο αρχείο `add_professor.php`, το οποίο επεξεργάζεται τα δεδομένα που εισήχθησαν. Σε αυτή περίπτωση γίνεται ανακατεύθυνση πίσω στην ίδια σελίδα, η οποία εξετάζει την τιμή του `session` που κρατάει το μήνυμα και αναλόγως το εμφανίζει.

```
1. <!-- get the value of error message that is given during the form check and
return it respectively -->
2. <?php
3. if( isset($_SESSION['MSG_ARR']) && is_array($_SESSION['MSG_ARR']) &&
count($_SESSION['MSG_ARR']) >0 )
4. {
5. foreach($_SESSION['MSG_ARR'] as $msg)
6. {
7. if ($msg == 'Αυτός ο καθηγητής είναι ήδη καταχωρημένος.' || $msg ==
'Αυτός ο καθηγητής δεν είναι χρήστης του συστήματος.')
8. {
9. echo '<div class="η_error">';
10.     echo '<p>',$msg,</p>';
11.     echo '</div>';
12.     unset($_SESSION['MSG_ARR']);
13. }
```

Κώδικας 1. 17: `Aprofessor.php` – εμφάνιση μηνυμάτων

Παρακάτω στο ίδιο αρχείο δημιουργούμε τη φόρμα, η οποία με το πάτημα του κουμπιού προσθήκης (<button type="submit">) αποστέλλει τα την είσοδο του χρήστη στο αρχείο add_professor.php μέσω της μεθόδου HTTP POST.

Σύμφωνα με αυτή τη μέθοδο (που δεν αναλύεται στον κώδικα του αρχείου), πριν το πρόγραμμα περιήγησης στείλει τις πληροφορίες, τις κωδικοποιεί το χρησιμοποιώντας ένα σύστημα που ονομάζεται κωδικοποίηση URL (URL encoding). Σε αυτή τη διαδικασία, τα ζεύγη ονομάτων/ τιμών ενώνονται με το σύμβολο "=" και τα διάφορα ζεύγη χωρίζονται με το "&" (π.χ. όνομα1=τιμή1&όνομα2=τιμή2&όνομα3 = τιμή3). Τα κενά αντικαθιστούνται από τον χαρακτήρα "+" και οποιοιδήποτε άλλοι μη αλφαριθμητικοί χαρακτήρες αντικαθιστούνται με δεκαεξαδικές τιμές. Μόλις κωδικοποιηθούν οι πληροφορίες αποστέλλονται στο διακομιστή με τη εκάστοτε μέθοδο (GET ή POST). Η μέθοδος POST μεταβιβάζει πληροφορίες μέσω κεφαλίδων HTTP (HTTP headers). Οι πληροφορίες κωδικοποιούνται όπως μόλις αναφέρθηκε και εισάγονται σε μια κεφαλίδα που λέγεται QUERY_STRING. Επίσης, οι πληροφορίες που αποστέλλονται με αυτή τη μέθοδο δεν είναι ορατές στους άλλους και δεν έχουν όριο στην ποσότητα προς αποστολή. Τέλος, είναι δυνατή οι αποστολή και λήψη τους μόνο από ένα αρχείο, σε αντίθεση με τη μέθοδο GET, η οποία στέλνει τα δεδομένα μέσω του URL και έτσι μπορεί να τα μεταβιβάσει σε πολλαπλές σελίδες.

```
1. <div class="element">
2. <label for="username">Username: </br>(μόλις συμπληρωθεί θα
   εμφανιστούν αυτόματα τα στοιχεία των παρακάτω πεδίων)</label>
3. <input id="username" name="username" maxlength="30" size="30"/>
4. </div>
5.
6. <script type="text/javascript">
7. $(document).ready(function(){
8.   setInterval(function(){
9.     var username = document.getElementById('username').value;
10.    if (username){
11.      $.ajax({
12.        type:"GET",
13.        url:"get_full_name.php?uname="+ username,
14.        data:{action:"showresults"},
15.        dataType: 'json',
16.        success:function(data){
17.          $('#first_name').val(data.first_name);
18.          $('#last_name').val(data.last_name);
```

```

19.     }
20.     });
21.     }
22.     }, 100);
23.     });
24.     </script>

```

Κώδικας 1. 18: Aprofessor.php – αυτόματη εμφάνιση στοιχείων

Το επόμενο βήμα είναι η δημιουργία των πεδίων εισαγωγής δεδομένων. Για τη διευκόλυνση του διαχειριστή, ώστε να εισάγει όντως έναν υπαρκτό χρήστη ως καθηγητή, υπάρχει το πεδίο "username". Σε αυτό ο διαχειριστής μπορεί να εισάγει προαιρετικά το όνομα χρήστη του καθηγητή και σύμφωνα με τα στοιχεία του στη βάση δεδομένων να εμφανιστούν αυτόματα τα πεδία "first_name" και "last_name" (ονοματεπώνυμο). Αυτό γίνεται και πάλι με τη βοήθεια της jQuery AJAX. Κάθε 0.1 δευτερόλεπτα (100 milliseconds) εκτελείται ο παραπάνω κώδικας JavaScript, ο οποίος το πρώτο πράγμα που κάνει είναι να διαβάσει την τιμή του πεδίου "username" και να την αναθέσει την μεταβλητή var (Κώδικας 1. 18, γραμμή 9). Έπειτα, αν η μεταβλητή αυτή είναι ορισμένη και δεν είναι κενή (Κώδικας 1. 18, γραμμή 10) καλείται η συνάρτηση \$.ajax() η οποία στέλνει την τιμή στο url που αντιστοιχεί στο αρχείο get_full_name.php μέσω της μεθόδου GET. Εκεί λαμβάνει τα δεδομένα με τον παρακάτω τρόπο, και τα αποθηκεύει σε ένα πίνακα που κωδικοποιείται στη μορφή JSON (Javascript Object Notation). Επί της ουσίας είναι μια σύνταξη για την αποθήκευση και ανταλλαγή δεδομένων, με τρόπο οργανωμένο και εύκολο στη πρόσβαση. Ιδιαίτερη προσοχή πρέπει να δώσουμε στο γεγονός ότι στις παραμέτρους της συνάρτησης \$.ajax() προσθέτουμε την έκφραση data-type: 'json', ώστε να αποκωδικοποιηθούν σωστά τα δεδομένα (Κώδικας 1. 18, γραμμή 15).

```

1. //get username value from ajax
2. $username = $_GET['uname'];
3.
4. if ($username != "")
5. {
6.     $query = $dbconn->prepare("SELECT first_name, last_name FROM users
WHERE username=:uname");
7.     $query->bindParam(':uname', $username);
8.     $query->execute();
9.     $rows = $query->fetch(PDO::FETCH_ASSOC);

```

```

10.     if ($query->rowCount() > 0)
11.     {
12.         echo json_encode(array('first_name' =>
13. $rows['first_name'], 'last_name' => $rows['last_name']));
14.     }
15.     else
16.     {
17.         echo json_encode(array('first_name' => "", 'last_name' => ""));
18.     }
19.     else
20.     {
21.         echo json_encode(array('first_name' => "", 'last_name' => ""));

```

Κώδικας 1. 19: Get_full_name.php

Αναζητούμε λοιπόν το όνομα και το επίθετο του χρήστη στον οποίο ανήκει το username που πληκτρολόγησε ο διαχειριστής. Αν βρεθεί αποτέλεσμα, το μετατρέπουμε σε πίνακα JSON. Αν πάλι όχι (ή αν για κάποιο λόγο η τιμή του username είναι κενή), τότε αποθηκεύουμε κενές τιμές στον πίνακα (Κώδικας 1. 19, γραμμές 16, 21). Πίσω στο αρχείο arprofessor.php μόλις η μέθοδος GET επιστρέψει τα δεδομένα επιτυχώς, εμφανίζουμε το όνομα στο πεδίο "first_name" και το επίθετο στο "last_name" αντίστοιχα (Κώδικας 1. 18, γραμμές 16-18).

Μόλις ο διαχειριστής πατήσει το κουμπί "Προσθήκη" είναι η ώρα για την επεξεργασία των στοιχείων που υπέβαλε, με σκοπό να προστεθούν στη βάση δεδομένων. Στον κώδικα του αρχείου add_professor.php που φαίνεται παρακάτω, αρχικά χρησιμοποιούμε την συνάρτηση trim της PHP για να εξαλείψουμε τα κενά και άλλους ανεπιθύμητους χαρακτήρες από την είσοδο. Αν αυτή αποτελείται από κενές μεταβλητές, τότε δεν προχωράμε στην προσθήκη, αλλά δημιουργούμε το ανάλογο προειδοποιητικό μήνυμα που θα εμφανιστεί στην αρχική φόρμα έπειτα από ανακατεύθυνση προς αυτή.

```

1. //get the user input values from the add professor form
2. $fname = trim($_POST['first_name']);
3. $lname = trim($_POST['last_name']);
4.
5. if($fname == "" || $lname == "")
6. {
7.     $msg_arr[] = 'Πρέπει να συμπληρώσετε όλα τα υποχρεωτικά πεδία.';
8.     $msgflag = true;
9. }

```

Κώδικας 1. 20: Add_professor.php – έλεγχος κενής εισόδου

Στη συνέχεια αναζητούμε στη βάση αν υπάρχει ένα τουλάχιστον id που να αντιστοιχεί σε υπάρχον καθηγητή, έτσι ώστε να εμφανίσουμε και πάλι το κατάλληλο μήνυμα (Κώδικας 1. 21, γραμμές 2-11). Αν ο καθηγητής που προστέθηκε είναι νέος, ελέγχουμε πρώτα αν είναι υπαρκτός χρήστης του συστήματος (Κώδικας 1. 21, γραμμές 15-19), ώστε να μη δημιουργηθεί το πρόβλημα που αναφέραμε νωρίτερα στην παράγραφο. Αν συμβαίνει κάτι τέτοιο αποθηκεύουμε στον πίνακα \$msg_arr[] ένα προειδοποιητικό μήνυμα. Διαφορετικά, αν δε τυχαίνει κάτι από τα προηγούμενα, προσθέτουμε τελικά τα στοιχεία του καθηγητή στον πίνακα Professors και αποθηκεύουμε ένα μήνυμα επιβεβαίωσης (Κώδικας 1. 21, γραμμές 28-33). Η τελευταία κίνηση είναι η ανακατεύθυνση στη σελίδα aprofessor.php όπου προβάλλεται το εκάστοτε μήνυμα.

```
1. //see if there is already a row with the same value in database
2. $query = $dbconn->prepare("SELECT id FROM professors WHERE
first_name=:fname AND last_name=:lname");
3. $query->bindParam(':fname', $fname);
4. $query->bindParam(':lname', $lname);
5. $query->execute();
6. $rows = $query->fetch(PDO::FETCH_NUM);
7. if($query->rowCount() > 0)
8. {
9.     $msg_arr[] = 'Αυτός ο καθηγητής είναι ήδη καταχωρημένος.';
10.    $msgflag = true;
11. }
12. else
13. {
14.    //check if this first name and last name belong to an existing user
15.    $query10 = $dbconn->prepare("SELECT id FROM users WHERE
first_name=:fname AND last_name=:lname");
16.    $query10->bindParam(':fname', $fname);
17.    $query10->bindParam(':lname', $lname);
18.    $query10->execute();
19.    $rows10 = $query10->fetch(PDO::FETCH_NUM);
20.    if($query10->rowCount() == 0)
21.    {
22.        $msg_arr[] = 'Αυτός ο καθηγητής δεν είναι χρήστης του συστήματος.';
23.        $msgflag = true;
24.    }
25.    else
```

```

26.      {
27.      //insert new department values
28.      $query1 = $dbconn->prepare("INSERT INTO professors
(first_name,last_name) VALUES (:fname,:lname)");
29.      $query1->bindParam(':fname', $fname);
30.      $query1->bindParam(':lname', $lname);
31.      $query1->execute();
32.      $msg_arr[] = 'Τα δεδομένα καταχωρήθηκαν επιτυχώς.';
33.      $msgflag = true;

```

Κώδικας 1. 21: Add_professor.php – έλεγχοι αποδεκτών στοιχείων

Σε αυτό το σημείο θα προσπεράσουμε τις λειτουργίες της επισκόπησης, επεξεργασίας και διαγραφής των επόμενων αντικειμένων και θα περάσουμε κατευθείαν στη διαχείριση των ερωτήσεων, στην προσθήκη τους πιο συγκεκριμένα.

3.2.4 Aquestion.php, aquestion_upload.php και download_csv.php

Στο αρχείο aquestion.php, υπάρχουν 2 ξεχωριστές φόρμες. Μια για τη προσθήκη πολλαπλών ερωτήσεων (πολλαπλής επιλογής), μέσω ενός αρχείου CSV, και μια για την προσθήκη μίας ερώτησης ξεχωριστά (πολλαπλής επιλογής ή ελεύθερου κειμένου).

```

1. <form enctype="multipart/form-data" id="upload" name="upload" ac-
tion="aquestion_upload.php" method="post">
2. <div class="element">
3. <a href="#csv_download">Κατέβασμα προτύπου CSV</a>
4. </div>
5. <div class="element">
6. <label for="csv">Ανέβασμα CSV με ερωτήσεις πολλαπλής επιλογής:
<br><br>(μάθημα, κατηγορία, ενότητα, ερώτηση, σωστή επιλογή/ απάντηση,
επιλογή 2, επιλογή 3, επιλογή 4, επιλογή 5, αριθμός δυσκολίας,
γλώσσα)<br><br>Το διαχωριστικό πρέπει να είναι το , (κόμμα)</label>
7. <input name="qupload" type="file" /><br><br>
8. <button type="submit" class="add">Προσθήκη</button>
9. <button type="reset" class="cancel">Ακύρωση</button>

```

Κώδικας 1. 22: Aquestion.php – φόρμα προσθήκης πολλαπλών ερωτήσεων

Η πρώτη φόρμα φαίνεται παραπάνω. Μια παράμετρος στον ορισμό της φόρμας που πρέπει να συμπεριληφθεί είναι η `enctype="multipart/form-data"`. Αυτό είναι απαραίτητο όταν χρησιμοποιούνται φόρμες που έχουν έλεγχο για ανέβασμα αρχείων, μέσω του στοιχείου `<input type="file"/>` (Κώδικας 1. 22, γραμμή 7). Είναι μια κωδικοποίηση που επιτρέπει ολόκληρα αρχεία (καθώς και πρόσθετες πληροφορίες που τα συνοδεύουν, όπως το μέγεθός τους) να συμπεριληφθούν στα δεδομένα που αποστέλλονται με τη μέθοδο POST. Εδώ ορίσαμε την τιμή που αντιστοιχεί στο χαρακτηριστικό ονόματος της εισόδου ίση με `"qupload"`. Επίσης, σε αυτό το σημείο δημιουργούμε και ένα εσωτερικό σύνδεσμο προς το κατέβασμα του προτύπου CSV. Με τον όρο "εσωτερικό" εννοούμε ότι ανακατευθύνει σε ένα μέρος της ίδιας σελίδας, για το οποίο θα εξηγήσουμε περισσότερα παρακάτω. Αυτό γίνεται με την κλασική δήλωση `<a href>` βάζοντας μια δίεση (#) δίπλα απ' το όνομα του στοιχείου στο οποίο θα κατευθυνθεί η σελίδα (Κώδικας 1. 22, γραμμή 3). Ας δούμε λοιπόν τι γίνεται στο βασικό μέρος του αρχείου `aquestion_upload.php`.

```
1. $file = $_FILES['qupload']['tmp_name'];
2. $filename = $_FILES['qupload']['name'];
3. $filesize = $_FILES['qupload']['size'];
4. $filetype = $_FILES['qupload']['type'];
5.
6. $row=0;
7. //if the type is csv
8. if ($filetype == 'text/csv')
9. {
10.     if($file != NULL)
11.     {
12.         //if the size doesn't exceed 5Mb
13.         if ($filesize <= 5000000)
14.         {
15.             //connect to database
16.             include 'dbase/connect.php';
17.             $fp=fopen($file, "r");
18.             //initiate the counter of failed questions
19.             $failed_upload = 0;
20.             while (($data = fgetcsv ($fp, 2050, ',')) !== FALSE)
21.             {
22.                 $row++;
23.                 $_SESSION['row'][$row] = $row;
24.                 $type = 'πολλαπλής επιλογής' ;
25.                 $subject = trim($data[0]) ;
26.                 $category = trim($data[1]) ;
```

```

27.     $topic = trim($data[2]) ;
28.     $question = trim($data[3]) ;
29.     $choice1 = trim($data[4]) ;
30.     $choice2 = trim($data[5]) ;
31.     $choice3 = trim($data[6]) ;
32.     $choice4 = trim($data[7]) ;
33.     $choice5 = trim($data[8]) ;
34.     $difficulty = trim($data[9]) ;
35.     $language = trim($data[10]) ;
36.     $answer = 1;

```

Κώδικας 1. 23: Aquestion_upload.php – άνοιγμα εισόδου

Υπάρχει μια καθολική μεταβλητή της PHP που ονομάζεται `$_FILES`. Αυτή η μεταβλητή είναι ένας δισδιάστατος πίνακας που διατηρεί όλες τις πληροφορίες που σχετίζονται με προστιθέμενο αρχείο. Έτσι λαμβάνουμε και αποθηκεύουμε τις ακόλουθες 4 μεταβλητές που παράγει η PHP για το αρχείο "qupload" (Κώδικας 1. 23, γραμμές 1-4):

- `$_FILES['qupload']['tmp_name']` – το αρχείο που ανέβηκε στον προσωρινό φάκελο του διακομιστή.
- `$_FILES['qupload']['name']` – το πραγματικό όνομα του ανεβασμένου αρχείου.
- `$_FILES['qupload']['size']` – το μέγεθος σε bytes του ανεβασμένου αρχείου.
- `$_FILES['qupload']['type']` – ο τύπος του ανεβασμένου αρχείου σε μορφή MIME (Multi-Purpose Internet Mail Extensions).

Εν συνεχεία χρησιμοποιούμε την συνάρτηση `fopen` για να ανοίξουμε το αρχείο που ανέβηκε, εφόσον σιγουρευτούμε ότι είναι όντως αρχείο `csv`, μη κενό και δε ξεπερνά σε μέγεθος τα 5Mb (Κώδικας 1. 23, γραμμές 8-17). Εδώ η παράμετρος "r" σημαίνει "read only", δηλαδή ότι το αρχείο θα διαβαστεί μόνο και δε θα τροποποιηθεί. Έπειτα μέσω της συνάρτησης `fgetcsv` αναλύουμε κάθε γραμμή του ανοιχτού αρχείου, η οποία έχει ως διαχωριστικό το κόμμα και δε ξεπερνά τους 2050 χαρακτήρες (λίγο παραπάνω απ' το συνολικό μέγεθος μιας γραμμής σύμφωνα με το μέγεθος του κάθε πεδίου στη βάση δεδομένων, συν τα αυτάκια και τα κόμματα). Η συνάρτηση αποθηκεύει κάθε πεδίο μιας γραμμής στον πίνακα `$data`. Έτσι αποθηκεύουμε κάθε πεδίο σε μεταβλητές. Παρακάτω γίνονται διάφοροι έλεγχοι για κάθε γραμμή αν λείπουν δεδομένα σε κάποιο πεδίο, αν η ερώτηση είναι ήδη καταχωρημένη ή αν το όνομα του μαθήματος, της κατηγορίας ή της ενότητας είναι λανθασμένα. Αν συμβαίνει κάτι τέτοιο τότε

αποθηκεύουμε το αντίστοιχο μήνυμα σε ένα πίνακα και ενημερώνουμε ένα μετρητή αποτυχημένων γραμμών του CSV (Κώδικας 1. 24, γραμμές 3-5)

```
1. else
2. {
3. $failed_upload = $failed_upload + 1;
4. $msg_arr[] = 'Το μάθημα, η κατηγορία ή η ενότητα της γραμμής
   '.$_SESSION['row'][$row].' είναι λανθασμένα: '.$_SESSION['question'][$row];
5. $msgflag = true;
6. }
7. }
8. }
9. if ($failed_upload > 0)
10. {
11.     $_SESSION['rows'] = $row;
12.     $_SESSION['failed_upload'] = $failed_upload;
13.     $_SESSION['success_upload'] = $row - $failed_upload;
14.     $msg_arr[] = 'Από τις '.$_SESSION['rows'].' ερωτήσεις, εισήχθησαν οι
   '.$_SESSION['success_upload'].' επιτυχώς, ενώ οι '.$_SESSION['failed_upload'].'
   είχαν προβλήματα.';
15.     $msgflag = true;
16. }
```

Κώδικας 1. 24: Aquestion_upload.php – μήνυμα αποτυχημένων εγγραφών

Αν υπάρχει έστω και μία αποτυχημένη γραμμή, τότε παράγουμε ένα μήνυμα που ενημερώνει τον διαχειριστή πόσες ερωτήσεις προστέθηκαν επιτυχώς και πόσες είχαν πρόβλημα από τις συνολικές (Κώδικας 1. 24, γραμμές 11-15).

