



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ &
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



Σχεδίαση και Υλοποίηση ιστοσελίδας για διαχείριση προγράμματος Πανεπιστημίου

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Κοσμάς Στάμος

Επιβλέπων: Μηνάς Δασυγένης

Αναπληρωτής Καθηγητής



HELLENIC DEMOCRACY
UNIVERSITY OF WESTERN MACEDONIA

FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRICAL &
COMPUTER ENGINEERING



Design and implementation of a website for the management of a university timetable

THESIS

Kosmas Stamos

Supervisor: Minas Dasygenis

Associate Professor



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
& ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

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 Διπλωματική Εργασία με τίτλο **“Σχεδίαση και Υλοποίηση ιστοσελίδας για διαχείριση προγράμματος Πανεπιστημίου”** καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Μηνά Δασυγένη 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, αρχεία 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) Κοσμάς Στάμος, Μηνάς Δασυγένης, 2025 Κοζάνη

Υπογραφή Φοιτητή: _____

Περίληψη

Η σύνταξη του προγράμματος ενός τμήματος πανεπιστημίου είναι μια περίπλοκη διαδικασία που απασχολεί τους καθηγητές και τη γραμματεία των εκάστοτε σχολών στην αρχή κάθε εξαμήνου, λόγω της περίπλοκης φύσης του προβλήματος και των πολλών παραμέτρων που πρέπει να ληφθούν υπόψιν.

Οι καθηγητές έχουν τις δικές τους προτιμήσεις για ώρες και μέρες διδασκαλίας, το κάθε προπτυχιακό πρόγραμμα σπουδών μπορεί να έχει από 4 έως 5 έτη διδασκαλίας ανάλογα με τη σχολή, ενώ το καθένα από αυτά μπορεί να απαρτίζεται από 4 και πάνω εξάμηνα. Επίσης κάθε πανεπιστήμιο έχει συγκεκριμένο αριθμό αιθουσών και ακόμα πιο περιορισμένο αριθμό «ειδικών αιθουσών». Οι αίθουσες αυτές μπορεί να έχουν υπολογιστές και ειδικό εξοπλισμό και αποτελούν πόρο που πρέπει να κατανομηθεί με σύνεση. Από τη μεριά των φοιτητών, το πρόγραμμα θα πρέπει να δομηθεί έτσι ώστε να μην είναι εξοντωτικό κάποιες μέρες της εβδομάδας και εντελώς κενό άλλες.

Συνεπώς η διαδικασία σχεδιασμού του πανεπιστημιακού προγράμματος, πέρα από τη σωστή διαχείριση περιορισμένων πόρων, απαιτεί καλή επικοινωνία μεταξύ των καθηγητών και σύνεση όσον αφορά την επιβάρυνση των μαθητών. Ένα καλά δομημένο πληροφοριακό σύστημα ή ιστοσελίδα, θα μπορούσε να απλουστεύσει τη διαδικασία αισθητά, μιας και θα μπορούσε να μειώσει την ανάγκη συνεχούς επικοινωνίας μεταξύ καθηγητών και γραμματείας για τη διαθεσιμότητα αιθουσών ή τυχών συγκρούσεις με άλλα μαθήματα για τις ίδιες ώρες ή αίθουσες. Η γραμματεία επίσης θα έχει πλήρη εικόνα των «συγκρούσεων» και θα μπορεί να τις διευθετήσει με μεγαλύτερη ευκολία. Επιπλέον, με τη προσθήκη διάφορων δυνατοτήτων, όπως δήλωση αναπληρώσεων, καθορισμό απαιτήσεων μαθημάτων, κράτηση ιστορικού η διαχείριση των πόρων θα γινόταν πολύ πιο εύκολη.

Για τους μαθητές που συχνά μπορεί να παρακολουθούν μαθήματα από πολλαπλά έτη ταυτόχρονα, η συγκέντρωση των μαθημάτων τους σε μία σελίδα καθώς και η δυνατότητα ενημέρωσης τους για οποιαδήποτε αναπλήρωση μαθήματος, θα ήταν αρκετά χρήσιμη για την ομαλή διαχείριση του εξαμήνου τους. Τέλος, η ίδια σελίδα θα μπορεί να χρησιμοποιηθεί και για τη δημιουργία του προγράμματος εξεταστικής περιόδου.

Λέξεις Κλειδιά: πρόγραμμα εξαμήνου, επίλυση συγκρούσεων ωρών, γραμματεία, πρόγραμμα εξεταστικής, καθηγητές, μαθητές, διαθεσιμότητα αιθουσών

Abstract

The creation of a timetable schedule for a university is a complex process that stumbles professors and university administrations of each department at the beginning of the semester, because of the complex nature of the problem and the many parameters that need to be accounted for.

The professors have their own preferences for specific time blocks and days that they want to teach, each semester might have four or five academic years per department, while each one of them might be comprised from four and above subjects. Moreover, every university has a set number of lecture halls and an even more limited number of laboratories. Those laboratories might have computers or special equipment and constitute a resource that needs to be wisely allocated. From the students' point of interest, the schedule should be constructed in a way where the hours are evenly distributed among the days of the school week.

Consequently, the designing process of the university timetable schedule, beyond the proper management of limited resources, requires good communication between professors and caution when it comes to student workload allocation. A well-structured informatics system or website could simplify the process considerably, since it would enable less and better communication between the professors and the school admin for the vacancy of classrooms and for the adjudication of time conflicts between subjects or classrooms. The admin will also have the full picture of those conflicts and will be in a better position to resolve them in a swift manner. Furthermore, with the addition of various extra features like course substitutions, subject requirements determination, schedule history, the management of resources would become easier.

For the students that often attend classes of multiple academic years simultaneously, the gathering of subjects in one website and the ability to stay informed about course substitutions would prove quite helpful for the smooth organization of their semesters. Finally, the same webpage could be used to create the schedule of the examination period.

Keywords: semester schedule, time conflict resolution, secretariat, exam schedule, professors, students, classroom availability

Ευχαριστίες

Αφιερώνεται στην οικογένειά μου και στην ανιψιά μου που τώρα που εγώ τελειώνω την ακαδημαϊκή μου πορεία, αυτή αρχίζει τη δική της.

Κοσμάς Στάμος
Σεπτέμβριος 2025

Περιεχόμενα

Περίληψη.....	5
Abstract	6
Ευχαριστίες.....	7
Περιεχόμενα	8
Κατάλογος Εικόνων	11
Κατάλογος Πινάκων.....	12
Κεφάλαιο 1: Εισαγωγή.....	13
1.1 Υπάρχουσες λύσεις και σχετική εργασία	13
1.2 Συνεισφορά και καινοτομία της εργασίας.....	17
1.3 Διάρθρωση της εργασίας	18
Κεφάλαιο 2: Θεωρητικό υπόβαθρο και εργαλεία υλοποίησης.....	19
2.1 Τεχνολογίες Λογισμικού.....	19
2.1.1 HTML	20
2.1.2 CSS	21
2.1.3 PHP	22
2.1.4 MySQL.....	23
2.1.5 JavaScript	24
2.1.6 AJAX (Asynchronous JavaScript and XML)	25
2.1.7 PDO (PHP Data Objects)	25
2.1.8 Node.js	26
2.1.9 Composer	26
2.2 Εργαλεία λογισμικού	27
2.2.1 XAMPP.....	27
2.2.2 Visual Studio Code	28
2.3 Πρόσθετες βιβλιοθήκες λογισμικού.....	29
2.4 Συμπεράσματα.....	30
Κεφάλαιο 3: Ανάλυση και σχεδίαση συστήματος.....	32
3.1 Ανάλυση απαιτήσεων	32
3.2 Ρόλοι και Δικαιώματα Πρόσβασης.....	33
3.2.1 Διαχειριστής	33
3.2.2 Γραμματεία.....	34
3.2.3 Καθηγητής.....	34
3.2.4 Φοιτητής.....	34
3.3 Σενάρια χρήσης περίπτωσης	34
3.3.1 Σενάριο δήλωσης χρονοθυρίδας μαθήματος.....	35
3.3.2 Σενάρια Διαχείρισης Χρήστη	36
3.3.3 Σενάριο Διαχείρισης πόρων πανεπιστημίου	37
3.4 Βάση Δεδομένων.....	38
3.4.1 Πίνακα users.....	38
3.4.2 Πίνακας roles	39

3.4.3 Πίνακας userroles.....	39
3.4.4 Πίνακας semesters.....	40
3.4.5 Πίνακας schools.....	40
3.4.6 Πίνακας rooms.....	41
3.4.7 Πίνακας notifications	42
3.4.8 Πίνακας classsschedules	43
3.4.9 Πίνακας emergencyclassrequests	44
3.4.10 Πίνακας classes.....	45
3.4.11 Πίνακας studentclasses.....	46
3.4.12 Πίνακας tutoravailability	46
3.4.13 Πίνακας tutorclasses.....	48
3.4.14 Πίνακας classschedule_rooms	48
3.4.15 Πίνακας timetables.....	49
3.4.16 Πίνακας classschedule_users	50
3.4.17 Πίνακας timetable_classes.....	50
3.4.18 Πίνακας userschools.....	51
3.4.19 Πίνακας tutoravailability_rooms.....	51
3.4.20 Πίνακας emergencyclassrequests_rooms.....	52
3.4.21 Πίνακας emergencyclassrequests_timetables	52
3.4.23 Πίνακας directions	53
3.5 Τεχνικές εξασφάλισης ασφάλειας	55
3.5.1 Χρήση ενιαίου αρχείου db_conn.php.....	55
3.5.2 Ασφάλεια κωδικών πρόσβασης	55
3.5.3 Ασφάλεια εισαγόμενων δεδομένων	56
3.5.4 Χρήση PDO και προστασία από SQL Injection.....	57
3.5.5 Εξασφάλιση χρήσης HTTPS για ασφάλεια της σελίδας.....	57
3.6 Συμπεράσματα.....	58
Κεφάλαιο 4: Υλοποίηση και δυνατότητες οντοτήτων.....	60
4.1 Δυνατότητες διαχειριστή και γραμματείας.....	60
4.1.1 Διαχείριση Τμήματος	63
4.1.2 Δημιουργία Προγράμματος	67
4.1.3 Διαχείριση Χρηστών.....	68
4.1.4 Αιτήματα καθηγητών	70
4.1.5 Διαχείριση Αναπληρώσεων	71
4.2 Δυνατότητες ρόλου καθηγητή.....	71
4.2.1 Τα Μαθήματά μου	71
4.2.2 Δήλωση Διαθεσιμότητας.....	72
4.2.3 Δήλωση Εκτάκτου μαθήματος.....	74
4.3 Δυνατότητες φοιτητή.....	75
4.4 Συμπεράσματα.....	76
Κεφάλαιο 5: Συμπεράσματα και προτάσεις.....	78
5.1 Ανάλυση S.W.O.T.....	78
5.2 Δυσκολίες και προβλήματα κατά την υλοποίηση.....	80
5.3 Μετρικά συστήματος.....	81

5.4 Συμπεράσματα.....	82
Παράρτημα Α – Οδηγοί Εγκατάστασης Προγραμματιστικών Εργαλείων	84
Α.1 Εγκατάσταση XAMPP	84
Α.2 Εγκατάσταση Visual studio Code και επεκτάσεων.....	86
Α.3 Εγκατάσταση διαχειριστή εξαρτήσεων Composer.....	86
Α.4 Εγκατάσταση περιβάλλοντος διαχείρισης πακέτων Node.js	87
Α4.1 Δημιουργία & Διαχείριση Έργου Node.js στο Visual Studio Code	88
Βιβλιογραφία	89

Κατάλογος Εικόνων

ΕΙΚΟΝΑ 1 FET UI.....	14
ΕΙΚΟΝΑ 2 FET TIMETABLING UI	14
ΕΙΚΟΝΑ 3 UNITIME UI.....	15
ΕΙΚΟΝΑ 4 UNITIME TIMETABLE SOLVER	16
ΕΙΚΟΝΑ 5 ΒΑΣΙΚΗ ΔΟΜΗ ΣΕΛΙΔΑΣ HTML	20
ΕΙΚΟΝΑ 6 ΕΤΙΚΕΤΑ LINK ΓΙΑ ΕΙΣΟΔΟ CSS ΑΡΧΕΙΟΥ	21
ΕΙΚΟΝΑ 7 ΠΑΡΑΔΕΙΓΜΑ ΑΛΛΗΛΕΠΙΔΡΑΣΗΣ CSS ΚΑΙ HTML	22
ΕΙΚΟΝΑ 8 ΠΑΡΑΔΕΙΓΜΑ ΕΝΣΩΜΑΤΩΣΗΣ PHP ΣΕ HTML	23
ΕΙΚΟΝΑ 9 ΔΙΑΧΕΙΡΙΣΗ ΕΤΙΚΕΤΩΝ DIV ΜΕ JAVASCRIPT	24
ΕΙΚΟΝΑ 10 ΤΡΟΠΟΣ ΛΕΙΤΟΥΡΓΙΑΣ AJAX	25
ΕΙΚΟΝΑ 11 ΧΑΜΡΡ CONTROL PANEL	28
ΕΙΚΟΝΑ 12 ΔΥΝΑΤΟΤΗΤΕΣ ΚΑΙ ΔΙΚΑΙΩΜΑΤΑ ΧΡΗΣΤΩΝ	35
ΕΙΚΟΝΑ 13 ΣΕΝΑΡΙΟ ΔΗΛΩΣΗΣ ΧΡΟΝΟΘΥΡΙΔΑΣ ΜΑΘΗΜΑΤΟΣ	36
ΕΙΚΟΝΑ 14 ΣΕΝΑΡΙΑ ΔΙΑΧΕΙΡΙΣΗΣ ΔΙΑΠΙΣΤΕΥΤΗΡΙΩΝ ΧΡΗΣΤΗ	37
ΕΙΚΟΝΑ 15 ΣΕΝΑΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ ΠΟΡΩΝ ΠΑΝΕΠΙΣΤΗΜΙΟΥ.....	38
ΕΙΚΟΝΑ 16 ΣΧΗΜΑ ΒΑΣΗΣ ΕΦΑΡΜΟΓΗΣ	54
ΕΙΚΟΝΑ 17 DB_CONN.PHP	55
ΕΙΚΟΝΑ 18 FILTER_VAR AND OTHER DATA INTEGRITY HELPERS	56
ΕΙΚΟΝΑ 19 HTMLSPECIALCHARS PROVISION WITH UTF-8 ENCODING	56
ΕΙΚΟΝΑ 20 PREPARE-BINDPARAM-EXECUTE PDO PATTERN.....	57
ΕΙΚΟΝΑ 21 ENFORCE_HTTPS.PHP	58
ΕΙΚΟΝΑ 22 ΜΠΑΡΑ ΠΛΟΗΓΗΣΗΣ	60
ΕΙΚΟΝΑ 23 ΠΡΟΓΡΑΜΜΑ - ΣΕΛΙΔΑ ΕΙΣΟΔΟΥ ΟΛΩΝ ΤΩΝ ΡΟΛΩΝ	61
ΕΙΚΟΝΑ 24 ΑΝΑΠΛΗΡΩΣΗ ΜΑΘΗΜΑΤΟΣ	62
ΕΙΚΟΝΑ 25 ΑΝΑΔΥΟΜΕΝΟ ΠΑΡΑΘΥΡΟ ΠΛΗΡΟΦΟΡΙΑΣ ΜΑΘΗΜΑΤΟΣ	62
ΕΙΚΟΝΑ 26 ΚΑΡΤΕΛΑ ΔΗΜΙΟΥΡΓΙΑΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ	63
ΕΙΚΟΝΑ 27 ΑΝΑΔΥΟΜΕΝΟ ΠΑΡΑΘΥΡΟ ΕΠΙΛΟΓΩΝ ΔΙΑΧΕΙΡΙΣΗΣ ΠΑΝΕΠΙΣΤΗΜΙΟΥ	63
ΕΙΚΟΝΑ 28 ΣΕΛΙΔΑ ΔΙΑΧΕΙΡΙΣΗΣ ΑΙΘΟΥΣΩΝ ΠΑΝΕΠΙΣΤΗΜΙΟΥ	64
ΕΙΚΟΝΑ 29 ΑΝΑΔΥΟΜΕΝΟ ΠΑΡΑΘΥΡΟ ΕΠΕΞΕΡΓΑΣΙΑΣ ΑΙΘΟΥΣΑΣ.....	64
ΕΙΚΟΝΑ 30 ΣΕΛΙΔΑ ΔΙΑΧΕΙΡΙΣΗΣ ΜΑΘΗΜΑΤΩΝ	65
ΕΙΚΟΝΑ 31 ΑΝΑΔΥΟΜΕΝΟ ΠΑΡΑΘΥΡΟ ΕΠΕΞΕΡΓΑΣΙΑΣ/ΠΡΟΣΘΗΚΗΣ ΜΑΘΗΜΑΤΟΣ	66
ΕΙΚΟΝΑ 32 ΣΕΛΙΔΑ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΘΗΓΗΤΩΝ.....	67
ΕΙΚΟΝΑ 33 ΑΝΑΔΥΟΜΕΝΟ ΠΑΡΑΘΥΡΟ ΑΝΑΘΕΣΗΣ ΚΑΘΗΓΗΤΩΝ	67
ΕΙΚΟΝΑ 34 ΑΝΑΔΥΟΜΕΝΟ ΠΑΡΑΘΥΡΟ ΔΗΜΙΟΥΡΓΙΑΣ ΝΕΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ.....	68
ΕΙΚΟΝΑ 35 ΛΙΣΤΑ ΧΡΗΣΤΩΝ.....	69
ΕΙΚΟΝΑ 36 ΑΝΑΔΥΟΜΕΝΟ ΠΑΡΑΘΥΡΟ ΕΠΕΞΕΡΓΑΣΙΑΣ ΧΡΗΣΤΗ.....	69
ΕΙΚΟΝΑ 37 ΑΝΑΔΥΟΜΕΝΟ ΠΑΡΑΘΥΡΟ ΠΡΟΣΘΗΚΗΣ ΧΡΗΣΤΗ.....	70
ΕΙΚΟΝΑ 38 ΔΙΑΧΕΙΡΙΣΗ ΑΙΤΗΜΑΤΩΝ ΚΑΘΗΓΗΤΩΝ	70
ΕΙΚΟΝΑ 39 ΔΙΑΧΕΙΡΙΣΗ ΑΝΑΠΛΗΡΩΣΕΩΝ	71
ΕΙΚΟΝΑ 40 ΜΠΑΡΑ ΠΛΟΗΓΗΣΗΣ ΚΑΘΗΓΗΤΗ	71
ΕΙΚΟΝΑ 41 ΤΑ ΜΑΘΗΜΑΤΑ ΜΟΥ.....	72
ΕΙΚΟΝΑ 42 ΑΡΧΙΚΗ ΣΕΛΙΔΑ "ΔΗΛΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΩΝ".....	72
ΕΙΚΟΝΑ 43 ΦΟΡΜΑ ΔΗΛΩΣΗΣ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΜΑΘΗΜΑΤΟΣ.....	73
ΕΙΚΟΝΑ 44 ΔΗΛΩΣΗ ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ "ΣΕ ΕΞΕΛΙΞΗ"	74
ΕΙΚΟΝΑ 45 ΔΗΛΩΣΗ ΈΚΤΑΚΤΟΥ ΜΑΘΗΜΑΤΟΣ.....	75
ΕΙΚΟΝΑ 46 ΕΓΓΡΑΦΗ ΣΕ ΜΑΘΗΜΑΤΑ	76
ΕΙΚΟΝΑ 47 ΑΠΕΝΕΡΓΟΠΟΙΗΣΗ UAC ΠΡΙΝ ΤΗΝ ΕΓΚΑΤΑΣΤΑΣΗ	84
ΕΙΚΟΝΑ 48 ΟΘΟΝΗ ΕΠΙΛΟΓΗΣ FEATURES ΤΟΥ ΧΑΜΡΡ	85
ΕΙΚΟΝΑ 49 ΧΑΜΡΡ CONTROL PANEL	85
ΕΙΚΟΝΑ 50 PHPMYADMIN.....	86

Κατάλογος Πινάκων

ΠΙΝΑΚΑΣ 1 ΒΟΗΘΗΤΙΚΕΣ ΒΙΒΛΙΟΘΗΚΕΣ ΛΟΓΙΣΜΙΚΟΥ	29
ΠΙΝΑΚΑΣ 2 USERS	39
ΠΙΝΑΚΑΣ 3 ROLES.....	39
ΠΙΝΑΚΑΣ 4 USERROLES	40
ΠΙΝΑΚΑΣ 5 SEMESTERS	40
ΠΙΝΑΚΑΣ 6 SCHOOLS.....	41
ΠΙΝΑΚΑΣ 7 ROOMS	41
ΠΙΝΑΚΑΣ 8 NOTIFICATIONS	42
ΠΙΝΑΚΑΣ 9 CLASSSSCHEDULES	43
ΠΙΝΑΚΑΣ 10 EMERGENCYCLASSREQUEST.....	44
ΠΙΝΑΚΑΣ 11 CLASSES	45
ΠΙΝΑΚΑΣ 12 STUDENTCLASSES	46
ΠΙΝΑΚΑΣ 13 TUTORAVAILABILITY	47
ΠΙΝΑΚΑΣ 14 TUTORCLASSES.....	48
ΠΙΝΑΚΑΣ 15 CLASSSSCHEDULE_ROOMS	48
ΠΙΝΑΚΑΣ 16 TIMETABLES.....	49
ΠΙΝΑΚΑΣ 17 CLASSCHEDULE_USERS	50
ΠΙΝΑΚΑΣ 18 TIMETABLE_CLASSES.....	51
ΠΙΝΑΚΑΣ 19 USERSCHOOLS.....	51
ΠΙΝΑΚΑΣ 20 TUTORAVAILABILITY_ROOMS.....	52
ΠΙΝΑΚΑΣ 21 EMERGENCYCLASSREQUESTS_ROOMS	52
ΠΙΝΑΚΑΣ 22 EMERGENCYCLASSREQUESTS_TIMETABLES	53
ΠΙΝΑΚΑΣ 23 ΠΙΝΑΚΑΣ DIRECTIONS.....	53
ΠΙΝΑΚΑΣ 24 ΜΕΤΡΙΚΑ ΑΡΧΕΙΩΝ ΚΩΔΙΚΑ -- ΧΩΡΙΣ ΒΙΒΛΙΟΘΗΚΕΣ	81
ΠΙΝΑΚΑΣ 25 ΜΕΤΡΙΚΑ ΑΡΧΕΙΩΝ ΚΩΔΙΚΑ ΜΑΖΙ ΜΕ ΤΙΣ ΒΙΒΛΙΟΘΗΚΕΣ	81
ΠΙΝΑΚΑΣ 26 ΕΝΤΟΛΕΣ COMPOSER	87

Κεφάλαιο 1: Εισαγωγή

Η κατάρτιση ενός ωρολογίου προγράμματος μαθημάτων σε πανεπιστημιακά και εκπαιδευτικά περιβάλλοντα αποτελεί μια σύνθετη και απαιτητική διαδικασία. Κάθε εξάμηνο, οι γραμματείες τμημάτων καλούνται να οργανώσουν δεκάδες μαθήματα, καθηγητές και αίθουσες, λαμβάνοντας υπόψιν πολλαπλούς περιορισμούς, όπως τη διαθεσιμότητα διδασκόντων, τη χωρητικότητα αιθουσών, τις ώρες διδασκαλίας και την αποφυγή επικαλύψεων μαθημάτων.

Η διαδικασία αυτή γίνεται κυρίως χειρωνακτικά ή με τη χρήση γενικών εργαλείων (π.χ. λογιστικά φύλλα) και τη συλλογική συνεννόηση των ακαδημαϊκών και της γραμματείας, γεγονός που την καθιστά ιδιαίτερα χρονοβόρα και επιρρεπή σε σφάλματα. Ένα χειρωνακτικά καταρτισμένο πρόγραμμα μπορεί εύκολα να περιέχει λάθη, όπως δύο διαφορετικά μαθήματα να προγραμματιστούν στην ίδια αίθουσα, ο ίδιος καθηγητής να διδάσκει την ίδια ώρα πολλαπλά μαθήματα ή ακόμη και μαθήματα κορμού ενός εξαμήνου να συμπίπτουν χρονικά, γεγονός που δεν θα επέτρεπε την πλήρη παρακολούθηση του προγράμματος σπουδών από τους φοιτητές. Επιπλέον, τυχόν αλλαγές της τελευταίας στιγμής, όπως η ακύρωση ή η αναπλήρωση μαθημάτων είναι δύσκολο να διαχειριστούν και να κοινοποιηθούν άμεσα σε όλους τους ενδιαφερόμενους και χρειάζεται να παρέμβει η γραμματεία για να ενημερώσει με ανακοίνωση στη σελίδα της σχολής τους διδασκόμενους.

Η δυσκολία του προβλήματος δεν είναι μόνο οργανωτική αλλά και αλγοριθμική: το ζήτημα της δημιουργίας βέλτιστου ωρολογίου προγράμματος με πολλούς περιορισμούς είναι γνωστό ότι ανήκει στα NP-πλήρη προβλήματα[1], πράγμα που σημαίνει ότι όσο αυξάνεται ο αριθμός μαθημάτων και περιορισμών, η υπολογιστική πολυπλοκότητα αυξάνεται ραγδαία. Για τους παραπάνω λόγους γίνεται εύκολα κατανοητό ότι θα πρέπει να βρεθεί ένας τρόπος ώστε η όλη διαδικασία να γίνει πιο εύκολη, ορθή και λιγότερο χρονοβόρα μειώνοντας την αλληλεπίδραση μεταξύ των καθηγητών, δίνοντάς τους τη δυνατότητα να έχουν όλη την πληροφορία αναφορικά με τη χρήση αιθουσών και χρονοθυρίδων του προγράμματος.

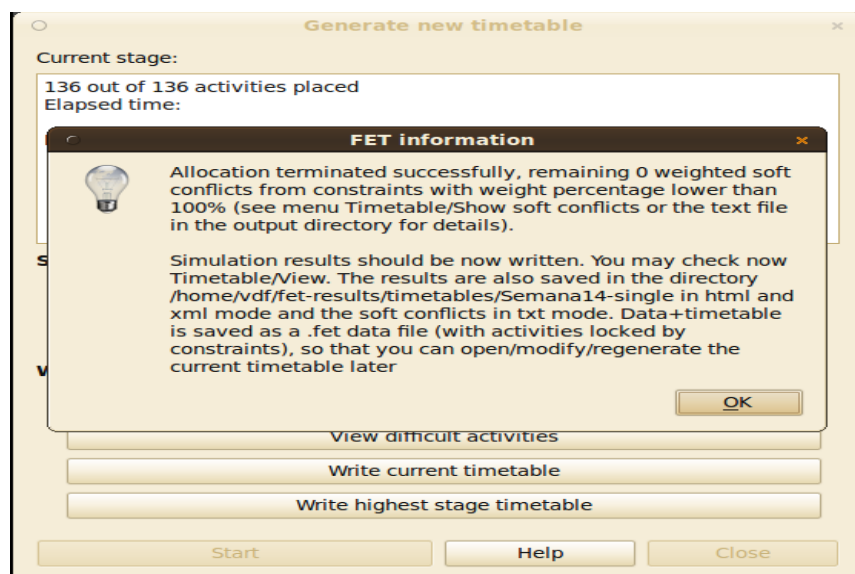
Η έλλειψη ενός ολοκληρωμένου πληροφοριακού συστήματος για τη διαχείριση του ωρολογίου προγράμματος μπορεί να οδηγήσει σε καθυστερήσεις, δυσκολία στην ενημέρωση αλλαγών και σε περιορισμένη δυνατότητα εξατομίκευσης της πληροφορίας (π.χ. ένας φοιτητής μπορεί να δει μόνο τα δικά του μαθήματα απ' όποιο εξάμηνο και να είναι αυτά). Είναι λοιπόν σαφές το κίνητρο για την ανάπτυξη μιας λύσης λογισμικού, συγκεκριμένα το **TMS (Timetable Management System)** που θα απλοποιήσει και θα αυτοματοποιήσει τη διαδικασία δημιουργίας του προγράμματος, εξασφαλίζοντας την ορθότητα του και διευκολύνοντας όλους τους εμπλεκόμενους.

1.1 Υπάρχουσες λύσεις και σχετική εργασία

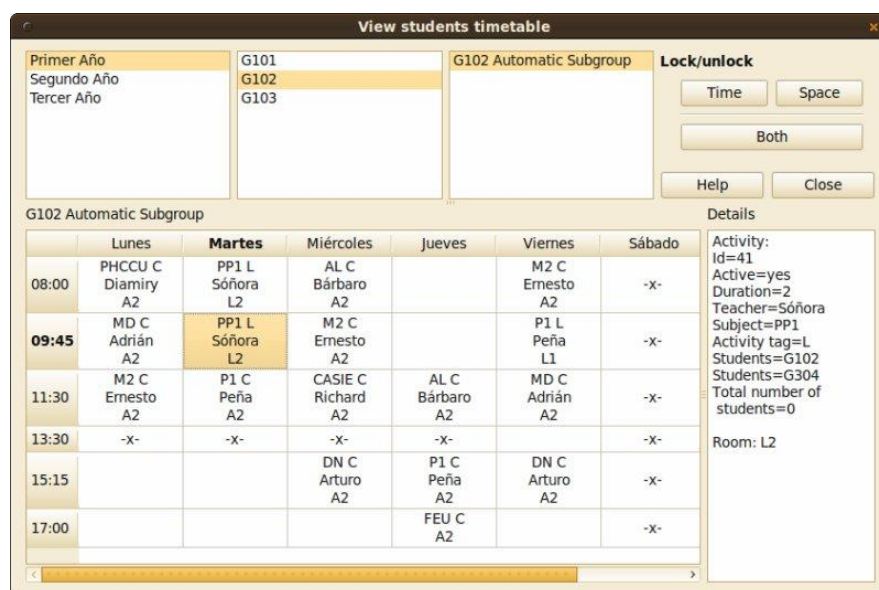
Η ανάγκη για αποδοτικό προγραμματισμό μαθημάτων έχει οδηγήσει στην ανάπτυξη αρκετών συστημάτων διαχείρισης ωρολογίου προγράμματος, τόσο εμπορικών όσο και ανοικτού κώδικα. Σε διεθνές επίπεδο, υπάρχουν γνωστές λύσεις λογισμικού που χρησιμοποιούνται ευρέως σε εκπαιδευτικούς οργανισμούς. Για παράδειγμα, το FET[2] είναι ένα δωρεάν λογισμικό ανοικτού κώδικα για τον αυτόματο προγραμματισμό του χρονοδιαγράμματος ενός σχολείου ή πανεπιστημίου. Το FET χρησιμοποιεί έναν γρήγορο και αποδοτικό αλγόριθμο για τη δημιουργία ωρολογίων προγραμμάτων, επιτρέποντας αυτόματη δημιουργία αλλά και χειροκίνητες

παρεμβάσεις αν χρειάζεται. Όπως φαίνεται στην Εικόνα 1 και στην Εικόνα 2, το FET είναι μια desktop εφαρμογή και όχι γραμμένο σαν web πλατφόρμα. Συγκεκριμένα είναι γραμμένη με τη χρήση της γλώσσας C++ και Qt και χρησιμοποιεί στοιχεία Windows Forms για το UI(user interface). Αυτό σημαίνει πως το σύστημα πρέπει να εγκατασταθεί τοπικά σε υπολογιστές, έκδοση κλειδιών για τον κάθε χρήστη της εφαρμογής σε διαφορετικό υπολογιστή. Δεν είναι σύστημα που υποστηρίζει πολλούς ρόλους χρηστών και ροές έγκρισης όπως το TMS, που καλύπτει τις ανάγκες λειτουργίας ενός πανεπιστημίου μέσα από ένα ενιαίο web περιβάλλον.