Ας εξηγήσουμε τώρα πως κατασκευάζεται το πρότυπο CSV που αναφέραμε λίγο νωρίτερα. Στον κώδικα του αρχείου aquestion.php στην δεύτερη φόρμα υπάρχει το παρακάτω σημείο (Κώδικας 1. 25)

```
1. //create the csv link
2. function csv_link()
3. {
4. if (document.getElementById('s_id').value && document.getElementById('s_id').value != "" && document.getElementById('c_id').value && document.getElementById('c_id').value != "" && document.getElementById('top_id').value && document.getElementById('difty').value && document.getElementById('difty').value
```

```

!=      "      "      &&      document.getElementById('lang').value      &&
document.getElementById('lang').value != " ")
5. {
6. var subvalue = document.getElementById('s_id').value;
7. var catvalue = document.getElementById('c_id').value;
8. var topvalue = document.getElementById('top_id').value;
9. var difty = document.getElementById('difty').value;
10.     var lang = document.getElementById('lang').value;
11.     var link_text = "Κατέβασμα";
12.     var link = "download_csv.php?sid=" + subvalue + "&cid=" + catvalue +
"&topid=" + topvalue + "&difty=" + difty + "&lang=" + lang;
13.     document.getElementById("csv_download").innerHTML = '<label
for="csv_download">Πρότυπο CSV:</label><a href="'+link+'">'+link_text+'</a>';
14.     }
15.     else
16.     {
17.         document.getElementById("csv_download").innerHTML = '<label
for="csv_download">Πρότυπο CSV:</label><div id="csv_warning"
class="n_warning"><p>Πρέπει να επιλέξετε το μάθημα, την κατηγορία, την
ενότητα, το επίπεδο δυσκολίας και την γλώσσα για να γίνει διαθέσιμος ο
σύνδεσμος.</p></div>';
18.     }
19.     }

```

Κώδικας 1. 25: Aquestion.php – εμφάνιση προτύπου CSV

Δημιουργούμε μια συνάρτηση με όνομα `csv_link()`, στην οποία ελέγχουμε αν υπάρχουν και έχουν μη κενές τιμές τα πεδία με `id "s_id, c_id, top_id, difty"` και `"lang"` που αντιστοιχούν στο μάθημα, την κατηγορία, την ενότητα, τη δυσκολία και τη γλώσσα. Σε αυτή την περίπτωση λοιπόν, αποθηκεύουμε τις τιμές αυτών των πεδίων σε μεταβλητές με τη δήλωση `"var μεταβλητή = document.getElementById('πεδίο').value;"` (Κώδικας 1. 25, γραμμές 6-10). Έχοντας αυτές τις τιμές δημιουργούμε τον σύνδεσμο προς το αρχείο `download_csv.php` (το οποίο παράγει το πρότυπο CSV), όπου δίνουμε ως παραμέτρους τις προηγούμενες μεταβλητές (Κώδικας 1. 25, γραμμή 12). Τελικά εμφανίζουμε τον σύνδεσμο στο πεδίο με `id "csv_download"`.

Πάμε να δούμε τώρα πως παράγεται το πρότυπο αρχείο CSV. Αφού πάρουμε τις τιμές των μεταβλητών του Κώδικας 1. 25 με τη μέθοδο `GET`, αναζητούμε στη βάση δεδομένων τα ονόματα του μαθήματος, της κατηγορίας και της ενότητας που επέλεξε ο διαχειριστής στη 2^η φόρμα προσθήκης ερωτήσεων (καθώς οι αντίστοιχες τιμές των μεταβλητών που

λαμβάνουμε είναι τα id τους). Στη συνέχεια δημιουργούμε το αρχείο CSV με τον τρόπο που φαίνεται στον Κώδικας 1. 26.

```
1. // Output array into CSV file
2. $output = fopen('php://output', 'w');
3. header('Content-Type: text/csv');
4. header('Content-Disposition: attachment; filename="'. $filename. '"');
5. // output the column headings
6. if (isset($_GET['q']) && isset($_GET['ch1']) && isset($_GET['ch2']) &&
isset($_GET['ch3']) && isset($_GET['ch4']) && isset($_GET['ch5']))
7. {
8.     if ($_GET['ch1'] != " && $_GET['ch2'] != " && $_GET['ch3'] != " &&
$_GET['ch4'] != " && $_GET['ch5'] != ")
9.     {
10.         fputcsv($output, array($rows['description'], $rows3['description'],
$rows2['description'], $_GET['q'], $_GET['ch1'], $_GET['ch2'], $_GET['ch3'],
$_GET['ch4'], $_GET['ch5'], $difficulty, $language));
11.     }
12.     else
13.     {
14.         fputcsv($output, array($rows['description'], $rows3['description'],
$rows2['description'], $_GET['q'], 'σωστή επιλογή/ απάντηση', 'επιλογή 2',
'επιλογή 3', 'επιλογή 4', 'επιλογή 5', $difficulty, $language));
15.     }
16. }
17. else
18. {
19.     fputcsv($output, array($rows['description'], $rows3['description'],
$rows2['description'], 'ερώτηση', 'σωστή επιλογή/ απάντηση', 'επιλογή 2',
'επιλογή 3', 'επιλογή 4', 'επιλογή 5', $difficulty, $language));
20. }
```

Κώδικας 1. 26: Download_csv.php – δημιουργία προτύπου CSV

Ανοίγουμε ένα νέο έγγραφο χρησιμοποιώντας τη συνάρτηση fopen με παράμετρο "w", δηλαδή "write only" (πρόκειται να γράψουμε κάτι μέσα στο έγγραφο). Μετά φτιάχνουμε τα κατάλληλα headers με τα χαρακτηριστικά του αρχείου, όπως τον τύπο και το όνομα, ώστε ο περιηγητής να ξέρει πώς να αντιμετωπίσει το αρχείο και πώς να το αποθηκεύσει στον υπολογιστή του χρήστη (Κώδικας 1. 26, γραμμές 2-4). Αν ο διαχειριστής βρίσκεται στη σελίδα προσθήκης νέας ερώτησης, τότε έχουμε την "default" (βασική) περίπτωση όπου δεν έχουν σταλθεί άλλες παράμετροι στο αρχείο. Τότε με την χρήση της συνάρτησης fputcsv γράφουμε στη μεταβλητή \$output (που αποτελεί το ανοιχτό έγγραφο)

τα ονόματα των αντικειμένων που βρήκαμε προηγουμένως συν το επίπεδο δυσκολίας και τη γλώσσα στην οποία είναι γραμμένη η νέα ερώτηση, ενώ συμπληρώνουμε τα υπόλοιπα πεδία με την περιγραφή τους (Κώδικας 1. 26, γραμμή 19). Διαφορετικά, αν ο διαχειριστής είναι στη σελίδα επεξεργασίας μιας ερώτησης και επιχειρήσει να παράγει το πρότυπο CSV, τότε μέσω του αντίστοιχου κώδικα JavaScript (παρόμοιος με τον κώδικα 3.23) περνάμε ως παραμέτρους στο αρχείο `download_csv.php` τις τιμές όλων των προ-συμπληρωμένων πεδίων, είτε η ερώτηση είναι πολλαπλής επιλογής ή ελεύθερου κειμένου (π.χ. των επιλογών-απαντήσεων της ερώτησης). Οπότε μέσω της συνάρτησης `isset` ελέγχουμε αν έχουν οριστεί οι μεταβλητές `$_GET` που στάλθηκαν στη σελίδα και σε αυτή τη περίπτωση τις περιλαμβάνουμε και αυτές στο έγγραφο CSV (κώδικας 3.24, γραμμές 6-16).

3.3 Σελίδες φοιτητών

Στις επόμενες παραγράφους θα εξετάσουμε τα βασικότερα σημεία της δημιουργίας των λειτουργιών του φοιτητή, δηλαδή τον τρόπο με τον οποίο αρχικοποιείται έως ότου λήξει μια εξέταση (είτε έχει γίνει με κανονική σύνδεση του χρήστη στο σύστημα ή στη περίπτωση που έγινε με χρήση ενός token), την τυχαία επιλογή ερωτήσεων και επιλογών και το χρονόμετρο.

3.3.1 `Steste.php` και `add_answers.php`

Ένας φοιτητής μπορεί να εισέλθει στη σελίδα εξέτασης με 2 τρόπους. Είτε έχοντας συνδεθεί με όνομα χρήστη και συνθηματικό και στη συνέχεια αναζητώντας και επιλέγοντας την εξέταση ή εισάγοντας token στο url της σελίδας. Θα περιγράψουμε εν συντομία τι γίνεται στις 2 αυτές περιπτώσεις στον κώδικα που δημιουργεί την εξέταση, μέχρι το σημείο που επιλέγονται τυχαία οι ερωτήσεις. Εκεί θα εξηγήσουμε τη διαδικασία πιο αναλυτικά. Καθώς λοιπόν το token περιλαμβάνει το id του διαγωνίσματος, και στις 2 περιπτώσεις στο αρχείο `steste.php` που παράγει την σελίδα εξέτασης δίνεται μέσω του url αυτό το id (με τη μορφή `"steste.php?tid="` ή `"steste.php?tokid="`, όπου `tid` = test id και `tokid` = token). Αρχικά ελέγχουμε αν είναι ορισμένη αυτή η τιμή (id ή token αντίστοιχα) και την φιλτράρουμε για να δούμε αν είναι αριθμός. Αν όχι τότε η εξέταση δε ξεκινάει καν. Διαφορετικά το πρώτο πράγμα που γίνεται είναι η διάσπαση του token ώστε να βρούμε το id του

καθώς και το id του διαγωνίσματος. Από κει και πέρα η βασική αρχή όταν αναζητούμε κάτι από τον πίνακα Testsequences ή προσθέτουμε κάτι σε αυτόν είναι να διαχωρίζουμε την περίπτωση που χρησιμοποιήθηκε token από αυτή που έγινε κανονική σύνδεση. Επίσης, έχοντας λάβει υπόψη ότι υπάρχει πιθανότητα να κλείσει για κάποιο άγνωστο λόγο το πρόγραμμα περιήγησης ή ακόμη και ο ίδιος ο υπολογιστής κατά την διάρκεια μιας εξέτασης, κρατάμε αποθηκευμένες στον πίνακα Testsequences κάποιες σταθερές και μεταβλητές οι οποίες δίνουν την δυνατότητα συνέχισης της εξέτασης από το τελευταίο σημείο στο οποίο σταμάτησε. Αυτές είναι οι εξής:

- Seed – ένας τυχαία παραγόμενος αριθμός που εισάγεται στην συνάρτηση rand, η οποία επιλέγει τις ερωτήσεις τυχαία. Χάρη στη σταθερά seed η λίστα των τυχαία επιλεγμένων ερωτήσεων παραμένει ίδια κάθε φορά που αλλάζει η σελίδα και ο χρήστης προχωρά σε νέα ερώτηση. Αυτός ο αριθμός δεν αλλάζει κατά τη διάρκεια όλης της εξέτασης.
- Page – η σελίδα στην οποία βρίσκεται ο φοιτητής. Αυτή είναι απαραίτητη επειδή χρησιμοποιείται ως όριο στην εμφάνιση της λίστας των τυχαίων ερωτήσεων. Π.χ. στη δεύτερη σελίδα ο κώδικας θα εμφανίσει μία ερώτηση ξεκινώντας από την ερώτηση 1 (αριθμός σελίδας – 1) της τυχαίας λίστας (δεδομένου ότι η λίστα ξεκινά απ' το 0). Η σελίδα αυξάνεται κατά 1 κάθε φορά που υποβάλλεται μια απάντηση και ορίζεται ξανά ως 1 κάθε φορά που ολοκληρώνεται ένας βρόγχος, δηλαδή όταν εμφανιστούν για μια φορά όλες οι ερωτήσεις που απομένουν για να απαντηθούν, ώστε να ξεκινήσει η επανεμφάνιση όσων παραλείφθηκαν.
- Skip_ids – Τα id των ερωτήσεων που παραλείπονται σε κάθε βρόγχο. Ορίζεται ως κενό στην αρχή κάθε νέου βρόγχου.
- Rem_questions – η λίστα των id των ερωτήσεων που απομένουν να εμφανιστούν στην εξέταση. Κάθε φορά που απαντάται ή παραλείπεται μια ερώτηση αυτή η λίστα μειώνεται κατά ένα id.
- Loop_ids – Είναι το ίδιο με το skip_ids με τη διαφορά ότι αρχικοποιείται όταν ο εξεταζόμενος μπει σε κάθε επόμενο βρόγχο μετά τον πρώτο. Με άλλα λόγια όταν δεν είναι κενό, εμφανίζονται οι ερωτήσεις που παραλείφθηκαν στον τρέχον βρόγχο.
- Answer_sequence – η ακολουθία των απαντήσεων με τη μορφοποίηση που αποθηκεύεται στη βάση δεδομένων, την οποία θα φανερώσουμε παρακάτω.

- Grade – Ο τρέχον βαθμός της εξέτασης. Τροποποιείται κάθε φορά που προστίθεται μια νέα απάντηση.

Οι τιμές αυτών αποθηκεύονται στον πίνακα μόλις αρχικοποιηθούν και κάθε φορά που τροποποιούνται. Έτσι σε κάθε νέα σελίδα αναζητούμε τον πίνακα για να δούμε αν έχουν αποθηκευτεί αυτές οι τιμές και σε τέτοια περίπτωση τις αποθηκεύουμε σε sessions. Έπειτα αρχικοποιείται ο πίνακας Testsequences όπου προστίθενται τα στοιχεία του χρήστη, το id του τεστ και η ημερομηνία εκκίνησης. Ακόμη, αποθηκεύουμε την τιμή του πεδίου start_point (παράγραφος 2.5.11), που χρησιμοποιείται για το χρονόμετρο (αρχικά είναι ίδια με την ημερομηνία εκκίνησης και αλλάζει αν ο διδάσκοντας τροποποιήσει τον διαθέσιμο χρόνο εξέτασης). Εν συνεχεία γίνονται τρεις έλεγχοι για το αν το διαγώνισμα έχει ήδη απαντηθεί, αν έχει λήξει και αν το token (σε περίπτωση που χρησιμοποιήθηκε) είχε ήδη χρησιμοποιηθεί, ώστε να λήξει απευθείας η διαδικασία σε αυτές τις περιπτώσεις. Μετά απ' όλα αυτά ξεκινά ουσιαστικά η παραγωγή της εξέτασης. Πρώτα λαμβάνονται τα χαρακτηριστικά του διαγωνίσματος από τον αντίστοιχο πίνακα της βάσης δεδομένων και αποθηκεύονται σε sessions. Ο βαθμός ανά ερώτηση υπολογίζεται ως 100/ αριθμός συνολικών ερωτήσεων. Ακόμη, βρίσκουμε την διεύθυνση IP του χρήστη που εξετάζεται, μέσω της σταθεράς `$_SERVER["REMOTE_ADDR"]` (ακόμα και αν μοιράζεται το Internet ή χρησιμοποιεί Proxy, μέσω των σταθερών `$_SERVER["HTTP_CLIENT_IP"]` και `$_SERVER["HTTP_X_FORWARDED_FOR"]` αντίστοιχα). Έχοντας αποθηκεύσει όλες αυτές τις τιμές σε sessions περιλαμβάνουμε το αρχείο `timer_sidebar.html` που δημιουργεί όλες τις πληροφορίες στα αριστερά τις σελίδας, όπως το χρονόμετρο, πόσες ερωτήσεις απομένουν κ.τ.λ. Αργότερα θα το αναλύσουμε περαιτέρω. Παρακάτω υπολογίζουμε πόσες ερωτήσεις αντιστοιχούν σε κάθε ενότητα σύμφωνα με τα ποσοστά που έχουν αποθηκευτεί στον πίνακα `Topicpercentage`. Εκεί τα ποσοστά έχουν την μορφή "αριθμός %", για παράδειγμα το η ενότητα με id 1 έχει ποσοστό 40. Έτσι αν οι συνολικές ερωτήσεις του διαγωνίσματος είναι 5, τότε σε αυτή την ενότητα θα αντιστοιχούν $(40/100)*5 = 2$ ερωτήσεις. Επιπλέον στρογγυλοποιούμε το αποτέλεσμα ώστε να μην έχει δεκαδικά ψηφία. Αυτό γίνεται για όλες τις ενότητες και κρατάμε το αποτέλεσμα σε έναν πίνακα `$_SESSION['res']`, ώστε να υπολογίσουμε και το σύνολο των ερωτήσεων που αντιστοιχούν σε όλες τις ενότητες (Κώδικας 1. 27, γραμμές 11-13).

1. **foreach** (`$rows0` as `$topic`)

```

2. {
3.     //find the percentage of each topic for the selected test.
4.     $query01 = $dbconn->prepare("SELECT percentage FROM topicper-
percentage WHERE test_id= :tid AND topic_id= :topicid");
5.     $query01->bindParam(':tid', $test);
6.     $query01->bindParam(':topicid', $topic['id']);
7.     $query01->execute();
8.     $rows01 = $query01->fetch(PDO::FETCH_ASSOC);
9.
10.    $_SESSION['topic'][$n] = $topic['id'];
11.    $_SESSION['res'][$n] = round(($rows01['percentage'] / 100) *
$rows['total_questions']);
12.    //calculate the total result of all the percentages
13.    $total_res = $total_res + $_SESSION['res'][$n];
14.    $n++;
15. }
16. //if the total result is bigger than the total questions remove the ra-
remaining questions from the bigger percentage
17. if ($total_res > $_SESSION['total_questions'])
18. {
19.     $rem = $total_res - $_SESSION['total_questions'];
20.     $max_key = array_search(max($_SESSION['res']), $_SESSION['res']);
21.     $_SESSION['res'][$max_key] = $_SESSION['res'][$max_key] - $rem;
22. }

```

Κώδικας 1. 27: Steste.php – υπολογισμός ερωτήσεων ανά ενότητα

Λόγω της στρογγυλοποίησης υπάρχει πιθανότητα να προκύψουν περισσότερες ερωτήσεις από τις συνολικές. Αν συμβαίνει κάτι τέτοιο τότε εκτελούμε τον εξής απλό αλγόριθμο (Κώδικας 1. 27, γραμμές 17-22): υπολογίζουμε την διαφορά των ερωτήσεων που προέκυψαν, βρίσκουμε ποια ενότητα έχει τις περισσότερες ερωτήσεις και αφαιρούμε την διαφορά από αυτή. Έτσι, αν για παράδειγμα προκύψουν 6 ερωτήσεις αντί για 5 που είναι οι συνολικές του διαγωνίσματος, αφαιρούμε μια ερώτηση απ' την ενότητα στην οποία αντιστοιχούν περισσότερες ερωτήσεις. Στη συνέχεια εμφανίζουμε το επισυναπτόμενο αν υπάρχει. Αν είναι εικόνα, αρχείο ήχου, ή κινούμενο αρχείο flash, τότε εμφανίζεται μέσα στο περιεχόμενο της σελίδας, χωρίς να χρειάζεται να το κατεβάσει ο χρήστης. Αν πάλι είναι pdf, τότε ο εξεταζόμενος μπορεί να το ανοίξει κατευθείαν στην ίδια ή διαφορετική καρτέλα. Τέλος, αν είναι κάποιας άλλης μορφής αρχείο, τότε υπάρχει ένας σύνδεσμος (ο οποίος ανακατευθύνει στο αρχείο download_attachment.php που δημιουργεί τα κατάλληλα headers) για να κατέβει το αρχείο.

Καθ' όλη τη διάρκεια της εξέτασης κρατάμε σε ένα session μια σταθερά με όνομα "start". Αυτή αρχικοποιείται και ορίζεται ως 1 μόλις εμφανιστεί η πρώτη σελίδα της εξέτασης. Είναι χρήσιμη επειδή, με το που ξεκινήσει η εξέταση πρέπει να γίνουν οι έλεγχοι που αναφέραμε προηγουμένως και επιπλέον πρέπει να αρχικοποιηθούν κάποιες σταθερές και να αποθηκευτούν τα στοιχεία της εξέτασης στη βάση δεδομένων μόνο για μια φορά. Έτσι από την δεύτερη σελίδα εξέτασης κι έπειτα, ελέγχοντας αν η σταθερά \$_SESSION['start'] έχει οριστεί, αυτές οι διαδικασίες δεν επαναλαμβάνονται. Στη περίπτωση λοιπόν που η εξέταση μόλις ξεκίνησε αρχικά δημιουργούμε μια άλλη σταθερά, την \$_SESSION['seed'], και της δίνουμε μια τυχαία αριθμητική τιμή μέσω της συνάρτησης rand() (φυσικά αποθηκεύουμε την τιμή της στο πεδίο seed του πίνακα Testsequences). Στη συνέχεια επιλέγουμε τις ερωτήσεις με τον τρόπο που φαίνεται παρακάτω (Κώδικας 1. 28).

```
1. $n = 1;
2. $sql="";
3. $union="";
4.
5. while (isset($_SESSION['topic'][$n]) && $_SESSION['topic'][$n] != "")
6. {
7.     //get the percentage of questions of each corresponding topic and
    difficulties in random order and unify them
8.     $sql .= $union . "(SELECT * FROM questions WHERE available = 1 AND
    topic_id=      " . $_SESSION['topic'][$n] . "      AND      difficulty      IN
    (" . $_SESSION['difficulties'] . ")      AND      RAND() < (SELECT
    (" . $_SESSION['res'][$n] . "/COUNT(*) * 10) FROM questions) ORDER BY
    RAND(" . $_SESSION['seed'] . ") LIMIT " . $_SESSION['res'][$n] . ")";
9.     $union = " UNION ";
10.    $n++;
11. }
12. //randomize the order again and get all the questions at once
13. $sql2 = $sql . " ORDER BY RAND(" . $_SESSION['seed'] . ")";
14. //produce a list of the questions
15. $query02 = $dbconn->prepare($sql2);
16. $query02->execute();
17. $rows02 = $query02->fetchAll(PDO::FETCH_ASSOC);
```

Κώδικας 1. 28: Steste.php – Επιλογή τυχαίων ερωτήσεων ανά ενότητα

Σύμφωνα με αυτόν τον κώδικα (γραμμές 5-11), για κάθε ενότητα επιλέγουμε τις ερωτήσεις που είναι διαθέσιμες, αντιστοιχούν σε αυτή

την ενότητα και ο αριθμός δυσκολίας τους βρίσκεται εντός του πεδίου δυσκολίας του διαγωνίσματος. Επιλέγουμε ένα μικρό τεμάχιο των ερωτήσεων [22] χρησιμοποιώντας την έκφραση `RAND() < (SELECT ((ποσοστό ερωτήσεων της ενότητας/ συνολικές ερωτήσεις) * 10) FROM questions)`. Αυτή η μέθοδος συνδυασμένη με το `LIMIT`, χρησιμοποιεί το `ORDER BY RAND` (για την τυχαία επιλογή) σε ένα μικρό ποσοστό των γραμμών του πίνακα, το οποίο βασίζεται στις πόσες γραμμές θέλουμε (όσες ερωτήσεις υπολογίστηκαν ότι αντιστοιχούν στην ενότητα – δηλώνεται με το `LIMIT`), διαιρεμένο με τον συνολικό αριθμό των γραμμών του πίνακα και έπειτα πολλαπλασιάζουμε αυτόν τον αριθμό με το 10 για να αποφύγουμε την επιστροφή λιγότερων γραμμών απ’ όσες ζητήσαμε. Αυτό το αίτημα SQL επαναλαμβάνεται για όλες τις ενότητες και συνδυάζονται μέσω της έκφρασης `UNION` [23] και αφότου ολοκληρωθεί επιλέγουμε και πάλι με τυχαία σειρά τις ερωτήσεις του αποτελέσματος (Κώδικας 1. 28, γραμμή 13). Η έκφραση `ORDER BY RAND` είναι υπερβολικά αργή σε πίνακες με πολλές γραμμές, αλλά η παραπάνω μέθοδος δίνει γρήγορα αποτελέσματα καθώς περιορίζεται στην επιλογή τυχαίων ερωτήσεων ενός ποσοστού των συνολικών μόνο. Αυτό είναι το θετικό της. Όμως λόγω της χρήσης του βρόχου `while` δεν υπάρχει η δυνατότητα να χρησιμοποιηθεί η μέθοδος `bindParam` στη δήλωση PDO. Έτσι εδώ χάνεται και η έννοια της ασφάλειας. Αυτή η παραχθείσα λίστα ερωτήσεων αποθηκεύεται εν συνεχεία στη βάση δεδομένων (ως ένα string που περιέχει τα id τους) και όπως θα δούμε παρακάτω σε κάθε επόμενη σελίδα επιλέγονται με τυχαία σειρά ερωτήσεις που περιέχονται σε αυτή τη λίστα. Έπειτα εκτελούμε και πάλι τις γραμμές 13-17 του κώδικα 3, προσθέτοντας την έκφραση `LIMIT ".$start_from.", 1` (όπου `$start_from = $page - 1`), ώστε να πάρουμε μόνο την πρώτη ερώτηση της λίστας.

Εφόσον έχουμε πλέον μία ερώτηση, ελέγχουμε αν είναι πολλαπλής επιλογής για να δημιουργήσουμε έναν πίνακα με τις επιλογές και τον αποθηκεύουμε στη μεταβλητή `$choices` (Κώδικας 1. 29, γραμμές 4-11). Σε αυτόν τον πολυδιάστατο πίνακα, για κάθε επιλογή-απάντηση της ερώτησης δημιουργούμε έναν πίνακα με 5 στοιχεία: την ιδιότητά της αν είναι σωστή ή λάθος (πάντα το `choice1` θέτουμε ως σωστό στη βάση δεδομένων), την περιγραφή της, το id της (δηλαδή την σειρά της στη βάση δεδομένων π.χ. το `choice3` έχει id 3) και το επισυναπτόμενό της (αν υπάρχει).

```
1. if ($question['type'] == 'πολλαπλής επιλογής')
2. {
3.     //Put choices into a temp array
```

```

4.     $choices = array
5.     (
6.         0=>array ('correct' => 1, 'answer' => $question['choice1'], 'id' => 1,
'blob' => $question['choice1blob'], 'blobid' => $question['choice1blobtype_id']),
7.         1=>array ('correct' => 0, 'answer' => $question['choice2'], 'id' => 2,
'blob' => $question['choice2blob'], 'blobid' => $question['choice2blobtype_id']),
8.         2=>array ('correct' => 0, 'answer' => $question['choice3'], 'id' => 3,
'blob' => $question['choice3blob'], 'blobid' => $question['choice3blobtype_id']),
9.         3=>array ('correct' => 0, 'answer' => $question['choice4'], 'id' => 4,
'blob' => $question['choice4blob'], 'blobid' => $question['choice4blobtype_id']),
10.        4=>array ('correct' => 0, 'answer' => $question['choice5'], 'id' => 5,
'blob' => $question['choice5blob'], 'blobid' => $question['choice5blobtype_id'])
11.    );
12.
13.    //Randomize the choices array
14.    shuffle($choices);
15.    $_SESSION['choices'][$_SESSION['page']]=$choices;
16.    }
17.    else
18.    {
19.        $choices = "";
20.        $_SESSION['choices'][$_SESSION['page']] = $choices;
21.    }
22.    //if the question is a multiple choice
23.    if (is_array($choices))
24.    {
25.        foreach($_SESSION['choices'][$_SESSION['page']] as $i => $choice)
26.        {
27.            //get the location of the correct one
28.            if ($choice['correct'] == 1)
29.            {
30.                $corr = $i + 1;
31.            }
32.        }
33.        ?>
34.        <input type="hidden" id="correct_answer" name="correct_answer"
value="<?php echo $corr; ?>" />
35.        <?php
36.        }

```

Κώδικας 1. 29: Steste.php – ανακάτεμα επιλογών ερώτησης

Στη συνέχεια ανακατεύουμε τον πίνακα αυτόν με τη χρήση της συνάρτησης `shuffle`. Έτσι οι επιλογές θα εμφανιστούν με διαφορετική τυχαία σειρά στον κάθε εξεταζόμενο, αλλά ακόμα και στον ίδιο αν ανανεώσει την σελίδα (ή αν κάνει κάποιο λάθος π.χ. πατήσει "Υποβολή" χωρίς να έχει επιλέξει κάτι, οπότε θα εμφανιστεί προειδοποιητικό μήνυμα). Αν η ερώτηση είναι ελεύθερου κειμένου, τότε θέτουμε την μεταβλητή `$choices` ως κενή. Επίσης, για κάθε επιλογή τσεκάρουμε αν έχει την ιδιότητα της σωστής απάντησης και τότε αποθηκεύουμε την θέση της όπως εμφανίστηκε στον χρήστη. Έπειτα με έναν απλό έλεγχο για το αν η μεταβλητή αυτή είναι πίνακας ή όχι (άρα είναι πολλαπλής επιλογής ή ελ/ρου κειμένου), εμφανίζουμε την ερώτηση με τις επιλογές της και τα επισυναπτόμενα αν υπάρχουν, ή ένα `<textarea>` όπου ο χρήστης μπορεί να γράψει την απάντηση, στο οποίο ελέγχουμε με JavaScript να μην ξεπεραστεί το όριο των 1024 λέξεων. Στο τέλος δημιουργούμε τα 3 κουμπιά που φαίνονται στον κώδικα 3 και προσφέρουν μια διαφορετική επιλογή το καθένα.

```
1. <div class="entry">
2. <button type="submit" name="add" id="add">Υποβολή</button>
3. <button type="submit" name="skip" id="skip">Παράληψη</button>
4. </div>
5. <div class="entry">
6. <button type="submit" name="end" id="end">Υποβολή τεστ</button>
7. </div>
```

Κώδικας 1. 30: Steste.php – κουμπιά επιλογής λειτουργίας

Σε αυτό το σημείο θα εξηγήσουμε πως γίνεται η επεξεργασία των απαντήσεων, πως προσθέτονται αυτές και πως παραλείπονται οι ερωτήσεις έως ότου να λήξει το διαγώνισμα. Αργότερα θα αναλύσουμε τα στοιχεία που εμφανίζονται στα αριστερά κάτω από το μενού κατά τη διάρκεια της εξέτασης και λειτουργούν ανεξάρτητα από τη φόρμα εμφάνισης των ερωτήσεων.

Στο αρχείο `add_answers.php` λοιπόν, καταρχήν λαμβάνουμε από το στοιχείο `<input type="hidden">` το token αν υπάρχει και το αναλύουμε ώστε να βρούμε το `id` του στον πίνακα `Tokens`. Έτσι για κάθε ενέργεια που κάνουμε, την χωρίζουμε στη περίπτωση που έχει οριστεί αυτό το "κρυφό" στοιχείο που περιέχει το token ή όχι. Η περίπτωση που εξετάζουμε πρώτα είναι ο εξεταζόμενος να πατήσει το κουμπί με όνομα "end". Τότε απλά τον ανακατευθύνουμε στη σελίδα λήξης της εξέτασης παρέχοντας είτε το `id` του διαγωνίσματος ή το token (`header("location:`

stestee.php?tokid=".\$_POST['token']."&msg=1");). Η δεύτερη περίπτωση ο εξεταζόμενος να πατήσει το <button name="add"> που υποδηλώνει την προσθήκη απάντησης. Τότε αν η ερώτηση ήταν πολλαπλής επιλογής αποθηκεύουμε την απάντηση σε μια μεταβλητή session. Η απάντηση έχει την μορφή "id ερώτησης > επιλογή που επιλέχτηκε |", όπου το δεύτερο σκέλος αναλύεται σε "θέση εμφάνισης της επιλογής (θέση εμφάνισης της σωστής απάντησης) θέση της επιλογής στη βάση δεδομένων". Έτσι προχωράμε σε αυτή την ανάλυση, ώστε να δούμε αν η επιλογή του εξεταζόμενου ήταν ίδια με την σωστή απάντηση και έτσι να ενημερώσουμε τον βαθμό. Η συνάρτηση explode μας βοηθάει να διασπάσουμε ένα string με βάση κάποιο διαχωριστικό του και να περάσουμε τα στοιχεία σε ένα πίνακα. Με αυτό τον τρόπο συγκρίνουμε τα στοιχεία 2 πινάκων, ενός που περιέχει την επιλογή του χρήστη και ενός άλλου που περιέχει την απάντηση (Κώδικας 1. 31, γραμμές 2-11).

```
1. //set the answer form
2. $_SESSION['answer'] .= $_POST['question_id'].>".$_POST['choice'].'|';
3.
4. //get the selected answer
5. $sel_arr = explode("(", $_POST['choice']);
6. $selected = $sel_arr[0];
7. $corr_selid= explode(")", $sel_arr[1]);
8. //get the correct answer
9. $correct = $corr_selid[0];
10. //if the selected one is equal to the correct one count one correct an-
    //swer and calculate the grade by adding the equivalent mark
11. if ($selected == $correct)
12. {
13.     $_SESSION['grade'] = $_SESSION['grade'] + $_SESSION['mark'];
14.     //add one correct to the counter
15.     $answered_correctly = $rows3['answered_correctly'] + 1;
16.
17.     //update the counter
18.     $query5 = $dbconn->prepare("UPDATE qualityresults SET ans-
        //wered_correctly=:acorrecly WHERE test_id=:tid AND question_id=:qid");
19.     $query5->bindParam(':acorrecly', $answered_correctly);
20.     $query5->bindParam(':tid', $_SESSION['test']);
21.     $query5->bindParam(':qid', $_POST['question_id']);
22.     $query5->execute();
23. }
24. else
25. {
26.     //else subtract the equivalent negative mark
```

```

27.     $_SESSION['grade']          =          $_SESSION['grade']          -
$_SESSION['negative_mark'];
28.
29.     //add one wrong to the counter
30.     $answered_wrong = $rows3['answered_wrong'] + 1;

```

Κώδικας 1. 31: Add_answers.php – ανάλυση απάντησης πολλαπλής επιλογής

Αν η απάντηση ήταν σωστή, τότε ενημερώνουμε την μεταβλητή `$_SESSION['grade']` προσθέτοντας την βαθμολογία ανά ερώτηση που υπολογίσαμε στο αρχείο `steste.php` και έπειτα (αφού λάβουμε τον μετρητή και τον αυξήσουμε κατά 1) ενημερώνουμε και το πεδίο `answered_correctly` του πίνακα `Qualityresults` της βάσης δεδομένων, ώστε να μπορούν να παραχθούν τα γραφήματα (Κώδικας 1. 31, γραμμές 11-23). Εάν η απάντηση ήταν λάθος τότε αφαιρούμε απ' τον βαθμό την αρνητική βαθμολογία (αν δεν υπάρχει τότε δεν αφαιρείται κάτι) και αυξάνουμε τον μετρητή `$answered_wrong` κατά 1 (γραμμές 27-30), ώστε να ενημερώσουμε το αντίστοιχο πεδίο του πίνακα `Qualityresults`. Σε περίπτωση που η ερώτηση ήταν ελεύθερου κειμένου, προσθέτουμε απλώς την απάντηση στο string `$_SESSION['answer']`. Το διαχωριστικό αυτού του string είναι η κάθετη γραμμή (|). Επειδή οι χαρακτήρες ">", "|", "#" χρησιμοποιούνται ως διαχωριστικά και για να δηλώσουν ότι η απάντηση έχει βαθμολογηθεί (#), τους αντικαθιστούμε με τα αντίστοιχα λεκτικά τους (π.χ. το ">" με "ΜΕΓΑΛΥΤΕΡΟ") προτού προσθέσουμε την απάντηση. Η τρίτη περίπτωση είναι ο εξεταζόμενος να πατήσει το κουμπί με όνομα "skip". Τότε προσθέτουμε στο string `$_SESSION['skip_ids']` το id της ερώτησης και ενημερώνουμε το πεδίο `skip_ids` της βάσης δεδομένων. Έτσι όταν έρθει η στιγμή να εμφανιστούν οι αναπάντητες ερωτήσεις θα χρησιμοποιηθεί αυτό το string. Αν συμβεί κάποιο λάθος του τύπου ο εξεταζόμενος να μην απαντήσει την ερώτηση αλλά να πατήσει κατά λάθος "Υποβολή", αποθηκεύουμε το αντίστοιχο μήνυμα σφάλματος και ανακατευθύνουμε τον χρήστη στην ίδια σελίδα που βρισκόταν, δίχως κάποια από τις προηγούμενες ενέργειες ή αυτές που θα ακολουθήσουν. Διαφορετικά συνεχίζουμε ως εξής: πρώτα απ' όλα μειώνουμε το string με τα ids των εναπομείναντα ερωτήσεων (`$_SESSION['rem_questions']`) κατά ένα id και ενημερώνουμε το αντίστοιχο πεδίο της βάσης δεδομένων. Αν βρισκόμαστε στον αρχικό βρόγχο ερωτήσεων, τότε η μεταβλητή `$_SESSION['loop_ids']` δεν έχει οριστεί ακόμα. Σε αυτή τη περίπτωση

κρατάμε σε ένα session τον αριθμό των συνολικών ερωτήσεων του βρόγχου, ο οποίος ισούται με τις συνολικές ερωτήσεις του διαγωνίσματος. Αν η σελίδα στην οποία βρίσκεται ο εξεταζόμενος είναι μικρότερη απ' αυτόν τον αριθμό (π.χ. ο χρήστης βρίσκεται στην ερώτηση 3 από τις 5 συνολικά), αυξάνουμε την μεταβλητή `$_SESSION['page']` κατά 1 ενημερώνοντας παράλληλα το πεδίο `page` του πίνακα `Testsequences` καθώς και το πεδίο `answer_sequence` με τις απαντήσεις. Σε αυτό το σημείο έχει ολοκληρωθεί η διαδικασία επεξεργασίας της απάντησης οπότε γίνεται ανακατεύθυνση στη σελίδα εξέτασης. Μέχρι στιγμής είδαμε τι γίνεται σε περίπτωση που ο εξεταζόμενος βρίσκεται ακόμα στον πρώτο βρόγχο εμφάνισης των ερωτήσεων.