Το TMS είναι γραμμένο εξ 'ολοκλήρου σε HTML, CSS, PHP και JavaScript ώστε να είναι προσβάσιμο σε διαφορετικούς χρήστες και χρειάζεται μόνο την δημοσίευσή του στον ακαδημαϊκό διακομιστή. Τέλος είναι αρκετά πιο ευέλικτο όσον αφορά τις επιτόπου δηλώσεις αναπληρώσεων και έκτακτων γεγονότων συγκριτικά με το FET που δεν προβλέπεται για τέτοια χρήση.



Εικόνα 1 FET UI



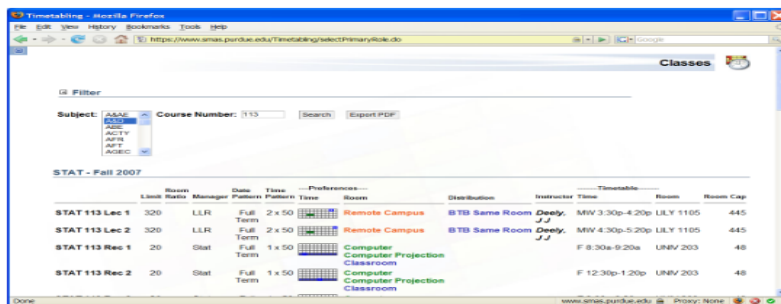
Εικόνα 2 FET timetabling UI

Παρόμοια, το UniTime[3] του οποίου στιγμιότυπα εμφανίζονται στην Εικόνα 3 και Εικόνα 4, είναι ένα ολοκληρωμένο σύστημα ακαδημαϊκού προγραμματισμού ανοικτού κώδικα που υλοποιήθηκε με τη συνεργασία καθηγητών, ακαδημαϊκού προσωπικού και καθηγητών και υποστηρίζει την κατάρτιση ωρολογίων προγραμμάτων μαθημάτων και εξετάσεων, τη διαχείριση αλλαγών και τη διαμοίραση χωρικών πόρων μεταξύ πολλαπλών τμημάτων που μπορεί να χρησιμοποιούν τις ίδιες εγκαταστάσεις.

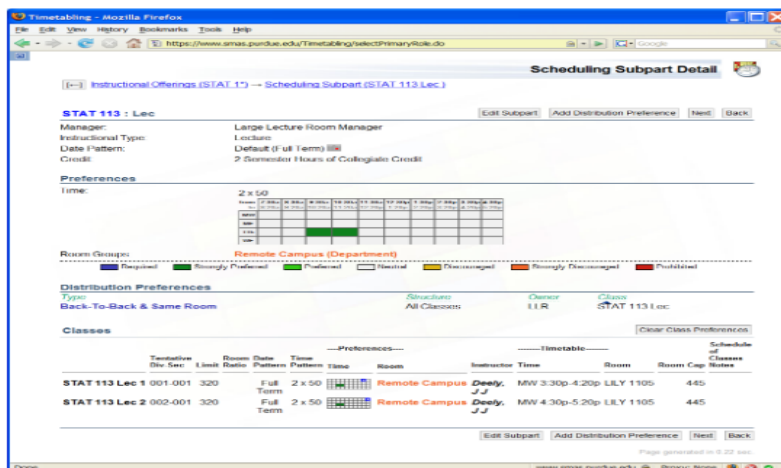
Μάλιστα, το UniTime επιτρέπει σε πολλαπλούς διαχειριστές από διαφορετικά τμήματα ενός πανεπιστημίου να συνεργάζονται για την δημιουργία του προγράμματος, προσφέροντας δυνατότητες για μείωση των συγκρούσεων μαθημάτων για τους φοιτητές. Διατίθεται ελεύθερα ως ανοικτό λογισμικό, έτσι ώστε και άλλα ιδρύματα να μπορούν να το αξιοποιήσουν ή/και να συμβάλουν στην εξέλιξή του, χωρίς κόστος αδειοδότησης.

Το UniTime είναι γραμμένο σε Java και απαιτεί την εγκατάσταση του Java Development Kit, MySQL και Apache Tomcat για ανάπτυξη και τη λειτουργία του. Το TMS από την άλλη, αποφεύγει application containers (Tomcat/Servlet), άρα έχει ελαφρύτερη υποδομή και άμεση συμβατότητα με Linux συστήματα, που υπάρχουν ήδη στα περισσότερα πανεπιστήμια. Το UniTime είναι ένα ολιστικό πρόγραμμα διαχείρισης ακαδημαϊκού προγραμματισμού (καθορίζει τα μαθήματα, τις εξετάσεις, το μοίρασμά αιθουσών κ.α) και απαιτεί σοβαρή παραμετροποίηση και αφιέρωση χρόνου για την κατατόπιση του χρήστη.

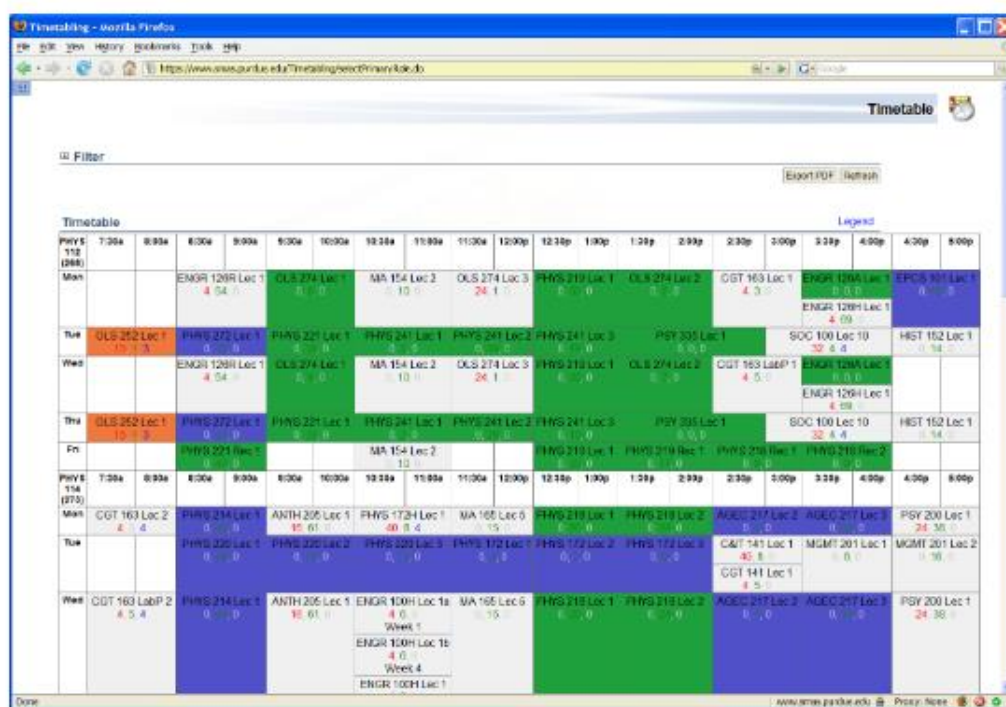
Το TMS κρατά συγκεκριμένο και στοχευμένο εύρος μειώνοντας την καμπύλη εκμάθησης και λειτουργικό κόστος. Το UniTime θα χρειάζεται και διαρκές «συγύρισμα» και rotation των αρχείων catalina.out, το αρχείο εσωτερικής καταγραφής logs της Java. Το UniTime διανέμεται ως WAR και αναπτύσσεται σε Tomcat, με απαιτήσεις έκδοσης/συμβατότητας Java-Tomcat. Το TMS αναπτύσσεται ως κλασική web εφαρμογή PHP, διευκολύνοντας γρήγορη εγκατάσταση/μετεγκατάσταση και χαμηλότερη κατανάλωση πόρων για τυπική χρήση.



User interface provides an easy and intuitive means of data entry.



Εικόνα 3 UniTime UI



Timetabling solver can provide a fully automated solution.

Suggestions

Score	Class	Date	Time	Room	Students
+15.2	POL 101 Lec 3	Full Term	TTh 12:00p → TTh 7:30a	BRNG 2280	+11
+31.7	POL 101 Lec 3	Full Term	TTh 12:00p → TTh 10:30a	BRNG 2280	+36 (h=3)
	HIST 342 Lec 1	Full Term	TTh 10:30a → TTh 1:30p	BRNG 2280 → BRNG 2290	
+36.6	POL 101 Lec 3	Full Term	TTh 12:00p → TTh 10:30a	BRNG 2280	+36 (h=4)
	HIST 342 Lec 1	Full Term	TTh 10:30a → TTh 7:30a	BRNG 2280	
+44.1	POL 101 Lec 3	Full Term	TTh 12:00p → TTh 10:30a	BRNG 2280	+34 (h=2)
	HIST 342 Lec 1	Full Term	TTh 10:30a → TTh 3:00p	BRNG 2280 → BRNG 2290	
	OBHR 330 Lec 4	Full Term	TTh 3:00p	BRNG 2290 → LWSN B155	

Εικόνα 4 UniTime timetable solver

Εκτός από τις λύσεις ανοικτού κώδικα, υπάρχουν και εμπορικά διαθέσιμα συστήματα που στοχεύουν κυρίως σε σχολεία αλλά και ανώτερες εκπαιδευτικές μονάδες. Χαρακτηριστικό παράδειγμα αποτελεί το λογισμικό Untis[4], μια εφαρμογή που εξελίσσεται από τη δεκαετία του 1970 και σήμερα χρησιμοποιείται από περισσότερα από 25.000 εκπαιδευτικά ιδρύματα παγκοσμίως. Το Untis προσφέρει ολοκληρωμένες λύσεις ωρολογίου προγράμματος, από την αυτόματη δημιουργία και βελτιστοποίηση του βασικού προγράμματος έως τη διαχείριση έκτακτων αλλαγών αντικαταστάσεις καθηγητών, αναπληρώσεις και άλλα, πολυγλωσσική υποστήριξη και διαδικτυακή πρόσβαση (μέσω του WebUntis).

Επισημαίνεται πως η εξάρτηση από εμπορικό λογισμικό ωρολογίου προγράμματος κρίθηκε μη βιώσιμη λόγω του σημαντικού ετήσιου κόστους συντήρησης και αδειοδότησης. Επιπλέον, πολλά από αυτά τα συστήματα είναι σχεδιασμένα με γενικό τρόπο για να καλύπτουν ποικιλία οργανισμών, πράγμα που σημαίνει ότι μπορεί να μην προσαρμόζονται εύκολα σε ειδικές απαιτήσεις ενός συγκεκριμένου πανεπιστημιακού τμήματος χωρίς περαιτέρω παραμετροποίηση ή πρόσθετη ανάπτυξη.

Στην Ελλάδα η χρήση ειδικών πληροφοριακών συστημάτων για το ωρολόγιο πρόγραμμα σε πανεπιστήμια βρίσκεται ακόμη σε αρχικό στάδιο. Παρόμοια συστήματα υπάρχουν σε ελάχιστα

πανεπιστήμια στην Ελλάδα, κάτι που αναδεικνύει την αξία μιας νέας υλοποίησης και την άμεση αξιοποίησή της από πολλά ιδρύματα. Οι περισσότερες σχολές εξακολουθούν να χρησιμοποιούν χειροκίνητες μεθόδους, συνεπώς ένα προσαρμοσμένο σύστημα που θα ανταποκρίνεται στις ανάγκες ενός ελληνικού πανεπιστημίου μπορεί να αποτελέσει σημαντική καινοτομία. Η παρούσα διπλωματική εργασία έρχεται να καλύψει αυτό το κενό, αξιοποιώντας τις βέλτιστες πρακτικές από τα υπάρχοντα συστήματα και προτείνοντας μια λύση προσαρμοσμένη στο περιβάλλον ενός πανεπιστημιακού τμήματος.

Στο πλαίσιο της έρευνας για την υλοποίηση, μελετήθηκε η διπλωματική του Γεώργιου Παπάκη[5] με το ίδιο θέμα και επεκτάθηκαν οι λειτουργικές απαιτήσεις και η λογική του κώδικα ώστε να ανταποκρίνεται σχεδόν ολοκληρωτικά στον τρόπο λειτουργίας του πανεπιστημίου Δυτικής Μακεδονίας και συγκεκριμένα της σχολής Μηχανικών Πληροφορικής και Τηλεπικοινωνιών.

1.2 Συνεισφορά και καινοτομία της εργασίας

Η συνεισφορά της παρούσας διπλωματικής εργασίας συνοψίζεται στα εξής βασικά σημεία:

- **Προσαρμογή στις ανάγκες πανεπιστημιακού τμήματος:** Σε αντίθεση με γενικά εργαλεία ωρολογίων προγραμμάτων, το TMS(Timetable Management System) που υλοποιήθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας, έχει σχεδιαστεί ειδικά για το περιβάλλον ενός πανεπιστημίου. Αυτό σημαίνει ότι υποστηρίζει τις ιδιαίτερες απαιτήσεις ενός τριτοβάθμιου ιδρύματος, όπως τη διάκριση μεταξύ προγράμματος μαθημάτων και εξεταστικής περιόδου, την ύπαρξη πολλαπλών τμημάτων/τομέων, καθώς και την ανάγκη διαφορετικών επιπέδων χρήστη (γραμματεία, διδάσκοντες, φοιτητές, επισκέπτες). Μέσω αυτής της προσαρμογής, το σύστημα γεφυρώνει το κενό που υπάρχει στην ελληνική ανώτατη εκπαίδευση όσον αφορά τέτοια πληροφοριακά εργαλεία.
- **Ολοκληρωμένη πλατφόρμα με διακριτούς ρόλους χρηστών:** Το σύστημα προσφέρει διαφορετικές λειτουργικότητες ανάλογα με τον ρόλο του χρήστη. Η γραμματεία έχει πλήρη δικαιώματα δημιουργίας και τροποποίησης του ωρολογίου προγράμματος, οι διδάσκοντες μπορούν να αλληλοεπιδρούν μέσω αιτημάτων και να βλέπουν μόνο ό,τι τους αφορά, και οι φοιτητές έχουν άμεση πρόσβαση στο εξατομικευμένο πρόγραμμά τους. Αυτός ο διαχωρισμός διασφαλίζει τάξη και ασφάλεια στη χρήση του συστήματος, αποφεύγοντας σύγχυση και μη εξουσιοδοτημένες αλλαγές.
- **Αυτοματοποιημένοι έλεγχοι εγκυρότητας:** Μια σημαντική καινοτομία είναι η ενσωμάτωση αυτοματοποιημένων ελέγχων κατά την καταχώριση στοιχείων του προγράμματος. Το σύστημα εμποδίζει την εισαγωγή συγκρουόμενων ωραρίων και ειδοποιεί άμεσα τον καθηγητή σε περίπτωση προσπάθειας δημιουργίας σύγκρουσης. Με αυτόν τον τρόπο, ολόκληρη η διαδικασία απλοποιείται για το γραμματειακό προσωπικό σε σχέση με τον εκ των υστέρων εντοπισμό λαθών, κάτι που βελτιώνει θεαματικά την αξιοπιστία του τελικού αποτελέσματος.
- **Δυνατότητα χρήσης της εφαρμογής για επέκταση και εκπαίδευση:** Το TMS εισάγει ένα ενιαίο περιβάλλον επικοινωνίας για θέματα δημιουργίας ωρολογίου προγράμματος. Μιας και η επιχειρησιακή λογική καταγράφεται εκτενώς στην παρούσα εργασία, μπορεί να χρησιμοποιηθεί για μελλοντική μετεκπαίδευση σε μαθήματα βάσεων δεδομένων, προγραμματισμού διαδικτύου, μαθήματα αλγορίθμων κ.α.

- **Βάση για μελλοντική αυτοματοποίηση:** Αν και στην παρούσα υλοποίηση η κατανομή των μαθημάτων στις ώρες γίνεται χειρωνακτικά από τη γραμματεία με την υποβοήθηση των ελέγχων, το σύστημα έχει σχεδιαστεί με τρόπο που μπορεί να επεκταθεί εύκολα. Αποτελεί δηλαδή μια βάση πάνω στην οποία μπορούν να αναπτυχθούν στο μέλλον πιο προηγμένοι αλγόριθμοι αυτοματοποιημένης κατάρτισης προγράμματος. Αυτό σημαίνει ότι με μελλοντική ενσωμάτωση επιπλέον λειτουργικοτήτων, το TMS θα μπορούσε να υποστηρίξει πλήρως αυτόματο προγραμματισμό με είσοδο μόνο τους περιορισμούς και στόχους, προσφέροντας ακόμη μεγαλύτερο βαθμό αυτοματοποίησης.

Συμπερασματικά, η εργασία αυτή προσφέρει ένα νέο εργαλείο που όχι μόνο καλύπτει πρακτικά τις ανάγκες της δημιουργίας του ωρολογίου προγράμματος σε ένα πανεπιστημιακό τμήμα, αλλά εισάγει και καινοτομίες που μπορούν να ωφελήσουν την ακαδημαϊκή κοινότητα. Η αξία του συστήματος έγκειται στην άμεση εφαρμοσιμότητά του μιας και μπορεί να χρησιμοποιηθεί από πολλαπλά ιδρύματα με μικρές προσαρμογές) και στη βελτίωση που επιφέρει σε μια κρίσιμη διοικητική διαδικασία της εκπαίδευσης.

1.3 Διάρθρωση της εργασίας

Η διαδικασία του σχεδιασμού και της υλοποίηση του πληροφοριακού συστήματος TMS αναλύεται στο παρακάτω κείμενο το οποίο αποτελείται από πέντε κεφάλαια και περιληπτικά αναφέρονται τα εξής για αυτά που ακολουθούν:

Κεφάλαιο 2: Ανάλυση και περιγραφή των γλωσσών προγραμματισμού που χρησιμοποιήθηκαν, των βιβλιοθηκών και των εργαλείων λογισμικού που χρησιμοποιήθηκαν κατά την ανάπτυξη.

Κεφάλαιο 3: Ανάλυση των απαιτήσεων του συστήματος, τα δομικά στοιχεία που το αποτελούν, περιγραφή της βάσης δεδομένων, διαγράμματα χρήσης-περίπτωσης για τις βασικότερες εργασίες μεταξύ χρηστών, καθώς επίσης και αναφορά των τεχνικών που χρησιμοποιήθηκαν για την εξασφάλιση της ασφάλειας των χρηστών και της σελίδας.

Κεφάλαιο 4: Διεπαφή χρήστη (user interface), στιγμιότυπα καθώς και πρακτικά παραδείγματα χρήσης των δυνατοτήτων της σελίδας για κάθε χρήστη ξεχωριστά.

Κεφάλαιο 5: Συμπεράσματα, ανάλυση SWOT(Strengths, Weaknesses, Opportunities, Threats) καθώς και τα μετρικά του συστήματος.

Το πρώτο κεφάλαιο κατέδειξε την πολυπλοκότητα και τα προβλήματα που συνοδεύουν την κατάρτιση ωρολογίων προγραμμάτων στα πανεπιστήμια. Έγινε αναφορά σε αντίστοιχα προϊόντα ανοιχτού κώδικα καθώς και εμπορικού χαρακτήρα και αναδείχθηκε η ανάγκη ανάπτυξης του TMS ως μιας λύσης με σαφή πλεονεκτήματα που αντιμετωπίζει πρακτικά εμπόδια, μειώνει τον φόρτο της γραμματείας και διευκολύνει καθηγητές και φοιτητές. Η εισαγωγή έθεσε το θεωρητικό υπόβαθρο και τους στόχους που αποτέλεσαν οδηγό για την υλοποίηση. Ακολουθεί το κεφάλαιο δύο με την παρουσίαση του τεχνικού θεωρητικού υποβάθρου, με την επεξήγηση όλων των τεχνολογιών και εργαλείων λογισμικού που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής.

Κεφάλαιο 2: Θεωρητικό υπόβαθρο και εργαλεία υλοποίησης

Στο παρόν κεφάλαιο παρουσιάζονται αναλυτικά οι τεχνολογίες και τα εργαλεία που αξιοποιήθηκαν για την ανάπτυξη και υλοποίηση του πληροφοριακού συστήματος. Το περιεχόμενο του κεφαλαίου έχει δύο κύριους άξονες:

- **Τεχνολογίες λογισμικού:** Αναλύονται οι γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για τη δημιουργία της βάσης δεδομένων, του back-end και του front-end της ιστοσελίδας καθώς και οι βοηθητικές βιβλιοθήκες που χρησιμοποιήθηκαν κατά την ανάπτυξη της.
- **Εργαλεία και βιβλιοθήκες τρίτων:** Παρουσιάζονται συνοπτικά πακέτα και βιβλιοθήκες που αξιοποιήθηκαν, όπως το «FullCalendar» για την απεικόνιση ημερολογίου, καθώς και επιλεγμένες βιβλιοθήκες από τα οικοσυστήματα των Node.js και Composer, με τεκμηρίωση και αναφορά στις επίσημες πηγές κάθε εργαλείου.

Η επιλογή των παραπάνω τεχνολογιών έγινε με γνώμονα τη λειτουργικότητα του συστήματος, την αποδεδειγμένη αξιοπιστία τους στη δημιουργία διαδικτυακών εφαρμογών, καθώς και την ευρεία τους τεκμηρίωση, που καθιστά δυνατή την ορθή υλοποίησή τους στο πλαίσιο της παρούσας διπλωματικής εργασίας.

Για την εγκατάσταση και τη χρήση του XAMPP και των βιβλιοθηκών Node.js και Composer παρέχονται αναλυτικές οδηγίες εγκατάστασης και αναφορά στις εκδόσεις και το περιβάλλον που έλαβε χώρα ο προγραμματισμός της σελίδας, στο παράρτημα της εργασίας.

2.1 Τεχνολογίες Λογισμικού

Στην υποενότητα που αυτή αναφερόμαστε στις γλώσσες προγραμματισμού:

- HTML (HyperText Markup Language)
- CSS (Cascading Style Sheets)
- PHP (Hypertext Preprocessor)
- JavaScript
- MySQL (My Structured Query Language)

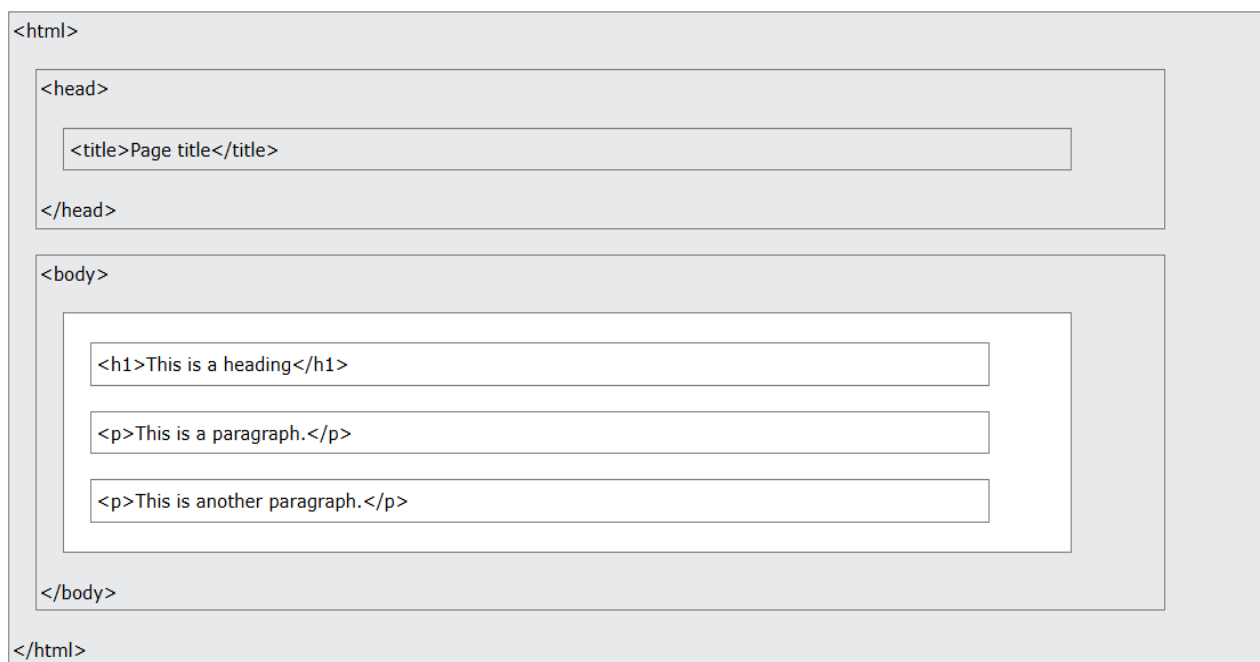
Μετά γίνεται αναφορά στην τεχνική AJAX (Asynchronous JavaScript and XML), στην επέκταση της PHP PDO (PHP Data Objects), στο οικοσύστημα της Node.js και στο διαχειριστή εξαρτήσεων Composer.

2.1.1 HTML

Η γλώσσα σήμανσης υπερκειμένου HTML (HyperText Markup Language) αποτελεί το θεμέλιο της σύγχρονης διαδικτυακής ανάπτυξης, καθώς προσδιορίζει τη δομή και την ιεραρχία των στοιχείων σε μία ιστοσελίδα. Αποτελεί το αναγκαίο υπόβαθρο για την οπτική απεικόνιση περιεχομένου, το οποίο προβάλλεται και ερμηνεύεται από τους περιηγητές ιστού (web browsers).

Η HTML περιλαμβάνει μία πληθώρα από ετικέτες (tags), όπως <div>, <p>, <table>, <form>, οι οποίες αποδίδουν νόημα και κατηγοριοποίηση στο περιεχόμενο και δημιουργούν τον σκελετό της ιστοσελίδας. [6]. Η κάθε ετικέτα μπορεί να συνοδεύεται από χαρακτηριστικά όπως id και class, τα οποία δεν εξυπηρετούν μόνο στον εντοπισμό των στοιχείων από CSS για σκοπούς μορφοποίησης, αλλά και στην αναφορά τους από scripts JavaScript για την επίτευξη δυναμικής συμπεριφοράς. Η συνέργεια αυτή καθιστά την HTML ιδιαίτερα ευέλικτη και καθοριστική για την λειτουργική και αισθητική διάσταση ενός ιστοτόπου.

Η δομή ενός εγγράφου HTML βασίζεται στη λογική του DOM (Document Object Model)[7], ξεκινώντας από την ετικέτα <!DOCTYPE html> και οργανώνοντας τα δεδομένα μέσω των ετικετών, head και body μέσα στην ευρύτερη html ετικέτα, όπως φαίνεται στην Εικόνα 5



Εικόνα 5 Βασική δομή σελίδας HTML

Στο παραπάνω παράδειγμα, η χρήση του «id="container"» και «class="main-content"» επιτρέπει στο αρχείο CSS να μορφοποιήσει τον συγκεκριμένο χώρο με μοναδικό ή επαναχρησιμοποιήσιμο στυλ, ενώ το αρχείο JavaScript μπορεί να αλληλεπιδράσει με το ίδιο στοιχείο για την ενεργοποίηση δυναμικών λειτουργιών όπως την αλλαγή περιεχομένου και τη διαχείριση συμβάντων χρήστη.

Η HTML αποτελεί τον «σκελετό» της πλειοψηφίας των ιστοσελίδων στο διαδίκτυο, λόγω της ευρείας υποστήριξής της από όλους τους περιηγητές, της εύκολης εκμάθησής και πλούσιας τεκμηρίωσης, αλλά και της δυνατότητας γρήγορης κατασκευής πρωτοτύπων. Επιπλέον, συνεργάζεται άμεσα με τεχνολογίες όπως CSS και JavaScript, υποστηρίζοντας επεκτάσιμη ανάπτυξη και διαδραστική λειτουργικότητα. Το γεγονός ότι αποτελεί το πιο αναγνωρισμένο πρότυπο για την κατασκευή σελίδων στο διαδίκτυο, καθιστά τη χρήση της απαραίτητη σε κάθε περιβάλλον ανάπτυξης ιστοσελίδων.