Το `$_SESSION['rem_questions']` έχει πλέον οριστεί, οπότε πίσω στη σελίδα `steste.php` ο κώδικας θα επαναληφθεί μέχρι το σημείο που επιλέγονται οι ερωτήσεις. Εκεί αντί για τον Κώδικας 1. 28, θα εκτελεστεί ο Κώδικας 1. 32. Όπως μπορούμε να δούμε, πλέον επιλέγουμε μία τυχαία ερώτηση από την λίστα με τις εναπομείναντα ερωτήσεις. Έτσι επαναλαμβάνεται η ίδια διαδικασία έως ότου να εμφανιστούν από μία φορά όλες οι ερωτήσεις.

```
1. //remove the last comma from string
2. $remq_ids = rtrim($_SESSION['rem_questions'], ",");
3. //now that the question list is set, select a question from this list, which
   doesn't contain previously selected questions. This way no question will be se-
   lected twice.
4. $sql = "(SELECT * FROM questions WHERE id IN (".$remq_ids.") ORDER BY
   RAND('".$_SESSION['seed']."') LIMIT 0,1)";
5. $query2 = $dbconn->prepare($sql);
6. $query2->execute();
7. $rows2 = $query2->fetchAll(PDO::FETCH_ASSOC);
```

Κώδικας 1. 32: `Steste.php` – επιλογή εναπομείναντα ερωτήσεων με τυχαία σειρά

Έστω ότι η ερωτήσεις ήταν 5 και ο φοιτητής έφτασε στην σελίδα με αριθμό 5 (ερώτηση 5 από τις συνολικές). Τότε μόλις υποβάλει την απάντησή του ή παραλείψει την ερώτηση επιτυχώς, θα γίνει η παρακάτω διαδικασία στο αρχείο `add_answers.php`. Εφόσον πλέον ισχύει `$_SESSION['page'] = $_SESSION['total_question_loop']`, αν επιπλέον το `$_SESSION['skip_ids']` είναι κενό, αυτό σημαίνει ότι ολοκληρώθηκε ο πρώτος βρόγχος δίχως να παραλειφθεί κάποια

ερώτηση και επομένως έχουν απαντηθεί όλες, άρα γίνεται ανακατεύθυνση προς την σελίδα λήξης της εξέτασης (όπου εμφανίζεται ο τελικός βαθμός). Αν όμως έχει παραληφθεί μία τουλάχιστον ερώτηση, τότε εκτελείτε ο Κώδικας 1. 33.

```
1. if (isset($_SESSION['skip_ids']) && $_SESSION['skip_ids'] != '')
2. {
3.     $_SESSION['loop_ids'] = $_SESSION['skip_ids'];
4.     $_SESSION['skip_ids'] = '';
5.     $_SESSION['page'] = 1;
6.     //update the page
7.     if (isset($_POST['token']))
8.     {
9.         $query3 = $dbconn->prepare("UPDATE testsequences SET skip_ids=
:skid, loop_ids= :loop_ids, page= :page WHERE token_id= :tokid AND test_id=
:tid");
10.        $query3->bindParam(':skid', $_SESSION['skip_ids']);
11.        $query3->bindParam(':loop_ids', $_SESSION['loop_ids']);
12.        $query3->bindParam(':page', $_SESSION['page']);
13.        $query3->bindParam(':tokid', $rows0['id']);
14.        $query3->bindParam(':tid', $_SESSION['test']);
15.        $query3->execute();
16.    }
17.    else
18.    {
19.        $query3 = $dbconn->prepare("UPDATE testsequences SET skip_ids=
:skid, loop_ids= :loop_ids, page= :page WHERE student_id= :uid AND test_id=
:tid");
20.        $query3->bindParam(':skid', $_SESSION['skip_ids']);
21.        $query3->bindParam(':loop_ids', $_SESSION['loop_ids']);
22.        $query3->bindParam(':page', $_SESSION['page']);
23.        $query3->bindParam(':uid', $_SESSION['user_id']);
24.        $query3->bindParam(':tid', $_SESSION['test']);
25.        $query3->execute();
26.    }
27.
28.    if (isset($_POST['token']))
29.    {
30.        //redirect to exam end page
31.        header("location: steste.php?tokid=".$_SESSION['token']);
32.    }
33.    else
34.    {
35.        //redirect to exam end page
36.        header("location: steste.php?tid=".$_SESSION['test']);
```

```

37.     }
38.     exit();
39.     }

```

Κώδικας 1. 33: Add_answers.php – περίπτωση υπαρχόντων παραλειπόμενων ερωτήσεων

Όπως φαίνεται εδώ, αρχικοποιούμε το session "loop_ids" και το θέτουμε ίσο με το \$_SESSION['skip_ids'] ενώ παράλληλα αδειάζουμε το τελευταίο, έτσι ώστε να δηλώσουμε ότι μπήκαμε σε επόμενο βρόγχο και αν υπάρξουν κι εκεί ερωτήσεις που παραλείφθηκαν, θα προστεθούν εκ νέου στο string \$_SESSION['skip_ids']. Η σελίδα εμφάνισης ορίζεται και πάλι ίση με 1, ώστε ο φοιτητής βλέποντας αυτό τον αριθμό στην οθόνη να καταλάβει ότι εμφανίζονται πλέον οι ερωτήσεις που παράλειψε. Έπειτα ενημερώνονται τα αντίστοιχα πεδία της βάσης δεδομένων (Κώδικας 1. 33, γραμμές 7-26) και στο τέλος γίνεται ανακατεύθυνση στη σελίδα, της οποίας τον αριθμό ορίσαμε. Πλέον στο αρχείο steste.php στο σημείο εμφάνισης της ερώτησης θα εκτελεστεί ο Κώδικας 1. 34.

```

1. //if the first loop has ended and there are remaining unanswered questions
2. if ((isset($_SESSION['start']) || $query20->rowCount() != 0) && isset($_SESSION['loop_ids']) && $_SESSION['loop_ids']!= "")
3. {
4.     //remove the last comma from string
5.     $loop_ids = rtrim($_SESSION['loop_ids'], ",");
6.     //get the percentage of the unanswered questions of each corresponding topic and difficulties in random order
7.     $sql = "(SELECT * FROM questions WHERE id IN (".$loop_ids.") ORDER BY RAND(".$_SESSION['seed'].") LIMIT ".$start_from.",1)";
8.     $query2 = $dbconn->prepare($sql);
9.     $query2->execute();
10.    $rows2 = $query2->fetchAll(PDO::FETCH_ASSOC);
11.    }

```

Κώδικας 1. 34: Steste.php – επιλογή παρελειπόμενων ερωτήσεων με τυχαία σειρά

Καθώς βρισκόμαστε στον δεύτερο και κάθε επόμενο βρόγχο, η μεταβλητή \$_SESSION['total_question_loop'] ορίζεται ίση με τον αριθμό

των id του session "loop_ids". Τελικά επαναλαμβάνοντας την παραπάνω διαδικασία, ολοκληρώνεται το διαγώνισμα (σε περίπτωση που δε λήξει ο χρόνος). Παρακάτω θα αναλύσουμε πως λειτουργεί το χρονόμετρο που εμφανίζεται στα αριστερά της οθόνης.

3.3.2 Timer_sidebar.html και get_time.php

Το αρχείο timer_sidebar.html καλείται από το αρχείο steste.php μόλις ληφθούν τα δεδομένα του διαγωνίσματος. Εκεί μέσα σε ένα στοιχείο <div> εμφανίζουμε το χρονόμετρο με τον εξής τρόπο: αρχικά καλούμε την μέθοδο \$.ajax της jQuery δημιουργώντας μια συνάρτηση JavaScript. Θέλουμε το χρονόμετρο να ενημερώνεται κάθε δευτερόλεπτο. Επειδή όπως θα δούμε χρησιμοποιούμε 2 συναρτήσεις, καλούμε την πρώτη κάθε δεύτερο του δευτερολέπτου (100 milliseconds) και την δεύτερη, που παίρνει τα αποτελέσματα της πρώτης, κάθε δευτερόλεπτο. Η πρώτη συνάρτηση φαίνεται παρακάτω (Κώδικας 1. 35).

```
1. <script type="text/javascript">
2. $(document).ready(function(){
3. setInterval(function(){
4. $.ajax({
5. type:"GET",
6. url:"get_time.php",
7. data:{action:"showresults"},
8. dataType: 'json',
9. success:function(data){
10.     $('#remHours').val(data.remHours);
11.     $('#remMinutes').val(data.remMinutes);
12.     $('#remSeconds').val(data.remSeconds);
13. }
14. });
15. }, 100);
16. });
17. </script>
18.
19. <input type="hidden" id="remHours" name="remHours" value="" />
20. <input type="hidden" id="remMinutes" name="remMinutes" value="" />
21. <input type="hidden" id="remSeconds" name="remSeconds" value="" />
```

Κώδικας 1. 35: Timer_sidebar.html – κάλεσμα αρχείου get_time.php και εμφάνιση αποτελεσμάτων

Όπως βλέπουμε εδώ, η συνάρτηση λαμβάνει δεδομένα από το αρχείο `get_time.php` και εμφανίζει τα αποτελέσματα που έχουν μορφοποίηση json σε τρία "κρυφά" από τον χρήστη divs. Στο πρώτο με id "remHours" εμφανίζονται οι ώρες που απομένουν, στο δεύτερο με id "remMinutes" τα λεπτά που απομένουν και στο τρίτο που έχει id "remSeconds" εμφανίζονται τα εναπομείναντα δευτερόλεπτα. Στην ουσία είναι αυτά που ζητάμε και περιμένουμε να εμφανίσει το χρονόμετρο. Το πώς τα βρίσκουμε αυτά θα το εξηγήσουμε λίγο αργότερα. Αμέσως μετά δημιουργούμε μια άλλη συνάρτηση με όνομα `timePassed()` (Κώδικας 1.36).

```
1. function timePassed() {
2.   var remainingSeconds = document.getElementById('remSeconds').value;
3.   var remainingMinutes = document.getElementById('remMinutes').value;
4.   var remainingHours = document.getElementById('remHours').value;
5.   document.getElementById('countdown').innerHTML = remainingHours + ":" +
   remainingMinutes + ":" + remainingSeconds;
6.   if (remainingSeconds == 0) {
7.     if (remainingMinutes == 0) {
8.       if (remainingHours == 0) {
9.         clearInterval(countdownTimer);
10.        var color = "#FF0000";
11.        document.getElementById('countdown').innerHTML = <font col-
   or="+color+">Τέλος χρόνου</font>";
12.        if (token_id != 0) {
13.          window.location = "stestee.php?tokid=" + token_id +
   "&msg=1";
14.        }
15.      } else {
16.        window.location = "stestee.php?tid=" + test_id + "&msg=1";
17.      }
18.    } else {
19.      remainingMinutes = 59;
20.      remainingHours--;
21.    }
22.  } else {
23.    remainingSeconds = 59;
24.    remainingMinutes--;
25.  }
26. } else {
27.   remainingSeconds--;
28. }
29. }
30.
```

31. `var countdownTimer = setInterval('timePassed()', 1000);`

Κώδικας 1. 36: Timer_sidebar.html – μείωση και τέλος χρόνου

Σε αυτή τη συνάρτηση λαμβάνουμε κάθε δευτερόλεπτο τις τιμές από τα παραπάνω divs (τα οποία όπως είπαμε ενημερώνονται κάθε 0,1 δευτερόλεπτο). Έπειτα αν συμβαίνει και οι 3 τιμές (εναπομείναντα ώρα, λεπτά και δευτερόλεπτα) να είναι μηδέν, τότε μέσω της JavaScript μεθόδου "window.location" ανακατευθύνουμε τον χρήστη στη σελίδα λήξης, αφού εμφανίσουμε με κόκκινα γράμματα το μήνυμα "Τέλος χρόνου". Διαφορετικά κάθε φορά που μηδενίζεται ένα από τα 3 στοιχεία, ορίζουμε εκ νέου την τιμή του ως 59, καθώς η μέτρηση είναι αντίστροφη, ενώ παράλληλα μειώνουμε τον μεγαλύτερο του μετρητή κατά 1. Για παράδειγμα μόλις τα δευτερόλεπτα γίνουν ίσα με 0, ενώ τα λεπτά δεν είναι 0, τότε τα δευτερόλεπτα γίνονται ίσα με 59 και τα λεπτά μειώνονται κατά 1. Έτσι συνεχίζονται να μειώνονται τα δευτερόλεπτα μέχρι να ξανασυμβεί το ίδιο και να γίνουν όλα 0. Ας δούμε σε αυτό το σημείο πως βρίσκουμε τον χρόνο που απομένει.

Για να υπολογίσουμε τον χρόνο που απομένει, αυτό που αρκεί είναι να βρούμε τον ακριβή χρόνο λήξης και να αφαιρέσουμε από αυτόν τον τρέχον χρόνο του διακομιστή. Έτσι αν για παράδειγμα ο χρόνος λήξης έχει υπολογιστεί στις 5:45:00 και ο τρέχον χρόνος είναι 5:23:00, τότε απομένουν 45-23=22 λεπτά. Για να βρούμε τον χρόνο λήξης χρειάζεται να ξέρουμε τον χρόνο έναρξης της εξέτασης και την διάρκεια της, ώστε να τα προσθέσουμε. Έτσι λοιπόν στο αρχείο get_time.php κάνουμε ακριβώς αυτό το πράγμα (Κώδικας 1. 37).

```
1. //get the start datetime
2. $query10 = $dbconn->prepare("SELECT start_point FROM testsequences
WHERE test_id= :tid AND student_id= :uid");
3. $query10->bindParam(':tid', $_SESSION['test']);
4. $query10->bindParam(':uid', $_SESSION['user_id']);
5. $query10->execute();
6. $rows10 = $query10->fetch(PDO::FETCH_ASSOC);
7. }
8.
9. //find the ending time based on start time + duration
10. $end_date = new DateTime($rows10['start_point']);
11. $end_date->add(new DateInterval("PT{$dbHours}H"));
12. $end_date->add(new DateInterval("PT{$dbMinutes}M"));
13. $end_date->add(new DateInterval("PT{$dbSeconds}S"));
14. $end_date = $end_date->format('Y-m-d H:i:s');
```

```

15.
16.      //find the ending year, month, day, hour, minute and second
17.      $send_datetime = explode(" ", $send_date);
18.      $send_date = explode("-", $send_datetime[0]);
19.      $send_year = $send_date[0];
20.      $send_month = $send_date[1];
21.      $send_day = $send_date[2];
22.      $send_time = explode(":", $send_datetime[1]);
23.      $send_hour = $send_time[0];
24.      $send_minute = $send_time[1];
25.      $send_second = $send_time[2];
26.      //make a unix timestamp of the ending time
27.      $endTime =
mktime($send_hour,$send_minute,$send_second,$send_month,$send_day,$send_y
ear);

```

Κώδικας 1. 37: Get_time.php – εύρεση χρόνου λήξης

Αρχικά παίρνουμε την τιμή του πεδίου start_point του πίνακα Testsequences. Αυτή η τιμή είναι ίση με την ημερομηνία και ώρα έναρξης της εξέτασης και μπορεί να τροποποιηθεί από τον διδάσκοντα ώστε να δώσει επιπλέον 5 λεπτά στον φοιτητή ή να λήξει τον διαθέσιμο χρόνο του. Όπως και να χει, αυτό που θέλουμε να πετύχουμε είναι να μετατρέψουμε τις ημερομηνίες σε μια μορφή κατάλληλη για να τις προσθέτουμε και να τις αφαιρούμε μεταξύ τους. Έτσι μετατρέπουμε την τιμή του start_point σε ένα αντικείμενο ημερομηνίας και ώρας, στο οποίο μπορούμε να προσθέσουμε τα στοιχεία του διαθέσιμου χρόνου υποβολής του διαγωνίσματος, μέσω της μεθόδου add(new DateInterval()) (γραμμές 10-14). Αυτή η προσθήκη μας δίνει τον χρόνο λήξης, τον οποίο αφού μορφοποιήσουμε ως "χρόνος – μήνας – μέρα ώρα : λεπτά : δευτερόλεπτα" προχωράμε να τον διασπάσουμε στα επιμέρους στοιχεία του (γραμμές 17-25). Τελικά χρησιμοποιώντας καθένα απ' αυτά μέσω της συνάρτησης mktime, μετατρέπουμε την ημερομηνία και ώρα λήξης σε μορφή Unix timestamp (στάμπα ώρας συστήματος Unix). Αυτή η στάμπα είναι ένας μακρύς ακέραιος που περιέχει τον αριθμό των δευτερολέπτων μεταξύ του Unix Epoch (το σημείο εκκίνησης του λειτουργικού συστήματος Unix που καθορίζει την τρέχουσα ημερομηνία και ώρα του υπολογιστή – 1 Ιανουαρίου 1970 00:00:00 GMT) και το καθορισμένο χρονικό διάστημα. Μετατρέποντας παρομοίως την τρέχουσα ημερομηνία και ώρα σε Unix timestamp μπορούμε να κάνουμε την αφαίρεση "ώρα λήξης – τρέχουσα ώρα" και να πάρουμε τον χρόνο που απομένει. Στη συνέχεια βρίσκουμε μέσω

αυτού του ακεραίου τα στοιχεία της ώρας και τα προβάλλουμε κωδικοποιημένα σε JSON (Κώδικας 1. 38). Καθώς η διαφορά είναι εκφρασμένη σε δευτερόλεπτα, την διαιρούμε δια 60 δευτερόλεπτα δια 60 λεπτά δια 24 ώρες, ώστε να βρούμε τις μέρες που απομένουν (γραμμή 4) και στη συνέχεια μέσω αυτών τις ώρες, τα λεπτά και τα δευτερόλεπτα (γραμμές 5-7).

```
1. //calculate their difference in seconds
2. $Diff = $endTime - $currTime;
3.
4. $remainingDay = floor($Diff/60/60/24);
5. $remainingHours = floor(($Diff-($remainingDay*60*60*24))/60/60);
6. $remainingMinutes = floor(($Diff-($remainingDay*60*60*24)-
($remainingHours*60*60))/60);
7. $remainingSeconds = floor(($Diff-($remainingDay*60*60*24)-
($remainingHours*60*60))-($remainingMinutes*60));
8.
9. echo json_encode(array('remHours' => $remainingHours,'remMinutes' =>
$remainingMinutes,'remSeconds' => $remainingSeconds));
```

Κώδικας 1. 38: Get_time.php – υπολογισμός εναπομείναντα χρόνου

Πλέον έχουμε ολοκληρώσει την ανάλυση των βασικότερων σημείων του κώδικα που γράφτηκε για τη δημιουργία της σελίδας. Ο αναγνώστης μπορεί να κατανοήσει πως δουλεύουν οι σημαντικότερες λειτουργίες του ιστότοπου. Στο κεφάλαιο που ακολουθεί θα τις περιγράψουμε όλες, ώστε να υπάρχει μια πλήρη εικόνα του ηλεκτρονικού συστήματος εξέτασης φοιτητών.

Κεφάλαιο 4

Παρουσίαση του ιστοχώρου

Στο προηγούμενο κεφάλαιο έγινε η υλοποίηση του ιστοτόπου με βάση τις προδιαγραφές του 2^{ου} κεφαλαίου. Σε αυτό το κεφάλαιο παρουσιάζονται οι λειτουργίες του ιστοτόπου που σχεδιάστηκαν για τον διαχειριστή και τον απλό χρήστη, σύμφωνα με όσα υλοποιήθηκαν στο 3^ο κεφάλαιο.

4.1 Λειτουργίες διαχειριστή

Όπως έχει αναφερθεί ο διαχειριστής έχει δύο ιδιότητες, καθώς μπορεί να είναι είτε καθολικός ή απλός διδάσκοντας. Η βασική διαφορά τους είναι ότι ο διδάσκοντας έχει πρόσβαση μόνο στα δικά του στοιχεία (μαθήματα, ερωτήσεις, διαγωνίσματα κ.τ.λ), ενώ ο καθολικός διαχειριστής μπορεί να επεξεργαστεί τα στοιχεία όλων των καθηγητών. Επιπλέον ο δεύτερος μπορεί να προσθέσει τα τμήματα, τους καθηγητές και τις κατηγορίες, καθώς και να επεξεργαστεί το καθένα απ' αυτά.

4.1.1 Σύνδεση και αποσύνδεση διαχειριστή

Η είσοδος του διαχειριστή στην διαδικτυακή εφαρμογή γίνεται με χρήση της φόρμας που υπάρχει στην αρχική σελίδα, στα αριστερά. Ο διαχειριστής πρέπει να εισάγει δεδομένα σε δύο πεδία, το όνομα χρήστη και το συνθηματικό, όπως αυτά έχουν οριστεί κατά την εγγραφή του στη διαδικτυακή πλατφόρμα του λέκτορα Μηνά Δασυγένη, "Laboratory of Digital Systems and Computer Architecture" (<https://arch.ict.e.uowm.gr/schedule/register.php>). Πριν την επεξεργασία τους, γίνεται έλεγχος από τον εξυπηρετητή με χρήση PHP για το αν τα πεδία είναι είτε άδεια ή λανθασμένα. Αν συμβαίνει μια από τις δύο αυτές περιπτώσεις τότε εμφανίζεται προειδοποιητικό μήνυμα σφάλματος, ενώ σε αντίθετη περίπτωση ο διαχειριστής ανακατευθύνεται στον προσωπικό του λογαριασμό.

Σχήμα 1. 5: Αποτυχία σύνδεσης

Αν αυτός που συνδέεται επιλέξει την λειτουργία "Remember Me" κατά τη σύνδεσή του, τότε θα παραμείνει συνδεδεμένος ακόμα και αν κλείσει το πρόγραμμα περιήγησης στο Διαδίκτυο, έως ότου να αποσυνδεθεί από το σύστημα.

Η αποσύνδεση χρήστη πραγματοποιείται με το πάτημα του σχετικού συνδέσμου στο πάνω μέρος της σελίδας. Κατά την έξοδο εμφανίζεται ενημερωτικό μήνυμα, όπου ο χρήστης μπορεί να επιλέξει επιβεβαίωση ή ακύρωση. Εφόσον επιβεβαιώσει την έξοδό του, ακυρώνονται όλα τα cookies που είχαν δημιουργηθεί.

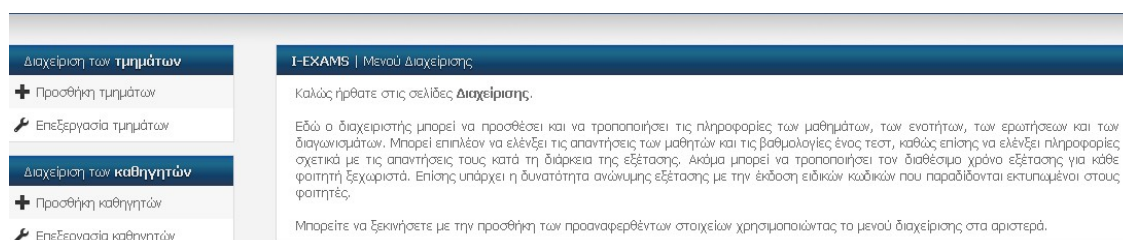
Σχήμα 1. 6: Αποσύνδεση χρήστη

4.1.2 Διαχείριση αντικειμένων

Η διαχείριση των αντικειμένων ακολουθεί το ίδιο μοτίβο στις περισσότερες των περιπτώσεων. Ο διαχειριστής επιλέγει το αντικείμενο που θέλει να προσθέσει ή να τροποποιήσει από το βασικό μενού στα αριστερά της ιστοσελίδας που βρίσκεται.

Κατά την προσθήκη του αντικειμένου καλείται να συμπληρώσει μια φόρμα με τα χαρακτηριστικά του αντικειμένου, κάποια απ' τα οποία είναι υποχρεωτικά και άλλα όχι. Μόλις επιλέξει την "Προσθήκη" γίνεται έλεγχος των δεδομένων και εφόσον πληρούν τις προϋποθέσεις, τότε καταχωρούνται επιτυχώς στη βάση και εμφανίζεται μήνυμα επιτυχίας. Διαφορετικά εμφανίζονται τα αντίστοιχα μηνύματα σφάλματος.

Για την επεξεργασία ενός αντικειμένου, ο διαχειριστής βλέπει έναν πίνακα με τα διαθέσιμα στοιχεία στον οποίο μπορεί να επιλέξει επεξεργασία ή διαγραφή. Κατά την επεξεργασία, εισέρχεται σε μια σελίδα όπου υπάρχει μια φόρμα με τα χαρακτηριστικά του αντικειμένου ήδη συμπληρωμένα. Εκεί γίνεται να τροποποιήσει τα χαρακτηριστικά και να πατήσει "Αποθήκευση". Κατά την διαγραφή, αν το αντικείμενο που έχει επιλεγεί συνδέεται με κάποιο άλλο αντικείμενο στη βάση δεδομένων, μέσω δευτερεύοντος κλειδιού, τότε εμφανίζεται μήνυμα στο πάνω μέρος της σελίδας που ενημερώνει ότι η διαγραφή του δεν είναι εφικτή, διαφορετικά διαγράφεται με επιτυχία.



Σχήμα 1. 7: Βασικό μενού

4.1.3 Διαχείριση τμημάτων

Η διαχείριση των τμημάτων γίνεται με την επιλογή του σχετικού συνδέσμου από το βασικό μενού στα αριστερά της ιστοσελίδας και περιλαμβάνει δύο υποκατηγορίες, την προσθήκη και την επεξεργασία τους.

Κατά την προσθήκη ενός νέου τμήματος υπάρχει μια φόρμα στην οποία ο διαχειριστής πρέπει να εισάγει πληροφορίες σε τρία πεδία, το όνομα τμήματος, το υποκοριστικό του και τον σύνδεσμο του ιστοτόπου του, από τα οποία μόνο το πρώτο είναι υποχρεωτικό. Με το πάτημα του κουμπιού "Προσθήκη" τα δεδομένα καταχωρούνται στη βάση δεδομένων.

Κατά την επεξεργασία των τμημάτων εμφανίζεται ένας πίνακας με όλα τα διαθέσιμα τμήματα ταξινομημένα κατά το όνομά τους. Εκεί ο διαχειριστής μπορεί να επιλέξει να επεξεργαστεί το τμήμα ή να το διαγράψει. Αν επιλέξει το δεύτερο και υπάρχει ήδη ένα μάθημα που συνδέεται με το συγκεκριμένο τμήμα μέσω του δευτερεύοντος κλειδιού, τότε εμφανίζεται μήνυμα λάθους ότι δεν είναι εφικτή η διαγραφή.

! Αυτό το τμήμα δε γίνεται να διαγραφεί επειδή υπάρχει ένα τουλάχιστον μάθημα με το οποίο συνδέεται. Πρέπει πρώτα να βρείτε και να διαγράψετε αυτό το μάθημα.

Επεξεργασία Τμημάτων

Λίστα διαθέσιμων τμημάτων

Σε αυτή τη σελίδα μπορείτε να δείτε όλα τα τμήματα που έχουν προστεθεί στη βάση δεδομένων, αλλά και να επεξεργαστείτε ξεχωριστά τα καθένα από αυτά.

Όνομα	Υποκοριστικό	Ιστότοπος	Επεξεργασία	Διαγραφή
Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών	ΤΜΠΤ	http://www.ictc.uowm.gr/		
Τμήμα Μηχανολόγων Μηχανικών	ΤΜΜ	http://www.mech.uowm.gr/gr/		

Σχήμα 1. 8: Αποτυχία διαγραφής τμήματος

Αν επιλέξει την επεξεργασία, τότε ο διαχειριστής εισέρχεται σε μια σελίδα με μια φόρμα, όπου υπάρχουν ήδη συμπληρωμένα τα στοιχεία του τμήματος. Εκεί μπορεί να τροποποιήσει τα δεδομένα των πεδίων και να επιλέξει "Αποθήκευση".

4.1.4 Διαχείριση καθηγητών

Η διαχείριση των καθηγητών είναι εφικτή μόλις ο διαχειριστής επιλέξει τον σχετικό σύνδεσμο από το βασικό μενού στα αριστερά της ιστοσελίδας.

Κατά την προσθήκη ενός νέου καθηγητή ο διαχειριστής προσθέτει δεδομένα μέσω μιας φόρμας σε δύο πεδία, το όνομα και το επίθετο του διδάσκοντα. Υπάρχει ένα επιπλέον προαιρετικό πεδίο που έχει σκοπό να διευκολύνει τον διαχειριστή. Σε αυτό συμπληρώνει το όνομα χρήστη (username) του καθηγητή, εφόσον το γνωρίζει, και τα στοιχεία των παραπάνω πεδίων εμφανίζονται αυτόματα. Εφόσον και τα δύο είναι συμπληρωμένα και σωστά, τα στοιχεία προσθέτονται στη βάση δεδομένων.

Στην επεξεργασία ενός καθηγητή, είναι διαθέσιμες οι ίδιες λειτουργίες όπως και στην επεξεργασία κάθε αντικείμενου. Μόνο που σε αυτή τη περίπτωση η διαγραφή ενός καθηγητή είναι πάντα επιτυχημένη, καθώς δεν υπάρχουν αντικείμενα στη βάση δεδομένων που να συνδέονται με τον πίνακα Professors. Τα μαθήματα δανείζονται το ονοματεπώνυμο του διδάσκοντα δίχως να αποθηκεύεται το id του στον πίνακα Subjects. Έτσι αν ένας διδάσκοντας διαγραφεί, τα υπόλοιπα αντικείμενά του θα παραμείνουν στη βάση. Επιπλέον, εδώ ο πίνακας με τους διαθέσιμους καθηγητές είναι ταξινομημένος με βάση το όνομά τους.

4.1.5 Διαχείριση μαθημάτων

Κατά την προσθήκη ενός νέου μαθήματος ο διαχειριστής προσθέτει τα χαρακτηριστικά στην ειδική φόρμα, όπως φαίνεται στο Σχήμα 1. 9. Βασική προϋπόθεση είναι να προϋπάρχουν τμήματα και καθηγητές. Επίσης, το εξάμηνο πρέπει να είναι αριθμός.

Αντίστοιχα κατά την επεξεργασία εμφανίζεται μια λίστα με τα διαθέσιμα μαθήματα ταξινομημένα κατά την ημερομηνία τελευταίας τροποποίησης του αντικειμένου. Κατά την διαγραφή ενός μαθήματος, αν υπάρχουν ενότητες και διαγωνίσματα που συνδέονται με αυτό, τότε εμφανίζεται μήνυμα σφάλματος.

Κατά την επισκόπηση των ενοτήτων εμφανίζεται ένας πίνακας με τις διαθέσιμες ενότητες ταξινομημένος κατά το όνομά της καθεμιάς (όπως και στις κατηγορίες). Μόλις ο διαχειριστής πατήσει το κουμπί της επεξεργασίας εισάγεται στην φόρμα όπου μπορεί να τροποποιήσει τα χαρακτηριστικά της επιλεγείσας ενότητας. Αν θελήσει να διαγράψει μια ενότητα τότε θα πρέπει να σιγουρευτεί ότι δεν υπάρχουν καταχωρημένες ερωτήσεις και διαγωνίσματα που συνδέονται με αυτή, διαφορετικά εμφανίζεται το σχετικό μήνυμα σφάλματος.

Επεξεργασία Ενοτήτων				
Λίστα διαθέσιμων ενοτήτων Σε αυτή τη σελίδα μπορείτε να δείτε όλες τις ενότητες που έχουν προστεθεί στη βάση δεδομένων, αλλά και να επεξεργαστείτε ξεχωριστά την καθεμιά από αυτές.				
Περιγραφή	Μάθημα	Κατηγορία	Επεξεργασία	Διαγραφή
Ενότητα 1	Προγραμματισμός Διαδικτύου	Εργαστήριο 1		
Κεφάλαιο 1	Αντικειμενοστραφής Προγραμματισμός	Εργαστήριο 1		
Κεφάλαιο 3	Αντικειμενοστραφής Προγραμματισμός	Εργαστήριο 1		
Κεφάλαιο 4	Αντικειμενοστραφής Προγραμματισμός	Εργαστήριο 1		

Σχήμα 1. 10: Πίνακας διαθέσιμων ενοτήτων

4.1.8 Διαχείριση ερωτήσεων

Οι σελίδες διαχείρισης των ερωτήσεων επιτρέπουν στον διδάσκοντα πέρα από την επεξεργασία τους, την προσθήκη πολλαπλών ερωτήσεων από ένα αρχείο CSV. Το αρχείο αυτό είναι ένα κείμενο που περιέχει σε κάθε γραμμή τις τιμές των χαρακτηριστικών για μια ερώτηση, χωρισμένες με κόμμα. Τα χαρακτηριστικά αυτά αναφέρονται στην φόρμα. Αν υπάρχουν λανθασμένα δεδομένα σε κάποια γραμμή (π.χ. ο διδάσκοντας έγραψε λάθος τον τίτλο του μαθήματος) ή αν η γραμμή περιέχει μια ήδη καταχωρημένη ερώτηση, τότε εμφανίζεται ένα σχετικό μήνυμα στο πάνω μέρος της φόρμας.

Η προσθήκη μίας μόνο ερώτησης γίνεται μέσω μιας φόρμας που ο αριθμός των πεδίων της εξαρτάται από το αν η ερώτηση είναι ελεύθερου κειμένου ή πολλαπλής επιλογής. Αν ισχύει το πρώτο τότε τα πεδία είναι 9, ενώ αν στον τύπο ο διδάσκοντας επιλέξει πολλαπλή επιλογή τότε εμφανίζονται άλλα 10 πεδία, συμπεριλαμβανομένων των

επισυναπτόμενων. Ο διδάσκοντας μπορεί να επιλέξει αν η ερώτηση θα είναι διαθέσιμη για τα διαγωνίσματα και επίσης υπάρχουν 5 επίπεδα δυσκολίας και 2 γλώσσες ("GR" για ελληνικά και "ENG" για αγγλικά) τις οποίες επιλέγει σύμφωνα με την γλώσσα στην οποία γράφει την ερώτηση. Ακόμη, μόλις επιλέξει το μάθημα, την ενότητα, το επίπεδο δυσκολίας και την γλώσσα της ερώτησης, γίνεται διαθέσιμος ένας σύνδεσμος με ένα πρότυπο CSV όπου υπάρχει μια γραμμή με συμπληρωμένα αυτά τα χαρακτηριστικά. Επίσης, είναι δυνατή η προσθήκη ενός επισυναπτόμενου στην ερώτηση και σε κάθε επιλογή-απάντηση. Αυτό το επισυναπτόμενο μπορεί να έχει την κατάληξη ενός από τα παρακάτω:

'.jpg', '.jpeg', '.gif', '.bmp', '.png', '.flv', '.wav', '.mp3', '.ogg', '.swf', '.wmv', '.mp4', '.pdf', '.txt', '.xlsx', '.doc', '.docx' και να μην υπερβαίνει τα 5 megabytes. Διαφορετικά εμφανίζεται ένα μήνυμα σφάλματος και το επισυναπτόμενο δεν προστίθεται.

Η νέα ερώτηση μπορεί να προστεθεί είτε απευθείας, είτε να προηγηθεί η προεπισκόπηση της και αφού ο διδάσκων εξακριβώσει ότι τα δεδομένα είναι όπως τα θέλει, μπορεί να προσθέσει την ερώτηση.

Κατά την επισκόπηση των ερωτήσεων, ο σχετικός πίνακας είναι ταξινομημένος κατά την ημερομηνία της πιο πρόσφατης τροποποίησης της κάθε ερώτησης. Για ευκολία στην αναζήτηση ερωτήσεων υπάρχει ένα πεδίο αναζήτησης με βάση την ερώτηση, τον τύπο, το μάθημα, την ενότητα ή την δυσκολία, όπου τα αποτελέσματα εμφανίζονται σε πραγματικό χρόνο.

⚠ Η ερώτηση της γραμμής 1 είναι ήδη καταχωρημένη.

⚠ Η ερώτηση της γραμμής 2 είναι ήδη καταχωρημένη.

❗ Το μάθημα, η κατηγορία ή η ενότητα της γραμμής 3 είναι λανθασμένα.

Προσθήκη Ερωτήσεων

Σε αυτή τη σελίδα ο διαχειριστής μπορεί να προσθέσει ερωτήσεις που ανήκουν σε συγκεκριμένο μάθημα και μπορούν αργότερα να συμπεριληφθούν στα διαγωνίσματά του.

Μπορούν να προστεθούν ερωτήσεις τύπου "πολλαπλής επιλογής" ή "ελεύθερου κειμένου", όπου οι ερωτήσεις 1ου τύπου βαθμολογούνται αυτόματα, ενώ οι 2ου τύπου βαθμολογούνται απ' τον διδάσκοντα. Υπάρχει επίσης η δυνατότητα προσθήκης πολλαπλών ερωτήσεων (πολλαπλής επιλογής) μέσω ενός αρχείου CSV.

Μετά την δημιουργία των ερωτήσεων για ένα μάθημα, αυτές μπορούν να προστεθούν άμεσα σε οποιοδήποτε διαγώνισμα του μαθήματος από την **Προσθήκη Διαγωνισμάτων**.

Προσθήκη πολλαπλών ερωτήσεων χρησιμοποιώντας αρχείο CSV

Κατέβασμα προτύπου CSV

Ανέβασμα CSV με ερωτήσεις πολλαπλής επιλογής:

(μάθημα, κατηγορία, ενότητα, ερώτηση, σωστή επιλογή/ απάντηση, επιλογή 2, επιλογή 3, επιλογή 4, επιλογή 5, αριθμός δυσκολίας, γλώσσα)

Το διαχωριστικό πρέπει να είναι το , (κόμμα)

Επιλογή αρχείου

Δεν επιλέχθηκε κανένα αρχείο.

+ Προσθήκη

- Ακύρωση

Σχήμα 1. 11: Προσθήκη αρχείου CSV με λανθασμένα στοιχεία

Είναι δυνατή η τροποποίηση οποιουδήποτε χαρακτηριστικού, καθώς και η αφαίρεση ενός επισυναπτόμενου και η επιλογή νέου. Επίσης, είναι διαθέσιμος ο σύνδεσμος του προτύπου CSV όπου είναι συμπληρωμένα όλα τα στοιχεία της υπάρχουσας ερώτησης. Κατά την διαγραφή μιας ερώτησης του πίνακα προκύπτει σφάλμα και εμφανίζεται το σχετικό μήνυμα αν υπάρχουν διαγωνίσματα στα οποία ανήκει η ερώτηση ή μηνύματα φοιτητών που την αφορούν.

Στις μετρήσεις ερωτήσεων ο διδάσκοντας βλέπει έναν αναλυτικό πίνακα με τον αριθμό των συνολικών ερωτήσεων (καθώς και ξεχωριστό αριθμό των ερωτήσεων πολλαπλής επιλογής και ελεύθερου κειμένου) για κάθε ενότητα.

4.1.9 Διαχείριση διαγωνισμάτων

Η διαχείριση των διαγωνισμάτων, όπως και των προηγούμενων αντικειμένων, χωρίζεται σε προσθήκη και επεξεργασία/ διαγραφή.