2.1.2 CSS

Η CSS (Cascading Style Sheets) αποτελεί την κύρια τεχνολογία μορφοποίησης για τις ιστοσελίδες HTML, επιτρέποντας τον πλήρη έλεγχο της οπτικής παρουσίασης των στοιχείων. Μέσω αυτής καθορίζεται η εμφάνιση του περιεχομένου όπως το χρώμα, οι γραμματοσειρές, τα περιθώρια, οι διαστάσεις, η στοίχιση και οι αποστάσεις ώστε να διασφαλίζεται η συνοχή στην εμφάνιση ολόκληρου του ιστότοπου.[8]

Η CSS παρέχει τη δυνατότητα συσχέτισης με τα HTML στοιχεία χρησιμοποιώντας χαρακτηριστικά όπως class και id. Αυτή η διασύνδεση επιτρέπει την εφαρμογή συγκεκριμένων μορφοποιήσεων σε επιμέρους ή πολλαπλά στοιχεία. Επιπλέον, η ύπαρξη ιεραρχίας μεταξύ των επιλεγμένων στοιχείων (specificity hierarchy) επιτρέπει την ακριβή προτεραιότητα στην εφαρμογή κανόνων. Ενδεικτικά, οι id έχουν υψηλότερη προτεραιότητα από class, ενώ η χρήση της ετικέτας <style> για την προσθήκη CSS απευθείας σε αρχείο HTML υπερισχύει όλων. Αυτή η δομή είναι κρίσιμη τόσο για τη διαχείριση πολύπλοκων σχεδίων. Η CSS εφαρμόζεται συνήθως μέσω εξωτερικών αρχείων .css, τα οποία συνδέονται στο HTML έγγραφο με την ετικέτα <link> στην κεφαλίδα όπως δείχνουμε στην Εικόνα 6.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Sandbox</title>
    <meta charset="UTF-8" />
    <link rel="stylesheet" href="src/styles.css" />
  </head>

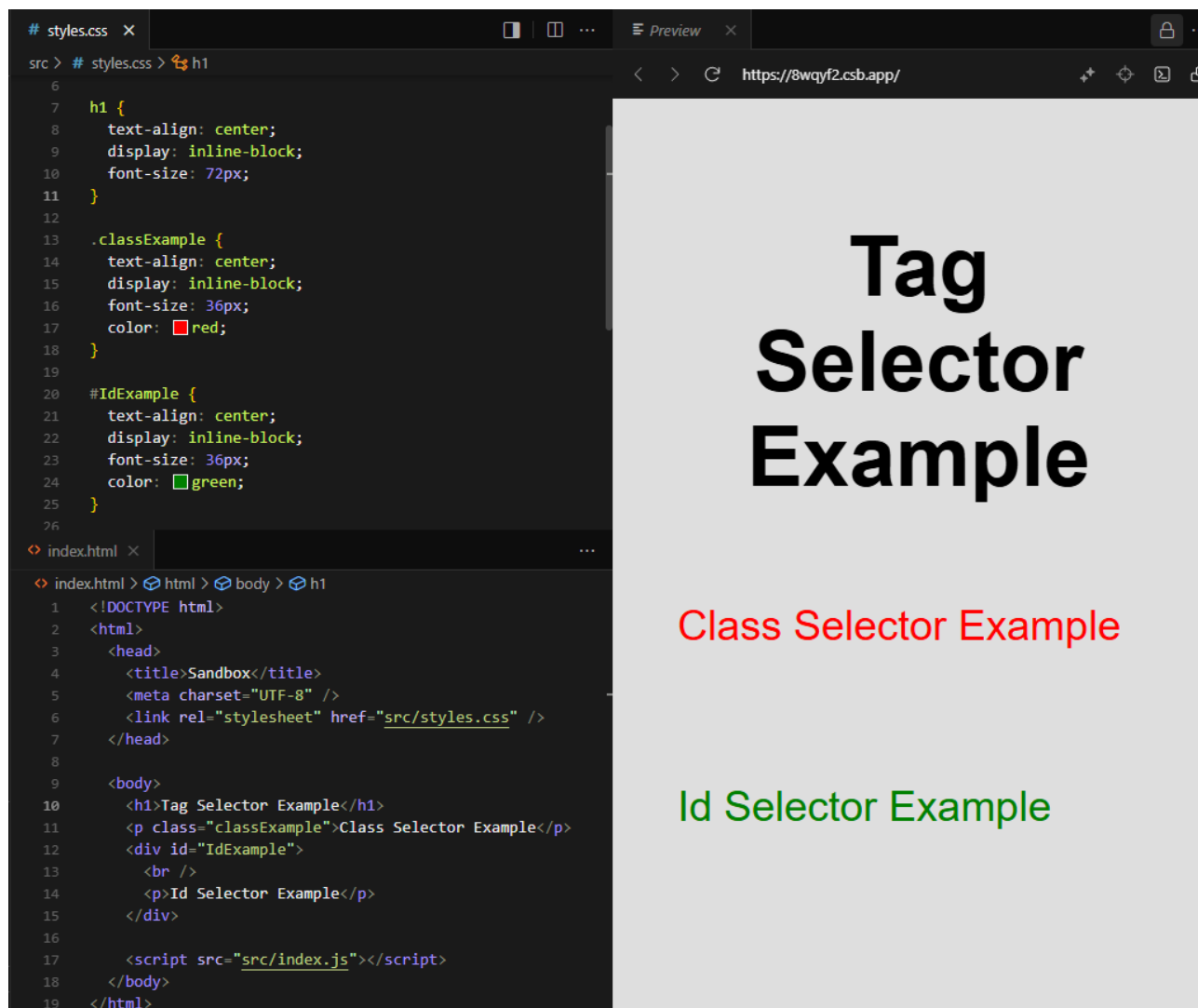
  <body>
    <h1>Tag Selector Example</h1>
    <p class="classExample">Class Selector Example</p>
    <div id="IdExample">
      <br />
      <p>Id Selector Example</p>
    </div>

    <script src="src/index.js"></script>
  </body>
</html>
```

Εικόνα 6 Ετικέτα link για είσοδο CSS αρχείου

Η σύνταξη της CSS βασίζεται σε επιλογείς (selectors)[9], οι οποίοι προσδιορίζουν τα HTML στοιχεία στα οποία εφαρμόζονται ιδιότητες. Ένα απλό παράδειγμα παρουσιάζεται στην Εικόνα 7.

Οι επιλογείς μπορεί να είναι στοιχειακοί όπως το p, h1), κατά class όπως η classExample ή κατά id όπως το #IdExample). Η χρήση της CSS σε συνδυασμό με τα class και id δίνει τη δυνατότητα στοχευμένης και επαναχρησιμοποιήσιμης μορφοποίησης, και παρέχει τη βάση για τη συνεργασία με βιβλιοθήκες JavaScript που εφαρμόζουν διαδραστικά χαρακτηριστικά. Η επιλογή της CSS στην παρούσα διπλωματική εργασία βασίστηκε στην αναγκαιότητα δημιουργίας ενός προσαρμοστικού, εύχρηστου και επεκτάσιμου περιβάλλοντος διεπαφής χρήστη. Η δυνατότητα διαχωρισμού της λογικής από την παρουσίαση, η καθαρότητα του τελικού κώδικα και η μαζική διαχείριση αλλαγών στην εμφάνιση κατέστησαν την CSS αναπόσπαστο εργαλείο υλοποίησης.



Εικόνα 7 Παράδειγμα αλληλεπίδρασης CSS και HTML

2.1.3 PHP

Η PHP (PHP: Hypertext Preprocessor)[10] αποτελεί μία από τις πιο διαδεδομένες γλώσσες προγραμματισμού για την ανάπτυξη διαδικτυακών εφαρμογών, ειδικά όταν αυτές απαιτούν δυναμική διαχείριση περιεχομένου. Πρόκειται για γλώσσα ανοιχτού κώδικα, η οποία εκτελείται στον διακομιστή (server-side) και επιστρέφει στον φυλλομετρητή ως απλό HTML, ενσωματώνοντας δυναμικά δεδομένα, λογική, ακόμη και αποτελέσματα από βάσεις δεδομένων.

Ένα από τα βασικά πλεονεκτήματα της PHP είναι η ευκολία με την οποία μπορεί να συνδυαστεί με την HTML. Η γλώσσα δίνει τη δυνατότητα εισαγωγής μεταβλητών, συναρτήσεων, βρόγχων και συνθηκών απευθείας μέσα σε HTML δομές. Αυτό επιτρέπει την άμεση απόδοση δυναμικού περιεχομένου, χωρίς την ανάγκη σύνθετων μεταγλωττιστικών διαδικασιών. Παρακάτω στην Εικόνα 8 φαίνεται ένα χαρακτηριστικό παράδειγμα όπου η PHP ενσωματώνεται σε HTML.

```

test.php
1  <?php
2      $classText = "Αυτό είναι περιεχόμενο από PHP σε class στοιχείο.";
3      $idText = "Αυτό είναι περιεχόμενο από PHP σε id στοιχείο.";
4  ?>
5
6  <p class="classExample"><?php echo $classText; ?></p>
7
8  <div id="IdExample">
9      <p><?php echo $idText; ?></p>
10 </div>

```

Εικόνα 8 Παράδειγμα ενσωμάτωσης PHP σε HTML

Η PHP προσφέρει σημαντική λειτουργικότητα και επεκτασιμότητα μέσω της ενσωματωμένης βιβλιοθήκης συναρτήσεων και της δυνατότητας αλληλεπίδρασης με βάσεις δεδομένων όπως η MySQL. Η συνεργασία της με την MySQL επιτρέπει την αποθήκευση, ανάκτηση, τροποποίηση και διαγραφή δεδομένων μέσω δομών τύπου POST και GET, καθιστώντας τη βασικό εργαλείο για την ανάπτυξη δυναμικών εφαρμογών.

Ενσωματώνεται σε ξεχωριστά αρχεία με κατάληξη «.php», τα οποία συνδέονται με HTML μέσω μηχανισμών υποβολής φόρμών και περιλαμβάνονται στο front-end με τη χρήση του τοπικού εξυπηρετητή του XAMPP με το οποίο εγκαθίσταται μαζί.

2.1.4 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (RDBMS), το οποίο οργανώνει τα δεδομένα σε δομές πινάκων με γραμμές και στήλες, σύμφωνα με το μοντέλο διασύνδεσης που δημιούργησε ο E. F. Codd το 1970. Η επιλογή της βασίστηκε στην αξιοπιστία της, την ευρεία αποδοχή στην κοινότητα προγραμματιστών και την πλήρη συμβατότητα με τη γλώσσα SQL, της οποίας επιτρέπει την εκτέλεση των αναγκαίων εργασιών μιας βάσης δεδομένων όπως η εισαγωγή, ανάκτηση, ενημέρωση και διαγραφή δεδομένων στη βάση. Οι λειτουργίες αυτές αναφέρονται κοινώς ως λειτουργίες CRUD (Create, Read, Update, Delete).

Για την ταξινόμηση των δεδομένων στους πίνακες της βάσης δεδομένων χρησιμοποιούνται οι έννοιες που ακολουθούν:

- **Οντότητα - Πίνακας:** Κάθε αντικείμενο πραγματικού κόσμου (π.χ. χρήστης, μάθημα) αναπαρίσταται από έναν πίνακα και κάθε στήλη του πίνακα αντιστοιχεί σε ένα χαρακτηριστικό της οντότητας.
- **Πληθικότητες:** Καθορισμός σχέσεων όπως **1:1**, **1:N** ή **M:N** για παράδειγμα ένας πελάτης μπορεί να έχει πολλές παραγγελίες (1:N πληθικότητα), ενώ μία παραγγελία μπορεί περιέχει πολλαπλά προϊόντα (**M:N** πληθικότητα).
- **Κλειδιά:** Κάθε πίνακας ορίζεται με ένα πρωτεύον κλειδί ή αλλιώς Primary Key το οποίο καθορίζει τη μοναδικότητα μιας γραμμής στον πίνακα ενώ οι συσχετίσεις μεταξύ πινάκων υλοποιούνται μέσω λεγόμενα ξένα κλειδιά ή αλλιώς Foreign Keys. Τα ξένα κλειδιά είναι ιδιαίτερα χρήσιμα για τη συλλογή συνδυαστικών δεδομένων με την εντολή «SQL JOIN».[11]
- **Κανονικοποίηση– Normalization:** Ακολουθούνται πρακτικές όπως 1NF–3NF για αποφυγή

πλεονάζουσας πληροφορίας, ανωμαλιών και διατήρηση συνέπειας μεταξύ των πινάκων.[12]

Μερικά παραδείγματα εντολών CRUD είναι τα εξής:

- CREATE TABLE course (ID INT AUTO_INCREMENT PRIMARY_KEY, name VARCHAR (50));
- SELECT * FROM course WHERE name = 'Chemistry';
- UPDATE course SET name = 'Mathematics' WHERE ID = 5;
- DELETE FROM course WHERE ID = 5;

Η χρήση της MySQL στην εργασία επιτεύχθηκε μέσω του phpMyAdmin, ένα GUI (General User Interface) περιβάλλον που επιτρέπει τη διαχείριση της βάσης και συγγραφή SQL εντολών, το οποίο συνοδευόταν με την εγκατάσταση του προγράμματος XAMPP, για το οποίο γίνεται λόγος σε μεταγενέστερη υποενότητα.

2.1.5 JavaScript

Η JavaScript (JS) είναι μια από τις δημοφιλέστερες γλώσσες προγραμματισμού για τη δημιουργία δυναμικών ιστοσελίδων με πληθώρα εφέ και δυνατοτήτων. Εκτελείται στον φυλλομετρητή, δηλαδή στην πλευρά του χρήστη (*client-side*) και δίνει τη δυνατότητα διαχείρισης των HTML στοιχείων της σελίδας μέσω του DOM (Document Object Model), ανταλλαγής ασύγχρονων δεδομένων με τον διακομιστή μέσω της μεθόδου AJAX (Asynchronous JavaScript and XML), χωρίς να χρειάζεται ανανέωση ολόκληρης της σελίδας.

Η JavaScript μπορεί να προστεθεί με δύο τρόπους:

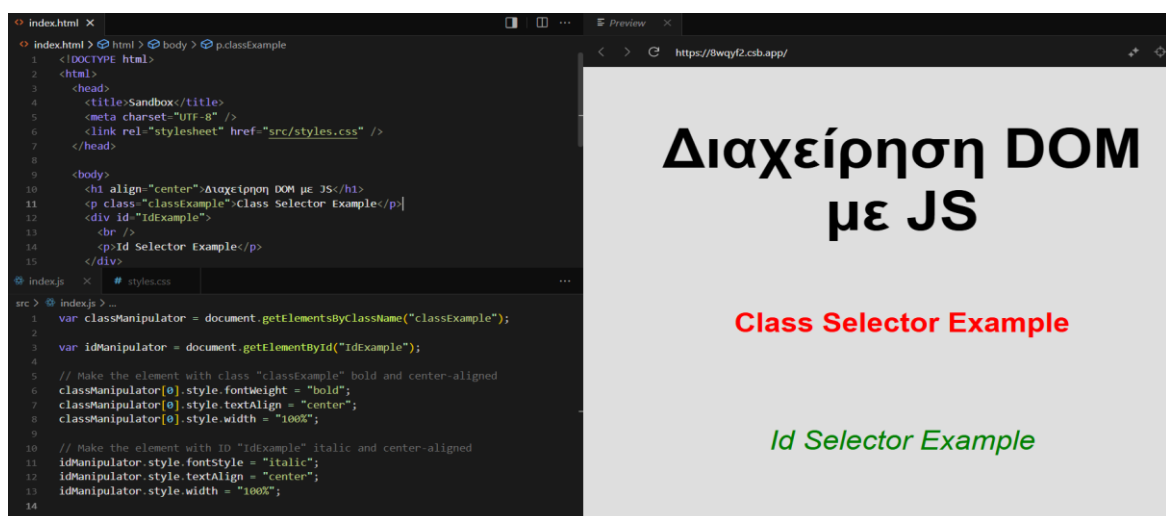
1. Ενσωμάτωση inline με χρήση script tags:

```
<script> alert("Hello World"); </script>
```

2. Εξωτερικά με χρήση .js αρχείου:

```
<script src="src/index.js" defer></script>
```

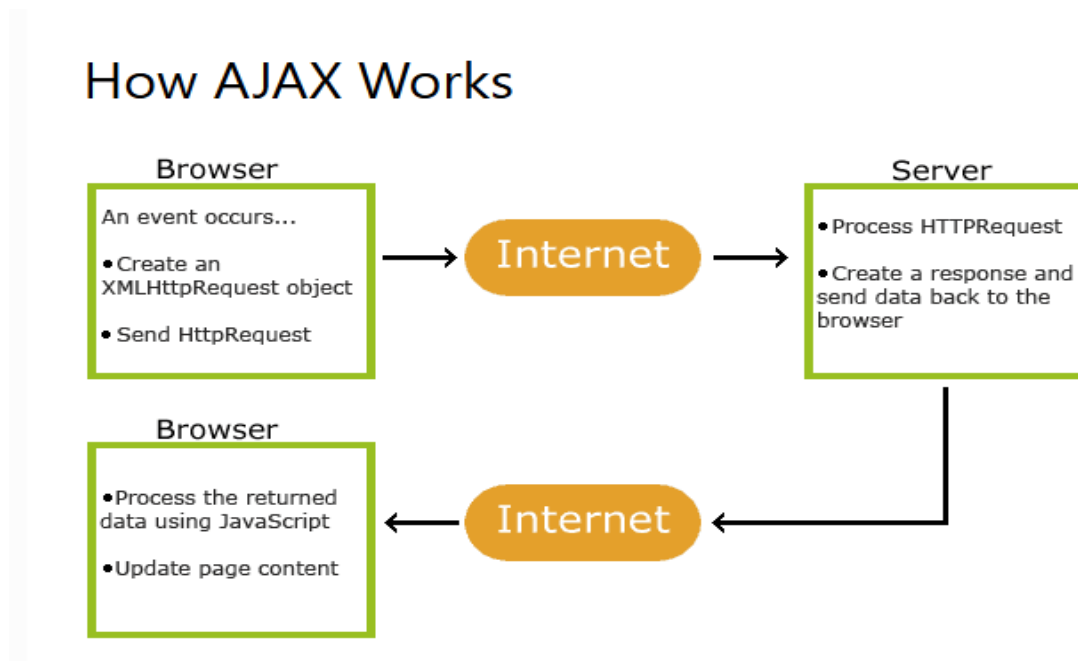
Ο δεύτερος τρόπος διευκολύνει στη συντήρηση του κώδικα της σελίδας. Ακολουθεί στην Εικόνα 9 παράδειγμα χρήσης JavaScript για τη διαχείριση στοιχείων του DOM με τη χρήση εντολών JavaScript.



Εικόνα 9 Διαχείριση ετικετών div με JavaScript

2.1.6 AJAX (Asynchronous JavaScript and XML)

Η AJAX είναι μια τεχνική που επιτρέπει την ασύγχρονη επικοινωνία μεταξύ του φυλλομετρητή και του διακομιστή, χωρίς να απαιτείται επαναφόρτωση της σελίδας, προσφέροντας ταχύτητα και βελτιστοποιημένη εμπειρία χρήστη. Ο τρόπος λειτουργίας της παρουσιάζεται στην Εικόνα 10.



Εικόνα 10 Τρόπος λειτουργίας AJAX

1. Ο χρήστης αλληλοεπιδρά με τη σελίδα (φορτώνει, κάνει κλικ, υποβάλλει μια φόρμα).
2. Η JavaScript δημιουργεί ένα αντικείμενο XMLHttpRequest.
3. Στέλνει αίτημα στον διακομιστή.
4. Ο διακομιστής το επεξεργάζεται και απαντά στο αίτημα.
5. Το αποτέλεσμα παραλαμβάνεται από τη JavaScript.
6. Η JavaScript ενημερώνει δυναμικά συγκεκριμένο μέρος της σελίδας, όπως για παράδειγμα κάποιον πίνακα. [13]

Η τεχνική AJAX προσφέρει σημαντικά πλεονεκτήματα, όπως η μερική συμπλήρωση και ανανέωση στοιχείων μιας σελίδας χωρίς την πλήρη επαναφόρτωση της, εξοικονόμηση πόρων δικτύου με τη μείωση αιτήσεων προς τον διακομιστή καθώς και ευελιξία μιας και επιτρέπει την άμεση επικύρωση στοιχείων πριν την πλήρη υποβολή μιας φόρμας.

2.1.7 PDO (PHP Data Objects)

Η βιβλιοθήκη **PDO (PHP Data Objects)** αποτελεί ένα σύγχρονο και αντικειμενοστραφές επίπεδο πρόσβασης σε βάσεις δεδομένων στην PHP, παρέχοντας ενιαίο προγραμματιστικό περιβάλλον για διάφορα συστήματα όπως MySQL. Η χρήση της PDO προάγει τη φορητότητα του κώδικα, επιτρέποντας τη μετακίνηση μεταξύ διαφορετικών βάσεων δεδομένων χωρίς την ανάγκη επανεγγραφής των ερωτημάτων.

Ένα από τα βασικά χαρακτηριστικά της PDO είναι η υποστήριξη «prepared statements», δηλαδή προετοιμασμένων εντολών SQL όπου η δομή ερωτήματος και οι παράμετροι διαχωρίζονται στην

εκτέλεση.

Αυτό προσφέρει ασφάλεια απέναντι σε επιθέσεις «SQL injection» και τη δυνατότητα επαναχρησιμοποίησης του ίδιου ερωτήματος με διαφορετικές τιμές, αυξάνοντας σημαντικά την απόδοση, καθώς η βάση δεδομένων εκτελεί μόνο μία φορά την ανάλυση και τον σχεδιασμό του ερωτήματος.[10].

Η PDO υποστηρίζει επίσης συναλλαγές (transactions) σύμφωνα με το πρότυπο ACID (Atomicity, Consistency, Isolation, Durability), επιτρέποντας την εκτέλεση πολλαπλών εντολών SQL ως μία ενιαία μονάδα εργασίας. [11]. Η PDO παρέχει επίσης ενοποιημένη διαχείριση σφαλμάτων μέσω εξαιρέσεων (PDOException), διευκολύνοντας τον καθαρό και δομημένο χειρισμό απρόσμενων καταστάσεων. Τέλος, η PDO επιτρέπει caching των prepared statements και μείωση των απαιτήσεων σε πόρους, συμβάλλοντας στη βελτιστοποίηση της απόδοσης

Η χρήση της PDO στη διπλωματική εργασία διασφαλίζει ότι ο κώδικας είναι ασφαλής έναντι SQL injection και εύκολος στη συντήρηση. [12]

2.1.8 Node.js

Η Node.js είναι ένα ανοιχτού κώδικα περιβάλλον εκτέλεσης JavaScript σε επίπεδο διακομιστή (server-side), το οποίο βασίζεται στη V8 μηχανή της Chromium και επιτρέπει τη συγγραφή back-end λογικής σε JavaScript. Χωρίς να απαιτεί νέο μαθησιακό βήμα από τους προγραμματιστές front-end, τους προσφέρει τη δυνατότητα να χρησιμοποιήσουν τη γνώση σε αυτούς JavaScript και για προγραμματισμό του back-end μιας λειτουργίας της σελίδας, υποστηρίζοντας έναν πλήρη κύκλο ανάπτυξης.

Το κύριο χαρακτηριστικό του Node.js είναι ο σύγχρονος, ασύγχρονος και χωρίς παγώματα τρόπος διαχείρισης εισόδων/εξόδων. Αντί να περιμένει η εκτέλεση ενός κώδικα λειτουργίας εισόδου/εξόδου, για παράδειγμα την ανάγνωση ενός αρχείου ή ερώτημα βάσης, ο κώδικας συνεχίζει, και όταν τα δεδομένα είναι διαθέσιμα. Έτσι, ο διακομιστής μπορεί να εξυπηρετεί χιλιάδες αιτήσεις ταυτόχρονα χωρίς μεγάλο αποτύπωμα μνήμης.[17]

Η δομή ενός Node.js προγράμματος βασίζεται συνήθως στο «CommonJS» πρότυπο, όπου κάθε αρχείο, βιβλιοθήκη ή πακέτο(module) εισάγεται με την εντολή require (...όνομα εισαγόμενου περιεχομένου). Το Node.js υποστηρίζεται από το Node Package Manager (npm), που παρέχει πρόσβαση σε εκατοντάδες χιλιάδες πακέτα, επιτρέποντας εύκολη ενσωμάτωση βιβλιοθηκών όπως Express, Socket.IO, Mongoose και άλλων, εμπλουτίζοντας τη λειτουργικότητα και επιταχύνοντας την ανάπτυξη της εφαρμογής.

Η χρήση του Node.js στην παρούσα διπλωματική εργασία έγινε για λόγους ευελιξίας και συνεργασίας με τα υπόλοιπα front-end στοιχεία, αλλά κυρίως για τη χρήση του npm πακέτου «@FullCalendar» που χρησιμοποιείται για τη δημιουργία του προγράμματος της σχολής.

2.1.9 Composer

Το Composer αποτελεί τον πρότυπο διαχειριστή εξαρτήσεων σε έργα PHP. Λειτουργεί σε επίπεδο «ανά-έργο», επιτρέποντας τον προγραμματιστή να δηλώσει τις βιβλιοθήκες και πλαίσια που χρειάζεται η εφαρμογή σε ένα αρχείο composer.json. Μέσω του Composer, όλα τα απαραίτητα πακέτα εγκαθίστανται στον φάκελο «vendor», ενώ το «composer.lock» εξασφαλίζει ότι όλοι οι χρήστες της εφαρμογής χρησιμοποιούν ακριβώς τις ίδιες, συμβατές εκδόσεις. Αυτή η προσέγγιση μειώνει τις ασυμβατότητες και ενισχύει τη διαλειτουργικότητα μεταξύ περιβαλλόντων.

Το Composer εντοπίζει και εγκαθιστά πακέτα μέσω του κύριου καταλόγου «Packagist», επιτρέποντας εύκολη ενσωμάτωση δημόσιων ή ιδιωτικών αποθετηρίων.[18]. Μια από τις πιο σημαντικές λειτουργίες του Composer είναι το αυτόματο autoloading μέσω PSR-4. Εκεί όπου παλιότερα απαιτούνταν οι εντολές include ή require, το Composer φορτώνει αυτόματα τις κλάσεις

όταν αυτές χρησιμοποιούνται. Αυτή η πρακτική οδηγεί σε πιο καθαρό, οργανωμένο, και συντηρήσιμο κώδικα PHP.[19]

Η εγκατάσταση του Composer γίνεται γρήγορα μέσω PHAR ή συστημικών πακέτων, ενώ μέσω εντολών όπως «composer require», «composer install», «composer update» και «composer remove», ο προγραμματιστής μπορεί να διαχειριστεί πλήρως τις εξαρτήσεις και τις εκδόσεις τους. Αυτή η δομή ενισχύει την επαγγελματική οργάνωση του έργου μιας και κάθε εξάρτηση είναι ξεκάθαρα δηλωμένη, φορτώνει αυτόματα με το αρχείο «vendor/autoload.php» και η εφαρμογή μπορεί να επεκταθεί με ασφάλεια, αποφεύγοντας χειροκίνητες συγχωνεύσεις βιβλιοθηκών.

2.2 Εργαλεία λογισμικού

Στην παρακάτω ενότητα αναλύονται τα εργαλεία XAMPP και ο κειμενογράφος Visual Studio που χρησιμοποιήθηκαν κατά την υλοποίηση της εφαρμογής.

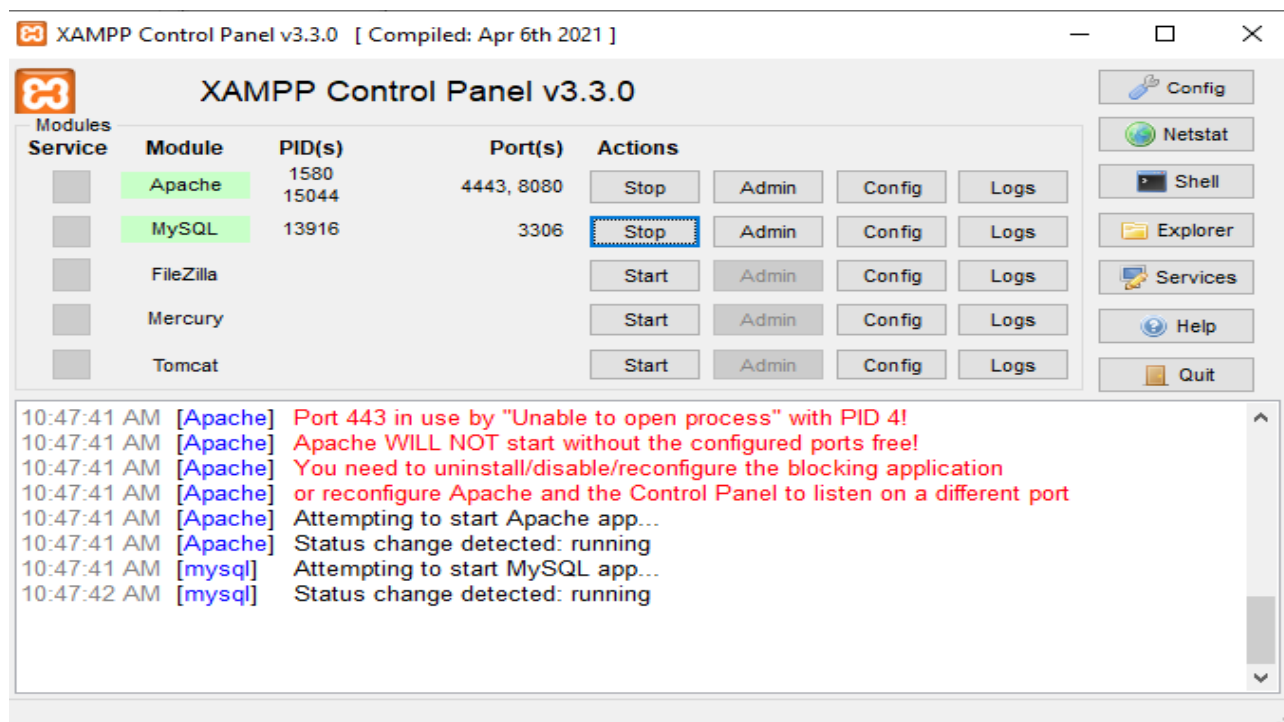
2.2.1 XAMPP

Το **XAMPP** αποτελεί ένα ολοκληρωμένο, ανοιχτού κώδικα πακέτο για τοπική διαχείριση περιβάλλοντος ανάπτυξης. Περιλαμβάνει το διακομιστή Apache, τη MariaDB/MySQL, την γλώσσα προγραμματισμού PHP, το εργαλείο **phpMyAdmin** για γραφική διαχείριση βάσεων δεδομένων, καθώς και άλλα βοηθητικά εργαλεία, όπως Perl και FileZilla και υποστήριξη για διακομιστές Tomcat και το διαχειριστικό περιβάλλον του παρουσιάζεται στην Εικόνα 11 Xampp Control Panel. Μέσω του XAMPP, η τοπική εκτέλεση αρχείων PHP μέσω `http://localhost/` γίνεται εύκολα, χωρίς να απαιτείται σύνδεση σε απομακρυσμένο διακομιστή ή σε περιβάλλον παραγωγής. Αρκεί τα αρχεία της εφαρμογής να τοποθετούνται στο φάκελο **htdocs**, ο οποίος λειτουργεί ως document root (/) του Apache. Για παράδειγμα, αν δημιουργήσουμε μέσα στο htdocs έναν φάκελο με όνομα «myapp», μπορούμε να εκτελέσουμε την εφαρμογή μέσω της διεύθυνσης `http://localhost/myapp`. Η ακριβής τοποθεσία και διαδρομή του φακέλου είναι σημαντική, καθώς ο Apache σε συνδυασμό με την PHP επεξεργάζεται μόνο αρχεία που βρίσκονται εντός του htdocs, σύμφωνα με την προκαθορισμένη ρύθμιση «DocumentRoot» στο βασικό του αρχείο ρυθμίσεων.

Οι ρυθμίσεις του XAMPP και του Apache—όπως η αλλαγή θύρας σύνδεσης (π.χ. από 80 σε 8080) ή ρύθμιση διεύθυνσης IP μπορούν να γίνουν είτε μέσω των αρχείων «httpd.conf» και «httpd-ssl.conf», είτε μέσω ενός αρχείου ρύθμισης «config.ini» του «Control Panel» του XAMPP. Το config.ini χρησιμοποιείται για διαχείριση ρυθμίσεων σε γραφικό περιβάλλον, συμπεριλαμβανομένης της αλλαγής θυρών, hostname ή άλλων εξελεγμένων επιλογών, χωρίς την ανάγκη χειροκίνητης επεξεργασίας αρχείων.[20]

Το XAMPP υποστηρίζει την εγκατάστασή του ως Windows Service (ή αντίστοιχη υπηρεσία σε Linux/macOS), επιτρέποντας την αυτόματη εκκίνηση μαζί με το λειτουργικό σύστημα. Το “Control Panel” παρέχει εργαλεία όπως «netstat» για τον έλεγχο της θύρας που ορίσαμε να τρέχει η εφαρμογή μας για την αποφυγή συγκρούσεων, Shell για πρόσβαση σε κονσόλα, καθώς και επιλογές για την ενεργοποίηση ή απενεργοποίηση των υπηρεσιών. Αυτό καθιστά τη διαχείριση εύχρηστη ακόμη και για χρήστες με λιγότερη εμπειρία.

Αν και το XAMPP αξιοποιείται κυρίως για τοπική ανάπτυξη και δοκιμές, η χρήση του στο πλαίσιο της διπλωματικής εργασίας είναι πλήρως δικαιολογημένη: επιταχύνει τις δοκιμές PHP/SQL, απλοποιεί την τοπική διαχείριση εφαρμογής και εξαλείφει περιττές εξαρτήσεις από εξωτερικά συστήματα. Ταυτόχρονα, παρέχει την προσομοίωση περιβάλλοντος παραγωγής με τη χρήση virtual hosts, SSL και του phpMyAdmin), καθιστώντας το μια κατάλληλη επιλογή για ανάπτυξη σταθερών, διαδικτυακών εφαρμογών.



Εικόνα 11 Xampp Control Panel

2.2.2 Visual Studio Code

Το Visual Studio Code (VS Code) είναι ένας δημοφιλής επεξεργαστής προγραμματισμού, με λειτουργικότητα cross-platform, ο οποίος αναπτύχθηκε από τη Microsoft με βάση την ανοιχτού κώδικα μηχανή "Monaco". Παρέχεται σε λειτουργικά συστήματα Windows, macOS και Linux, και αποτελεί μια ευέλικτη αλλά ισχυρή λύση για τη συγγραφή κώδικα σε PHP, JavaScript, CSS, HTML και άλλες γλώσσες.

Το VS Code παρέχει ενσωματωμένο τερματικό για το οποιοδήποτε έργο, όπου ο προγραμματιστής μπορεί να τρέξει εντολές `npm`, `composer` ή άλλα εργαλεία, χωρίς να χρειάζεται να φύγει από το προγραμματιστικό του περιβάλλον. Η εγγενής υποστήριξη για `npm`, η οποία είναι εκ των έσω, διευκολύνει την εγκατάσταση και διαχείριση πακέτων JavaScript και Node.js μέσω της εντολής «`npm install`», ενώ η εγκατάσταση `Composer` πακέτων μπορεί επίσης να πραγματοποιηθεί απευθείας από το περιβάλλον ανάπτυξης.

Το VS Code υποστηρίζει εκτενές οικοσύστημα επεκτάσεων (*Extensions*) που βελτιώνουν σημαντικά την εμπειρία του προγραμματιστή. Για παράδειγμα, διαθέτει τις πιο δημοφιλείς επεκτάσεις για PHP, όπως «`PHP Intelephense`» και «`PHP Debug`», οι οποίες παρέχουν αυτόματη συμπλήρωση κώδικα (*IntelliSense*) και εντοπισμό σφαλμάτων. Υπάρχουν επίσης εργαλεία για μορφοποίηση κώδικα, αυτόματου σχολιασμού και εργαλεία διαχείρισης βάσεων, ενώ οι χρήστες JavaScript και Node.js μπορούν να επωφεληθούν από επεκτάσεις όπως `ESLint`, `IntelliSense`, `Chrome Debugger` κ.α.

Η εγκατάσταση των επεκτάσεων γίνεται εύκολα και χωρίς να απαιτείται έξοδος από το περιβάλλον ανάπτυξης. Με αυτόν τον τρόπο, το VS Code μεταμορφώνεται σε ένα πλήρως εξοπλισμένο προγραμματιστικό περιβάλλον. Με τον συνδυασμό του ενσωματωμένου τερματικού, των ισχυρών επεκτάσεων και της υποστήριξης για κοινά εργαλεία όπως `composer` και `npm`, το VS Code παρέχει ένα αποδοτικό, ευέλικτο και προσαρμόσιμο περιβάλλον ανάπτυξης που ενδείκνυται για τη δημιουργία σύγχρονων και ολοκληρωμένων εφαρμογών, με τη χρήση διαφορετικών γλωσσών προγραμματισμού.

2.3 Πρόσθετες βιβλιοθήκες λογισμικού

Στην ενότητα αυτή παρουσιάζονται συνοπτικά βιβλιοθήκες λογισμικού που χρησιμοποιήθηκαν κατά τη δημιουργία της εφαρμογής. Όλες οι παρακάτω βιβλιοθήκες που παρουσιάζονται στον Πίνακα 1 προστέθηκαν είτε με τη χρήση του διαχειριστή πακέτων «npm», είτε με τον διαχειριστή εξαρτήσεων της PHP «Composer»

Πίνακας 1 Βοηθητικές βιβλιοθήκες λογισμικού

Όνομα βιβλιοθήκης	Πρόγραμμα εγκατάστασης	Περιγραφή - Χρήση
dompdf	Composer (PHP)	Renderer PDF: μετατρέπει HTML/CSS σε PDF (π.χ. τιμολόγια, εκτυπώσεις)
masterminds/html5	Composer (PHP)	HTML5 αναλυτής για PHP· συχνή εξάρτηση του dompdf
nesbot/carbon	Composer (PHP)	Βελτιωμένα αντικείμενα ημερομηνίας/ώρας (διαφορές, μορφοποιήσεις).
phenx/php-font-lib	Composer (PHP)	Βιβλιοθήκη γραμματοσειρών για απόδοση PDF
phenx/php-svg-lib	Composer (PHP)	Βιβλιοθήκη SVG για απόδοση/μετατροπή σε PDF
sabberworm/php-css-parser	Composer (PHP)	Αναλυτής CSS που διαβάζει stylesheet για το dompdf
setasign/fpdi	Composer (PHP)	Εισαγωγή/συγχώνευση σελίδων από υπάρχοντα PDF
symfony/*	Composer (PHP)	Εργαλεία/Polyfills της Symfony (helpers, events κ.λπ.) για παροχή συμβατότητας σε βοηθητικές συναρτήσεις και επεκτάσεις PHP
tecnickcom/tcpdf	Composer (PHP)	Μηχανή δημιουργίας PDF (συμπληρωματική του dompdf)
jasny/*	Composer (PHP)	Βοηθητικές βιβλιοθήκες PHP (utilities, helpers) για πιστοποίηση και εξουσιοδότηση χρηστών

improved/*	Composer (PHP)	Μικρές utility βιβλιοθήκες/βελτιώσεις PHP
psr/*	Composer (PHP)	Τυποποιημένες διεπαφές PSR (logs, containers, cache). Απαραίτητο για τη χρήση του autoloader
composer (vendor/composer)	Composer (PHP)	Autoloader & metadata του Composer για φόρτωση πακέτων.
@fullcalendar	Node.JS (JavaScript)	Plugins του FullCalendar (core, dayGrid κ.λπ.) για διαδραστικό ημερολόγιο
fullcalendar	Node.JS (JavaScript)	Η κύρια συλλογή πακέτων του FullCalendar για εμφάνιση ημερολογίου στο UI
preact	Node.JS (JavaScript)	Ελαφριά βιβλιοθήκη UI τύπου React· χρησιμοποιείται από το FullCalendar

2.4 Συμπεράσματα

Στο παρόν κεφάλαιο τεκμηριώθηκε το θεωρητικό υπόβαθρο και τα εργαλεία που συγκροτούν τη στοίβα ανάπτυξης του συστήματος. Η επιλογή των τεχνολογιών **HTML**, **CSS**, **PHP**, **JavaScript** και **MySQL** δεν έγινε αποσπασματικά, αλλά ως ενιαίο σύνολο με σαφείς ρόλους: η HTML ορίζει τον σκελετό και τη ιεραρχία του περιεχομένου, η CSS διασφαλίζει συνεκτική και επεκτάσιμη μορφοποίηση, ενώ η JavaScript προσφέρει διαδραστικότητα στην πλευρά του πελάτη με αξιοποίηση του DOM. Η χρήση της **AJAX** επιτρέπει στοχευμένες, ασύγχρονες ενημερώσεις τμημάτων της σελίδας χωρίς πλήρη επαναφόρτωση, βελτιώνοντας την εμπειρία χρήστη και μειώνοντας τον περιττό φόρτο αιτημάτων. Στην πλευρά του διακομιστή, η PHP υλοποιεί την επιχειρησιακή λογική και παράγει δυναμικά περιεχόμενο, ενώ η MySQL προσφέρει ώριμη, σχεσιακή οργάνωση δεδομένων με πρακτικές κανονικοποίησης και τυπικές CRUD λειτουργίες.

Κομβικός άξονας του κεφαλαίου ήταν η ασφάλεια και η ποιότητα πρόσβασης στα δεδομένα μέσω της **PDO** (PHP Data Objects). Η υιοθέτηση prepared statements και δεσμευμένων παραμέτρων αποτρέπει επιθέσεις SQL injection, ενώ η υποστήριξη συναλλαγών (ACID) και ο ομοιόμορφος χειρισμός σφαλμάτων μέσω εξαιρέσεων συμβάλλουν σε πιο ανθεκτική και συντηρήσιμη υλοποίηση. Έτσι, η σύνδεση της εφαρμογής με τη βάση δεν αποτελεί απλώς τεχνική λεπτομέρεια, αλλά μέρος μιας συνειδητής στρατηγικής για αξιόπιστη και ασφαλή λειτουργία.

Παράλληλα, η ένταξη εργαλείων και βιβλιοθηκών τρίτων ενισχύει ουσιαστικά τη λειτουργικότητα. Το **Node.js** (npm) χρησιμοποιείται ως οικοσύστημα πακέτων για την ενσωμάτωση του **FullCalendar**, το οποίο προσφέρει διαδραστική ημερολογιακή απεικόνιση στο UI· η παρουσία του preact ως εξάρτηση συνεισφέρει σε ελαφριά και αποδοτική απόδοση των στοιχείων. Στον χώρο της PHP, ο **Composer** οργανώνει μεθοδικά τις εξαρτήσεις, διασφαλίζει την αναπαραγωγή των

σωστών εκδόσεων μέσω composer.lock και παρέχει autoloading (PSR-4), γεγονός που καθιστά τον κώδικα καθαρό, φορητό και επεκτάσιμο. Οι βιβλιοθήκες dompdf με τα συναφή parsers και βοηθητικά πακέτα, Carbon για τον τυποποιημένο χειρισμό ημερομηνιών και ωρών και τα πακέτα PSR και Symfony προσθέτουν κρίσιμες δυνατότητες χωρίς να επιβαρύνουν την αναγνωσιμότητα ή τη συντήρηση του έργου.

Τέλος, τα εργαλεία **XAMPP** και **Visual Studio Code** διευκολύνουν τον κύκλο ανάπτυξης. Το πρώτο παρέχει ενοποιημένο τοπικό περιβάλλον (Apache, MariaDB/MySQL, PHP, phpMyAdmin) για γρήγορες δοκιμές μέσω localhost, ενώ το δεύτερο προσφέρει επεκτάσεις για PHP/JS, ενσωματωμένο τερματικό και ομαλή συνεργασία με τον διαχειριστή πακέτων npm και Composer. Συνολικά, η επιλεγμένη στοίβα επιτυγχάνει ισορροπία ανάμεσα σε απλότητα εγκατάστασης, επεκτασιμότητα, ασφάλεια και ευχρηστία.

Με αυτό το υπόβαθρο, το επόμενο κεφάλαιο μπορεί να εστιάσει απρόσκοπτα στην ανάλυση απαιτήσεων, στη σχεδίαση του συστήματος και στην παρουσίαση των ροών αλληλεπίδρασης με τη χρήση διαγραμμάτων περιπτώσεων χρήσης. Αναφέρονται επίσης οι πολιτικές ασφάλειας που χρησιμοποιούνται για τη διασφάλιση της ασφάλειας και τις ακεραιότητας των δεδομένων της εφαρμογής.

Κεφάλαιο 3: Ανάλυση και σχεδίαση συστήματος

Στο παρόν κεφάλαιο αναλύονται διεξοδικά οι απαιτήσεις της εφαρμογής διαχείρισης ωρολογιακού προγράμματος. Αρχικά παρουσιάζονται οι θεμελιώδεις λειτουργίες και δυνατότητες του συστήματος, για κάθε χρήση ξεχωριστά. Ακολουθούν διαγράμματα χρήσης-περίπτωσης που αφορούν τις αλληλεπιδράσεις μεταξύ των χρηστών στο πλαίσιο χρήσης της εφαρμογής. Επίσης, γίνεται η επισκόπηση των πινάκων της βάσης δεδομένων που χρησιμοποιήθηκαν και του διαγράμματος οντοτήτων–συσχετίσεων. Τέλος καταγράφονται οι μέθοδοι που χρησιμοποιήθηκαν για τη διασφάλιση της ασφάλειας της σελίδας και των δεδομένων του χρήστη αυτής.

3.1 Ανάλυση απαιτήσεων

Στο παρόν τμήμα περιγράφονται οι βασικές λειτουργικές απαιτήσεις της εφαρμογής διαχείρισης ωρολογίου προγράμματος. Η παρούσα ιστοσελίδα υποστηρίζει την εγγραφή χρηστών μέσω του ιδρυματικού τους email και ο κάθε χρήστης θα εμπίπτει σε έναν εκ των τεσσάρων ρόλων χρήστη.

Οι ρόλοι αυτοί είναι οι εξής:

- Διαχειριστής
- Γραμματεία
- Καθηγητής
- Φοιτητής

και θα αναλυθούν εκτενώς στην επόμενη ενότητα.

Για την ομαλή λειτουργία του συστήματος, στους χρήστες που έχουν ρόλο με περισσότερες διαχειριστικές δυνατότητες, προσφέρεται η δυνατότητα καταχώρησης και επικαιροποίησης στοιχείων για τις αίθουσες της σχολής, τα μαθήματα, τους καθηγητές . Συγκεκριμένα, οι πληροφορίες που είναι απαραίτητες είναι οι εξής:

- Τα στοιχεία των καθηγητών, μαζί με τα μαθήματα που καλούνται να διδάξουν.
- Οι αίθουσες διδασκαλίας του τμήματος.
- Πληροφορίες για τα εργαστήρια, όπως ο εξοπλισμός και η χωρητικότητά τους.
- Οι απαιτήσεις των μαθημάτων σε ώρες θεωρίας και εργαστηρίου όταν αυτό υπάρχει.

Για το σχεδιασμό του προγράμματος, η εφαρμογή παρέχει διαδραστικό πίνακα σύνθεσης σε μορφή ημερολογίου που ελέγχεται άμεσα για πιθανές συγκρούσεις στις δηλώσεις που κάνουν οι καθηγητές. Τα σενάρια σύγκρουσης είναι τα εξής:

- Ένας καθηγητής δεν μπορεί να έχει παραπάνω από ένα μάθημα την ίδια ώρα.
- Δεν επιτρέπεται να υπάρχει χρονική επικάλυψη σε μαθήματα κορμού του ίδιου εξαμήνου. Μαθήματα επιλογής, είναι εφικτό να διεξάγονται την ίδια ώρα, αλλά δεν είναι επιθυμητό.
- Σε μία αίθουσα δεν μπορεί να διεξάγεται παραπάνω από ένα μάθημα.
- Η μόνη επιτρεπτή περίπτωση επικάλυψης μαθημάτων του ίδιου εξαμήνου είναι αυτή όπου έχουμε δύο μαθήματα επιλογής ή διαφορετικών κατευθύνσεων.

Σε περίπτωση σύγκρουσης, εμφανίζεται ξεκάθαρο μήνυμα ειδοποίησης, ενώ υποστηρίζεται ο ορισμός προγράμματος για την εξεταστική περίοδο με την ίδια ακριβώς μέθοδο δήλωσης-επιβεβαίωσης από τους χρήστες με ρόλο «γραμματεία». Σε ένα μάθημα μπορούν να αντιστοιχηθούν παραπάνω από ένας καθηγητές και ένα μάθημα μπορεί να διεξαχθεί σε παραπάνω από μία αίθουσα ταυτόχρονα.

Το σύστημα προσφέρει σε όλους τους ρόλους εκτός από αυτό του «μαθητή», πρόσβαση σε σελίδα που μπορούν να επιβεβαιώσουν τη διαθεσιμότητα μιας αίθουσας κάποια συγκεκριμένη χρονική στιγμή, τις ώρες που είναι ελεύθεροι άλλοι καθηγητές, τα μαθήματα που έχουν δηλωθεί ανά εξάμηνο μέσω των φίλτρων προγράμματος, για να διευκολυνθεί η επιλογή χρονοθυρίδων κατά τη δήλωση μαθημάτων χωρίς συγκρούσεις. Επίσης, παρέχεται η δυνατότητα αρχειοθέτησης ολοκληρωμένων προγραμμάτων και η δυνατότητα επαναχρησιμοποίησής τους σε επόμενα ακαδημαϊκά έτη. Για να προγραμματιστούν έκτακτες αναπληρώσεις μαθημάτων, οι καθηγητές πρέπει να εξακριβώνουν τη διαθεσιμότητα αίθουσας το συγκεκριμένο χρονικό διάστημα που επιθυμούν, και να λάβουν επιβεβαίωση ή απόρριψη του αιτήματός τους από τη γραμματεία. Παρέχονται επίσης πολλαπλά φίλτρα για την απεικόνιση του τελικού προγράμματος, ανά εξάμηνο, αίθουσα, καθηγητή ή εξάμηνο. Τέλος, όλοι οι χρήστες έχουν πρόσβαση στην εκτύπωση του τελικού προγράμματος σε μορφή pdf.

3.2 Ρόλοι και Δικαιώματα Πρόσβασης

Η πρόσβαση στις λειτουργίες της εφαρμογής καθορίζεται αυστηρά από τα εκχωρημένα προνόμια κάθε χρήστη. Κάθε απόπειρα εκτέλεσης ενέργειας χωρίς την απαιτούμενη εξουσιοδότηση απορρίπτεται άμεσα, διασφαλίζοντας την ακεραιότητα και την ασφάλεια του συστήματος. Σε αυτή την ενότητα θα αναφέρουμε με λεπτομέρεια στις δυνατότητες και τις αρμοδιότητες των τεσσάρων ρόλων χρήστη της εφαρμογής.

3.2.1 Διαχειριστής

Ο χρήστης με ρόλο Διαχειριστή έχει πρόσβαση σε όλες τις διαχειριστικές λειτουργίες της σελίδας, όπως η καταχώρηση και η ενημέρωση των στοιχείων πανεπιστημίων και των τμημάτων τους, η προσθήκη χρηστών υψηλών δικαιωμάτων, όπως καθηγητές και μέλη της γραμματείας της σχολής. Είναι επίσης σε θέση να βοηθήσει με θέματα που αφορούν τους λογαριασμούς άλλων χρηστών, όπως για παράδειγμα διαγραφή χρήστη, τη μετακίνηση καθηγητή σε άλλο πανεπιστημιακό ίδρυμα, την προσθήκη ή αφαίρεση μαθημάτων σε καθηγητή κ.α.

3.2.2 Γραμματεία

Ο χρήστης με ρόλο Γραμματεία έχει την ευθύνη για τη διαχείριση του συνόλου των ακαδημαϊκών δεδομένων του τμήματος. Συγκεκριμένα, μέσω της εφαρμογής καταχωρούνται και επικαιροποιούνται τα χαρακτηριστικά των μαθημάτων όπως οι διδακτικές ώρες, το εξάμηνο που ανήκουν, οι υπεύθυνοι διδάσκοντες καθώς και τα τεχνικά στοιχεία των αιθουσών και εργαστηρίων όπως η χωρητικότητα και ο εξοπλισμός που παρέχουν. Επίσης, ο χρήστης γραμματεία είναι υπεύθυνος για την καταχώρηση των διδασκόντων και των στοιχείων τους, στα εκάστοτε πανεπιστημιακά τμήματα.

Όσον αφορά τη δημιουργία του προγράμματος, η Γραμματεία αξιοποιεί τα φίλτρα του προγράμματος για τον έλεγχο διαθεσιμότητας καθηγητών, αιθουσών και χρονοθυρίδων για τη δημιουργία του εξαμηνιαίου και εξεταστικού ωρολογίου προγράμματος με βάση τα αιτήματα των καθηγητών για ώρες και αίθουσες διδασκαλίας. Κατά την εισαγωγή των δεδομένων ελέγχονται αυτόματα οι συγκρούσεις στις δηλώσεις χρονοθυρίδων και σε περίπτωση παραβίασης κανόνα εμφανίζεται άμεση ειδοποίηση σφάλματος. Επιπλέον, διαχειρίζεται αιτήματα έκτακτων αναπληρώσεων, ελέγχοντας για διαθεσιμότητα αίθουσας και χρονοθυρίδας και αποστέλλει στον αρμόδιο διδάσκοντα επιβεβαίωση ή απόρριψη του αιτήματός του αυτόματα μέσω του συστήματος ειδοποιήσεων. Τέλος, ολοκληρωμένα ωρολόγια προγράμματα αρχειοθετούνται και μπορούν να επαναχρησιμοποιηθούν σε επόμενα ακαδημαϊκά έτη ή και να εκτυπωθούν όπως ακριβώς ήταν μειώνοντας σημαντικά το φόρτο εργασίας.

3.2.3 Καθηγητής

Ο χρήστης με ρόλο Καθηγητή, έχουν πρόσβαση στη δική τους διαχειριστική επιφάνεια για να μπορεί να δηλώσει τις ώρες που επιθυμεί για τα μαθήματά τους και να βλέπει ή να εκτυπώσει το τελικό τους πρόγραμμα, καθώς και το πρόγραμμα της εξεταστικής. Του δίνεται επίσης η δυνατότητα να ελέγξει τις διαθεσιμότητες σε αίθουσες, αλλά και τις ώρες διδασκαλίας κάποιου άλλου καθηγητή, ώστε να είναι σε θέση να προσαρμόσει τα αιτήματά τους για χρονοθυρίδες ή αίθουσες. Τέλος, μπορεί να δημιουργήσει αιτήματα για αναπλήρωση μαθήματος λαμβάνοντας υπόψιν το υπάρχον σταθερό πρόγραμμα του εξαμήνου.

3.2.4 Φοιτητής

Ο χρήστης με ρόλο Φοιτητή, δεν κατέχει διαχειριστικά και λειτουργικά δικαιώματα και δεν αλληλοεπιδρά με κάποιον άλλο χρήστη της ιστοσελίδας. Μπορεί απλά να δηλώσει ή να αφαιρέσει τα μαθήματα που θέλει να εμφανίζονται στο πρόγραμμά του, για το τρέχον, παλαιότερο ή μελλοντικό εξάμηνο φοίτησης και να έχει στη διάθεση του το δικό του εξατομικευμένο πρόγραμμα. Επίσης του δίνεται η δυνατότητα να εκτυπώσει το πρόγραμμά του σε μορφή pdf.

3.3 Σενάρια χρήσης περίπτωσης

Στην παρούσα ενότητα παρουσιάζεται η συνολική επισκόπηση των δυνατοτήτων κάθε ρόλου στο σύστημα με χρήση διαγραμμάτων, προκειμένου να αποτυπωθεί με σαφήνεια το πλήρες εύρος των λειτουργιών που υποστηρίζει η εφαρμογή. Στο ενιαίο διάγραμμα που απεικονίζεται στην Εικόνα 12, παρουσιάζονται οι ενέργειες που μπορεί να εκτελέσει ο κάθε χρήστης. Όπως φαίνεται και στην εικόνα, ο ρόλος χρήστη «γραμματεία» και «διαχειριστής» έχουν όλες τις ίδιες

διαχειριστικές δυνατότητες και όπως θα φανεί και παρακάτω σε αυτή την ενότητα, οι κύριες ροές αλληλεπίδρασης είναι μεταξύ των καθηγητών και τις γραμματείας.

Στις επιμέρους ενότητες του κεφαλαίου παρουσιάζονται με διαγράμματα περιπτώσεων-χρήσης οι αλληλεπιδράσεις αυτές και στο κεφάλαιο 4 που εξετάζονται όλες οι σελίδες στις οποίες έχει πρόσβαση ο κάθε ρόλος, θα φανεί με ακόμη περισσότερη τεχνική λεπτομέρεια ο τρόπος που υλοποιήθηκαν αυτές.



Εικόνα 12 Δυνατότητες και δικαιώματα χρηστών

3.3.1 Σενάριο δήλωσης χρονοθυρίδας μαθήματος

Η δήλωση χρονοθυρίδας από τους καθηγητές και η επικύρωση ή απόρριψη του αιτήματος από τη γραμματεία, είναι μία από τις βασικότερες λειτουργίες της ιστοσελίδας και το σενάριο χρήσης περίπτωσης παρουσιάζεται στην Εικόνα 13. Στην παρούσα περίπτωση-χρήσης απεικονίζεται η

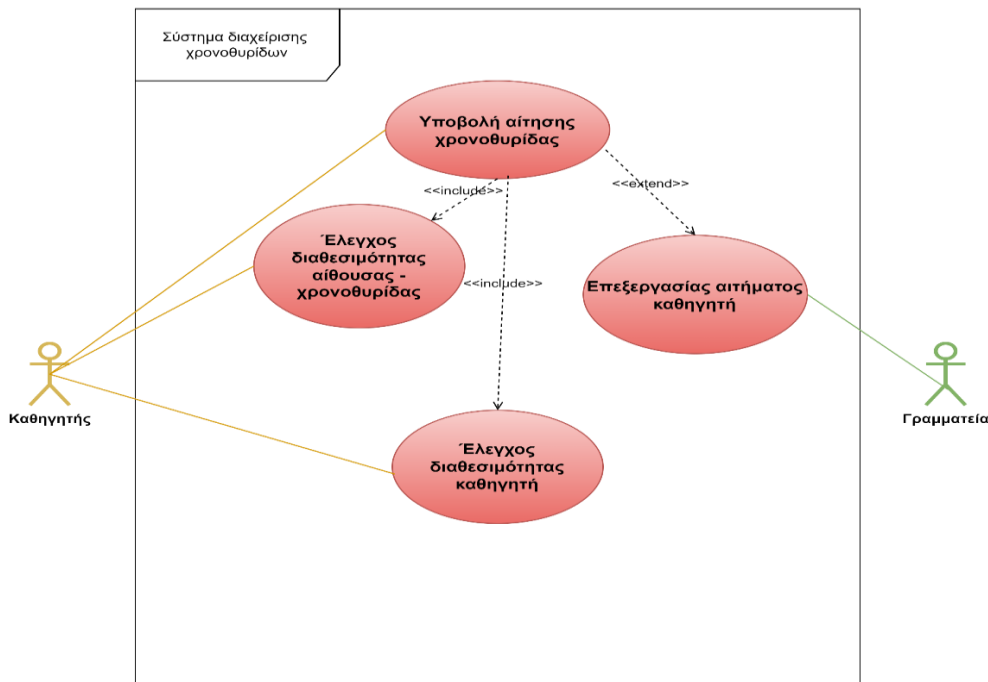
διαδικασία με την οποία ο χρήστης με ρόλο Καθηγητή υποβάλλει αίτημα για κράτηση χρονοθυρίδας στο ωρολόγιο πρόγραμμα, καθώς και η μεταγενέστερη επεξεργασία του αιτήματος από τη Γραμματεία. Το διάγραμμα οργανώνεται γύρω από τον κεντρικό κύκλο «Υποβολή αίτησης χρονοθυρίδας», ο οποίος περιλαμβάνει δύο βασικούς ελέγχους διαθεσιμότητας:

- Κατά την εισαγωγή του αιτήματος το σύστημα ελέγχει αμέσως εάν η επιλεγμένη αίθουσα είναι ελεύθερη στο επιθυμητό χρονικό διάστημα.
- Παράλληλα, εξακριβώνει ότι ο Καθηγητής δεν έχει άλλη διδασκαλία την ίδια ώρα, αποτρέποντας τις επικαλύψεις στο ατομικό του πρόγραμμα.

Μόλις οι έλεγχοι ολοκληρωθούν επιτυχώς, το αίτημα καταγράφεται στο σύστημα. Κατόπιν, ενεργοποιείται, μέσω σχέσης <<extend>> ακολουθεί η «Επεξεργασία αιτήματος», από τη Γραμματεία:

- Η Γραμματεία λαμβάνει την καταχωρημένη αίτηση και επαληθεύει εκ νέου την ορθότητα του αιτήματος χρησιμοποιώντας το «αιθουσολόγιο» για επιβεβαίωση διαθεσιμότητας.
- Με βάση τα αποτελέσματα του ελέγχου η Γραμματεία είτε εγκρίνει την αίτηση, οπότε η χρονοθυρίδα επιβεβαιώνεται, είτε την απορρίπτει.

Το συγκεκριμένο σενάριο διασφαλίζει την ορθή διαχείριση συγκρούσεων χρόνου διδασκαλίας, ελαχιστοποιώντας την ανάγκη χειροκίνητης επιβεβαίωσης και επιταχύνοντας τον προγραμματισμό των ωρολογίων.



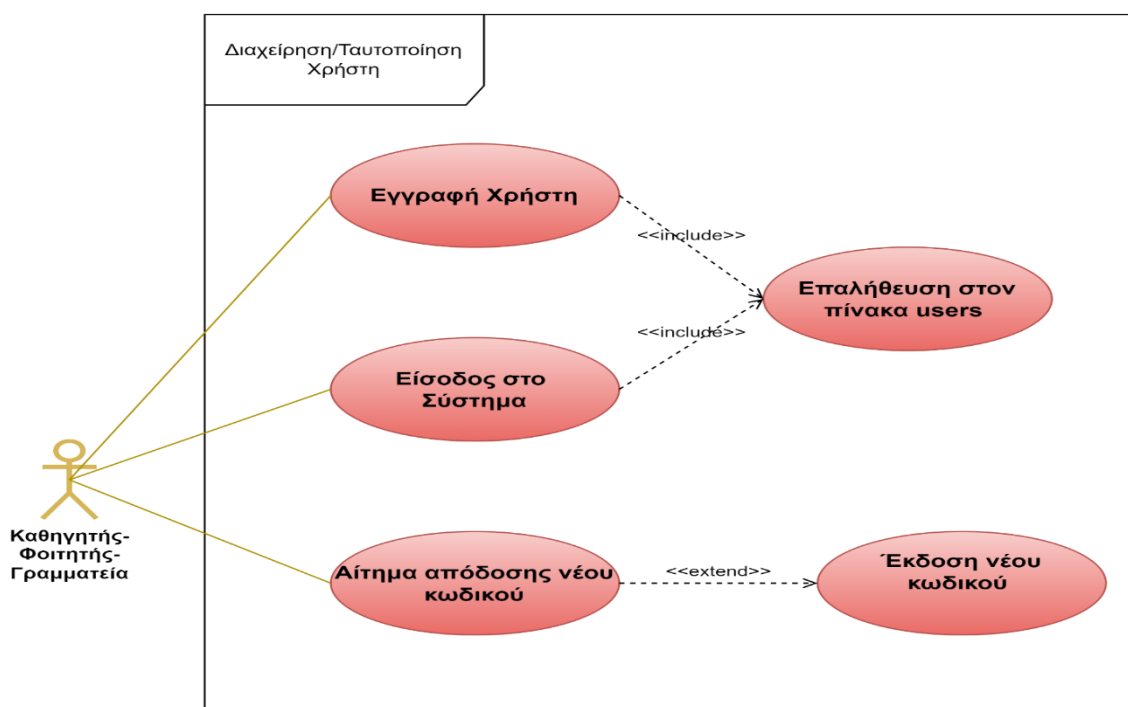
Εικόνα 13 Σενάριο δήλωσης χρονοθυρίδας μαθήματος

3.3.2 Σενάρια Διαχείρισης Χρήστη

Στο ενιαίο διάγραμμα χρήσης-περίπτωσης της Εικόνα 14, αποτυπώνονται όλες οι βασικές

διεργασίες πιστοποίησης και εξουσιοδότησης των χρηστών. Ο χρήστης αρχικά χρησιμοποιεί τη λειτουργία «Εγγραφή χρήστη», όπου το σύστημα εκτελεί υποχρεωτικά τον έλεγχο «Επαλήθευση στον πίνακα users» (<<include>>) της βάσεως δεδομένων για την επαλήθευση της ύπαρξης του χρήστη στο σύστημα. Προκειμένου να διασφαλιστεί η εγκυρότητα των διαπιστευτηρίων. Σε περίπτωση που ο χρήστης ξεχάσει τον κωδικό του, υποβάλει το «Αίτημα απόδοσης νέου κωδικού», το οποίο επεκτείνει (<<extend>>) τη διαδικασία «Έκδοση Νέου Κωδικού», όπου ο χρήστης καλείται να εισάγει νέο κωδικό.

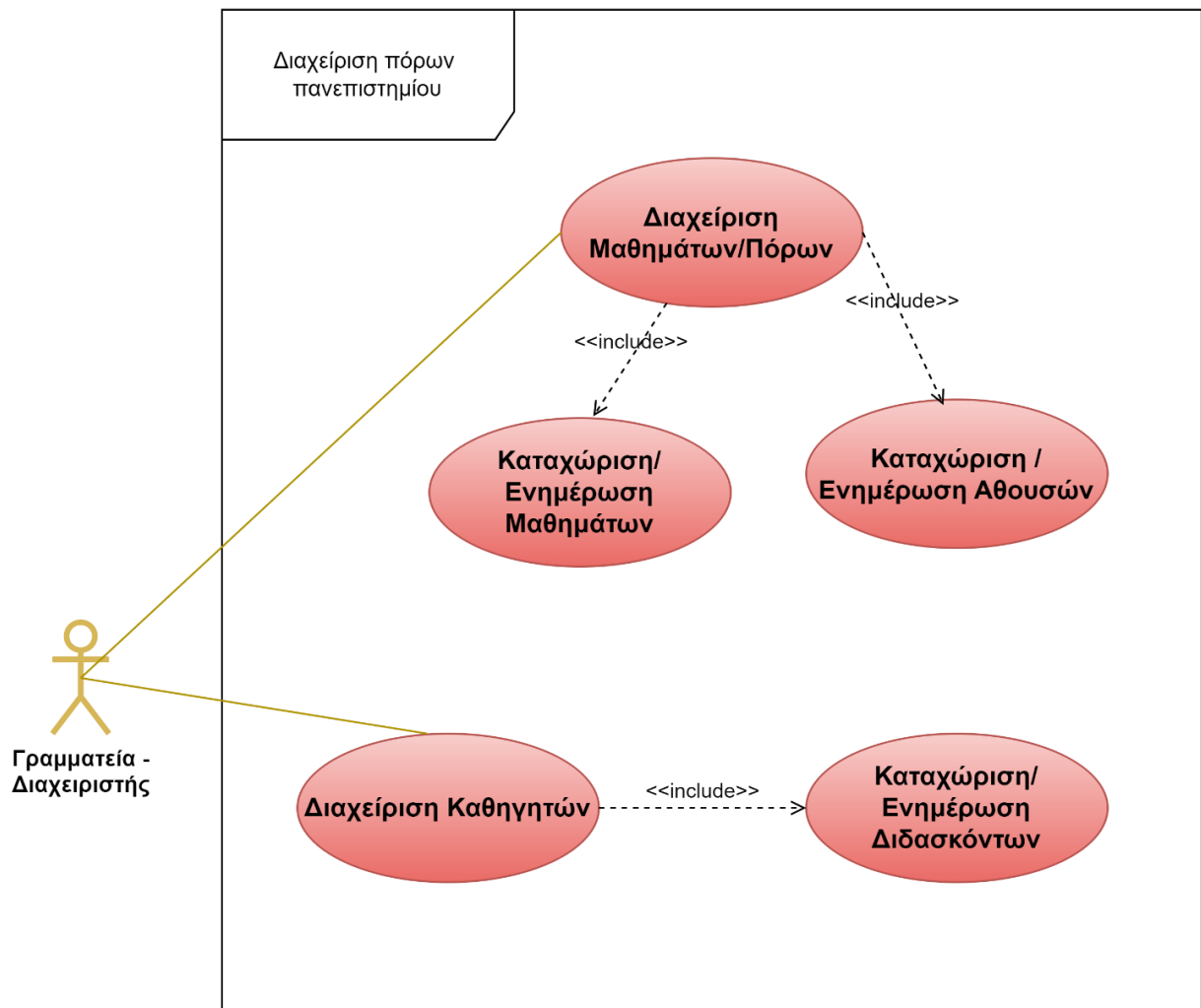
Μια λεπτομέρεια της ροής αυτής είναι ότι όταν ο οποιοσδήποτε χρήστης δημιουργήσει λογαριασμό, του αποδίδεται ο ρόλος του μαθητή ως προεπιλογή. Μετά με τη σειρά της η γραμματεία μπορεί μέσω της σελίδας διαχείρισης χρηστών, να αναζητήσει τον χρήστη και να του αποδώσει τον κατάλληλο ρόλο.



Εικόνα 14 Σενάρια διαχείρισης διαπιστευτηρίων χρήστη

3.3.3 Σενάριο Διαχείρισης πόρων πανεπιστημίου

Στο διάγραμμα χρήσης-περίπτωσης της Εικόνα 15, οι χρήστες με ρόλο Γραμματεία και Διαχειριστή αναλαμβάνουν το σύνολο των εργασιών καταχώρισης και συντήρησης των ακαδημαϊκών πόρων του τμήματος. Αρχικά το σύστημα επιτρέπει την εισαγωγή ή τροποποίηση πληροφοριών όπως ο κωδικός κάθε μαθήματος, ο αριθμός διδακτικών ωρών, το εξάμηνο και οι υπεύθυνοι καθηγητές. Παράλληλα, είναι εφικτή η διαχείριση των χώρων διδασκαλίας, όπου καταγράφονται τα χαρακτηριστικά κάθε αίθουσας και εργαστηρίου. Η σχέση «include» υποδηλώνει ότι κάθε μία από αυτές τις λειτουργίες αποτελεί ενσωματωμένο βήμα στην ευρύτερη χρήση-περίπτωση «Διαχείριση Μαθημάτων και Πόρων», διασφαλίζοντας την άρτια λειτουργία της ιστοσελίδας.



Εικόνα 15 Σενάριο διαχείρισης πόρων πανεπιστημίου

3.4 Βάση Δεδομένων

Η βάση δεδομένων είναι ένα από τα πιο βασικά κομμάτια της οποιαδήποτε ιστοσελίδας, μιας και αποθηκεύει όλα τα στοιχεία της και κατά το σχεδιασμό της αποτυπώνονται, οι σχέσεις μεταξύ των οντοτήτων του οποιουδήποτε συστήματος. Ακολουθεί η αναλυτική περιγραφή του κάθε πίνακα της βάσης καθώς και η απεικόνιση όλων των σχέσεων μεταξύ των πινάκων της βάσης.

3.4.1 Πίνακα users

Ο πίνακα users, η δομή του οποίου απεικονίζεται στον Πίνακα 2, συγκεντρώνει τα βασικά στοιχεία κάθε χρήστη όπως το username, τον κρυπτογραφημένο κωδικό πρόσβασης, το email και το τμήμα στο οποίο ανήκει ο χρήστης της ιστοσελίδας, επιτρέποντας την πιστοποίηση και την ανάθεση ρόλων στο σύστημα.

Πίνακας 2 users

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ID	INT (11)	Όχι	AUTO_INCREMENT, PRIMARY KEY
Username	VARCHAR (255)	Όχι	UNIQUE
Password	VARCHAR (255)	Όχι	
Email	VARCHAR (255)	Όχι	UNIQUE
SchoolID	INT (11)	Ναι	FK από τον πίνακα schools (ID)

Περιγραφή πεδίων:

- **Username, Email:** Διαπιστευτήρια χρήστη.
- **Password:** Κρυπτογραφημένος κωδικός.
- **SchoolID:** Τμήμα χρήστη (FK)

3.4.2 Πίνακας roles

Ο πίνακας roles περιέχει τους ορισμένους ρόλους Διαχειριστής, Γραμματεία, Καθηγητής, Φοιτητής που μπορούν να ανατεθούν σε χρήστες. Η ύπαρξη αυτού του πίνακα διευκολύνει την επέκταση ή τροποποίηση των ρόλων σε μελλοντικές εκδόσεις του συστήματος. Η δομή του παρουσιάζεται στον Πίνακας 3.

Πίνακας 3 roles

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ID	INT (11)	Όχι	AUTO_INCREMENT, PRIMARY KEY
Name	VARCHAR (255)	Όχι	

Περιγραφή πεδίων:

- **ID:** Πρωτεύον κλειδί τύπου ρόλου.
- **Name:** Όνομα ρόλου (admin, secretary, tutor, student).

3.4.3 Πίνακας userroles

Ο πίνακας userroles είναι συνδετικός πίνακας μεταξύ των πινάκων user και roles, εξασφαλίζοντας ότι κάθε χρήστης διαθέτει το σωστό σύνολο δικαιωμάτων βάσει των αναγκών της εφαρμογής. Η δομή του παρουσιάζεται στον Πίνακας 4.

Πίνακας 4 userroles

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
UserID	INT (11)	Όχι	Μέρος του PK, FK στον πίνακα users (ID)
RoleID	INT (11)	Όχι	Μέρος του PK, FK στον πίνακα roles (ID)

Περιγραφή πεδίων:

- **UserID, RoleID:** Σύνδεση χρήστη με ρόλους (συνδεδετικός πίνακας).

3.4.4 Πίνακας semesters

Για την ομαδοποίηση των μαθημάτων κατά εξάμηνο, ο πίνακας semesters καταγράφει τα διαθέσιμα εξάμηνα (π.χ. Άνοιξη 2025, Φθινόπωρο 2025), επιτρέποντας στο σύστημα να ταξινομεί και να φιλτράρει τα δεδομένα ανάλογα με την ακαδημαϊκή περίοδο. Η δομή του παρουσιάζεται στον Πίνακα 5.

Πίνακας 5 semesters

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ID	INT (11)	Όχι	AUTO_INCREMENT, PRIMARY KEY
SemesterName	VARCHAR (255)	Όχι	
Period	Enum('Χειμερινό', 'Εαρινό')	Όχι	

Περιγραφή πεδίων:

- **SemesterName:** Όνομα εξαμήνου.
- **Period:** Περίοδος εξαμήνου, χρησιμοποιείται για φιλτράρισμα και κατηγοριοποίηση.

3.4.5 Πίνακας schools

Ο πίνακας schools αναπαριστά τα επιμέρους τμήματα ή σχολές του ιδρύματος. Μεταξύ άλλων, περιλαμβάνει το όνομα του τμήματος, το οποίο χρησιμοποιείται ως ξένο κλειδί από πίνακες, όπως τους classes και rooms. Η δομή του παρουσιάζεται στον Πίνακα 6.

Πίνακας 6 schools

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ID	INT(11)	Όχι	AUTO_INCREMENT, PRIMARY KEY
Name	VARCHAR(255)	Όχι	UNIQUE

Περιγραφή πεδίων:

- **ID:** Πρωτεύον κλειδί.
- **Name:** Όνομα τμήματος.

3.4.6 Πίνακας rooms

Κάθε αίθουσα ή εργαστήριο του τμήματος αποτυπώνεται στον πίνακα rooms, όπου καταχωρούνται όνομα, χωρητικότητα, κτίριο και η συσχέτισή του με το τμήμα. Η ακριβής καταγραφή των πόρων αυτών είναι απαραίτητη για τον έλεγχο διαθεσιμότητας και τον προγραμματισμό των μαθημάτων. Η δομή του παρουσιάζεται στον Πίνακα 7.

Πίνακας 7 rooms

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ID	INT(11)	Όχι	AUTO_INCREMENT, PRIMARY KEY
Name	VARCHAR(255)	Όχι	
SchoolID	INT(11)	Ναι	FK από τον πίνακα schools (ID)
Capacity	INT(11)	Όχι	
Building	VARCHAR(255)	Όχι	
Equipment	Text	Ναι	

Περιγραφή πεδίων:

- **ID:** Πρωτεύον κλειδί.
- **Name:** Όνομα αίθουσας.
- **SchoolID:** Τμήμα στο οποίο ανήκει η αίθουσα (FK στον πίνακα schools).
- **Capacity:** Χωρητικότητα αίθουσας.
- **Building:** Κτίριο διεξαγωγής του μαθήματος.

- **Equipment:** Εξοπλισμός αίθουσας, χρήση της στήλης κυρίως για αίθουσες εργαστηρίου

3.4.7 Πίνακας notifications

Για την έγκαιρη ενημέρωση χρηστών σχετικά με αιτήματα ή αλλαγές στο ωρολόγιο πρόγραμμα, ο πίνακας notifications αποθηκεύει όλα τα μηνύματα συστήματος. Κάθε εγγραφή συνδέεται με συγκεκριμένο χρήστη, μάθημα και χρονοθυρίδα και καταγράφει αν απαιτείται ενέργεια από τον παραλήπτη. Η δομή του παρουσιάζεται στον Πίνακα 8.

Πίνακας 8 notifications

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ID	INT(11)	Όχι	AUTO_INCREMENT, PRIMARY KEY
UserID	INT(11)	Ναι	FK από τον πίνακα users (ID)
ClassID	INT(11)	Ναι	FK από τον πίνακα classes (ID)
ClassscheduleID	INT(11)	Ναι	FK από τον πίνακα classsschedules (ID)
Message	TEXT	Ναι	
Action_Required	TINYINT(1)	Ναι	
Time	TIMESTAMP	Όχι	DEFAULT τιμή: current_timestamp()
deleted_at	TIMESTAMP	Ναι	Soft delete

Περιγραφή πεδίων:

- **ID:** Πρωτεύον κλειδί.
- **UserID:** Παραλήπτης ειδοποίησης (FK στον πίνακα *users*).
- **ClassID:** Σχετικό μάθημα (FK στον πίνακα *classes*).
- **ClassscheduleID:** Σχετική χρονοθυρίδα (FK στον πίνακα *classsschedules*).
- **Message:** Κείμενο της ειδοποίησης.
- **Action_Required:** Σημαία εάν απαιτείται ενέργεια.
- **Time:** Χρονική στιγμή που έγινε η καταχώριση.
- **deleted_at:** Soft delete. Με φιλτράρισμα για τιμή null, δεν εμφανίζονται οι ειδοποιήσεις που σβήστηκαν από τον χρήστη.

3.4.8 Πίνακας classsschedules

Ο πίνακας classsschedules που απεικονίζεται στον Πίνακα 9, καταγράφει την ακριβή τοποθέτηση κάθε μαθήματος στο ωρολόγιο πρόγραμμα. Για κάθε συνεδρία θεωρίας ή εργαστηρίου αποθηκεύονται η αίθουσα, η ημερομηνία και οι ώρες έναρξης/λήξης, καθώς και η τρέχουσα κατάσταση (ενεργή ή διαγραμμένη εγγραφή). Με αυτόν τον τρόπο υποστηρίζεται η απευθείας σύνδεση μεταξύ της χρονοθυρίδας του μαθήματος και της αίθουσας.

Πίνακας 9 classsschedules

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ID	INT(11)	Όχι	AUTO_INCREMENT, PRIMARY KEY
ClassID	INT(11)	Ναι	FK από τον πίνακα classes(ID)
StartTime	TIME	Όχι	
EndTime	TIME	Όχι	
ScheduleDate	DATE	Όχι	
SessionType	ENUM('Theory','Lab')	Όχι	
Status	TINYINT(1)	Όχι	DEFAULT 1
deleted_at	TIMESTAMP	Ναι	
TimetableID	INT(11)	Όχι	FK από τον πίνακα timetables(ID)

Περιγραφή πεδίων:

- **ID:** Πρωτεύον κλειδί, AUTO_INCREMENT.
- **ClassID:** Αναφορά στο μάθημα (FK από τον πίνακα classes).
- **StartTime, EndTime:** Ώρες έναρξης και λήξης.
- **ScheduleDate:** Ημερομηνία διεξαγωγής.
- **SessionType:** Τύπος συνεδρίας (θεωρία ή εργαστήριο).
- **Status:** Κατάσταση εγγραφής (ενεργή=1).
- **deleted_at:** Σήμανση για soft delete.
- **TimetableID:** Για διασύνδεση με τον πίνακα timetables και αντιστοίχιση συνεδρίας με το σωστό πρόγραμμα

3.4.9 Πίνακας emergencyclassrequests

Ο πίνακας emergencyclassrequests χρησιμοποιείται για την καταχώριση αιτημάτων επαναληπτικών συνεδριών (αναπληρώσεων) από τους καθηγητές. Κάθε αίτηση περιλαμβάνει πληροφορίες για την αρχική χρονοθυρίδα, την προτεινόμενη νέα ημερομηνία και ώρα και την κατάσταση επεξεργασίας του αιτήματος (εκκρεμεί, εγκρίθηκε, απορρίφθηκε). Η δομή του παρουσιάζεται στον Πίνακα 10.

Πίνακας 10 emergencyclassrequest

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ID	INT(11)	Όχι	AUTO_INCREMENT , PRIMARY KEY
UserID	INT(11)	Ναι	FK από τον πίνακα users(ID)
ClassID	INT(11)	Ναι	FK από τον πίνακα classes(ID)
Comments	TEXT	Ναι	
ClassscheduleID	INT(11)	Όχι	FK από τον πίνακα classsschedules(ID)
Status	ENUM('Pending','Confirmed','Denied','Expired')	Όχι	DEFAULT 'Pending'
newStartTime	TIME	Όχι	
newEndTime	TIME	Όχι	
newDate	DATE	Όχι	
deleted_at	TIMESTAMP	Ναι	

Περιγραφή πεδίων:

- **ID:** Πρωτεύον κλειδί.
- **UserID:** Αναφορά στο χρήστη (FK στον πίνακα users), υποβάλλων.
- **ClassID:** Μάθημα για την επαναληπτική συνεδρία (FK στον πίνακα classes).
- **Comments:** Σχόλια χρήστη.

- **ClassscheduleID:** Αναφορά στην αρχική χρονοθυρίδα (FK στον πίνακα *classsschedules*).
- **Status:** Κατάσταση της αίτησης.
- **newStartTime, newEndTime, newDate:** Προτεινόμενες νέες ώρες και ημερομηνία.
- **deleted_at:** Χρήση για soft delete δυνατότητα.

3.4.10 Πίνακας classes

Ο πίνακας classes λειτουργεί ως κεντρικός κατάλογος των μαθημάτων που προσφέρονται από το τμήμα. Κάθε εγγραφή αποτυπώνει βασικές πληροφορίες για ένα μάθημα όπως όνομα, απαιτούμενες διδακτικές και εργαστηριακές ώρες, αν περιέχει θεωρία, εργαστήριο ή και τα δύο και το εξάμηνο στο οποίο προσφέρεται. Επιπλέον, με τα ξένα κλειδιά *SchoolID* και *SemesterID* διασφαλίζεται η σύνδεση με το κατάλληλο τμήμα και εξάμηνο. Η δομή του απεικονίζεται στον Πίνακα 11.

Πίνακας 11 classes

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ID	INT(11)	Όχι	AUTO_INCREMENT, PRIMARY KEY
SchoolID	INT(11)	Ναι	FK από τον πίνακα schools (ID)
Name	VARCHAR(255)	Όχι	
Description	TEXT	Ναι	
SemesterID	INT(11)	Ναι	FK από τον πίνακα semesters (ID)
TotalTheoryHours	INT(11)	Όχι	
TotalLabHours	INT(11)	Όχι	
Requirements	TEXT	Ναι	
ClassType	ENUM('Theory','Lab','Both')	Όχι	
CourseType	Enum('κ','ε')	Όχι	K = κορμού / E = επιλογής
DirectionId	INT(11)	Ναι	FK από τον πίνακα directions (id)

Περιγραφή πεδίων:

- **ID:** Πρωτεύον κλειδί του πίνακα, αυξάνεται αυτόματα (AUTO_INCREMENT).
- **SchoolID:** Αναφορά στο τμήμα (FK στον πίνακα *schools*), υποδεικνύει σε ποιο τμήμα ανήκει το μάθημα.
- **Name:** Όνομα του μαθήματος.
- **Description:** Σύντομη περιγραφή του μαθήματος.
- **SemesterID:** Αναφορά στο εξάμηνο (FK στον πίνακα *semesters*), υποδεικνύει σε ποιο

εξάμηνο διδάσκεται.

- **TotalTheoryHours:** Συνολικές ώρες θεωρίας.
- **TotalLabHours:** Συνολικές ώρες εργαστηρίου.
- **Requirements:** Επιπλέον απαιτήσεις ή υποσημειώσεις.
- **ClassType:** Τύπος μαθήματος (θεωρία, εργαστήριο ή και τα δύο).
- **CourseType:** Χρήση για διαχωρισμό μαθημάτων επιλογής και κορμού
- **DirectionId:** Χρήση για τη διασύνδεση των μαθημάτων με των πίνακα των κατευθύνσεων(FK).

3.4.11 Πίνακας studentclasses

Ο πίνακας studentclasses συνδέει τους φοιτητές με τα μαθήματα που έχουν δηλώσει, καταγράφοντας την ημερομηνία εγγραφής. Αυτή η σχέση many-to-many επιτρέπει τη δημιουργία εξατομικευμένου ωρολογίου προγράμματος για κάθε φοιτητή και απεικονίζεται στον Πίνακα 12.

Πίνακας 12 studentclasses

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
StudentID	INT(11)	Όχι	PART of PK, FK στον πίνακα users(ID)
ClassID	INT(11)	Όχι	PART of PK, FK στον πίνακα classes(ID)
EnrollmentDate	DATE	Όχι	DEFAULT CURDATE()
deleted_at	TIMESTAMP	Ναι	

Περιγραφή πεδίων:

- **StudentID:** Αναφορά σε χρήστη με ρόλο φοιτητή (FK).
- **ClassID:** Αναφορά σε μάθημα (FK).
- **EnrollmentDate:** Ημερομηνία εγγραφής στο μάθημα.
- **deleted_at:** Soft delete. Αν έχουμε null δεν διαγράφηκε

3.4.12 Πίνακας tutoravailability

Οι διαθεσιμότητες των καθηγητών αποθηκεύονται στον πίνακα *tutoravailability*, όπου κάθε γραμμή καθορίζει ημέρα, ώρες, μάθημα και πιθανές πρόσθετες παρατηρήσεις. Χρησιμοποιείται για τον αυτοματοποιημένο έλεγχο σύγκρουσης κατά την υποβολή αιτήσεων χρονοθυρίδων. Η δομή του παρουσιάζεται στον Πίνακα 13.

Πίνακας 13 tutoravailability

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ID	INT(11)	Όχι	AUTO_INCREMENT, PRIMARY KEY
UserID	INT(11)	Όχι	FK από τον πίνακα users (ID)
ClassID	INT(11)	Όχι	FK από τον πίνακα classes (ID)
TimetableID	INT (11)	Όχι	FK από τον πίνακα timetables (ID)
DayOfWeek	ENUM('Monday',..., 'Friday')	Ναι	
StartTime	TIME	Ναι	
EndTime	TIME	Ναι	
Comments	TEXT	Ναι	
type	INT(11)	Όχι	
deleted_at	TIMESTAMP	Ναι	
Status	enum('Pending', 'Confirmed', 'Denied', 'NotSet')	Ναι	

Περιγραφή πεδίων:

- **UserID:** Αναφορά σε χρήστη-καθηγητή (FK).
- **ClassID:** Αναφορά σε μάθημα (FK).
- **TimetableID:** Σύνδεση με τον πίνακα προγραμμάτων της σχολής για αντιστοίχιση με τη σωστή περίοδο διδασκαλίας
- **DayOfWeek, StartTime, EndTime:** Διαθεσιμότητες.
- **Comments:** Τυχόν σχόλια που θέλει να προσθέσει ο καθηγητής για τη δηλωμένη χρονοθυρίδα.
- **type:** Σημαία για υπόδειξη 0: θεωρία, 1: Εργαστήριο, χρήση της πληροφορίας για φιλτράρισμα και ελέγχους
- **deleted_at:** soft delete δυνατότητα για αρχειοθέτηση.
- **Status:** Χρήση της στήλης για ανάλυση κατάστασης αιτημάτων καθηγητών και φιλτράρισμα.

3.4.13 Πίνακας **tutorclasses**

Ο πίνακας **tutorclasses** απεικονίζει τη σχέση διδασκαλίας μεταξύ καθηγητών και μαθημάτων, επιτρέποντας σε κάθε μάθημα να έχει έναν ή περισσότερους υπεύθυνους διδάσκοντες. Η δομή του παρουσιάζεται στον Πίνακα 14.

Πίνακας 14 *tutorclasses*

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ClassRoomID	INT(11)	Όχι	PART of PK, FK από τον πίνακα classes (ID)
UserID	INT(11)	Όχι	PART of PK, FK από τον πίνακα users (ID)
Status	ENUM('Pending','NotSet','Confirmed','Rejected')	Ναι	DEFAULT 'NotSet'
deleted_at	TIMESTAMP	Ναι	

Περιγραφή πεδίων:

- **ClassRoomID**: Αναφορά σε session (FK).
- **UserID**: Αναφορά σε χρήστη-καθηγητή (FK).
- **Status**: Enumerator για την κατηγοριοποίηση των διαθεσιμοτήτων του καθηγητή.
- **deleted_at**: soft delete για αρχικοποίηση.

3.4.14 Πίνακας **classschedule_rooms**

Ο πίνακας **classschedule_rooms** είναι ένας συνδετικός πίνακας μεταξύ των πινάκων **classschedule** και **rooms** και ο λόγος που δημιουργήθηκε είναι για να ικανοποιηθεί η ανάγκη για την παρουσίαση πολλαπλών αιθουσών στο τελικό πρόγραμμα, στην περίπτωση που ένα μάθημα λαμβάνει χώρα σε παραπάνω από μία αίθουσα. Η δομή του παρουσιάζεται στον Πίνακα 15.

Πίνακας 15 *classschedule_rooms*

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ClassscheduleID	INT (11)	Όχι	FK από τον πίνακα classschedules

RoomID	INT (11)	Όχι	FK από τον πίνακα rooms
---------------	----------	-----	-------------------------

Περιγραφή πεδίων:

- **ID:** Μοναδικό αναγνωριστικό εγγραφής. PRIMARY KEY
- **SchoolID:** Πεδίο τύπου int(11). (FK)
- **Name:** Το όνομα του προγράμματος
- **StartDate:** Αρχή προγράμματος.
- **EndDate:** Τέλος προγράμματος.
- **Type:** Πεδίο τύπου Enum('Teaching','Exam')
- **Period:** Πεδίο τύπου enum('Χειμερινό','Εαρινό','Επαναληπτική')
- **Year:** Πεδίο τύπου year(4).

3.4.15 Πίνακας timetables

Ο πίνακας timetables χρησιμοποιείται για τη δημιουργία προγραμμάτων, την διασύνδεση των μαθημάτων και των διαθεσιμοτήτων των καθηγητών ώστε να εμφανιστούν στο πρόγραμμα ανάλογα με τις επιλογές του χρήστη. Το κάθε πανεπιστήμιο μπορεί να δηλώσει τα δικά του διαστήματα για το κάθε πρόγραμμα εξαμήνου ή εξεταστικής περιόδου. Τα δεδομένα που αποθηκεύονται σε αυτόν παρουσιάζονται στον Πίνακα 16.

Πίνακας 16 timetables

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ID	INT (11)	Όχι	PRIMARY KEY
SchoolID	INT (11)	Όχι	FK από τον πίνακα schools
Name	varchar(255)	Όχι	
StartDate	Date	Όχι	Αρχή προγράμματος
EndDate	Date	Όχι	Τέλος προγράμματος
Type	enum('Teaching','Exam')	Όχι	Προσδιορισμός προγράμματος
Period	enum('Χειμερινό','Εαρινό','Επαναληπτική')	Όχι	Περίοδος προγράμματος
Year	year(4)	Όχι	

Περιγραφή πεδίων:

- **ID:** Μοναδικό αναγνωριστικό εγγραφής. PRIMARY KEY
- **SchoolID:** Πεδίο τύπου int(11). (FK)
- **Name:** Το όνομα του προγράμματος
- **StartDate:** Αρχή προγράμματος.
- **EndDate:** Τέλος προγράμματος.
- **Type:** Πεδίο τύπου Enum('Teaching','Exam')
- **Period:** Πεδίο τύπου enum('Χειμερινό','Εαρινό','Επαναληπτική')
- **Year:** Πεδίο τύπου year(4).

3.4.16 Πίνακας classschedule_users

Ο πίνακας classschedule_users είναι ένας συνδετικός πίνακας μεταξύ των πινάκων classschedule και users και ο λόγος που δημιουργήθηκε είναι για να ικανοποιηθεί η ανάγκη για τη διασύνδεση πολλαπλών καθηγητών με ένα μάθημα, για να . Η δομή του παρουσιάζεται στον

Πίνακας 17.

Πίνακας 17 classschedule_users

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
ClassscheduleID	INT (11)	Όχι	FK από τον πίνακα classschedule(ID)
UserID	INT (11)	Όχι	FK από τον πίνακα users(ID)

Περιγραφή πεδίων:

- **ClassscheduleID,UserID:** Συνδυαστικό πρωτεύον κλειδί για τον συνδετικό πίνακα.

3.4.17 Πίνακας timetable_classes

Ο πίνακας timetable_classes είναι ένας συνδετικός πίνακας μεταξύ των πινάκων timetables και classes και ο λόγος που δημιουργήθηκε είναι για να ικανοποιηθεί η ανάγκη για τη διασύνδεση των μαθημάτων με το πρόγραμμα μαθημάτων. Ένα μάθημα δύναται να χρειάζεται πολλαπλές χρονοθυρίδες για να εκπληρωθούν οι απαραίτητες ώρες διδασκαλίας του, οπότε κρίθηκε αναγκαστική η δημιουργία του συνδετικού πίνακα . Η δομή του παρουσιάζεται στον Πίνακας 18.

Πίνακας 18 timetable_classes

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
TimetableID	INT (11)	Όχι	FK από τον πίνακα timetables(ID)
ClassID	INT (11)	Όχι	FK από τον πίνακα classes(ID)

Περιγραφή πεδίων:

- **ClassID, TimetableID:** Συνδυαστικό πρωτεύον κλειδί για τον συνδετικό πίνακα.

3.4.18 Πίνακας userschools

Ο πίνακας userschools είναι ένας συνδετικός πίνακας μεταξύ των πινάκων users και schools και ο λόγος που δημιουργήθηκε είναι για να ικανοποιηθεί η ανάγκη για τη διασύνδεση των καθηγητών με τα πανεπιστήμια. Ένας καθηγητής, παρόλο που έχει ένα κύριο πανεπιστήμιο στο οποίο διδάσκει, μπορεί να διδάσκει με συνδιδασκαλία ή χωρίς σε παραπάνω από ένα τμήματα οπότε η γραμματοεία θα πρέπει να έχει τη δυνατότητα να τον καταχωρήσει σε όλα τα τμήματα. Η δομή του παρουσιάζεται στον Πίνακας 19.

Πίνακας 19 userschools

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
UserID	INT (11)	Όχι	FK από τον πίνακα users(ID)
SchoolID	INT (11)	Όχι	FK από τον πίνακα schools(ID)
Main	tinyint(1)	Ναι	

Περιγραφή πεδίων:

- **UserID, SchoolID:** Συνδυαστικό πρωτεύον κλειδί για τον συνδετικό πίνακα.
- **Main:** Σημαία που υποδηλώνει το κύριο πανεπιστήμιο του διδάσκοντα

3.4.19 Πίνακας tutoravailability_rooms

Ο πίνακας tutoravailability_rooms είναι ένας συνδετικός πίνακας μεταξύ των πινάκων tutoravailability και rooms και ο λόγος που δημιουργήθηκε είναι για να ικανοποιηθεί η ανάγκη για τη διασύνδεση των αιτημάτων των καθηγητών για πολλαπλές αίθουσες διδασκαλίας για το

μάθημά τους. Η δομή του παρουσιάζεται στον Πίνακα 20.

Πίνακας 20 *tutoravailability_rooms*

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
UserID	INT (11)	Όχι	FK από τον πίνακα users(ID)
SchoolID	INT (11)	Όχι	FK από τον πίνακα schools(ID)
Main	tinyint(1)	Ναι	

Περιγραφή πεδίων:

- **UserID, SchoolID**: Συνδυαστικό πρωτεύον κλειδί για τον συνδετικό πίνακα.
- **Main**: Σημαία που υποδηλώνει το κύριο πανεπιστήμιο του διδάσκοντα

3.4.20 Πίνακας *emergencyclassrequests_rooms*

Ο πίνακας *emergencyclassrequests_rooms* είναι ένας συνδετικός πίνακας μεταξύ των πινάκων *emergencyclassrequests* και *rooms* και ο λόγος που δημιουργήθηκε είναι για να ικανοποιηθεί η ανάγκη για τη διασύνδεση των αιτημάτων των δηλώσεων των αναπληρώσεων με πολλαπλές αίθουσες διδασκαλίας για το μάθημά τους. Η δομή του παρουσιάζεται στον

Πίνακα 21.

Πίνακας 21 *emergencyclassrequests_rooms*

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
EmergencyClassRequestID	INT (11)	Όχι	FK από τον πίνακα <i>emergencyclassrequests</i> (ID)
RoomID	INT (11)	Όχι	FK από τον πίνακα <i>rooms</i> (ID)

Περιγραφή πεδίων:

- **EmergencyClassRequestID, RoomID**: Συνδυαστικό πρωτεύον κλειδί για τον συνδετικό πίνακα.

3.4.21 Πίνακας *emergencyclassrequests_timetables*

Ο πίνακας *emergencyclassrequests_timetables* είναι ένας συνδετικός πίνακας μεταξύ των πινάκων *emergencyclassrequests* και *timetables* και ο λόγος που δημιουργήθηκε είναι για να ικανοποιηθεί η ανάγκη για τη διασύνδεση των αιτημάτων των δηλώσεων των αναπληρώσεων με

το συγκεκριμένο πρόγραμμα που ανήκουν για λόγους ιστορικότητας αλλά και φιλτραρίσματος στη σελίδα που εμφανίζεται το πρόγραμμα της σχολής. Η δομή του παρουσιάζεται στον Πίνακα 22.

Πίνακας 22 *emergencyclassrequests_timetables*

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
EmergencyClassRequestID	INT (11)	Όχι	FK από τον πίνακα emergencyclassrequests(ID)
TimetableID	INT (11)	Όχι	FK από τον πίνακα timetables(ID)

Περιγραφή πεδίων:

- **EmergencyClassRequestID, TimetableID** : Συνδυαστικό πρωτεύον κλειδί για τον συνδετικό πίνακα.

3.4.23 Πίνακας directions

Ο πίνακας directions δημιουργήθηκε για να ικανοποιήσει την ανάγκη για καταχώρηση των εξειδικευμένων κατευθύνσεων που μπορεί να υπάρχουν σε κάποια τμήματα. Είναι συνδεδεμένος με τον πίνακα schools με περιορισμό ξένου κλειδιού με πληθικότητα 1-M. Η δομή του παρουσιάζεται στον Πίνακα 23.

Πίνακας 23 Πίνακας directions

Όνομα πεδίου	Τύπος	Null	Πρόσθετα
id	INT (11)	Όχι	Primary Key
schoolid	INT (11)	Ναι	Foreign Key από τον πίνακα schools(id)
name	varchar(255)	Όχι	Το όνομα της κατεύθυνσης

Περιγραφή πεδίων:

- **schoolid**: Η σύνδεση με τον πίνακα schools με ξένο κλειδί (FK) στο πεδίο id.
- **name**: Το όνομα της κατεύθυνσης. Το πεδίο δεν μπορεί να μείνει κενό.

Η βάση δεδομένων ακολουθεί 3NF Κανονικοποίηση:

- **Πρώτη Κανονική Μορφή (1NF)**. Όλα τα γνωρίσματα είναι ατομικά, δεν εμφανίζονται επαναλαμβανόμενες ομάδες και οι τιμές των χαρακτηριστικών αποθηκεύονται σε κλιμακωτούς τύπους (π.χ. time, date, enum, varchar, int). Χαρακτηριστικά παραδείγματα: οι ώρες (starttime, endtime) και οι ημερομηνίες (scheduledate) είναι σε ξεχωριστές στήλες, ενώ οι ποιοτικές κατηγορίες αποτυπώνονται με **ENUM** (π.χ. sessiontype, classtype).

- [illegible]

54

3.5 Τεχνικές εξασφάλισης ασφάλειας

Η ασφάλεια είναι ένα από τους πιο σημαντικούς παράγοντες που πρέπει να λαμβάνει υπόψιν του ο προγραμματιστής κατά την υλοποίηση του οποιαδήποτε προγραμματιστικού έργου και πρέπει να την παρέχει στους χρήστες του και τα δεδομένα τους. Στις υποενότητες που ακολουθούν παρουσιάζονται οι τεχνικές που χρησιμοποιούνται στην εφαρμογή για την διασφάλιση της ασφάλειας των προσωπικών δεδομένων των χρηστών που αποθηκεύονται στη βάση δεδομένων καθώς και η χρήση HTTPS(HTTP-secure) για τη σύνδεση στην ιστοσελίδα.

3.5.1 Χρήση ενιαίου αρχείου db_conn.php

Ένα από τα πιο κρίσιμα αρχεία του πληροφοριακού συστήματος είναι εκείνο που διαχειρίζεται τη σύνδεση με τη βάση δεδομένων. Στο **db_conn.php** ορίζονται οι βασικές παράμετροι σύνδεσης, όπως το όνομα της βάσης δεδομένων, τα στοιχεία πρόσβασης του χρήστη και το socket επικοινωνίας. Με βάση αυτές τις πληροφορίες δημιουργείται το αντικείμενο PDO, μέσω του οποίου πραγματοποιούνται όλες οι συναλλαγές με τη βάση. Επιπλέον, στο στάδιο της αρχικοποίησης της σύνδεσης ορίζονται οι απαραίτητες ρυθμίσεις ώστε το σύστημα να χειρίζεται σωστά ελληνικούς χαρακτήρες, εξασφαλίζοντας έτσι την ακεραιότητα των δεδομένων. Το αρχείο αυτό περιλαμβάνεται όπου χρειαστεί να εξάγουμε ή να εισάγουμε δεδομένα στη βάση δεδομένων και η δομή του παρουσιάζεται στην Εικόνα 17.