Μετρήσεις Ερωτήσεων			
Σε αυτή τη σελίδα υπάρχει μια λίστα με τις συνολικές ερωτήσεις ανά ενότητα. Για κάθε ενότητα που ανήκει σε μία κατηγορία και ένα μάθημα του διδάσκοντα, φαίνεται ο αριθμός των ερωτήσεων πολλαπλής επιλογής και ελεύθερου κειμένου. Στο κάτω μέρος του πίνακα υπάρχει μια στήλη με τον συνολικό αριθμό ερωτήσεων.			
Μάθημα	Κατηγορία	Ενότητα	Αριθμός ερωτήσεων
Αντικειμενοστραφής Προγραμματισμός	Εργαστήριο 1	Κεφάλαιο 1	3 (2 πολ/πλής επιλογής/ 1 ελεύθερου κειμένου)
Αντικειμενοστραφής Προγραμματισμός	Εργαστήριο 1	Κεφάλαιο 3	4 (4 πολ/πλής επιλογής/ 0 ελεύθερου κειμένου)
Αντικειμενοστραφής Προγραμματισμός	Θεωρία 1	Κεφάλαιο 1 Εισαγωγή	2 (2 πολ/πλής επιλογής/ 0 ελεύθερου κειμένου)
Αντικειμενοστραφής Προγραμματισμός	Εργαστήριο 1	Κεφάλαιο 4	2 (0 πολ/πλής επιλογής/ 2 ελεύθερου κειμένου)
Συνολικές ερωτήσεις			
11 (8 πολ/πλής επιλογής/ 3 ελεύθερου κειμένου)			

Σχήμα 1. 12: Μετρήσεις ερωτήσεων

Κατά την προσθήκη ενός νέου διαγωνίσματος υπάρχει μια φόρμα με 15 πεδία, ένα για κάθε χαρακτηριστικό του (τίτλος, συνολικές ερωτήσεις, ημερομηνία και ώρα έναρξης κ.α.). Για την εισαγωγή της ημερομηνίας έναρξης/ λήξης εμφανίζεται ένα ημερολόγιο με javascript, ενώ για την ώρα έναρξης/ λήξης εμφανίζεται ένα χρονόμετρο. Ο διδάσκοντας προσθέτει τα δεδομένα τα οποία αρχικά περνάνε από διάφορους ελέγχους που, εκτός του να είναι συμπληρωμένα όλα τα υποχρεωτικά πεδία, είναι οι εξής:

- Οι ερωτήσεις του επιλεγμένου μαθήματος να είναι περισσότερες από τον επιλεγμένο συνολικό αριθμό ερωτήσεων.
- Τα ποσοστά ενοτήτων να έχουν συνολικό άθροισμα ίσο με 100 %.
- Τα επιλεγμένα ποσοστά ενοτήτων των ερωτήσεων σε συνδυασμό με τα επιλεγμένα επίπεδα δυσκολίας να αντιστοιχούν σε επαρκή αριθμό ερωτήσεων.
- Ο αριθμός ερωτήσεων των επιπέδων δυσκολίας που επιλέχθηκαν να μη ξεπερνά τον αριθμό των υπαρχόντων ερωτήσεων.
- Οι υπάρχοντες ερωτήσεις της επιλεγμένης κατηγορίας να είναι ίσες ή να ξεπερνούν τον επιλεγμένο συνολικό αριθμό ερωτήσεων.
- Η ημερομηνία έναρξης να είναι νεότερη από την ημερομηνία προσθήκης του διαγωνίσματος.
- Σε περίπτωση που η ημερομηνία είναι ίδια με αυτήν της προσθήκης του διαγωνίσματος, η ώρα έναρξής του να είναι νεότερη της ώρας προσθήκης του.

- Η ημερομηνία λήξης πρέπει να είναι μεγαλύτερη από την ημερομηνία έναρξης.
- Ο διαθέσιμος χρόνος εξέτασης να μην είναι μηδενικός.

Αφού ολοκληρωθούν όλοι αυτοί οι έλεγχοι, ο διδάσκοντας μεταφέρεται στη σελίδα προεπισκόπησης του διαγωνίσματος, όπου υπάρχει ένα προειδοποιητικό μήνυμα αν έχει παραβιαστεί κάποιος κανόνας από τους παραπάνω. Διαφορετικά φαίνονται όλα τα στοιχεία που μόλις συμπληρώθηκαν στη φόρμα (Σχήμα 1. 14). Εκεί ο διδάσκοντας βλέπει αν τα στοιχεία είναι όμοια με αυτά που επιθυμεί και έπειτα τα προσθέτει στη βάση δεδομένων.

✔ Τα δεδομένα καταχωρήθηκαν επιτυχώς.

Επεξεργασία Ερώτησης

Κατέβασμα προτύπου CSV

*** Μάθημα:**

Αντικειμενοστρεφής Προγραμματισμός ▼

*** Κατηγορία:**

Εργαστήριο 1 ▼

*** Ενότητα:**

Κεφάλαιο 1 ▼

Διαθέσιμη στα διαγωνίσματα: ☒

*** Επίπεδο δυσκολίας:**

2 ▼

Γλώσσα:

GR ▼

*** Τύπος Ερώτησης:**

πολλαπλής επιλογής ▼

*** Ερώτηση:**

Πότε χρησιμοποιούμε την Κληρονομικότητα;

Επισυναπτόμενα:

Επιλογή αρχείου

Δεν επιλέχθηκε κανένα αρχείο.

test.pdf

✖

Επιλογές:

*** 1) Σωστή επιλογή/ απάντηση:**

Όταν έχουμε μια ακολουθία αντικειμένων τα οποία πηγαίνουν από το γενικότερο στο ειδικότερο.

Επισυναπτόμενο 1:

Επιλογή αρχείου

Δεν επιλέχθηκε κανένα αρχείο.

Σχήμα 1. 13: Επιτυχημένη επεξεργασία ερώτησης

Προεπισκόπηση:

Πρόκειται να προσθέσετε ένα διαγώνισμα με τα εξής στοιχεία:

- Μάθημα: Αντικειμενοστραφής Προγραμματισμός
- Τίτλος διαγωνίσματος: **τεστ 1**
- Περιγραφή: **Αυτό το διαγώνισμα στοχεύει να σας δείξει πως λειτουργεί ο ιστότοπος.**
- Ποσοστά ενοτήτων: **Κεφάλαιο 1 40% Κεφάλαιο 3 50% Κεφάλαιο 4 10%**
- Κατηγορία: **Εργαστήριο 1**
- Συνολικές ερωτήσεις: **5**
- Ημερομηνία έναρξης: **2015-04-20**
- Ώρα έναρξης (Ώρα:Λεπτά): **15:40**
- Ημερομηνία λήξης: **2015-05-31**
- Ώρα λήξης (Ώρα:Λεπτά): **15:40**
- Διαθέσιμος χρόνος εξέτασης (Ώρα:Λεπτά/ Λεπτά): **00:15**
- Επίπεδα δυσκολίας: **1,2,3,4**
- Αρνητική βαθμολογία:
- Ορατή τελική βαθμολογία στους χρήστες: **Όχι**
- Διαθέσιμο στους ελεύθερους χρήστες: **Ναι**
- Επισυνυπτόμενο: **test.pdf**



Σχήμα 1. 14: Προεπισκόπηση διαγωνίσματος

Κατά την επισκόπηση των διαθέσιμων διαγωνισμάτων εμφανίζεται ένας πίνακας ταξινομημένος κατά τον τίτλο τους. Μόλις ο διαχειριστής επιλέξει την επεξεργασία ενός από αυτά, γίνεται εφικτή η τροποποίηση των χαρακτηριστικών του μέσω μιας ειδικής φόρμας. Και πάλι πριν την αποθήκευση των δεδομένων γίνονται πρώτα οι κατάλληλοι έλεγχοι (ίδιοι με πριν) και έπειτα εμφανίζεται η προεπισκόπηση τους από όπου αποθηκεύονται οι αλλαγές. Για να διαγραφεί ένα διαγώνισμα θα πρέπει να μην υπάρχουν tokens και μηνύματα που το αφορούν.

4.1.10 Διαχείριση σειράς κωδικών

Στη διαχείριση σειράς κωδικών συναντάμε ένα διαφορετικό μοτίβο λειτουργιών απ' ότι στα προηγούμενα αντικείμενα. Είναι δυνατή μόνο η προσθήκη και η διαγραφή μιας σειράς κωδικών, αλλά όχι η επεξεργασία τους. Είναι διαθέσιμες όμως άλλες δύο λειτουργίες, η επανεκτύπωση σειράς κωδικών και ο έλεγχος χρήσης τους.

Κατά την προσθήκη μιας νέας σειράς κωδικών ο διδάσκοντας επιλέγει το διαγώνισμα ενός μαθήματος και το πλήθος των κωδικών που θέλει να παραχθούν. Αμέσως μετά την προσθήκη τους στη βάση δεδομένων

παράγεται ένα PDF με πλαίσια τόσα όσα και το πλήθος κωδικών, που το καθένα περιέχει στα αριστερά μία εικόνα QR Code που "συνδέεται" με την εξέταση με χρήση του token και στα δεξιά τα στοιχεία του καθηγητή, του μαθήματος και του διαγωνίσματος, καθώς και τον σύνδεσμο για την έναρξη αυτού του διαγωνίσματος χρησιμοποιώντας το εκάστοτε token. Κατά την διαγραφή, επιλέγει απλώς την σειρά κωδικών του τεστ που θέλει να διαγράψει.

Κατά την επανεκτύπωση σειράς κωδικών, ο διδάσκοντας επιλέγει την σειρά έκδοσης που επιθυμεί και παράγεται το αντίστοιχο PDF με τα tokens που μπορεί να εκτυπώσει.

Στον έλεγχο σειράς κωδικών ο διδάσκοντας έχει δύο επιλογές: Να δει για ένα συγκεκριμένο διαγώνισμα πόσοι κωδικοί από την κάθε σειρά έκδοσης χρησιμοποιήθηκαν, ή να ελέγξει για μία συγκεκριμένη σειρά πόσοι κωδικοί έχουν χρησιμοποιηθεί από τους συνολικούς.

Έλεγχος Σειράς Κωδικών	
Λίστα χρησιμοποιημένων κωδικών	
Σε αυτή τη σελίδα μπορείτε να δείτε όλους τους κωδικούς που έχουν χρησιμοποιηθεί.	
Αριθμός σειράς	Χρησιμοποιημένοι κωδικοί
383157	15/15
272129	3/5

Σχήμα 1. 15: Έλεγχος σειράς κωδικών

4.1.11 Διαχείριση αποτελεσμάτων

Η διαχείριση των αποτελεσμάτων περιλαμβάνει τις απαντήσεις των διαγωνισμάτων, τον ενεργό έλεγχο εξέτασης, τις απαντήσεις ελεύθερου κειμένου και την επισκόπηση των μηνυμάτων.

Στις απαντήσεις των διαγωνισμάτων αρχικά υπάρχει μια λίστα των διαγωνισμάτων που είναι σε ισχύ, δηλαδή έχουν ημερομηνία έναρξης ή και λήξης παλιότερη από αυτή που ο διδάσκων βλέπει τις απαντήσεις. Από εκεί πατώντας τον σύνδεσμο "Αποτελέσματα" μεταφέρεται σε μια σελίδα όπου μπορεί να δει τις εξετάσεις που έχουν υποβάλει οι φοιτητές (είτε συνδέθηκαν με όνομα χρήστη και συνθηματικό ή χρησιμοποίησαν token, όχι όμως των ελεύθερων χρηστών). Σε κάθε

γραμμή του πίνακα που υπάρχει, φαίνονται πληροφορίες όπως το ονοματεπώνυμο και το ΑΕΜ του φοιτητή αν συνδέθηκε κανονικά, αλλιώς η σειρά έκδοσης και ο αριθμός token. Επίσης, φαίνεται η IP του χρήστη, η ημερομηνία εκκίνησης και υποβολής της εξέτασης καθώς και η συνολική διάρκεια. Ακόμη, είναι ορατός ο αριθμός των συνολικά σωστών απαντήσεων, των συνολικά λάθους και των συνολικά αβαθμολόγητων, δηλαδή των ερωτήσεων ελεύθερου κειμένου (εάν υπήρχαν) που δε βαθμολογήθηκαν από τον διδάσκοντα μέχρι εκείνη τη χρονική στιγμή. Τέλος φαίνεται ο βαθμός (ο οποίος δεν είναι τελικός αν υπάρχουν αβαθμολόγητες απαντήσεις). Στα δεξιά όλων αυτών των πληροφοριών υπάρχουν τρεις σύνδεσμοι που θα εξηγήσουμε με τη σειρά τι κάνουν, αλλά πρώτα πρέπει να αναφερθεί ότι στο κάτω μέρος του πίνακα φαίνεται ο μέσος όρος όλων των βαθμολογιών, καθώς και η μικρότερη και μεγαλύτερη βαθμολογία. Ο πρώτος σύνδεσμος, οι "απαντήσεις" οδηγούν στην τελική σελίδα των αναλυτικών απαντήσεων. Εκεί υπάρχουν και πάλι ορισμένα στοιχεία του φοιτητή, καθώς και 3 στήλες. Η πρώτη δείχνει τον αύξοντα αριθμό της κάθε ερώτησης όπως αυτή εμφανίστηκε στον φοιτητή κατά τον πρώτο βρόχο (όπως θα εξηγηθεί στη συνέχεια αν ο φοιτητής παρακάμψει ερωτήσεις, τότε αυτές εμφανίζονται ξανά στο τέλος με αύξοντα αριθμό που ξεκινάει και πάλι απ' το 1). Η δεύτερη στήλη είναι το id της ερώτησης και αποτελεί υπερσύνδεσμο προς τη φόρμα επεξεργασίας αυτής της ερώτησης, ώστε ο διδάσκοντας να έχει τη δυνατότητα να την δει ή και να την τροποποιήσει αν επιθυμεί. Η τρίτη και τελευταία στήλη περιέχει το αποτέλεσμα της απάντησης εκφραζόμενο με μια εικόνα που δηλώνει αν ήταν σωστή ή λανθασμένη/ αναπάντητη (ή παραμένει αβαθμολόγητη). Αν η απάντηση ήταν λάθος τότε δίπλα από την εικόνα υπάρχει σε παρένθεση ο αριθμός της επιλογής που υπέβαλε ο εξεταζόμενος, όπως αυτός φαίνεται στην φόρμα επεξεργασίας της ερώτησης. Ο τρίτος σύνδεσμος διαγράφει τα αποτελέσματα από τη βάση δεδομένων. Πρέπει να δίνεται μεγάλη προσοχή κατά αυτή την ενέργεια, ιδίως όταν υπάρχουν ακόμα αβαθμολόγητες απαντήσεις, αλλά κι επειδή αυτομάτως σημαίνει ότι ούτε οι φοιτητές θα είναι σε θέση να δουν τον βαθμό τους μέσω της δικής τους ιστοσελίδας. Ο δεύτερος σύνδεσμος αφορά την μεταφορά των αποτελεσμάτων σε PDF με σκοπό την εκτύπωση. Το PDF περιέχει όλες τις παραπάνω πληροφορίες και μπορεί να παραχθεί επίσης για όλους τους εξεταζόμενους ενός διαγωνίσματος από την αρχική σελίδα των απαντήσεων διαγωνισμάτων.

Όνοματεπ... φοιτητή	AEM	IP	Test ID	Σειρά έκδοσης	Αρ. Token	Συνολικά σωστές	Συνολικά λάθος	Αβαθμολό...	Βαθμός %	Μεταφορά σε PDF
Ξενοδόκης Ηλίας	118	46.12.35.78	8			3	2	0	59.92	
Αρ. ερώτησης			ID ερώτησης			Αποτέλεσμα				
1			14							
2			17							
3			19			 Απάντησε(5)				
4			16			 Αναπάντητη				
5			23							

Σχήμα 1. 16: Επισκόπηση απαντήσεων εξέτασης

Κατά τον ενεργό έλεγχο εξέτασης ο διαχειριστής αρχικά βρίσκεται σε μια σελίδα όπου βλέπει μια λίστα με τα ενεργά διαγωνίσματα, όσα δηλαδή βρίσκονται σε ισχύ και δεν έχουν λήξει ακόμα. Εκεί πατώντας το εικονίδιο "+" αυξάνεται ο διαθέσιμος χρόνος υποβολής όλων των ενεργών εξετάσεων (εκτός των ελεύθερων) κατά 5 λεπτά. Γίνεται να πατηθεί πολλές φορές, ώστε να προστεθούν περισσότερα λεπτά. Αντίθετα πατώντας το εικονίδιο "-" μηδενίζεται ο διαθέσιμος χρόνος υποβολής όλων των φοιτητών που εξετάζονται εκείνη τη στιγμή και τερματίζεται η εξέταση. Αν πατήσει το εικονίδιο της στήλης με τίτλο "έλεγχος" μπορεί να ελέγξει την κατάσταση όλων των εξετάσεων που λαμβάνουν χώρα εκείνη τη στιγμή (εκτός των ελεύθερων) και να δει τα στοιχεία του φοιτητή, τον αριθμό των εναπομείναντα ερωτήσεων, τον αριθμό ερωτήσεων ελεύθερου κειμένου, τις συνολικά σωστές και λάθος απαντήσεις και την διάρκεια εξέτασης του καθενός. Επίσης, γίνεται να αυξήσει τον χρονικό περιθώριο υποβολής ή να το μηδενίσει για τον κάθε φοιτητή ξεχωριστά.

Κατά την επισκόπηση απαντήσεων ελεύθερου κειμένου ο διδάσκοντας βλέπει έναν πίνακα όπου περιέχονται για τις εξετάσεις των διαγωνισμάτων των μαθημάτων του, οι οποίες έχουν υποβληθεί, τα στοιχεία του φοιτητή ή το token, αναλόγως τον τρόπο σύνδεσης, καθώς και τα στοιχεία του διαγωνίσματος και ο αριθμός των ερωτήσεων ελεύθερου κειμένου που έχουν απαντηθεί.

Προστέθηκαν 5 λεπτά επιτυχώς.

Ενεργός έλεγχος εξέτασης

Σε αυτή τη σελίδα ο καθηγητής μπορεί να ελέγξει για κάθε φοιτητή, που είναι ενεργός στο διαγώνισμα, τα στοιχεία του, τις ερωτήσεις που απάντησε, τον αριθμό σωστών - λάθους, τον αριθμό υπολειπόμενων ερωτήσεων, και τον υπολειπόμενο χρόνο. Επίσης κατά τη διάρκεια της εξέτασης, μπορεί να δώσει είτε για όλους τους φοιτητές, είτε για ένα τεστ, παράταση +5 λεπτά ή να ρυθμίσει "Time Over" για άμεση λήξη της εξέτασης του φοιτητή ή όλων.

Όνομα... φοιτητή	AEM	IP	Test ID	Σειρά έκδοσης	Αρ. Token	Εκκίνηση	Διάρκεια	Απαντ... ερωτή...	Υπολο... ερωτή...	Ερωτή... ελεύθε... κειμένου	Σωστές	Λάθος	+5 Λεπτά	Τέλος χρόνου
Ξενουδ... Ηλίας	118	79.103...	8			2015-03-21 18:13:59	0 ώρες & 2 λεπτά	3	2	2	3	0		

Σχήμα 1. 17: Ενεργός έλεγχος εξέτασης

Στη συνέχεια πατώντας το εικονίδιο στα δεξιά του πίνακα για έναν συγκεκριμένο φοιτητή φαίνονται οι ερωτήσεις και οι απαντήσεις που έχει δώσει. Ο διδάσκοντας έχει τη δυνατότητα να βαθμολογήσει την κάθε απάντηση ως σωστή ή λανθασμένη και πατώντας "Βαθμολόγηση" ενημερώνεται ο βαθμός της εξέτασης.

Βαθμολόγηση απαντήσεων

Τι ονομάζουμε αντικείμενα στον αντικειμενοστραφή προγραμματισμό?

αντικείμενα ονομάζουμε όλες τις οντότητες, πραγματικές και αφηρημένες.

Σωστή: ☒ Λάθος: ☐

Βαθμολόγηση

Σχήμα 1. 18: Βαθμολόγηση απαντήσεων ελεύθερου κειμένου

Κατά την επισκόπηση των μηνυμάτων ο διδάσκοντας μεταφέρεται σε μια σελίδα όπου υπάρχει ένας πίνακας με όλα τα μηνύματα που έχουν αποστείλει οι φοιτητές για κάθε διαγώνισμά του, ακόμη και καθώς οι εξετάσεις βρίσκονται σε εξέλιξη. Ο πίνακας περιέχει τις πληροφορίες του διαγωνίσματος και ένα σύνδεσμο προς το μήνυμα. Εκεί φαίνεται το μήνυμα και το id της ερώτησης που αφορά, το οποίο μπορεί να το

πατήσει ο διδάσκοντας και να μεταφερθεί στη σελίδα επεξεργασίας της ερώτησης. Έτσι, αν τροποποιήσει την ερώτηση ώστε να καλύψει το μήνυμα του φοιτητή, τότε ο τελευταίος έχοντας παρακάμψει τη συγκεκριμένη ερώτηση κατά τη διάρκεια της εξέτασης θα την ξανασυναντήσει διορθωμένη στον δεύτερο βρόχο.

Μάθημα	Διαγώνισμα	Ερώτηση	Ονοματεπώνυμ... φοιτητή	ΑΕΜ	Σειρά έκδοσης	Αρ. Token	Μήνυμα	Διαγραφή
Αντικειμενοστρ... Προγραμματισμός	τεστ 2	15	Ξενουδάκης Ηλίας	118				

Σχήμα 1. 19: Πίνακας διαθέσιμων μηνυμάτων

4.1.12 Επισκόπηση γραφημάτων

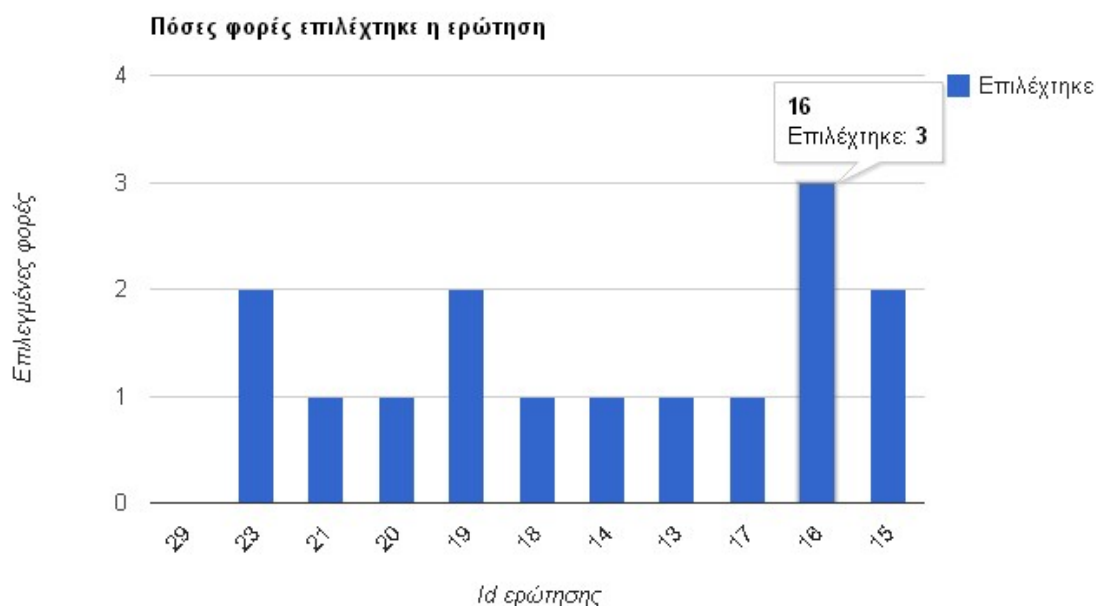
Εδώ υπάρχουν όλα τα γραφήματα που αφορούν τα αποτελέσματα. Χωρίζονται σε γραφήματα διαγωνισμάτων, γραφήματα ερωτήσεων, βαθμολογίες ανά τεστ και βαθμολογίες ανά φοιτητή. Κάθε γράφημα λόγω του ότι παράγεται σε πραγματικό χρόνο και ανανεώνεται κάθε ένα δευτερόλεπτο, κάνει τόσο χρόνο να εμφανιστεί αρχικά.

Στη πρώτη κατηγορία ο διδάσκοντας βλέπει για τα διαγωνίσματα που έχουν ήδη ξεκινήσει ήδη, γραφήματα με στήλες για όλες τις ερωτήσεις (αναφέρονται μόνο τα id) του κάθε διαγωνίσματος που αφορούν:

- Πόσες φορές επιλέχτηκε η κάθε ερώτηση.
- Πόσες φορές απαντήθηκε σωστά.
- Πόσες λάθος.
- Πόσες δεν απαντήθηκε καθόλου.

Δηλαδή συνολικά 4 διαφορετικά γραφήματα όλες τις απαραίτητες πληροφορίες, διαφορετικό χρώμα το καθένα για να ξεχωρίζουν και με τις στήλες τους να αποτελούν σύνδεσμο προς την σελίδα επεξεργασίας της κάθε ερώτησης. Τα τρία πρώτα γραφήματα ενημερώνονται αφού ξεκινήσει η εξέταση και μόλις απαντηθεί η κάθε ερώτηση, το τελευταίο δε με το που τελειώσει η εξέταση. Για τις ερωτήσεις ελεύθερου κειμένου, το γράφημα ενημερώνεται μόλις ο καθηγητής τις βαθμολογήσει.

Στα γραφήματα ερωτήσεων αφορούν τις ίδιες πληροφορίες που εξηγήθηκαν παραπάνω, αλλά για καθεμία μεμονωμένη ερώτηση από τις συνολικές που υπάρχουν στη βάση δεδομένων, την οποία επιλέγει ο διδάσκοντας από τον πίνακα των διαθέσιμων ερωτήσεων που εμφανίζεται μόλις επιλέξει αυτή την κατηγορία.



Σχήμα 1. 20: Γράφημα διαγωνίσματος

Στη κατηγορία "Βαθμολογίες ανά τεστ" υπάρχουν σε έναν πίνακα όλα τα διαγωνίσματα που έχουν ξεκινήσει ήδη. Μόλις ο καθηγητής επιλέξει ένα και πατήσει το εικονίδιο της στήλης "Γράφημα", εμφανίζεται ένα γράφημα με στήλες, όπου στον οριζόντιο άξονα αναφέρονται τα ονοματεπώνυμα των φοιτητών που ολοκλήρωσαν το διαγώνισμα και στον κατακόρυφο άξονα οι βαθμοί τους. Οι στήλες αποτελούν υπερσύνδεσμο προς τη σελίδα επισκόπησης των απαντήσεων των εξετάσεων.

Στις βαθμολογίες ανά φοιτητή, μόλις ο διδάσκοντας επιλέξει έναν συγκεκριμένο φοιτητή, εμφανίζονται δύο αντικείμενα. Στο πάνω μέρος της ιστοσελίδας εμφανίζεται ένα γράφημα με όλες τις βαθμολογίες του φοιτητή ανά εξέταση, ενώ από κάτω υπάρχει ένας πίνακας με όλες τις εξετάσεις του ώστε ο διδάσκοντας να επιλέγει όποια θέλει και να βλέπει ένα γράφημα με τον βαθμό του φοιτητή. Και πάλι η στήλη του γραφήματος μπορεί να πατηθεί και οδηγεί στη σελίδα επισκόπησης των απαντήσεων.

4.2 Λειτουργίες χρήστη

Ο απλός χρήστης του συστήματος ή αλλιώς φοιτητής έχει 3 ιδιότητες. Πρώτον, μπορεί να είναι ελεύθερος χρήστης, δηλαδή να μην έχει συνδεθεί στο σύστημα. Δεύτερον, να έχει συνδεθεί με όνομα χρήστη

και συνθηματικό. Τρίτον, να έχει εξουσιοδοτηθεί για μια εξέταση με την απόκτηση ενός κωδικού ανώνυμης περιήγησης (token).

4.2.1 Ελεύθερη εξέταση

Μια ελεύθερη εξέταση μπορεί να πραγματοποιηθεί ανά πάσα στιγμή από οποιονδήποτε πατώντας στον σχετικό σύνδεσμο του οριζοντίου μενού στο πάνω μέρος του ιστοτόπου. Οι ελεύθερες εξετάσεις δε περιέχουν τις ερωτήσεις ελεύθερου κειμένου που ανήκουν στο διαγώνισμα. Αρχικά ο χρήστης βλέπει έναν πίνακα με όλα τα διαθέσιμα διαγωνίσματα και πληροφορίες για το καθένα, όπως τον διαθέσιμο χρόνο υποβολής και την αρνητική βαθμολογία, αν υπάρχει. Αφού επιλέξει το διαγώνισμα που θέλει και πατήσει το εικονίδιο της στήλης "Έναρξη", ξεκινάει η εξέταση.

Οι ερωτήσεις εμφανίζονται μία μια σε κάθε σελίδα με τυχαία σειρά και τυχαία ταξινόμηση των επιλογών (των ερωτήσεων πολλαπλής επιλογής). Είναι αριθμημένες με βάση την σειρά εμφάνισής τους μέχρι να τελειώσουν την πρώτη φορά. Ο χρήστης μπορεί να απαντήσει σε μία ερώτηση και να πατήσει "Υποβολή" ή να μη την απαντήσει και να επιλέξει "Παράληψη" για να την παρακάμψει. Αν για κάποιο λόγο πατήσει λάθος κουμπί εμφανίζεται προειδοποιητικό μήνυμα. Αν όλα γίνουν σωστά προχωράει στην επόμενη ερώτηση. Μόλις τελειώσουν οι ερωτήσεις, εμφανίζονται ξανά όσες είχαν παραληφθεί, με τυχαία σειρά και αριθμημένες από την αρχή, δηλαδή το 1.

Αν υπάρχουν επισυναπτόμενα, τότε αυτά εμφανίζονται με το όνομα τους σε μπλε γράμματα δίπλα από την ερώτηση ή τις επιλογές της και πατώντας τα αποθηκεύονται στον υπολογιστή. Ο χρήστης δεν έχει την δυνατότητα να ξαναδεί προηγούμενη ερώτηση που έχει απαντηθεί. Καθ' όλη τη διάρκεια της εξέτασης στα αριστερά των ερωτήσεων φαίνονται τρία στοιχεία:

- Το αντίστροφο χρονόμετρο που δείχνει τον διαθέσιμο χρόνο υποβολής.
- Οι υπολειπόμενες ερωτήσεις.
- Η IP (Internet Protocol) διεύθυνση του υπολογιστή που χρησιμοποίησε ο χρήστης.

Μόλις απαντηθούν όλες οι ερωτήσεις η εξέταση τελειώνει. Εναλλακτικά, οποιαδήποτε στιγμή επιθυμεί ο χρήστης μπορεί να πατήσει το κουμπί "Υποβολή τεστ" και να τερματίσει την εξέταση. Και στις 2 περιπτώσεις ο χρήστης μεταφέρεται στην τελική σελίδα, όπου υπάρχει ένα ενημερωτικό μήνυμα ότι η εξέταση τερματίστηκε με

επιτυχία και από κάτω φαίνεται ο βαθμός της. Η ίδια εξέταση γίνεται να πραγματοποιηθεί παραπάνω από μία φορά έως ότου να μην είναι πια διαθέσιμο το διαγώνισμα.

Διαθέσιμος χρόνος 0:14:53	ΤΕΣΤ 1
Απομένουν 5 ερωτήσεις	Στις επόμενες ερωτήσεις δίνονται 5 επιλογές από τις οποίες μόνο μία είναι σωστή. Επιλέξτε την σωστή απάντηση κάνοντας κλικ στον αντίστοιχο κύκλο και πατήστε "Υποβολή" για να προχωρήσετε στην επόμενη ερώτηση ή αγνοήστε την ερώτηση πατώντας "Παράληψη". Οι ερωτήσεις που παραλήφθηκαν θα εμφανιστούν ξανά στο τέλος.
Η διεύθυνση IP σας 127.0.0.1	Πατήστε "Υποβολή Τεστ" για να τερματίσετε την εξέταση.
	Περιγραφή: Αυτό το διαγώνισμα στοχεύει να σας δείξει πως λειτουργεί ο ιστότοπος.
	Ερώτηση 1 από αυτές που απομένουν.
	<u>Ενότητα: Κεφάλαιο 1</u>
	Πότε χρησιμοποιούμε την Κληρονομικότητα;
	☐ Όταν μια κλάση κληρονομεί μια υπερκλάση.
	☐ Όταν μια κλάση κληρονομεί μια υποκλάση.
	☐ Όταν έχουμε μια ακολουθία αντικειμένων τα οποία προκύπτουν από το ειδικότερο στο γενικότερο.
	☐ Όταν έχουμε μια ακολουθία αντικειμένων τα οποία προκύπτουν από το γενικότερο στο ειδικότερο.
	☐ Όταν οι ιδιότητες ενός αντικειμένου είναι και αυτές με τη σειρά τους αντικείμενα.
	Υποβολή Παράληψη
	Υποβολή τεστ

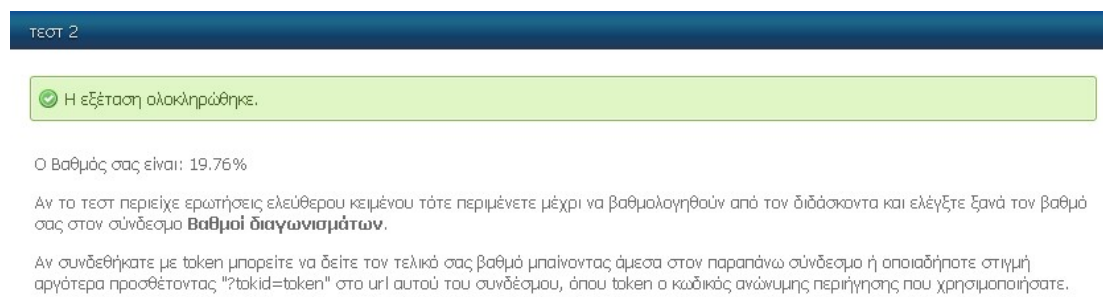
Σχήμα 1. 21: Ελεύθερη εξέταση

4.2.2 Σύνδεση και έναρξη νέας εξέτασης

Κατά τη σύνδεση (και αποσύνδεση) χρήστη ακολουθείται η ίδια διαδικασία όπως και με την σύνδεση διαχειριστή, μόνο που μόλις τα στοιχεία σύνδεσης ελεγχτούν, ο χρήστης μεταφέρεται στη δική του αρχική σελίδα. Από εκεί έχει 2 επιλογές: να ξεκινήσει μια νέα εξέταση ή να δει τις βαθμολογίες του για τις εξετάσεις που έχει ήδη υποβάλει.

Κατά την έναρξη νέας εξέτασης η διαδικασία είναι παρόμοια με την ελεύθερη εξέταση, με λίγες διαφορές που θα εξηγήσουμε σε αυτή τη παράγραφο. Αρχικά ο εγγεγραμμένος φοιτητής επιλέγει το μάθημα στο οποίο θέλει να εξεταστεί. Αν υπάρχουν διαθέσιμα διαγωνίσματα εκείνη τη στιγμή, τότε εμφανίζονται μαζί με τις πληροφορίες τους σε έναν πίνακα. Στη τελευταία στήλη του πίνακα αυτού υπάρχει ένα εικονίδιο που μόλις πατηθεί σηματοδοτεί την έναρξη της εξέτασης. Από εκεί και πέρα οι ερωτήσεις εμφανίζονται όπως και στην ελεύθερη εξέταση, με τη διαφορά ότι πλέον εμφανίζονται και ερωτήσεις ελεύθερου κειμένου. Κάτω από τον χώρο που έχει ο φοιτητής για να γράψει την απάντηση,

υπάρχει ένας μετρητής χαρακτήρων, ώστε να μη ξεπεραστεί το όριο των 1024 χαρακτήρων. Ακόμη, στα αριστερά υπάρχει ένα πλαίσιο, στο οποίο ο φοιτητής μπορεί να γράψει ένα μήνυμα που αφορά την ερώτηση και να το αποστείλει στον διδάσκοντα πατώντας "Αποστολή". Το όριο για ένα μήνυμα είναι 255 χαρακτήρες, ώστε να είναι δυνατή η αποστολή ενός μέτριου μεγέθους και κατανοητού μηνύματος, αλλά όχι υπερβολικά μεγάλου και χρονοβόρου. Στα αριστερά της σελίδας, εκτός από το βασικό μενού του χρήστη, υπάρχουν και πάλι ένα αντίστροφο χρονόμετρο, οι υπολειπόμενες ερωτήσεις και η IP του χρήστη. Μόλις τελειώσει η εξέταση (με έναν από τους τρόπους που αναφέρθηκαν στην ελεύθερη εξέταση) ο χρήστης μεταφέρεται στη σελίδα τερματισμού, όπου υπάρχει ένα μήνυμα επιβεβαίωσης και ο βαθμός του φοιτητή. Ο βαθμός αυτός δεν είναι τελικός αν στην εξέταση υπήρχαν ερωτήσεις ελεύθερου κειμένου, οι οποίες βαθμολογούνται αργότερα από τον διδάσκοντα. Έτσι ο φοιτητής θα πρέπει να ελέγξει και πάλι τον βαθμό του σε μετέπειτα χρονική στιγμή από τον σύνδεσμο "Βαθμοί διαγωνισμάτων" του βασικού μενού. Όλες οι εξετάσεις δε μπορούν να επαναληφθούν μετά την ολοκλήρωσή τους. Όποιος επιχειρήσει κάτι τέτοιο λαμβάνει ένα σχετικό μήνυμα στη σελίδα του, το οποίο ενημερώνει ότι η εξέταση έχει ήδη υποβληθεί.



Σχήμα 1. 22: Ολοκλήρωση εξέτασης

4.2.3 Εξέταση με χρήση token

Αν κάποιος φοιτητής κατέχει έναν κωδικό ανώνυμης περιήγησης ή αλλιώς token, μαζί με το QR code που το συνοδεύει (σε έντυπη μορφή, δοσμένο από τον διδάσκοντα), τότε είναι εξουσιοδοτημένος να ξεκινήσει μια ανώνυμη εξέταση μέσω αυτών. Κάθε token και το αντίστοιχο QR code αντιπροσωπεύει μία μοναδική εξέταση ενός

μαθήματος, η οποία δε γίνεται να επαναληφθεί αφότου χρησιμοποιηθούν αυτά τα στοιχεία. Επιπλέον αυτά τα στοιχεία αντικαθιστούν την κανονική σύνδεση ενός φοιτητή και μπορούν να χρησιμοποιηθούν με δύο τρόπους. Ο πρώτος είναι να σαρωθεί η εικόνα QR με μία φορητή συσκευή (π.χ. κινητό ή υπολογιστής ταμπλέτα) που είναι συνδεδεμένη στο διαδίκτυο και έτσι θα ανοίξει αμέσως η σελίδα με την πρώτη ερώτηση της συγκεκριμένης εξέτασης. Ο δεύτερος τρόπος είναι να εισαχθεί στο πρόγραμμα περιήγησης η ηλεκτρονική διεύθυνση με την token που έχει δοθεί στον φοιτητή και έχει την μορφή `"/steste.php?tokid=token"`, όπου token ο κωδικός που αποτελείται από το id του διαγωνίσματος στη βάση δεδομένων, την σειρά έκδοσης και τον τυχαίο εξαψήφιο αριθμό. Μόλις ξεκινήσει η εξέταση και μέχρι την ολοκλήρωσή της ακολουθείται ακριβώς η ίδια διαδικασία που περιγράφηκε στην παράγραφο 3.2.2.

4.2.4 Βαθμοί διαγωνισμάτων

Πατώντας τον αντίστοιχο σύνδεσμο από το βασικό μενού του χρήστη, αυτός μεταφέρεται στη σελίδα με τις βαθμολογίες του. Υπάρχουν δύο περιπτώσεις και η σελίδα διαφέρει ανάλογα με αυτές. Η πρώτη περίπτωση είναι ο φοιτητής να συνδέθηκε στο σύστημα με όνομα χρήστη και συνθηματικό και η δεύτερη να χρησιμοποίησε token.

Αν συνέβη το πρώτο τότε εμφανίζονται οι βαθμοί από όλες τις εξετάσεις του φοιτητή. Συγκεκριμένα υπάρχει ένας πίνακας που περιέχει πληροφορίες για τα διαγωνίσματα που υπέβαλλε ο φοιτητής, όπως την αρνητική βαθμολογία που έχουν, την ημερομηνία και ώρα έναρξης και υποβολής της εξέτασης, τις συνολικά σωστές, λάθος και αναπάντητες ερωτήσεις και τέλος τον βαθμό που σημείωσε ο φοιτητής. Ο πίνακας αυτός είναι ταξινομημένος κατά την ημερομηνία/ ώρα υποβολής των εξετάσεων και περιέχει το μέγιστο 15 γραμμές ανά σελίδα (όπως όλοι οι πίνακες που υπάρχουν στον ιστότοπο). Κάτω από αυτόν τον πίνακα υπάρχει ένας δεύτερος με μία γραμμή και τρεις στήλες, οι οποίες αντιπροσωπεύουν τον μέσο όρο, τη μέγιστη και την ελάχιστη βαθμολογία του φοιτητή.