```
<?php

$servername = [REDACTED];
$user = [REDACTED];
$password = [REDACTED];
$dbName = [REDACTED];

error_reporting(E_ALL);
ini_set("display_errors", 1);

try {
    $conn = new PDO("mysql:host=$servername;dbname={$dbName};charset=utf8", $user, $password);
    $conn->query('set character_set_client=utf8');
    $conn->query('set character_set_connection=utf8');
    $conn->query('set character_set_results=utf8');
    $conn->query('set character_set_server=utf8');
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    echo "Connection failed: " . $e->getMessage();
}
```

Εικόνα 17 db_conn.php

3.5.2 Ασφάλεια κωδικών πρόσβασης

Οι κωδικοί πρόσβασης των χρηστών δεν αποθηκεύονται σε απλή μορφή κειμένου, αλλά υφίστανται ισχυρή κρυπτογράφηση με τον αλγόριθμο **bcrypt**[21] που βασίζεται στον Blowfish. Κατά τη δημιουργία νέου λογαριασμού, ο κωδικός μετατρέπεται σε κρυπτογράφημα μέσω της συνάρτησης **password_hash()**, ενώ κατά τη διαδικασία σύνδεσης πραγματοποιείται έλεγχος ταυτοποίησης με τη συνάρτηση **password_verify()**. Η επιλογή του **bcrypt** εξασφαλίζει υψηλό επίπεδο προστασίας απέναντι σε επιθέσεις brute force και λεξικών, καθώς είναι ειδικά σχεδιασμένος για αποθήκευση κωδικών.

Επιπλέον, κατά τη διαδικασία αλλαγής κωδικού χρησιμοποιείται εκ νέου η ίδια διαδικασία κρυπτογράφησης του νέου κωδικού που θα αποθηκευτεί στη βάση δεδομένων.

3.5.3 Ασφάλεια εισαγόμενων δεδομένων

Η επικύρωση και “καθαρισμός” των δεδομένων που εισάγονται από τον χρήστη αποτελεί απαραίτητη πρακτική για την αποτροπή επιθέσεων. Στην εφαρμογή γίνεται εκτενής χρήση συναρτήσεων όπως **filter_var()** και **htmlspecialchars()** όπως φαίνεται στις Εικόνα 18 και Εικόνα 19, οι οποίες απομακρύνουν ανεπιθύμητους χαρακτήρες και εξασφαλίζουν ότι τα δεδομένα πληρούν τις απαιτούμενες προδιαγραφές. Με τον τρόπο αυτό μειώνεται σημαντικά ο κίνδυνος επιθέσεων **Cross-Site Scripting (XSS)**[22] ή εισαγωγής μη έγκυρων τιμών στις φόρμες επικοινωνίας.

```
/* --- AJAX: Edit course --- */
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['editCourseId'])) {
    header(header: 'Content-Type: application/json; charset=utf-8');

    $courseId = filter_var(value: $_POST['editCourseId'] ?? 0, filter: FILTER_VALIDATE_INT);
    $name      = mb_substr(string: trim(string: $_POST['editCourseName'] ?? ''), start: 0, length: 255);
    $desc      = mb_substr(string: trim(string: $_POST['editCourseDescription'] ?? ''), start: 0, length: 1000);
    $semester  = filter_var(value: $_POST['editCourseSemester'] ?? 0, filter: FILTER_VALIDATE_INT);
    $theory    = filter_var(value: $_POST['editCourseTheoryHours'] ?? 0, filter: FILTER_VALIDATE_INT);
    $lab       = filter_var(value: $_POST['editCourseLabHours'] ?? 0, filter: FILTER_VALIDATE_INT);
    $reqs      = mb_substr(string: trim(string: $_POST['editCourseRequirements'] ?? ''), start: 0, length: 1000);
    $type      = mb_substr(string: trim(string: $_POST['editCourseType'] ?? ''), start: 0, length: 32);
    $direction = (isset($_POST['editCourseDirection']) && $_POST['editCourseDirection'] !== '')
        ? (int)$_POST['editCourseDirection'] : null;

    // Business rules
```

Εικόνα 18 filter_var and other data integrity helpers

```
$rowNumber = 1;
if (!empty($classrooms)) {
    foreach ($classrooms as $room) {
        echo '<tr data-classroom-id="' . $room['id'] . '">';
        echo '<td>' . $rowNumber++ . '</td>';
        echo '<td class="classroom-name">' . htmlspecialchars(string: $room['name'], flags: ENT_QUOTES, encoding: 'UTF-8') . '</td>';
        echo '<td class="classroom-capacity">' . htmlspecialchars(string: $room['capacity'], flags: ENT_QUOTES, encoding: 'UTF-8') . '</td>';
        echo '<td class="classroom-building">' . htmlspecialchars(string: $room['building'], flags: ENT_QUOTES, encoding: 'UTF-8') . '</td>';
        echo '<td class="classroom-equipment">' . htmlspecialchars(string: $room['equipment'] ?? '', flags: ENT_QUOTES, encoding: 'UTF-8') . '</td>';
        echo '<td>';
        <button class="btn btn-warning btn-sm edit-classroom-btn"
            data-id="' . $room['id'] . '"
            data-name="' . htmlspecialchars(string: $room['name'], flags: ENT_QUOTES, encoding: 'UTF-8') . '"
            data-capacity="' . htmlspecialchars(string: $room['capacity'], flags: ENT_QUOTES, encoding: 'UTF-8') . '"
            data-building="' . htmlspecialchars(string: $room['building'], flags: ENT_QUOTES, encoding: 'UTF-8') . '"
            data-equipment="' . htmlspecialchars(string: $room['equipment'] ?? '', flags: ENT_QUOTES, encoding: 'UTF-8') . '">
                Επεξεργασία
            </button>
            <button class="btn btn-danger btn-sm delete-classroom-btn" data-id="' . $room['id'] . '">Διαγραφή</button>
        </td>';
        echo '</tr>';
    }
}
```

Εικόνα 19 htmlspecialchars provision with UTF-8 encoding

3.5.4 Χρήση PDO και προστασία από SQL Injection

Όλες οι αλληλεπιδράσεις με τη βάση δεδομένων πραγματοποιούνται μέσω της βιβλιοθήκης **PDO**, η οποία προσφέρει ασφάλεια και ευελιξία στη διαχείριση των ερωτημάτων. Στις συναρτήσεις εισαγωγής, τροποποίησης και ανάκτησης δεδομένων χρησιμοποιούνται προετοιμασμένες εντολές (prepared statements) με τη μέθοδο **bindParam()** και χωρίς τη χρήση ανώνυμων παραμέτρων όπως παρουσιάζεται στην Εικόνα 20. Η τεχνική αυτή αποτρέπει την εισαγωγή κακόβουλου SQL κώδικα στα πεδία εισόδου και προστατεύει το σύστημα από επιθέσεις SQL injection. Ο διαχωρισμός της δομής του ερωτήματος από τα δεδομένα εγγυάται ότι τα τελευταία δεν εκτελούνται ως κώδικας, αλλά αντιμετωπίζονται αποκλειστικά ως τιμές.

```
// AJAX: Edit classroom
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['editClassroomId'])) {
    $classroomId = intval(value: $_POST['editClassroomId']);
    $name = trim(string: $_POST['editClassroomName'] ?? '');
    $capacity = intval(value: $_POST['editClassroomCapacity'] ?? 0);
    $building = trim(string: $_POST['editClassroomBuilding'] ?? '');
    $equipment = trim(string: $_POST['editClassroomEquipment'] ?? ''); // Add this

    if ($name !== '' && $capacity > 0 && $building !== '') {
        $stmt = $conn->prepare(query: "UPDATE rooms SET name = :name, capacity = :capacity, building = :building, equipment = :equipment WHERE id = :id");
        $stmt->bindParam(param: ':name', var: &$name, type: PDO::PARAM_STR);
        $stmt->bindParam(param: ':capacity', var: &$capacity, type: PDO::PARAM_INT);
        $stmt->bindParam(param: ':building', var: &$building, type: PDO::PARAM_STR);
        $stmt->bindParam(param: ':equipment', var: &$equipment, type: PDO::PARAM_STR); // Add this
        $stmt->bindParam(param: ':id', var: &$classroomId, type: PDO::PARAM_INT);
        $stmt->execute();
        echo json_encode(value: ['success' => true]);
    } else {
        echo json_encode(value: ['success' => false, 'error' => "Όλα τα πεδία είναι υποχρεωτικά και η χωρητικότητα πρέπει να είναι θετικός αριθμός."]);
    }
    exit();
}
```

Εικόνα 20 Prepare-bindparam-execute PDO pattern

3.5.5 Εξασφάλιση χρήσης HTTPS για ασφάλεια της σελίδας

Η ασφαλής πρόσβαση σε διαδικτυακές εφαρμογές προϋποθέτει τη χρήση του πρωτοκόλλου HTTPS, το οποίο εξασφαλίζει την κρυπτογράφηση της επικοινωνίας μεταξύ χρήστη και διακομιστή. Για τον σκοπό αυτό δημιουργήθηκε ειδικό αρχείο με όνομα «enforce_https.php», το οποίο ενσωματώνεται σε όλα τα επιμέρους αρχεία του πληροφοριακού συστήματος. Η λειτουργία του είναι απλή: σε περίπτωση που ο χρήστης επιχειρήσει να συνδεθεί μέσω του μη ασφαλούς πρωτοκόλλου HTTP, η εφαρμογή τον ανακατευθύνει αυτόματα στη σωστή διεύθυνση με χρήση HTTPS. Με αυτόν τον τρόπο διασφαλίζεται ότι όλες οι ανταλλαγές δεδομένων πραγματοποιούνται σε περιβάλλον κρυπτογράφησης, ενισχύοντας την εμπιστευτικότητα και την προστασία των προσωπικών δεδομένων. Η δομή του αρχείου παρουσιάζεται στην Εικόνα 21.