Αν χρησιμοποιήθηκε token, τότε ο φοιτητής μπορεί να δει τον βαθμό της εξέτασης που έδωσε, αν επισκεφτεί τον σύνδεσμο αυτό αμέσως μόλις τελειώσει η εξέταση. Φυσικά βλέπει έναν μόνο βαθμό, οπότε δεν υπάρχει ο δεύτερος πίνακας σε αυτή τη περίπτωση. Αν όμως επισκεφτεί άλλη σελίδα του ιστοτόπου, τότε για να δει τον βαθμό της εξέτασης που έδωσε κάνοντας χρήση token θα πρέπει να προσθέσει στο URL της

σελίδας με τις βαθμολογίες το "?tokid=token", δηλαδή το URL να έχει την μορφή "../steste.php?tokid=token" όπου token ο κωδικός ανώνυμης περιήγησης που χρησιμοποιήθηκε στην εξέταση. Με αυτόν τον τρόπο ο φοιτητής μπορεί να δει και βαθμολογίες του από παλιότερες εξετάσεις τις οποίες έδωσε χρησιμοποιώντας token.

Βαθμοί διαγωνισμάτων								
Σε αυτή τη σελίδα ο εξεταζόμενος μπορεί να δει τους βαθμούς του για όλα τα διαγωνίσματα που έχει υποβάλει, καθώς και τον μέσο όρο, τη μέγιστη και ελάχιστη βαθμολογία του(εφόσον δεν έχει χρησιμοποιήσει token). Αν επιθυμεί να δει τον βαθμό του για ένα τεστ στο οποίο πήρε μέρος κάνοντας χρήση token, τότε αρκεί να προσθέσει το "?tokid=token" στο τέλος του url αυτής της σελίδας, όπου token ο κωδικός ανώνυμης περιήγησης που του δόθηκε.								
Μάθημα	Διαγώνισμα	Αρνητική βαθμολογία	Ημερομηνία εκκίνησης	Ημερομηνία υποβολής	Συνολικά σωστές	Συνολικά λάθος	Αβαθμολόγητες	Βαθμός %
Αντικειμενοστρ... Προγραμματισμός	τεστ 2	0,08	2015-03-28 17:33:53	2015-03-28 17:34:40	1	3	1	19.76
Μέσος όρος %			Μέγιστη βαθμολογία %			Ελάχιστη βαθμολογία %		
19.76			19.76			19.76		

Σχήμα 1. 23: Βαθμοί διαγωνισμάτων

Εδώ τελειώνει η περιγραφή των λειτουργιών του ιστότοπου. Ο αναγνώστης έχει πλέον μια πλήρη εικόνα της ανάπτυξης του ιστότοπου, καθώς έχει ήδη γνώση των τεχνολογιών που χρησιμοποιήθηκαν, των απαιτήσεων και προδιαγραφών του συστήματος, της βάσης δεδομένων που δημιουργήθηκε και των βασικότερων σημείων του κώδικα που αναπτύχθηκε. Μετά απ' όλα αυτά ακολουθεί μια σύνοψη του παρόντος έργου καθώς και ιδέες για μελλοντικές επεκτάσεις.

Κεφάλαιο 5

Επίλογος

Στο παρόν κεφάλαιο θα συνοψίσουμε ότι έχει ειπωθεί μέχρι στιγμής. Θα γίνει αναφορά στα συμπεράσματα που προέκυψαν από την υλοποίηση της παρούσας εργασίας και στις μελλοντικές επεκτάσεις που μπορούν να λάβουν χώρα μέσω των αποτελεσμάτων της.

5.1 Σύνοψη και συμπεράσματα

Στη παρούσα διπλωματική εργασία αναπτύχθηκε ένα σύστημα διαδικτυακής εξέτασης φοιτητών. Πρόκειται για έναν ιστότοπο που υλοποιήθηκε γράφοντας κώδικα σε γλώσσα PHP, η οποία σε επικοινωνία με την JavaScript και μια βάση δεδομένων MySQL δημιουργεί δυναμικό περιεχόμενο στις σελίδες του. Η πρόσβαση στη MySQL και η διαχείριση των δεδομένων της έγινε μέσω της επέκτασης της PHP, PDO, η οποία χρησιμοποιεί έτοιμες δηλώσεις με παραμετροποιημένα ερωτήματα, πράγμα που την καθιστά κατάλληλη για την αποφυγή της κακόβουλης τεχνικής SQL injection πρώτης τάξης. Ο ιστόχωρος σχεδιάστηκε για να παρέχει στους διδάσκοντες ενός Πανεπιστημίου εργαλεία διαχείρισης αντικειμένων όπως είναι τα μαθήματα, οι ερωτήσεις και τα διαγωνίσματα. Καθένα από αυτά τα αντικείμενα σχετίζεται άμεσα με ένα άλλο. Έτσι δημιουργήθηκε μια βάση δεδομένων με 14 πίνακες, οι οποίοι συνδέονται μεταξύ τους με εξωτερικά κλειδιά. Αυτό που ξεχωρίζει το σύστημα από άλλα παρόμοια που έχουν αναπτυχθεί είναι τα εξής 3 χαρακτηριστικά του:

- Οι εξετάσεις περιέχουν ερωτήσεις πολλαπλής επιλογής αλλά και ελεύθερου κειμένου. Οι πρώτες βαθμολογούνται αυτόματα ενώ οι τελευταίες από τον διδάσκοντα.
- Κατά τη διάρκεια μιας εξέτασης οι ερωτήσεις επιλέγονται και εμφανίζονται μία μια με τυχαία σειρά σε κάθε σελίδα. Έτσι ένας φοιτητής δε θα δώσει ποτέ την ίδια εξέταση ενός διαγωνίσματος με έναν άλλο.
- Οι φοιτητές έχουν τη δυνατότητα να δώσουν μια εξέταση ανώνυμα με τη χρήση ενός κωδικού ανώνυμης περιήγησης (token) που

εισάγεται στο url της σελίδας εξέτασης και δίνεται εκτυπωμένο σε αυτούς.

Άλλα βασικά χαρακτηριστικά του ιστότοπου είναι ότι ο διδάσκοντας μπορεί στο διάστημα που υπάρχουν ενεργές εξετάσεις να αυξήσει ή να μηδενίσει τον διαθέσιμο χρόνο υποβολής ενός ή όλων των φοιτητών που δίνουν την εξέταση. Επίσης, μπορεί να δει γραφήματα πραγματικού χρόνου που αφορούν τις βαθμολογίες των φοιτητών και των διαγωνισμάτων, καθώς και συγκεντρωτικούς αριθμούς σχετικά με το πόσες φορές απαντήθηκαν σωστά ή λάθος οι ερωτήσεις ενός διαγωνίσματος. Τέλος, όταν ολοκληρωθεί μια εξέταση ο διδάσκοντας μπορεί να ελέγξει για όσους έδωσαν την εξέταση, αναλυτικά ποιες ερωτήσεις επιλέχτηκαν με τη σειρά που επιλέχτηκαν και τι απάντησε ο φοιτητής στη καθεμία. Μπορεί να βαθμολογήσει τις απαντήσεις ελεύθερου κειμένου μαρκάροντας τες ως σωστές ή λάθος και ακόμη, να παράγει ένα PDF με τα αποτελέσματα.

Κατά την ανάπτυξη του συστήματος υπήρχαν αρκετές δυσκολίες, που κατά κύριο λόγο αφορούσαν την σωστή επικοινωνία μεταξύ PHP και JavaScript ή AJAX και την δυναμική άντληση δεδομένων και ενημέρωση των σελίδων, αλλά ξεπεράστηκαν σχετικά εύκολα. Ακόμη, υπήρξε μεγάλη δυσκολία κατά την υλοποίηση της σελίδας εξέτασης και πιο συγκεκριμένα στη παραγωγή ερωτήσεων με τυχαία σειρά. Το πρόβλημα που παρουσιάστηκε είναι ότι οι ερωτήσεις δεν εμφανίζονταν όπως θα περιμέναμε, αλλά κάποιες επαναλαμβάνονταν παραπάνω από μία φορά. Τελικά υπήρξε ικανοποιητική λύση, όμως έπρεπε να γίνει ένα είδος συμβιβασμού. Για τη σωστή λειτουργία του κώδικα παραγωγής των σελίδων εξέτασης, θυσιάστηκε ο σωστός τρόπος χρήσης της PDO με αποτέλεσμα να μη χρησιμοποιούνται "δεσμευμένοι" παράμετροι. Ως εκ τούτου, ο συγκεκριμένος κώδικας είναι ευάλωτος σε επιθέσεις SQL injection. Όμως ενισχύσαμε την ασφάλεια με άλλες τεχνικές όπως το φιλτράρισμα των στοιχείων του χρήστη κατά την είσοδό του στο σύστημα, την επικύρωση των δεδομένων εισόδου και την εξάλειψη ανεπιθύμητων χαρακτήρων από αυτή.

Ας δούμε τώρα μερικές μετρικές του συστήματος που υλοποιήθηκε:

- Δημιουργήθηκαν 143 αρχεία.
- Γράφτηκαν 23,755 γραμμές κώδικα (με σχόλια).
- Χρησιμοποιήθηκαν 26 συναρτήσεις JavaScript.

Ο ιστότοπος δοκιμάστηκε εκτενώς σε πραγματικές συνθήκες. Υπήρξαν 4 χρήστες (ένας καθολικός διαχειριστής, δύο απλοί-διδάσκοντες και ένας φοιτητής), που δημιούργησαν το μέγιστο 55 εγγραφές στη βάση δεδομένων με 11 ερωτήσεις, 3 διαγωνίσματα, 10 tokens και 3

αποτελέσματα εξετάσεων. Ο κώδικας ελέγχτηκε με XHTML validator δίχως λάθη. Το σύστημα έτρεξε τοπικά σε Windows XP αλλά και σε διακομιστή με FreeBSD χωρίς κανένα πρόβλημα και λειτουργεί αποδοτικά σε όλα τα σύγχρονα προγράμματα περιήγησης Ιστού (Chrome, Mozilla Firefox και Internet Explorer). Έτσι συμπεραίνουμε ότι ο ιστότοπος είναι καινοτόμος, ασφαλής σε ικανοποιητικό βαθμό, αποδοτικός και ανταγωνίζεται πλήρως παρόμοια συστήματα που κυκλοφορούν στο διαδίκτυο ή διατίθενται στην αγορά με αμοιβή.

5.2 Μελλοντικές επεκτάσεις

Ο ιστόχωρος που αναπτύχθηκε είναι πλήρης και αυτάρκης, ωστόσο θα μπορούσαν να προστεθούν περαιτέρω λειτουργίες στο μέλλον για να εξυπηρετηθούν άλλες ανάγκες πέρα από αυτές που καλύπτονται. Ακόμη, μπορούν να υπάρξουν μελλοντικές εργασίες που βασίζονται στη παρούσα. Ο ιστόχωρος είναι ευέλικτος ώστε να διευκολύνει τις αλλαγές.

5.2.1 Υποστήριξη πολλαπλών σωστών απαντήσεων

Μια λειτουργία που θα μπορούσε να προστεθεί στο μέλλον είναι να υποστηρίζονται παραπάνω από μία σωστές απαντήσεις για τις ερωτήσεις πολλαπλής επιλογής με ιδανικά δύο. Έτσι αν και οι δύο έχουν επιλεχτεί, η απάντηση θα θεωρείται σωστή, ενώ σε αντίθετη περίπτωση λάθος. Αν έχει επιλεχτεί μόνο η μία, τότε ο φοιτητής θα λαμβάνει το μισό της βαθμολογίας. Αυτή η προσθήκη είναι χρήσιμη στη περίπτωση που υπάρχουν ερωτήσεις που δεν έχουν αριθμητική απάντηση, αλλά αλφαριθμητική ή οποία διαφέρει ανάλογα με την περίπτωση. Για να γίνει αυτή η αλλαγή, πρέπει αρχικά το πεδίο answer του πίνακα Questions να τροποποιηθεί ώστε να είναι τύπου enum και να περιέχει τις 2 τα id των 2 πλέον σωστών απαντήσεων. Έπειτα θα άλλαζε ο τρόπος επιλογής της απάντησης από radio button σε checkbox που έχει ως τιμή το id της απάντησης και κατά την επεξεργασία της θα γινόταν έλεγχος για το αν έχουν επιλεχθεί τα σωστά id συνδυασμένα ή μη.

5.2.2 Δημιουργία εφαρμογής για Android

Ένα χρήσιμο μελλοντικό έργο που θα μπορούσε να πραγματοποιηθεί είναι η δημιουργία μιας εφαρμογής για συσκευές με λειτουργικό Android, που θα είναι βασισμένη στο παρόν σύστημα. Αν και ο ιστότοπος που αναπτύχθηκε είναι προσβάσιμος από φορητές συσκευές που είναι συνδεδεμένες στο διαδίκτυο, θα ήταν αρκετά χρήσιμο να υπήρχε μια ξεχωριστή εφαρμογή όπου οι καθηγητές θα ανέβαζαν μερικές απλές ερωτήσεις πολλαπλής επιλογής και οι φοιτητές θα τις απαντούσαν μέσω της δικής τους συσκευής. Οι τελευταίοι θα μπορούσαν επίσης να ενημερώσουν απλώς τη βάση δεδομένων της εφαρμογής και έπειτα να πάρουν μέρος σε μια εξέταση δίχως να είναι συνδεδεμένοι στο διαδίκτυο. Για να γίνει κάτι τέτοιο πρέπει να χρησιμοποιηθεί η δομή του υπάρχοντος ιστοχώρου ώστε να γραφτεί η εφαρμογή Android μέσω ενός πακέτου ανάπτυξης λογισμικού (SDK) Android, όπως είναι το Eclipse.

Βιβλιογραφία

- [1]. Webgift. *Διαφορές στατικής με δυναμική ιστοσελίδα*. [Ηλεκτρονικό] Νοέμβριος 2014. <https://www.webgift.gr/eblog/istologio-ipiresies-diadiktou/diafores-statikis-dinamikis-istoselidas.html>.
- [2]. Wikipedia. *Web browser*. [Ηλεκτρονικό] Νοέμβριος 2014. https://el.wikipedia.org/wiki/Web_browser.
- [3]. **J.C., Meloni**. *Μάθετε PHP, MySQL και APACHE Όλα σε Ένα, Τέταρτη Έκδοση*. 2009.
- [4]. **Ακης, Καργιοφύλλης**. Wlearn: πρόσβαση στη γνώση. *Γενικά για την XHTML*. [Ηλεκτρονικό] Δεκέμβριος 2014. <http://www.wlearn.gr/index.php/home-xhtml>.
- [5]. **Γαβαλάς Δαμιανός**. Τμήμα Πολιτισμικής Τεχνολογίας & Επικοινωνίας-Πανεπιστήμιο Αιγαίου-Δικτυακά Πολυμέσα II. *Βασικές έννοιες σχεδιασμού στο web*. [Ηλεκτρονικό] Δεκέμβριος 2014. http://dgavalas.ct.aegean.gr/DP_II/slides/DP_II_02.pdf.
- [6]. **Μιχαήλια Κομβούτη-Βέρου**. Οικονομικό Πανεπιστήμιο Αθηνών-Τμήμα Πληροφορικής-Εισαγωγή στην Επιστήμη των Υπολογιστών. *Εισαγωγή στο CSS*. [Ηλεκτρονικό] Δεκέμβριος 2014. http://pages.cs.aueb.gr/courses/epl131/files/CSS_notes.pdf.
- [7]. Προγραμματισμός στο διαδίκτυο. *Τι είναι η PHP?* [Ηλεκτρονικό] Ιανουάριος 2015. <http://webprog.pblogs.gr/2007/08/ti-einai-h-php.html>.
- [8]. ΜΑΘΗΜΑΤΑ (TUTORIALS) ΑΠΟ ΤΟΝ Σ.Τ.Ε.Μ.Π. ΚΑΙ ΤΑ ΚΕΝΤΡΑ ΠΛΗ.ΝΕ.Τ. ΚΟΖΑΝΗΣ & ΠΤΟΛΕΜΑΪΔΑΣ. *Η Γλώσσα Προγραμματισμού PHP (Αναλυτικά)*. [Ηλεκτρονικό] Ιανουάριος 2015. <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-Php-Analytical.html>.
- [9]. AUSWEB Web Hosting. *Features of PHP5*. [Ηλεκτρονικό] Ιανουάριος 2015. <https://ausweb.com.au/tutorials/features-of-php5>.
- [10]. Wikipedia. *JavaScript*. [Ηλεκτρονικό] Ιανουάριος 2015. <https://el.wikipedia.org/wiki/JavaScript>.

- [11]. studentguru. *Α0. Εισαγωγή στην Javascript (σύνταξη,τρόποι εισαγωγής Javascript σε Html,τελεστές)*. [Ηλεκτρονικό] Μάιος 2015. <http://studentguru.gr/w/tutorials/a0-javascript-javascript-html>.
- [12]. wikibooks. *Βασικές γνώσεις PHP και MySQL*. [Ηλεκτρονικό] Ιανουάριος 2015. https://el.wikibooks.org/wiki/Βασικές_γνώσεις_PHP_και_MySQL.
- [13]. **R. Elmarsi, S.B Navathe**. *Θεμελιώδεις αρχές συστημάτων Βάσεων Δεδομένων, Τόμος Α, 3η Έκδοση*. s.l. : Εκδόσεις Δίαυλος, 2001.
- [14]. **Abraham Silberschatz, HenryF. Korth, S. Sudarshan**. *Συστήματα Βάσεων Δεδομένων «Η πλήρη θεωρία των βάσεων δεδομένων», 6η έκδοση*. s.l. : Εκδόσεις Μ.Γκιούρδας, 2011.
- [15]. Wikibooks. *Βασικές γνώσεις PHP και MySQL/Εισαγωγή στην MySQL*. [Ηλεκτρονικό] Φεβρουάριος 2015. https://el.wikibooks.org/wiki/Βασικές_γνώσεις_PHP_και_MySQL/Εισαγωγή_στην_MySQL.
- [16]. **Σταυρακούδης, Αθανάσιος**. *Βάσεις δεδομένων και SQL*. s.l. : Εκδόσεις Κλειδάριθμος, 2010.
- [17]. **Wurzer, Erik**. *Why you Should be using PHP's PDO for Database Access*. [Ηλεκτρονικό] Φεβρουάριος 2015. <http://code.tutsplus.com/tutorials/why-you-should-be-using-phps-pdo-for-database-access--net-12059>.
- [18]. *Prepared statements and stored procedures*. [Ηλεκτρονικό] Φεβρουάριος 2015. <http://php.net/manual/en/pdo.prepared-statements.php>.
- [19]. **Β. Γερογιάννης, Γ. Κακαρόντζας, Α. Καμέας, Γ. Στάμελος, Π. Φιτσιλής**. *Αντικειμενοστρεφής Ανάπτυξη Λογισμικού με τη UML*. s.l. : Εκδόσεις Κλειδάριθμος, 2006.
- [20]. **maximem**. *Validate and sanitize data with PHP Filter, part 1*. [Ηλεκτρονικό] Απρίλιος 2015. <http://phpchunk.net/2011/06/validate-sanitize-data-php-filter-part-1>.
- [21]. **Ullrich, Johannes**. *How To Fix SQL Injection: PHP*. [Ηλεκτρονικό] Απρίλιος 2015. <http://software-security.sans.org/developer-how-to/fix-sql-injection-in-php-using-prepared-statements>.
- [22]. **Hazan, Ilan**. *HOW TO SELECT RANDOM ROWS IN MYSQL*. [Ηλεκτρονικό] Ιανουάριος 2015. <http://www.rndblog.com/how-to-select-random-rows-in-mysql>.

[23]. W3Schools. *SQL UNION Operator*. [Ηλεκτρονικό] Οκτώβριος 2014.
http://www.w3schools.com/sql/sql_union.asp.