```

<?php
function force_HTTPS()
{
if (empty($_SERVER['HTTPS']) || $_SERVER['HTTPS'] === "off") {
    {
        $redirect = 'https://' . $_SERVER['HTTP_HOST'] . $_SERVER['REQUEST_URI'];
        header('HTTP/1.1 301 Moved Permanently');
        header('Location: ' . $redirect);
        exit();
    }
}
}
?>

```

Εικόνα 21 enforce_https.php

3.6 Συμπεράσματα

Στο παρόν κεφάλαιο αποτυπώθηκε με συστηματικό τρόπο η επιχειρησιακή λογική του συστήματος και πώς αυτή μεταφράζεται σε συγκεκριμένη αρχιτεκτονική και κανόνες λειτουργίας. Πρώτον, τεκμηριώθηκαν οι **λειτουργικές απαιτήσεις** και οι τέσσερις ρόλοι πρόσβασης αυτοί του Διαχειριστή, της Γραμματείας, του Καθηγητή και του Φοιτητή, με σαφή οριοθέτηση των δικαιωμάτων τους. Η γραμματεία αναλαμβάνει την κεντρική διαχείριση μαθημάτων, αιθουσών και χρονοθυρίδων, οι καθηγητές υποβάλλουν αιτήματα (τακτικά και έκτακτα) και ελέγχουν διαθεσιμότητες, ενώ οι φοιτητές καταρτίζουν την εξατομικευμένη προβολή τους και εξάγουν το πρόγραμμά τους σε PDF. Ο **μηχανισμός προληπτικών ελέγχων** αποτελεί βασική σχεδιαστική επιλογή: κατά την καταχώριση εφαρμόζονται κανόνες για τη χωρητικότητα και τον εξοπλισμό αιθουσών, την αποκλειστικότητα των διδασκόντων και την αποφυγή επικαλύψεων μαθημάτων εξαμήνου, ώστε να αποτρέπονται τα εσφαλμένα αιτήματα στη ρίζα τους και να μειώνεται δραστικά ο κύκλος «σφάλματος–διόρθωσης».

Δεύτερον, τα **διαγράμματα χρήσης-περίπτωσης** ανέδειξαν τον τρόπο με τον οποίο υλοποιούνται οι κύριες ροές: δήλωση χρονοθυρίδας από τον καθηγητή και επεξεργασία/έγκριση από τη γραμματεία, διαχείριση διαπιστευτηρίων και ρόλων, καθώς και ολοκληρωμένη διαχείριση ακαδημαϊκών πόρων (μαθήματα, αίθουσες/εργαστήρια, εξοπλισμός, ωράρια). Η ενσωμάτωση **μηχανισμού αιτημάτων–εγκρίσεων** για αναπληρώσεις, σε συνδυασμό με φίλτρα προβολής (ανά εξάμηνο, αίθουσα, διδάσκοντα), επιτρέπει ορατότητα και διαφάνεια στην καθημερινή λειτουργία και επιταχύνει τη λήψη αποφάσεων από τη γραμματεία.

Τρίτον, η **βάση δεδομένων** σχεδιάστηκε ώστε να αντανakλά πιστά τις παραπάνω απαιτήσεις, ακολουθώντας τις αρχές κανονικοποίησης 3NF για την επίτευξη της συνέπειας και την αποφυγή πλεονασμών δεδομένων. Οι βασικές οντότητες (users, roles, classes, rooms, semesters, timetables) πλαισιώνονται από **συνδυαστικούς πίνακες** (π.χ. userroles, timetable_classes, classschedule_users, classschedule_rooms) που επιτρέπουν ευέλικτες συσχετίσεις πολλών-προς-πολλά (πολλοί διδάσκοντες ανά μάθημα, πολλαπλές αίθουσες ανά συνεδρία). Η παρουσία **πεδίων soft delete (deleted_at)** διευκολύνει την αρχειοθέτηση χωρίς απώλεια ιστορικότητας, ενώ ο πίνακας **notifications** υποστηρίζει τη στοχευμένη ενημέρωση χρηστών για αλλαγές και ενέργειες που απαιτούνται.

Τέταρτον, οι **τεχνικές ασφάλειας** που υιοθετήθηκαν (ενιαίο **db_conn.php** για ελεγχόμενη σύνδεση, κρυπτογράφηση κωδικών με **bcrypt**, επικύρωση και «απολύμανση εισόδου» με τη χρήση **filter_var()** και **htmlspecialchars()**, χρήση PDO με prepared statements και **bindParam()**, και επιβολή πρόσβασης μέσω **HTTPS**) θωρακίζουν την ακεραιότητα και την εμπιστευτικότητα των

δεδομένων. Σε συνδυασμό με τους περιορισμούς ακεραιότητας σε επίπεδο σχήματος (πρωτεύοντα/ξένα κλειδιά, ENUM πεδία για ελεγχόμενα λεξιλόγια), το σύστημα διατηρεί συνεπές επιχειρησιακό μοντέλο ακόμα και υπό έντονη χρήση.

Συνολικά, το Κεφάλαιο 3 απέδειξε ότι οι απαιτήσεις της εφαρμογής δεν αποτελούν απλή λίστα λειτουργιών, αλλά μετασχηματίζονται σε **συνεκτική σχεδίαση**: ρόλοι και ροές με καθαρές ευθύνες, σχήμα δεδομένων που υποστηρίζει τις πραγματικές ανάγκες (εξάμηνα, εξεταστικές, πολλαπλές αναθέσεις), προληπτικοί έλεγχοι που μειώνουν σφάλματα και μηχανισμοί ενημέρωσης που διασφαλίζουν διαφάνεια. Με αυτή τη θεμελίωση, το επόμενο κεφάλαιο εστιάζει στην **πρακτική όψη της υλοποίησης**: παρουσιάζονται αναλυτικά οι οθόνες και τα σενάρια χρήσης ανά ρόλο (Διαχειριστής/Γραμματεία, Καθηγητής, Φοιτητής), ώστε να γίνει εμφανές πώς η σχεδίαση του παρόντος κεφαλαίου υλοποιείται και λειτουργεί στην πράξη.

Κεφάλαιο 4: Υλοποίηση και δυνατότητες οντοτήτων

Στο παρόν κεφάλαιο παρουσιάζονται αναλυτικά οι λειτουργικότητες και οι δυνατότητες που παρέχει το πληροφοριακό σύστημα **TMS** στους διαφορετικούς τύπους χρηστών. Η ανάπτυξη και σχεδίαση του συστήματος στηρίχθηκε σε μια φιλοσοφία διαφοροποιημένων ρόλων, έτσι ώστε κάθε κατηγορία χρήστη να διαθέτει ξεχωριστά δικαιώματα και εργαλεία διαχείρισης, προσαρμοσμένα στις ανάγκες και ευθύνες της.

Η παρουσίαση γίνεται με παραδείγματα οθονών και στιγμιότυπα του περιβάλλοντος χρήστη, ώστε να καταδειχθεί με σαφήνεια ο τρόπος λειτουργίας της εφαρμογής. Για κάθε κατηγορία χρήστη εξετάζονται οι δυνατότητες πρόσβασης και οι διαθέσιμες ενέργειες.

- Στην **ενότητα 4.1** παρουσιάζονται οι δυνατότητες του διαχειριστή και της γραμματείας, οι οποίες περιλαμβάνουν την εισαγωγή μαθημάτων, τον ορισμό αιθουσών, την κατανομή καθηγητών και τη δημιουργία του βασικού ωρολογίου προγράμματος.
- Στην **ενότητα 4.2** παρουσιάζονται οι δυνατότητες που προσφέρονται στον καθηγητή, όπως η προβολή και διαχείριση των μαθημάτων που διδάσκει, η δυνατότητα υποβολής αιτημάτων για αίθουσες και η άμεση ενημέρωση για αλλαγές στο πρόγραμμά του.
- Στην **ενότητα 4.3** αναλύονται οι λειτουργίες που απευθύνονται στον φοιτητή, κυρίως η εξατομικευμένη προβολή του προσωπικού του ωρολογίου προγράμματος, το φιλτράρισμά του ανά εξάμηνο ή αίθουσα και η ενημέρωση για αλλαγές σε πραγματικό χρόνο.

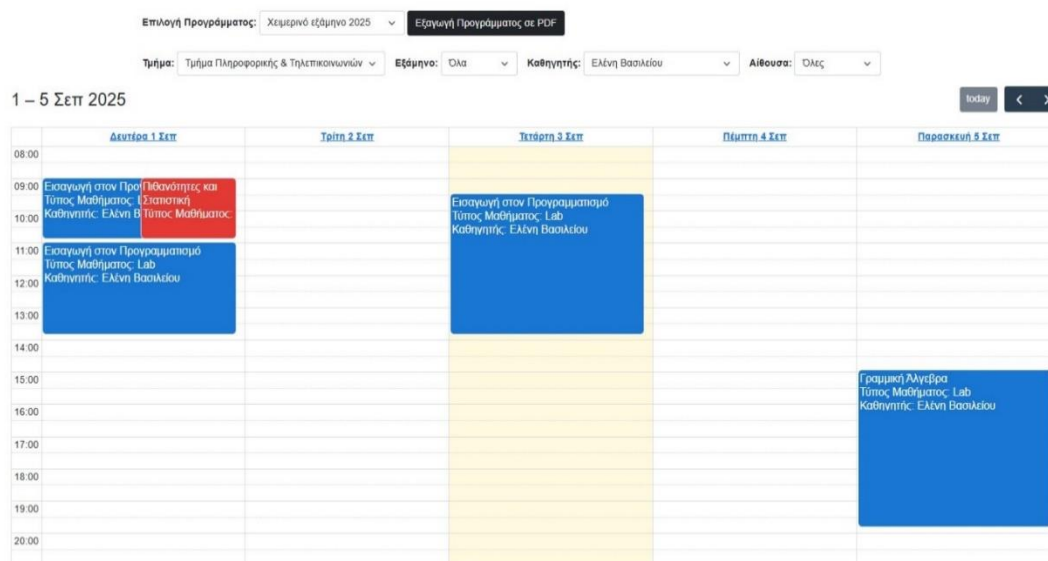
Με αυτόν τον τρόπο επιτυγχάνεται μια συνολική εικόνα του συστήματος μέσα από το πρίσμα των διαφορετικών χρηστών, αναδεικνύοντας τον τρόπο με τον οποίο η εφαρμογή διευκολύνει την οργάνωση και τη λειτουργία ενός πανεπιστημιακού τμήματος.

4.1 Δυνατότητες διαχειριστή και γραμματείας

Η κατηγορίες χρηστών «Διαχειριστή» και «Γραμματεία» έχουν τα ίδια δικαιώματα για να αναλαμβάνουν τις διαχειριστικές δράσεις που χρειάζεται για το στήσιμο ενός πανεπιστημιακού τμήματος και για τη διαχείριση του προγράμματος και μοιράζονται τις ίδιες σελίδες στις οποίες έχουν πρόσβαση από τη μπάρα πλοήγησης στο πάνω μέρος της σελίδας, όπως φαίνεται στην Εικόνα 22.

Το καμπανάκι δίπλα από την επιλογή αποσύνδεσης, είναι η περιοχή που εμφανίζει όλες τις ειδοποιήσεις που αφορούν τον εκάστοτε χρήστη και είναι κοινό στοιχείο διεπαφής (UI) για όλους τους ρόλους χρήστη.

Το πρόγραμμα που εμφανίζεται στον χρήστη είναι πλαισιωμένο με πολλαπλά φίλτρα τα οποία εξαρτώνται από το πανεπιστήμιο που είναι επιλεγμένο αρχικά. Πρώτο είναι πάντα επιλεγμένο το κύριο πανεπιστήμιο του καθηγητή, μιας και δίνεται η δυνατότητα στους καθηγητές να διδάσκουν σε πολλαπλά εκπαιδευτικά ιδρύματα. Το πρόγραμμα μπορεί να φιλτραριστεί με βάση το επιλεγμένο πρόγραμμα, το εξάμηνο, τον καθηγητή και τις αίθουσες του πανεπιστημίου για να δίνεται πλήρη και πολυδιάστατη εικόνα στους χρήστες, γεγονός που τους βοηθάει στην ορθή αντιμετώπιση των αιτημάτων των καθηγητών. Ακολουθεί στην Εικόνα 23 η εικόνα που βλέπουν οι χρήστες του ρόλου.



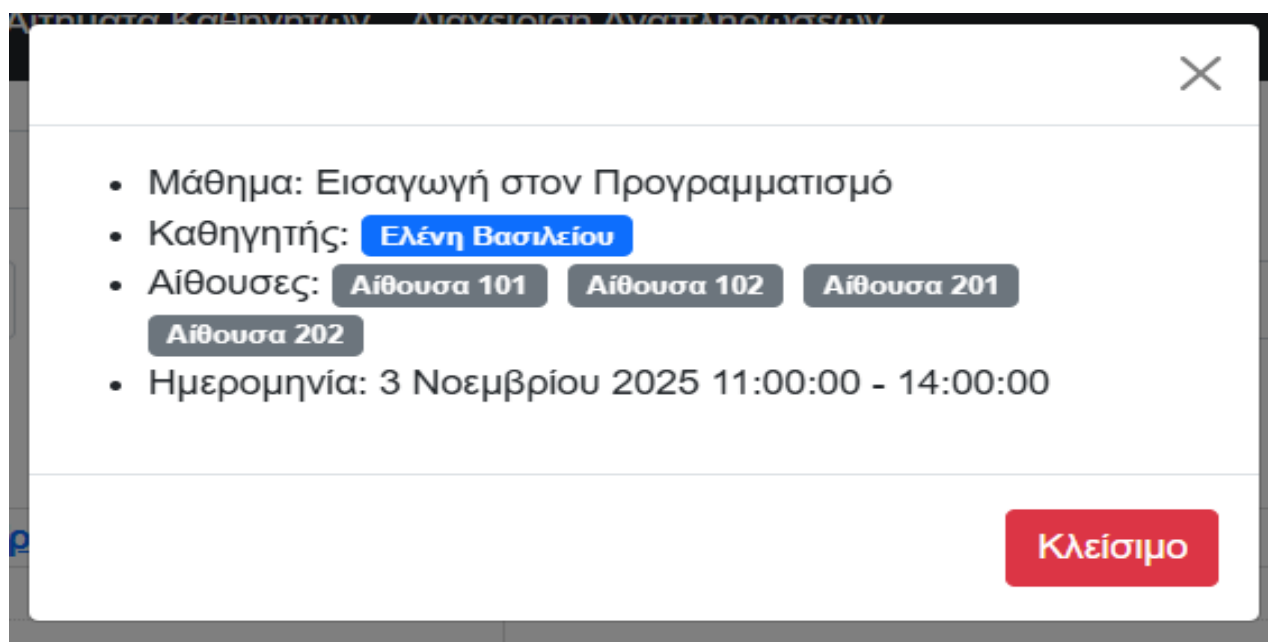
Εικόνα 23 Πρόγραμμα - Σελίδα εισόδου όλων των ρόλων

Τα χρωματιστά κελιά αντιπροσωπεύουν χρονοθυρίδες μαθημάτων και χρησιμοποιούνται διαφορετικά χρώματα για τις περιπτώσεις συγκρούσεων όπως φαίνεται στην παραπάνω εικόνα. Σε περίπτωση αναπλήρωσης το συμβάν παρουσιάζεται με πορτοκαλί χρώμα και μετά τη λήξη της, δεν εμφανίζεται ξανά στο πρόγραμμα. Ένα παράδειγμα αναπλήρωσης εμφανίζεται στην Εικόνα 24.

Τρίτη 4 Νοε	Τετάρτη 5 Νοε	Πέμπτη 6 Νοε
	Εισαγωγή στον Προγραμματισμό Τύπος Μαθήματος: Lab Καθηγητής: Ελένη Βασιλείου	Έκτακτο - Γραμμική Άλγεβρα Τύπος Μαθήματος: Lab Καθηγητής: Δημήτριος Καραγιάννης,

Εικόνα 24 Αναπλήρωση μαθήματος

Το κάθε συμβάν είναι διαδραστικό και ο χρήστης μπορεί να κάνει κλικ σε αυτό και να του εμφανιστεί ένα modal με περισσότερα και πιο συγκεκριμένα στοιχεία για το μάθημα, όπως φαίνεται στην Εικόνα 25.

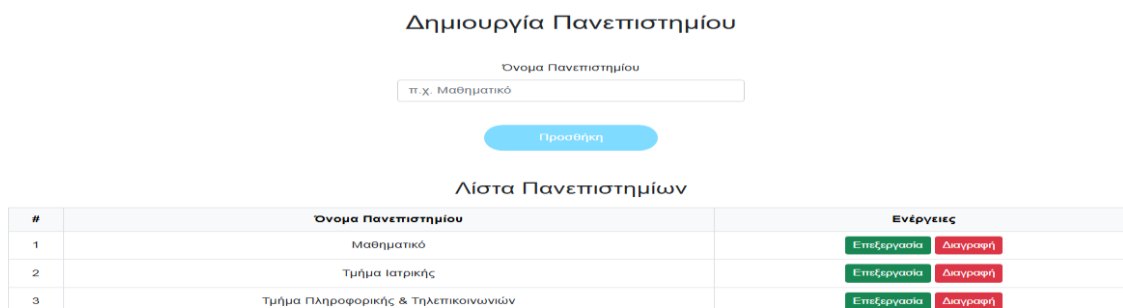


Εικόνα 25 Αναδυόμενο Παράθυρο πληροφορίας μαθήματος

Στις επόμενες υποενότητες Ακολουθεί η ανάλυση κάθε μίας επιλογής της μπάρας πλοήγησης ξεχωριστά και των σελίδων που μπορεί να έχουν πρόσβαση οι χρήστες «Γραμματεία» και «Διαχειριστής», καθώς και οι λειτουργίες τους, για την πλήρη κατανόηση του συστήματος.

4.1.1 Διαχείριση Τμήματος

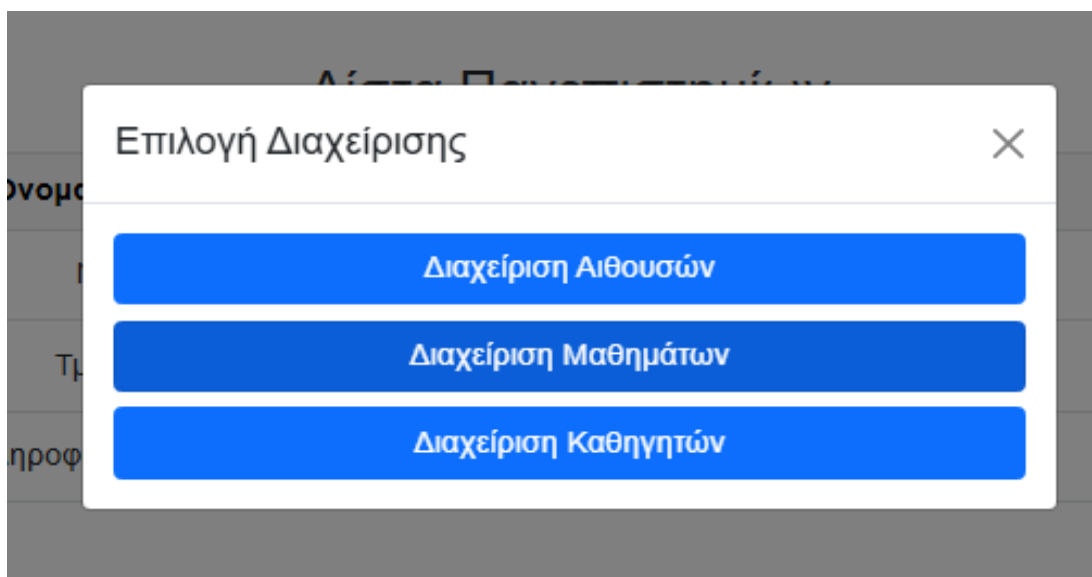
Σε αυτή τη σελίδα μπορούμε να δημιουργήσουμε νέο πανεπιστημιακό τμήμα με το πλήκτρο προσθήκη και να διαγράψουμε τα υπάρχοντα τμήματα με το κόκκινο κουμπί διαγραφής δίπλα από το κάθε τμήμα όπως φαίνεται στην Εικόνα 26.



#	Όνομα Πανεπιστημίου	Ενέργειες
1	Μαθηματικό	<button>Επεξεργασία</button> <button>Διαγραφή</button>
2	Τμήμα Ιατρικής	<button>Επεξεργασία</button> <button>Διαγραφή</button>
3	Τμήμα Πληροφορικής & Τηλεπικοινωνιών	<button>Επεξεργασία</button> <button>Διαγραφή</button>

Εικόνα 26 Καρτέλα δημιουργίας πανεπιστημίου

Με το που επιλέξουμε το πλήκτρο επεξεργασία για κάποιο υπάρχον τμήμα, εμφανίζεται ένα modal με διαφορετικές επιλογές όπως φαίνεται στην Εικόνα 27.



Εικόνα 27 Αναδυόμενο παράθυρο επιλογών διαχείρισης πανεπιστημίου

Συγκεκριμένα ο χρήστης μπορεί να επιλέξει ανάμεσα στις σελίδες:

- Διαχείριση Αιθουσών
- Διαχείριση Μαθημάτων
- Διαχείριση Καθηγητών

Στη σελίδα διαχείρισης αιθουσών ο χρήστης μπορεί να προσθέσει αίθουσες πανεπιστημίου μέσω της συμπλήρωσης της σχετικής φόρμας και υπάρχει πρόνοια για τη διευκρίνιση του κτιρίου που βρίσκεται η αίθουσα, σε περίπτωση που κάποια μαθήματα ή εργαστήρια λαμβάνουν χώρα σε διαφορετικά τμήματα της εκάστοτε πανεπιστημιούπολης.

Η στήλη εξοπλισμού περιέχει πληροφορίες που αφορούν τον εξοπλισμό της κάθε αίθουσας και συνεισφέρει στην ευκολότερη κατανομή των μαθημάτων κυρίως των εργαστηρίων, που μπορεί να χρειάζονται εξειδικευμένο εξοπλισμό για τη διεξαγωγή τους. Ο χρήστης επίσης μπορεί να επεξεργαστεί (ίδια φόρμα με αυτή της προσθήκης αίθουσας) και να διαγράψει αίθουσες με το κίτρινο και κόκκινο κουμπί αντίστοιχα. Στις εικόνες Εικόνα 28 και Εικόνα 29 φαίνονται όλες οι λειτουργίες της συγκεκριμένης σελίδας.

Διαχείριση Αιθουσών για το "Τμήμα Πληροφορικής & Τηλεπικοινωνιών"

Προσθήκη Αίθουσας

Λίστα Αιθουσών

#	Όνομα Αίθουσας	Χωρητικότητα	Κτίριο	Εξοπλισμός	Ενέργειες
1	Αίθουσα 101	30	Κτίριο Α	25 Η/Υ	Επεξεργασία Διαγραφή
2	Αίθουσα 102	35	Κτίριο Α		Επεξεργασία Διαγραφή
3	Αίθουσα 201	40	Κτίριο Β		Επεξεργασία Διαγραφή
4	Αίθουσα 202	25	Κτίριο Β		Επεξεργασία Διαγραφή
5	Αίθουσα 301	25	Κτίριο Β		Επεξεργασία Διαγραφή
6	Αίθουσα 302	25	Κτίριο Β		Επεξεργασία Διαγραφή

Εικόνα 28 Σελίδα Διαχείρισης Αιθουσών πανεπιστημίου

Εικόνα 29 Αναδυόμενο παράθυρο επεξεργασίας αίθουσας

Η επόμενη διαθέσιμη επιλογή για τη διαχείριση τμήματος είναι αυτή της διαχείρισης μαθημάτων. Στην αρχική σελίδα βλέπουμε το κουμπί προσθήκης μαθήματος καθώς και τον πίνακα των ήδη καταχωρημένων μαθημάτων για το πανεπιστήμιο που επιλέξαμε με στήλες όπως το όνομα του μαθήματος, την περιγραφή του αν υπάρχει, το εξάμηνό του, τις ώρες θεωρίας και εργαστηρίου αν υπάρχουν τα προαπαιτούμενα μαθήματα και τις διαχειριστικές ενέργειες επεξεργασίας και διαγραφής του μαθήματος, όπως φαίνεται στην Εικόνα 30.

Διαχείριση Μαθημάτων για το "Τμήμα Πληροφορικής & Τηλεπικοινωνιών"

Προσθήκη Μαθήματος

Λίστα Μαθημάτων

#	Όνομα	Περιγραφή	Εξάμηνο	Θεωρία Ωρες	Εργαστήριο Ωρες	Προαπαιτούμενα	Τύπος	Ενέργειες
1	Αντικειμενοστραφής Προγραμματισμός	Αντικειμενοστραφής Προγραμματισμός: Κατανόηση των αρχών του αντικειμενοστραφούς προγραμματισμού και της εφαρμογής τους.	Εξάμηνο 3	0	0		Lab	Επεξεργασία Διαγραφή
2	Αντικειμενοστραφής προγραμματισμός 2	Java	Εξάμηνο 3	4	4		Both	Επεξεργασία Διαγραφή
3	Αντικειμενοστραφής προγραμματισμός 3	C#	Εξάμηνο 5	4	4		Both	Επεξεργασία Διαγραφή
4	Αρχιτεκτονική Υπολογιστών I	Αρχιτεκτονική Υπολογιστών I: Εξετάζει τη δομή και τη λειτουργία των σύγχρονων υπολογιστών.	Εξάμηνο 2	0	0		Both	Επεξεργασία Διαγραφή
5	Ασύρματα Δίκτυα Αισθητήρων	Ασύρματα Δίκτυα Αισθητήρων: Εξερεύνηση των τεχνολογιών και των εφαρμογών των ασύρματων δικτύων αισθητήρων.	Εξάμηνο 6	0	0		Theory	Επεξεργασία Διαγραφή
		Γραμμική Άλγεβρα: Εκμάθηση						

Εικόνα 30 Σελίδα Διαχείρισης Μαθημάτων

Το modal με τις φόρμες προσθήκης και επεξεργασίας των στοιχείων των μαθημάτων παρουσιάζεται στην Εικόνα 31. Όταν ο χρήστης επιλέξει τον τύπο του μαθήματος, δηλαδή αν είναι μάθημα που έχει μόνο θεωρία ή μόνο εργαστήριο απενεργοποιείται η επιλογή ωρών εργαστηρίου ή θεωρίας αντίστοιχα.

Επεξεργασία Μαθήματος

✕

Όνομα Μαθήματος

Αντικειμενοστραφής Προγραμματισμός

Περιγραφή

Αντικειμενοστραφής Προγραμματισμός: Κατανόηση των αρχών του αντικειμενοστραφούς προγραμματισμού και της

Εξάμηνο

Εξάμηνο 3

Τύπος

Εργαστήριο

Ώρες Θεωρίας

0

Ώρες Εργαστηρίου

0

Προαπαιτούμενα

Άκυρο

Αποθήκευση

Εικόνα 31 Αναδυόμενο παράθυρο επεξεργασίας/προσθήκης μαθήματος

Η τελευταία λειτουργία της σελίδας διαχείρισης τμήματος αφορά την διαχείριση των καθηγητών και την κατανομή τους στο εκάστοτε μάθημα που διδάσκουν. Επειδή μπορούν παραπάνω από ένας καθηγητής να διδάσκει ένα μάθημα, δίνεται η δυνατότητα προσθήκης πολλαπλών καθηγητών για κάθε μάθημα, αρκεί ο καθηγητής να είναι δηλωμένος ως μέλος της σχολής.

Επίσης καθηγητές άλλων πανεπιστημιακών τμημάτων μπορούν να διδάσκουν μαθήματα σε άλλη σχολή, όπως τις περιπτώσεις συνδιδασκαλίας μαθημάτων, μπορεί να επιλεγεί καθηγητής και από άλλο πανεπιστημιακό τμήμα, αρκεί πάλι να είναι ο καθηγητής να είναι δηλωμένος και ως μέλος των σχολών που έχουν στον οδηγό σπουδών τους το συγκεκριμένο μάθημα. Με αυτό τον τρόπο ένα μάθημα υπό καθεστώς συνδιδασκαλίας, θα εμφανίζεται στο πρόγραμμα και των δύο ή παραπάνω τμημάτων όπου διδάσκεται το μάθημα.

Η τελευταία δυνατότητα που προσφέρεται για τη διαχείριση του πανεπιστημιακού τμήματος είναι η κατανομή των μαθημάτων στους καθηγητές ενός τμήματος. Αυτό επιτυγχάνεται από τη σελίδα «Διαχείριση καθηγητών» που εμφανίζεται στην Εικόνα 32.

Διαχείριση Καθηγητών ανά Πανεπιστήμιο

Επιλέξτε πανεπιστήμιο: Τμήμα Πληροφορικής & Τηλεπικοινωνιών ▾

Εξάμηνο: -- Όλα -- ▾

Μαθήματα

#	Μάθημα	Καθηγητές	Ενέργειες
1	Αντικειμενοστραφής Προγραμματισμός	Αλέξανδρος Παπαδόπουλος Δημήτριος Καραγιάννης	Επεξεργασία
2	Αντικειμενοστραφής προγραμματισμός 2	Κανέναν	Επεξεργασία
3	Αντικειμενοστραφής προγραμματισμός 3	Κανέναν	Επεξεργασία
4	Αρχιτεκτονική Υπολογιστών I	Αλέξανδρος Παπαδόπουλος Δημήτριος Καραγιάννης	Επεξεργασία
5	Ασύρματα Δίκτυα Αισθητήρων	Δημήτριος Καραγιάννης Ελένη Βασιλείου	Επεξεργασία
6	Γραμμική Άλγεβρα	Δημήτριος Καραγιάννης Ελένη Βασιλείου	Επεξεργασία
7	Δίκτυα Επικοινωνιών I	Κωνσταντίνος Παπανικολάου	Επεξεργασία

Εικόνα 32 Σελίδα Διαχείρισης Καθηγητών

Όπως φαίνεται μπορεί να επιλεγθούν παραπάνω καθηγητές ανά μάθημα και πατώντας το κουμπί «Επεξεργασία» μπορούν να αφαιρεθούν οι να προστεθούν καθηγητές. Στο αναδυόμενο παράθυρο ο χρήστης μπορεί και να αναζητήσει καθηγητές για να επιλέξει, και να επιλέξει πολλούς καθηγητές πατώντας το πλήκτρο ctrl. Το παράθυρο φαίνεται στην Εικόνα 33.

Ανάθεση Καθηγητών στο μάθημα:
Αντικειμενοστραφής Προγραμματισμός

Επιλέξτε Καθηγητές

Αναζήτηση καθηγητή...

Αλέξανδρος Παπαδόπουλος

Δημήτριος Καραγιάννης

Ελένη Βασιλείου

Κωνσταντίνος Παπανικολάου

Κώστας Παναγιώτου

Κλείσιμο

Αποθήκευση

Εικόνα 33 Αναδυόμενο παράθυρο ανάθεσης καθηγητών

4.1.2 Δημιουργία Προγράμματος

Η επόμενη επιλογή στην μπάρα πλοήγησης είναι η «Δημιουργία Προγράμματος». Είναι μια από τις πιο βασικές λειτουργίες που θα πρέπει να γίνεται πρώτη χρονικά πάντα από τη γραμματεία, καθώς η αντιστοίχιση των δηλώσεων μαθημάτων, η εμφάνιση του τελικού προγράμματος και το

φιλτράρισμα των δεδομένων του, οι αναπληρώσεις, η καταγραφή ιστορικών δεδομένων για παλαιότερα προγράμματα και η εκτύπωση του προγράμματος εξαρτώνται άμεσα από το επιλεγμένο πρόγραμμα. Η φόρμα συμπλήρωσης του προγράμματος επιτρέπει στον χρήστη να επιλέξει τον τύπο του προγράμματος (δηλαδή αν είναι πρόγραμμα διδασκαλίας ή εξεταστικής) και την περίοδο του προγράμματος (Χειμερινό, Εαρινό, Εξεταστικής) γεγονός που βοηθάει στο αποτελεσματικό φιλτράρισμα των δηλώσεων μαθημάτων, στην σωστή εμφάνιση και αντιστοίχιση των μαθημάτων στην εκάστοτε χρονική περίοδο και στην διατήρηση ιστορικότητας για παλιά προγράμματα. Η φόρμα με ένα παράδειγμα συμπλήρωσης παρουσιάζεται στην Εικόνα 34.

Δημιουργία Νέου Προγράμματος

Πανεπιστήμιο

Τμήμα Πληροφορικής & Τηλεπικοινωνιών (Κύρια Σχολή)

Όνομα Προγράμματος

Εαρινό Πρόγραμμα 2026

Ημερομηνία Έναρξης

03/01/2026

Ημερομηνία Λήξης

07/01/2026

Τύπος

Διδασκαλία

Περίοδος

Εαρινό

Έτος

2026

Αποθήκευση

Εικόνα 34 Αναδυόμενο παράθυρο δημιουργίας νέου προγράμματος

4.1.3 Διαχείριση Χρηστών

Στην σελίδα «Διαχείρισης Χρηστών» ο χρήστης μπορεί να φιλτράρει τους χρήστες της εφαρμογής, με βάση το Πανεπιστημιακό τμήμα και τον ρόλο τους και να αναζητήσει συγκεκριμένους χρήστες με το όνομά τους. Στον πίνακα της Εικόνα 35, οι χρήστες εμφανίζονται με το username, τον ρόλο, το email και τα πανεπιστήμια που είναι καταχωρημένοι και υπάρχει και η στήλη «Ενέργειες» με τα κουμπιά «Επεξεργασία» και «Διαγραφή». Πατώντας το κουμπί επεξεργασίας, αναδύεται το παράθυρο της Εικόνα 36 , όπου ο διαχειριστής ή το γραμματειακό προσωπικό μπορεί να προσδώσει ρόλο και να επιλέξει το κύριο πανεπιστήμιο για τον καθηγητή, γεγονός που βοηθά στην προεπιλογή του πανεπιστημίου στο πρόγραμμα και στα μαθήματά του, και να τον κατανείμει και σε παραπάνω από ένα πανεπιστήμιο.

Λίστα χρηστών

Αναζήτηση:

Πανεπιστήμιο:

Ρόλος:

Προσθήκη Χρήστη

#	Username	Ρόλος	Email	Πανεπιστήμια	Ενέργειες	
3	secretary@secretary.com	Γραμματεία	secretary@secretary.com	Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Τμήμα Ιατρικής	Επεξεργασία	Διαγραφή
4	Αλέξανδρος Παπαδόπουλος	Καθηγητής	alekrap@gmail.com	Τμήμα Πληροφορικής & Τηλεπικοινωνιών	Επεξεργασία	Διαγραφή
5	Δημήτριος Καραγιάννης	Καθηγητής	jimkaragiannis@yahoo.com	Τμήμα Πληροφορικής & Τηλεπικοινωνιών	Επεξεργασία	Διαγραφή
6	Ελένη Βασιλείου	Καθηγητής	helenvas@outlook.com	Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Τμήμα Ιατρικής	Επεξεργασία	Διαγραφή
7	Κωνσταντίνος Παπανικολάου	Καθηγητής	kospap@gmail.com	Τμήμα Πληροφορικής & Τηλεπικοινωνιών	Επεξεργασία	Διαγραφή
10	Ελένη Κωστάκη	Μαθητής	eleni.kostas@email.com	Τμήμα Πληροφορικής & Τηλεπικοινωνιών	Επεξεργασία	Διαγραφή
15	Νίκος Παπαδόπουλος	Μαθητής	nikos.papadopoulos@email.com	Τμήμα Πληροφορικής & Τηλεπικοινωνιών	Επεξεργασία	Διαγραφή
16	Κώστας Παναγιώτου	Καθηγητής	kws.pan@uowm.ece.gr	Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Τμήμα Ιατρικής	Επεξεργασία	Διαγραφή
18	Kosmas Stamos	Μαθητής	st0355@uowm.ece.gr	Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Τμήμα Ιατρικής	Επεξεργασία	Διαγραφή

Εικόνα 35 Λίστα Χρηστών

Επεξεργασία Χρήστη

Ρόλος

Αναζήτηση Σχολείων

Μαθηματικό
Τμήμα Ιατρικής
Τμήμα Πληροφορικής & Τηλεπικοινωνιών

Κύριο Πανεπιστήμιο

Τμήμα Πληροφορικής & Τηλεπικοινωνιών

Κλείσιμο

Αποθήκευση

Εικόνα 36 Αναδυόμενο παράθυρο επεξεργασίας χρήστη

Επίσης με το κουμπί προσθήκη εμφανίζεται το παράθυρο της Εικόνα 37 για την προσθήκη χρήστη.

Προσθήκη Χρήστη

Όνομα Χρήστη

Email

Κωδικός Πρόσβασης

Ρόλος

Διαχειριστής

Σχολεία

Μαθηματικό
Τμήμα Ιατρικής
Τμήμα Πληροφορικής & Τηλεπικοινωνιών

Κύριο Σχολείο

Επιλέξτε κύριο σχολείο

Κλείσιμο

Προσθήκη Χρήστη

4.1.4 Αιτήματα καθηγητών

Η σελίδα «Αιτήματα Καθηγητών» συγκεντρώνει τα αιτήματα για διαθεσιμότητα από όλους τους καθηγητές για τους οποίους είναι υπεύθυνοι οι χρήστες με ρόλο γραμματεία ή διαχειριστή. Όπως φαίνεται στην Εικόνα 38, υπάρχουν φίλτρα ανά πανεπιστήμιο, ανά εξάμηνο, ανά αίθουσα και ανά καθηγητή τα οποία εφαρμόζονται πατώντας το κουμπί «Φιλτράρισμα». Ο χρήστης έχει όλες τις απαραίτητες πληροφορίες για την αξιολόγηση του αιτήματός του εκάστοτε καθηγητή και είναι σε θέση ελέγχοντας το πρόγραμμα να αποφασίσει αν θα εγκρίνει το αίτημα ή όχι.

Διαχείριση Αιτημάτων Καθηγητών

Τμήμα Πληροφορικής & Τηλεπικοινωνιών

Όλα τα εξάμηνα

Όλες οι αίθουσες

Όλοι οι καθηγητές

Φιλτράρισμα

Σχολή	Καθηγητής	Μάθημα	Εξάμηνο	Τύπος	Κατηγορία	Ημέρα	Ώρες	Αίθουσες	Σχόλια	Ενέργειες
Τμήμα Πληροφορικής & Τηλεπικοινωνιών	Ελένη Βασιλείου	Τεχνητή Νοημοσύνη Ι	Εξάμηνο 5	Θεωρία	Κορμού	Παρασκευή	09:00:00 - 11:00:00	Αίθουσα 302		Έγκριση
										Απόρριψη

4.1.5 Διαχείριση Αναπληρώσεων

Η σελίδα «Διαχείριση Αναπληρώσεων» έχει ακριβώς την ίδια λειτουργία και δομή με τη σελίδα «Αιτήματα Καθηγητών». Περιέχει ακριβώς τα ίδια φίλτρα, τον ίδιο πίνακα με πληροφορίες για το αίτημα αναπλήρωσης του καθηγητή και ο χρήστης μπορεί από αυτή να εγκρίνει ή να απορρίψει τα εκάστοτε αιτήματα. Η δομή της φαίνεται στην Εικόνα 39.

Διαχείριση Αναπληρώσεων

Τμήμα Πληροφορικής & Τηλεπικοινωνιών

Όλα τα εξάμηνα

Όλες οι αίθουσες

Όλοι οι καθηγητές

Φιλτράρισμα

Σχολή	Καθηγητής	Μάθημα	Εξάμηνο	Τύπος	Ημερομηνία	Ωρες	Αίθουσες	Σχόλια	Ενέργειες
Τμήμα Πληροφορικής & Τηλεπικοινωνιών	Ελένη Βασιλείου	Πιθανότητες και Στατιστική	Εξάμηνο 3	Θεωρία	2025-09-24	16:00:00 - 18:00:00	Αίθουσα 102, Αίθουσα 201	Αναπλήρωση λόγω ασθένειας	Έγκριση Απόρριψη

Εικόνα 39 Διαχείριση Αναπληρώσεων

4.2 Δυνατότητες ρόλου καθηγητή

Ο ρόλος του καθηγητή περιορίζεται στη δήλωση χρονοθυρίδων για τα μαθήματα που διδάσκει και για αναπληρώσεις μαθημάτων. Στην αρχική σελίδα που εμφανίζεται το πρόγραμμα του καθηγητή, αφού το συμβουλευθεί, θα έχει τις εξής δυνατότητες από τη μπάρα πλοήγησης, όπως φαίνεται στην Εικόνα 40.

Αρχική Δήλωση Διαθεσιμότητας Τα Μαθήματά μου Δήλωση Έκτακτου Μαθήματος

Εικόνα 40 Μπάρα πλοήγησης καθηγητή

4.2.1 Τα Μαθήματά μου

Στη σελίδα «Τα μαθήματά μου» ο καθηγητής έχει στη διάθεση του όλες τις πληροφορίες που αφορούν τα μαθήματά του, όπως φαίνεται στην Εικόνα 41. Ο καθηγητής μπορεί να επιλέξει το πανεπιστήμιό που επιθυμεί και να φιλτράρει περαιτέρω τις επιλογές μαθημάτων με τα φίλτρα εξαμήνου και αίθουσας. Κάτι που πρέπει να σημειωθεί, είναι ότι το φίλτρο αίθουσας, αποκτάει δεδομένα για τα μαθήματα, μόνο αν η διδασκαλία κάποιου μαθήματος σε αυτήν επικυρωθεί από τη γραμματεία, κατόπιν αιτήσεως του καθηγητή. Οπότε αν ο χρήστης προσπαθήσει να φιλτράρει τα μαθήματά του μέσω του φίλτρου αιθουσών και δεν λάβει αποτελέσματα, ο λόγος που συμβαίνει αυτό είναι επειδή η συγκεκριμένη αίθουσα δε χρησιμοποιείται για κάποιο μάθημα που διδάσκει ο καθηγητής.

Τα ενεργά μαθήματά μου

Φίλτρο Σχολής		Φίλτρο Εξαμήνου		Φίλτρο Αίθουσας		
Τμήμα Πληροφορικής & Τηλεπικοινωνιών (Κύρια)		Εξάμηνο 8		Όλες οι αίθουσες		
#	Όνομα Μαθήματος	Σχολή	Περιγραφή	Εξάμηνο	Αίθουσες	Τύπος/ Ώρες
1	Ειδικά Θέματα Υπολογιστικών Συστημάτων και Εφαρμογών: Υπολογιστικά Συστήματα Μεγάλης Κλίμακας	Τμήμα Πληροφορικής & Τηλεπικοινωνιών	Ειδικά Θέματα Υπολογιστικών Συστημάτων και Εφαρμογών: Υπολογιστικά Συστήματα Μεγάλης Κλίμακας: Εξετάζει τα θέματα και τις προκλήσεις που σχετίζονται με μεγάλης κλίμακας υπολογιστικά συστήματα.	Εξάμηνο 8	-	Θεωρία: 0 ώρες
2	Επεξεργασία Εικόνων	Τμήμα Πληροφορικής & Τηλεπικοινωνιών	Επεξεργασία Εικόνων: Εκμάθηση των τεχνικών και των εργαλείων για την επεξεργασία και ανάλυση εικόνων.	Εξάμηνο 8	-	Θεωρία: 5 ώρες, Εργαστήριο: 5 ώρες
3	Κρυπτογραφία	Τμήμα Πληροφορικής & Τηλεπικοινωνιών	Κρυπτογραφία: Κατανόηση των μεθόδων προστασίας των πληροφοριών μέσω της κρυπτογραφίας.	Εξάμηνο 8	-	Θεωρία: 0 ώρες

Εικόνα 41 Τα μαθήματά μου

4.2.2 Δήλωση Διαθεσιμότητας

Στη σελίδα «Δήλωση διαθεσιμότητας» λαμβάνει χώρα μια από τις πιο σημαντικές διεργασίες της εφαρμογής, η δήλωση χρονοθυρίδων διαθεσιμότητας για τα μαθήματα των καθηγητών. Ο καθηγητής πρέπει πρώτα να επιλέξει το πρόγραμμα και το εξάμηνο για το οποίο θέλει να δηλώσει διαθεσιμότητα για τα μαθήματά του όπως φαίνεται στην Εικόνα 42.

Επιλέξτε Σχολή: Τμήμα Πληροφορικής & Τηλεπικοινωνιών
 Πρόγραμμα: Χειμερινό εξάμηνο 2025 (Χειμερινό 2025)
 Εξάμηνο: Εξάμηνο 1

Μάθημα	Περιγραφή	Εξάμηνο	Ώρες Θεωρίας	Ώρες Εργαστηρίου	Ανάγκες Εξοπλισμού	Τύπος Μαθήματος
Γραμμική Άλγεβρα	Γραμμική Άλγεβρα: Εκμάθηση των βασικών εννοιών της γραμμικής άλγεβρας και των εφαρμογών της.	1	0	5		Εργαστήριο

Τύπος: Επιλέξτε...
 Ημέρα: Επιλέξτε...
 Ώρα Έναρξης: --:--
 Ώρα Λήξης: --:--
 Αίθουσες:

Αίθουσα 101
 Αίθουσα 102
 Αίθουσα 201
 Αίθουσα 202
 Αίθουσα 302

 Υποβολή Αφαίρεση

Δηλωμένες Διαθεσιμότητες:

Ημέρα	Ώρα Έναρξης	Ώρα Λήξης	Διάρκεια	Τύπος	Αίθουσες	Σχόλια	Κατάσταση	Ενέργεια
Friday	15:00	20:00	5 ώρες	Εργαστήριο	Αίθουσα 302		Εγκρίθηκε	Διαγραφή

Μάθημα	Περιγραφή	Εξάμηνο	Ώρες Θεωρίας	Ώρες Εργαστηρίου	Ανάγκες Εξοπλισμού	Τύπος Μαθήματος
Εισαγωγή στον Προγραμματισμό	Εισαγωγή στον Προγραμματισμό: Μια βασική εισαγωγή στις αρχές του προγραμματισμού και των αλγορίθμων.	1	0	10	4455	Εργαστήριο

Τύπος: Επιλέξτε...
 Ημέρα: Επιλέξτε...
 Ώρα Έναρξης: --:--
 Ώρα Λήξης: --:--
 Αίθουσες:

Αίθουσα 101
 Αίθουσα 102
 Αίθουσα 201
 Αίθουσα 202
 Αίθουσα 302

 Υποβολή Αφαίρεση

Δηλωμένες Διαθεσιμότητες:

Εικόνα 42 Αρχική σελίδα "Δήλωση Διαθεσιμότητας"

Αφού εφαρμοστούν τα φίλτρα, εμφανίζονται οι φόρμες για τη δήλωση μαθημάτων καθώς και οι ώρες εργαστηρίου και θεωρίας για το καθένα. Επίσης, ο χρήστης έχει στη διάθεσή του πίνακα

ιστορικότητας για τις διαθεσιμότητες που έχουν δηλωθεί για το συγκεκριμένο μάθημα όπως παρουσιάζεται στην Εικόνα 43.

Μάθημα	Περιγραφή	Εξάμηνο	Ωρες Θεωρίας	Ωρες Εργαστηρίου	Ανάγκες Εξοπλισμού	Τύπος Μαθήματος
Εισαγωγή στον Προγραμματισμό	Εισαγωγή στον Προγραμματισμό: Μια βασική εισαγωγή στις αρχές του προγραμματισμού και των αλγορίθμων.	1	0	10	4455	Εργαστήριο

Τύπος: Επιλέξτε...

Ημέρα: Επιλέξτε...

Ωρα Έναρξης: --:--

Ωρα Λήξης: --:--

Αίθουσες

Αίθουσα 101
 Αίθουσα 102
 Αίθουσα 201
 Αίθουσα 202
 Αίθουσα 204

Υποβολή Αφαίρεση

Δηλωμένες Διαθεσιμότητες:

Ημέρα	Ωρα Έναρξης	Ωρα Λήξης	Διάρκεια	Τύπος	Αίθουσες	Σχόλια	Κατάσταση	Ενέργεια
Wednesday	09:00	14:00	5 ώρες	Εργαστήριο	Αίθουσα 101, Αίθουσα 102, Αίθουσα 201		Απορρίφθηκε	
Monday	09:00	14:00	5 ώρες	Εργαστήριο	Αίθουσα 101, Αίθουσα 102, Αίθουσα 201, Αίθουσα 202		Εγκρίθηκε	Διαγραφή
Wednesday	09:30	14:00	4.5 ώρες	Εργαστήριο	Αίθουσα 101, Αίθουσα 102, Αίθουσα 201		Εγκρίθηκε	Διαγραφή

Εικόνα 43 Φόρμα δήλωσης διαθεσιμότητας μαθήματος

Οι καταστάσεις που μπορεί να έχει μια δήλωση μαθήματος είναι οι εξής:

- **Not Set:** Η αρχική κατάσταση διαθεσιμότητας για τα μαθήματα.
- **Pending:** Η κατάσταση που αποκτά η διαθεσιμότητα πριν εγκριθεί ή απορριφθεί από το γραμματειακό προσωπικό.
- **Denied:** Η κατάσταση που αποκτά η διαθεσιμότητα όταν απορριφθεί από το γραμματειακό προσωπικό.
- **Confirmed:** Η κατάσταση που αποκτά η διαθεσιμότητα όταν εγκριθεί από το γραμματειακό προσωπικό.

Στον πίνακα «Δηλωμένες Διαθεσιμότητες» ο χρήστης μπορεί να ελέγχει την κατάσταση των δηλώσεών του για κάθε μάθημα και να λαμβάνει υπόψιν τις υπολειπόμενες διαθέσιμες ώρες μαθημάτων έχει στη διάθεσή του να δηλώσει, αφαιρώντας τις ώρες που έχουν εγκριθεί από τις συνολικές ώρες του μαθήματος. Το σύστημα επίσης εφαρμόζει τις ακόλουθες συνθήκες αυτοματοποιημένης απόρριψης για τις δηλώσεις των χρηστών.

- Η δήλωση διαθεσιμότητας δεν μπορεί να ξεπερνάει τις διαθέσιμες ώρες διδασκαλίας του μαθήματος.
- Η δήλωση διαθεσιμότητας θα πρέπει να λαμβάνει χώρα στο ωράριο του πανεπιστημίου.
- Ένας καθηγητής δεν μπορεί να διδάσκει ταυτόχρονα παραπάνω από ένα μαθήματα. Απορρίπτονται και οι επικαλυπτόμενες δηλώσεις αυτομάτως.

- Σε μια αίθουσα δεν είναι εφικτό να διδάσκονται παραπάνω από ένα μαθήματα την ίδια ώρα.
- Για μαθήματα του ίδιου εξαμήνου δεν επιτρέπεται να διδάσκονται τις ίδιες ώρες μαθήματα κορμού.

Όταν ο χρήστης δηλώσει μια διαθεσιμότητα επιτυχώς, αυτή εμφανίζεται στον πίνακα «Δηλωμένων Διαθεσιμοτήτων» με την πορτοκαλί ετικέτα «Σε εξέλιξη», όπως παρουσιάζεται στην Εικόνα 44. Η γραμματεία ενημερώνεται για τη δήλωση του καθηγητή και πρέπει με τη σειρά της να την εγκρίνει ή να την απορρίψει. Ανεξαρτήτως της απόφασής της ο καθηγητής θα ενημερωθεί με το σύστημα ενημερώσεων και του παρέχεται και η δυνατότητα να ελέγξει την κατάσταση της δήλωσής του, στην σελίδα «Δήλωση Διαθεσιμότητας», στον πίνακα «Δηλωμένες Διαθεσιμότητες» για το συγκεκριμένο μάθημα.

Επιλέξτε Σχολή: Τμήμα Πληροφορικής & Τηλεπικοινωνιών
Πρόγραμμα: Χειμερινό εξάμηνο 2025 (Χειμερινό 2025)
Εξάμηνο: Εξάμηνο 5

Μάθημα	Περιγραφή	Εξάμηνο	Ώρες Θεωρίας	Ώρες Εργαστηρίου	Ανάγκες Εξοπλισμού	Τύπος Μαθήματος
Τεχνητή Νοημοσύνη I	Τεχνητή Νοημοσύνη I: Εισαγωγή στις αρχές και τις μεθόδους της τεχνητής νοημοσύνης.	5	10	0	Lorem ipsum dolor sit amet consectetur adipisicing elit. Minima atque accusamus sit suscipit, cum illum harum commodi beatae maxime reiciendis provident sed dolores quibusdam asperiores est non quis cumque aliquam delectus optio magni facere distinctio! Distinctio cumque molestias officia maxime! Necessitatibus molestias aliquid quaerat quidem iste in quibusdam nostrum repudiandae?	Θεωρία

Τύπος
Επιλέξτε...

Ημέρα
Επιλέξτε...

Ώρα Έναρξης
--:--

Ώρα Λήξης
--:--

Αίθουσες
Αίθουσα 101
Αίθουσα 102
Αίθουσα 201
Αίθουσα 202
Αίθουσα 204

Υποβολή
Αφαίρεση

Δηλωμένες Διαθεσιμότητες:

Ημέρα	Ώρα Έναρξης	Ώρα Λήξης	Διάρκεια	Τύπος	Αίθουσες	Σχόλια	Κατάσταση	Ενέργεια
Friday	09:00	11:00	2 ώρες	Θεωρία	Αίθουσα 302		Σε εξέλιξη	Διαγραφή
Wednesday	09:00	14:00	5 ώρες	Θεωρία	Αίθουσα 302		Απορρίφθηκε	

Εικόνα 44 Δήλωση διαθεσιμότητας "Σε εξέλιξη"

4.2.3 Δήλωση Εκτάκτου μαθήματος

Η συγκεκριμένη σελίδα χρησιμοποιείται για τη δήλωση αναπληρώσεων μαθημάτων και είναι μια απλουστευμένη μορφή της σελίδας δηλώσεων διαθεσιμότητας όπως παρουσιάζεται στην Εικόνα 45. Αφού χρησιμοποιήσει τα φίλτρα που παρέχονται και επιλέξει το μάθημα για το οποίο επιθυμεί να αιτηθεί αναπλήρωση, πρέπει να συμπληρώσει την ημερομηνία, την ώρα και μία ή περισσότερες αίθουσες για τη διεξαγωγή του έκτακτου μαθήματος.

⚠ Δήλωση Έκτακτου Μαθήματος

Επιλέξτε Σχολή: Τμήμα Πληροφορικής & Τηλεπικοινωνιών ▾ Πρόγραμμα: Χειμερινό εξάμηνο 2025 (Χειμερινό) ▾ Εξάμηνο: Όλα ▾

Μάθημα: Εισαγωγή στον Προγραμματισμό ▾ Αίθουσα: Αίθουσα 101, Αίθουσα 102, Αίθουσα 201, Αίθουσα 202
Κρατήστε Ctrl (ή Cmd σε Mac) για να επιλέξετε πολλές αίθουσες.

Νέα Ημερομηνία: 09/18/2025 📅 Ώρα Έναρξης: 09:00 AM 🕒 Ώρα Λήξης: 12:00 PM 🕒

Σχόλια/Αιτιολόγηση
Παρακαλώ αιτιολογήστε την ανάγκη για έκτακτο μάθημα...

📌 Υποβολή Αιτήματος

Εικόνα 45 Δήλωση Έκτακτου Μαθήματος

Η δήλωση δεν περνάει από ελέγχους υπέρβασης των ωρών διδασκαλίας, αλλά μόνο από τους τέσσερις βασικούς λογικούς ελέγχους που αναφέρουμε στο κεφάλαιο για τη σελίδα «Δηλώσεις Διαθεσιμότητας». Το αίτημα καταφθάνει στην κατάλληλη σελίδα του χρήστη διαχειριστή ή γραμματείας και μπορεί πάλι να απορριφθεί ή να εγκριθεί. Ο καθηγητής μπορεί να δηλώσει αναπλήρωση μόνο για τα μαθήματα, για τα οποία έχει εγκριθεί έστω και μια ώρα διδασκαλίας στο πρόγραμμα.

4.3 Δυνατότητες φοιτητή

Ο ρόλος του χρήστη φοιτητή είναι ο πιο περιορισμένος συγκριτικά με τους άλλους τρεις προαναφερθέντες ρόλους. Η αρχική του σελίδα είναι η ίδια ακριβώς με αυτή των ρόλων διαχειριστή, γραμματεία και καθηγητή, δηλαδή η σελίδα του προγράμματος με όλα τα διαθέσιμα φίλτρα. Έχει όμως μια μόνο βασική λειτουργική διαφορά. Ο φοιτητής βλέπει μόνο τα μαθήματα που έχει επιλέξει ο ίδιος προς παρακολούθηση. Του δίνεται επομένως ο πλήρης έλεγχος της εξατομίκευσης του προσωπικού του προγράμματος, καθώς και η δυνατότητα εκτύπωσής του.

Η μόνη σελίδα που υπάρχει στην μπάρα πλοήγησης του φοιτητή ονομάζεται «Εγγραφή σε μαθήματα» και παρουσιάζεται στην Εικόνα 46. Περιέχει επίσης πολλαπλά φίλτρα και δυνατότητα αναζήτησης μαθήματος με το όνομά του και ο φοιτητής μπορεί να δηλώσει μαθήματα χωρίς κανένα περιορισμό, και να τα αφαιρέσει αν μελλοντικά δεν την παρακολουθήσει τους.

Εγγραφή σε Μαθήματα

Επιλέξτε πανεπιστήμιο, εξάμηνο και τα μαθήματα που θέλετε να παρακολουθήσετε.

Πανεπιστήμιο

Τμήμα Πληροφορικής & Τηλεπικοινωνιών

Εξάμηνο

Όλα τα εξάμηνα

Αναζήτηση Μαθήματος

Αναζήτηση...

Επιλογή	Όνομα Μαθήματος	Περιγραφή	Εξάμηνο	Τύπος	Κατηγορία	Ενέργειες
☒	Εισαγωγή στον Προγραμματισμό	Εισαγωγή στον Προγραμματισμό: Μια βασική εισαγωγή στις αρχές του προγραμματισμού και των αλγορίθμων.	1	Εργαστήριο	Κορμού	Αφαίρεση
☒	Γραμμική Άλγεβρα	Γραμμική Άλγεβρα: Εκμάθηση των βασικών εννοιών της γραμμικής άλγεβρας και των εφαρμογών της.	1	Εργαστήριο	Κορμού	Αφαίρεση
☒	Αρχιτεκτονική Υπολογιστών I	Αρχιτεκτονική Υπολογιστών I: Εξετάζει τη δομή και τη λειτουργία των σύγχρονων υπολογιστών.	2	Εργαστήριο/Θεωρία	Κορμού	Αφαίρεση
☐	Αντικειμενοστραφής Προγραμματισμός	Αντικειμενοστραφής Προγραμματισμός: Κατανόηση των αρχών του αντικειμενοστραφούς προγραμματισμού και της εφαρμογής τους.	3	Εργαστήριο	Κορμού	-
☒	Πιθανότητες και Στατιστική	Πιθανότητες και Στατιστική: Βασικές έννοιες των πιθανοτήτων και η εφαρμογή της στατιστικής στην πληροφορική.	3	Θεωρία	Κορμού	Αφαίρεση
☐	Δίκτυα Επικοινωνιών I	Δίκτυα Επικοινωνιών I: Εισαγωγή στα βασικά της επικοινωνίας δεδομένων και των δικτύων.	4	Εργαστήριο/Θεωρία	Κορμού	-
☒	Σχεδίαση και Χρήση Βάσεων Δεδομένων	Σχεδίαση και Χρήση Βάσεων Δεδομένων: Εκμάθηση των αρχών της σχεδίασης και της εφαρμογής βάσεων δεδομένων.	4	Θεωρία	Κορμού	Αφαίρεση
☐	Ψηφιακή Επεξεργασία Σήματος	Ψηφιακή Επεξεργασία Σήματος: Μελέτη των τεχνικών επεξεργασίας ψηφιακών σημάτων και των εφαρμογών τους.	5	Εργαστήριο/Θεωρία	Κορμού	-

Εικόνα 46 Εγγραφή σε μαθήματα

4.4 Συμπεράσματα

Το τέταρτο κεφάλαιο ανέδειξε την πλήρη λειτουργία της εφαρμογής, από την οπτική όλων των ρόλων. Η διεπαφή με το ημερολόγιο και τα φίλτρα κατά πανεπιστήμιο, πρόγραμμα, εξάμηνο, διδάσκοντα και αίθουσα επιτρέπει στους χρήστες να έχει άμεση και πολυδιάστατη εικόνα του ωρολογίου προγράμματος. Τα συμβάντα είναι διαδραστικά, με χρωματική σήμανση για συγκρούσεις και αναπληρώσεις και με αναδυόμενα παράθυρα που συγκεντρώνουν τις κρίσιμες λεπτομέρειες του μαθήματος. Η ροή εργασίας είναι συνεκτική και λιτή, ώστε ο χρήστης να βρίσκει γρήγορα ό,τι χρειάζεται χωρίς περιττά βήματα.

Οι ενότητες διαχείρισης τμήματος επιβεβαιώνουν την πληρότητα της πλατφόρμας για πραγματικές ακαδημαϊκές ανάγκες. Η γραμματεία προσθέτει και ενημερώνει αίθουσες, μαθήματα και διδάσκοντες, ορίζει χωρητικότητα και εξοπλισμό για εργαστήρια, αναθέτει πολλούς διδάσκοντες στο ίδιο μάθημα και υποστηρίζει περιπτώσεις συνδιδασκαλίας μεταξύ τμημάτων. Η δημιουργία προγραμμάτων με σαφή προσδιορισμό τύπου και περιόδου διασφαλίζει ορθό φιλτράρισμα, ιστορικότητα και επαναχρησιμοποίηση. Η διαχείριση χρηστών συγκεντρώνει ρόλους και συσχετίσεις με τα ιδρύματα, διευκολύνοντας τον έλεγχο πρόσβασης και την καθημερινή υποστήριξη.

Στην πλευρά των διδασκόντων, η δήλωση διαθεσιμότητας και οι αναπληρώσεις υλοποιούνται με άμεση ορατότητα του τρέχοντος προγράμματος και με αυτόματους ελέγχους που αποτρέπουν

υπερβάσεις ωρών, διπλές αναθέσεις και επικαλύψεις αιθουσών ή μαθημάτων κορμού. Η κατάσταση κάθε αιτήματος ενημερώνεται με σαφήνεια και οι ειδοποιήσεις φτάνουν στους εμπλεκόμενους χωρίς καθυστέρηση. Για τους φοιτητές, η εξατομίκευση του προσωπικού προγράμματος και η εκτύπωση σε μορφή PDF καθιστούν το σύστημα πρακτικό στην καθημερινότητα.

Συνολικά, το κεφάλαιο απέδειξε ότι ο σχεδιασμός του προηγούμενου μέρους μεταφράζεται σε λειτουργικό προϊόν με ξεκάθαρους ρόλους, προληπτικούς ελέγχους και διαφάνεια στην επικοινωνία. Η χρηστικότητα και η οργάνωση των ροών μειώνουν τον χρόνο διαχείρισης και αυξάνουν την αξιοπιστία του τελικού ωρολογίου. Στο επόμενο κεφάλαιο ακολουθούν τα συνολικά συμπεράσματα, η ανάλυση SWOT και τα μετρικά της υλοποίησης, μαζί με προτάσεις για μελλοντικές βελτιώσεις.

Κεφάλαιο 5: Συμπεράσματα και προτάσεις

Στο κεφάλαιο αυτό παρουσιάζονται τα συμπεράσματα που προέκυψαν από την εκπόνηση της παρούσας διπλωματικής εργασίας, καθώς και προτάσεις για την περαιτέρω εξέλιξη του συστήματος. Η εργασία είχε ως αντικείμενο τον σχεδιασμό και την ανάπτυξη του **TMS (Timetable Management System)**, ενός πληροφοριακού συστήματος διαχείρισης ωρολογίου προγράμματος, ειδικά προσαρμοσμένου στις ανάγκες πανεπιστημιακού περιβάλλοντος.

Η ανάπτυξη του έργου ακολούθησε μια σαφή μεθοδολογική πορεία, η οποία αποτυπώθηκε στα προηγούμενα κεφάλαια:

- Στο **Κεφάλαιο 2** παρουσιάστηκαν οι γλώσσες προγραμματισμού, οι βιβλιοθήκες και τα εργαλεία που χρησιμοποιήθηκαν κατά την ανάπτυξη.
- Στο **Κεφάλαιο 3** αναλύθηκαν οι απαιτήσεις του συστήματος, έγινε η περιγραφή του μοντέλου της βάσης δεδομένων και αποδόθηκαν τα διαγράμματα περιπτώσεων χρήσης, καθώς και οι τεχνικές ασφάλειας.
- Στο **Κεφάλαιο 4** δόθηκε έμφαση στη διεπαφή χρήστη, με αναλυτική παρουσίαση οθονών και λειτουργιών, διακριτά για κάθε κατηγορία χρήστη.

Μέσα από αυτή τη διαδικασία υλοποιήθηκε ένα σύστημα που βελτιώνει τη διαδικασία κατάρτισης και διαχείρισης του ωρολογίου προγράμματος, ενσωματώνοντας αυτοματοποιημένους ελέγχους εγκυρότητας και προσφέροντας εξατομικευμένες δυνατότητες πρόσβασης σε φοιτητές, καθηγητές και γραμματεία. Παράλληλα, αναδεικνύονται τα πλεονεκτήματα αλλά και οι περιορισμοί της προτεινόμενης προσέγγισης.

Για την αποτύπωση αυτών των στοιχείων ακολουθεί στην **ενότητα 5.1** ανάλυση SWOT, η οποία συνοψίζει σε τέσσερις κατηγορίες τα δυνατά σημεία, τις αδυναμίες, τις ευκαιρίες και τις απειλές του συστήματος. Στην συνέχεια, στην **ενότητα 5.2** παρουσιάζονται τα προβλήματα που παρουσιάστηκαν κατά την υλοποίηση. Τέλος, στην **ενότητα 5.3** παρατίθενται οι μετρικές ανάπτυξης του έργου, ώστε να δοθεί μια ποσοτική εικόνα του μεγέθους και του φόρτου υλοποίησης.

5.1 Ανάλυση S.W.O.T

Η ανάλυση SWOT αποτελεί ένα εργαλείο στρατηγικής αξιολόγησης που βοηθά στον εντοπισμό των Δυνατών Σημείων (Strengths), των Αδυναμιών (Weaknesses), των Ευκαιριών (Opportunities) και των Απειλών (Threats) ενός έργου ή συστήματος. Στην παρούσα ενότητα επιχειρείται μια αποτύπωση των βασικών στοιχείων που χαρακτηρίζουν το πληροφοριακό σύστημα TMS και πως αυτά μπορούν να βελτιωθούν και να επεκταθούν πέρα από τα πλαίσια αυτής της εργασίας.

Strengths (Δυνατά Σημεία)

- Ειδικά σχεδιασμένο για πανεπιστημιακά περιβάλλοντα, καλύπτοντας τις ανάγκες γραμματείας, καθηγητών και φοιτητών.
- Υποστήριξη διαφοροποιημένων ρόλων χρηστών με προσαρμοσμένες δυνατότητες.
- Αυτοματοποιημένοι έλεγχοι που αποτρέπουν σφάλματα κατά την καταχώριση προγράμματος (συγκρούσεις μαθημάτων, διπλές κρατήσεις αιθουσών, επικαλύψεις διδασκόντων).
- Ευκολία χρήσης με φιλικό περιβάλλον διεπαφής και απλές διαδικασίες εισαγωγής/τροποποίησης.
- Εξατομικευμένη προβολή προγράμματος για φοιτητές και καθηγητές και πολλαπλά φίλτρα που παρέχει στον χρήστη πολλαπλές επιλογές για την θέαση και εκτύπωση του προγράμματος.
- Δυνατότητα εκτύπωσης σε μορφή pdf μέσα από το πρόγραμμα για ευκολία του χρήστη.

Weaknesses (Αδυναμίες)

- Απουσία αλγοριθμικής ή αυτόματης δημιουργίας ωρολογίου προγράμματος με βάση προτεραιότητες ή βελτιστοποίηση.
- Σε περιπτώσεις όπου ένα εργαστηριακό μάθημα χωρίζεται σε τμήματα, δεν υπάρχει τρόπος κατά τη δήλωση διαθεσιμότητας από τη πλευρά του καθηγητή να δηλώσει τις ώρες στο πρόγραμμα για κάθε τμήμα ξεχωριστά. Δηλαδή, αν ένα εργαστήριο δύο ωρών διεξαχθεί σε τρία τμήματα, θα πρέπει η γραμματεία να το λάβει υπόψιν και να ενημερώσει τις ώρες διδασκαλίας εργαστηρίου για τον καθηγητή ώστε να έχει έξι ώρες διαθέσιμες αντί για δύο.
- Έλλειψη λειτουργίας αποστολής ενημερώσεων μέσω email: Οι χρήστες ενημερώνονται μόνο εντός του συστήματος μέσω ειδοποιήσεων, στις οποίες έχουν πρόσβαση όταν πατήσουν το κουμπί με το καμπανάκι στο δεξί μέρος της μπάρας πλοήγησής τους.
- Περιορισμένη διασύνδεση με άλλα πληροφοριακά συστήματα του πανεπιστημίου (π.χ. σύστημα δηλώσεων μαθημάτων, πλατφόρμα e-class).
- Έλλειψη διασύνδεσης με την ακαδημαϊκή βάση χρηστών μέσω SSO(Single Sign-On).

Opportunities (Ευκαιρίες)

- Δυνατότητα μελλοντικής ενσωμάτωσης αλγορίθμων αυτόματης βελτιστοποίησης του προγράμματος με βάση σύνθετα κριτήρια.
- Σύνδεση με υπάρχουσες πανεπιστημιακές πλατφόρμες ώστε να υπάρχει ενιαίο περιβάλλον πληροφόρησης.
- Δυνατότητα διαχωρισμού των εργαστηριακών μαθημάτων σε επιμέρους τμήματα, προκειμένου να επιτυγχάνεται η πιο ρεαλιστική προσομοίωση των πραγματικών

συνθηκών. Ιδιαίτερα στα μαθήματα εργαστηρίου που απαιτούν εξειδικευμένες αίθουσες με περιορισμένο αριθμό θέσεων ή εξοπλισμού, η λειτουργία αυτή καθίσταται απαραίτητη ώστε να αποφεύγεται ο υπερπληθυσμός και να διασφαλίζεται η ομαλή διεξαγωγή του εργαστηρίου.

- Ανάπτυξη mobile εφαρμογής που θα επιτρέπει σε φοιτητές και καθηγητές να έχουν πρόσβαση στο πρόγραμμα και στις ειδοποιήσεις σε πραγματικό χρόνο από το κινητό τους.
- Δυνατότητα υιοθέτησης του συστήματος και από άλλα τμήματα ή ιδρύματα, συμβάλλοντας στη διάδοση και περαιτέρω εξέλιξή του.
- Σύνδεση με την βάση χρηστών της σχολής μέσω SSO για την άμεση χρήση με τους ιδρυματικούς λογαριασμούς από τους χρήστες της εφαρμογής.

Threats (Απειλές)

- Εξάρτηση από την τεχνική υποστήριξη και συντήρηση του συστήματος: τυχόν έλλειψη πόρων μπορεί να επηρεάσει τη λειτουργικότητα.
- Πιθανές αντιστάσεις στην υιοθέτηση του συστήματος από μέλη ΔΕΠ ή γραμματείες που έχουν συνηθίσει σε χειροκίνητες μεθόδους.
- Ανταγωνισμός από εμπορικά ή διεθνώς διαδεδομένα εργαλεία διαχείρισης ωρολογίων που διαθέτουν περισσότερες λειτουργίες.

5.2 Δυσκολίες και προβλήματα κατά την υλοποίηση

Τα βασικά προβλήματα και οι δυσκολίες που εμφανίστηκαν έχουν να κάνουν με περιορισμούς που παρουσίασαν κάποιες βιβλιοθήκες που χρησιμοποιούνται για το σύστημα. Αναλυτικότερα:

- Η βιβλιοθήκη «@FullCalendar» παρόλο που είναι από τις πιο πλήρεις και δημοφιλείς βιβλιοθήκες για τη σύνταξη σύνθετων προγραμμάτων, δεν κατέχει τη δυνατότητα διάσπασης γεγονότων (event-splitting) από μόνη της. Δηλαδή, σε περίπτωση που δύο ή περισσότερα μαθήματα συμπίπτουν για μία ή ακόμη και παραπάνω ώρες στο πρόγραμμα μιας μέρας, δεν υπάρχει τρόπος διαχωρισμού των κελιών.
Κατά συνέπεια, χρειάστηκε να δημιουργηθεί λογική που ελέγχει για τυχόν συγκρούσεις και διαχωρίζει τα συμβάντα δυναμικά για όσες χρονοθυρίδες χρειαστεί. Ύστερα από έρευνα για κάποια επιλογή βιβλιοθήκης που να παρέχει αυτή τη δυνατότητα, δεν βρέθηκε κάποια καλύτερη ή βολική λύση για αυτό το θέμα.
- Η βιβλιοθήκη «dompdf» που χρησιμοποιείται για την εκτύπωση του προγράμματος δεν υποστηρίζει την εκτύπωση σύνθετων στοιχείων CSS και HTML, γεγονός που κατέστησε απαραίτητη τη δυναμική δημιουργία με απλά στοιχεία HTML και χωρίς τη χρήση ιδιαίτερων στοιχείων CSS. Επίσης δεν υπάρχει η δυνατότητα χρήσης εξωτερικών αρχείων CSS για τον πίνακα που δημιουργεί το πρόγραμμα-προς εκτύπωση.

5.3 Μετρικά συστήματος

Οι μετρικές του λογισμικού αποτελούν κρίσιμο στοιχείο για την ποσοτική αξιολόγηση της υλοποίησης. Για τον σκοπό αυτόν, χρησιμοποιήθηκε το εργαλείο **CLOC (Count Lines of Code)**[23], ένα ευρέως αναγνωρισμένο εργαλείο ανοικτού κώδικα που παρέχει ακριβείς μετρήσεις γραμμών κώδικα, σχολίων και κενών γραμμών. Η χρήση του επέτρεψε την αποτύπωση αντικειμενικών δεικτών για το μέγεθος και την πολυπλοκότητα του αναπτυγμένου κώδικα. Τα αποτελέσματα αυτών των μετρήσεων παρουσιάζονται αναλυτικά στον Πίνακα 24. Αξίζει να σημειωθεί, ότι αυτά είναι τα μετρικά της εργασίας χωρίς να συμπεριληφθούν οι φάκελοι “vendor” και “node_modules” που περιέχουν τις βιβλιοθήκες και όλες τις εξαρτήσεις τους.

Πίνακας 24 Μετρικά αρχείων κώδικα -- χωρίς βιβλιοθήκες

Language	Files	Blank Lines	Comments	Code
PHP	57	984	514	7,875
JavaScript	10	186	111	1,983
SQL	1	170	330	655
CSS	1	51	31	378
JSON	2	1	0	79
Text	1	0	0	6
SUM:	72	1,392	986	10,976

Τα μετρικά μαζί με τις βιβλιοθήκες και τις εξαρτήσεις τους παρουσιάζονται στον Πίνακα 25. Τα περισσότερα αρχεία αποτελούν εξαρτήσεις από τις βιβλιοθήκες που χρησιμοποιήθηκαν με το Node.js και το Composer.

Πίνακας 25 Μετρικά αρχείων κώδικα μαζί με τις βιβλιοθήκες

Language	Files	Blank Lines	Comments	Code
PHP	1556	23223	64775	147005
JavaScript	483	2706	6402	69885
JSON	55	1	0	27384
TypeScript	112	663	686	8595
Text	18	1001	0	4330
Markdown	47	1311	1	3819
HTML	60	72	0	3278
XSD	3	91	48	2842
SVG	4	1	1	1890
CSS	3	143	66	790
SQL	1	170	330	655
YAML	4	39	0	118
XML	4	8	3	79
INI	1	3	2	10
DOS Batch	2	0	0	9
SUM:	2353	29432	72314	270689

Όπως φαίνεται από τους δύο πίνακες, οι βιβλιοθήκες που χρησιμοποιούνται, εισάγουν αρκετά αρχεία χωρίς όμως να μειώνουν σημαντικά την απόδοση της εφαρμογής και το όφελος που προσφέρουν στην ταχύτητα υλοποίησης δικαιολογεί την ύπαρξή τους στον πηγαίο κώδικα της εργασίας.

5.4 Συμπεράσματα

Το κεφάλαιο αυτό συνθέτει τα βασικά ευρήματα της εργασίας και αποτιμά την ωριμότητα του TMS σε σχέση με τους στόχους που τέθηκαν. Η ανάπτυξη του συστήματος απέδειξε ότι είναι εφικτή μια ολοκληρωμένη, πρακτική λύση για τη διαχείριση του ωρολογίου προγράμματος σε πανεπιστημιακό περιβάλλον. Η αρχιτεκτονική με διακριτούς ρόλους, οι προληπτικοί έλεγχοι εγκυρότητας κατά την καταχώριση και η διαδραστική διεπαφή συγκροτούν ένα προϊόν που μειώνει τα λάθη, επιταχύνει τη ροή εργασιών και προσφέρει σαφή ορατότητα σε γραμματεία, διδάσκοντες και φοιτητές.

Η ανάλυση SWOT κατέδειξε ότι τα δυνατά σημεία του TMS εδράζονται στην ευθυγράμμιση με τις καθημερινές ανάγκες ενός τμήματος και στη χρηστικότητα της διεπαφής. Η ύπαρξη εξατομικευμένων προβολών και πλούσιων φίλτρων επιταχύνει τη λήψη αποφάσεων και διευκολύνει την οργάνωση σε επίπεδο εξαμήνου και εξεταστικής περιόδου. Παράλληλα, αναγνωρίστηκαν σαφείς περιορισμοί που οριοθετούν το παρόν στάδιο ωρίμανσης. Η απουσία αυτόματης δημιουργίας προγράμματος με αλγοριθμικούς τρόπους, η έλλειψη ξεχωριστής διαχείρισης υπομημάτων εργαστηρίων μέσα από τις δηλώσεις των διδασκόντων και η μη ενσωμάτωση αποστολής ενημερώσεων μέσω ηλεκτρονικού ταχυδρομείου συγκροτούν τεχνικά κενά τα οποία ορίζουν τον άξονα των επόμενων βημάτων. Οι ευκαιρίες επανασχεδιασμού και επέκτασης είναι σαφείς. Προβλέπονται ενισχυμένες διασυνδέσεις με ιδρυματικές πλατφόρμες, υιοθέτηση ενιαίας ταυτοποίησης, ανάπτυξη κινητής εφαρμογής και διεύρυνση των μονάδων παραμετροποίησης, ώστε το σύστημα να προσαρμόζεται ευχερέστερα σε διαφορετικά ακαδημαϊκά περιβάλλοντα. Οι απειλές αφορούν κυρίως τον ανταγωνισμό από ώριμες λύσεις και την ανάγκη σταθερής τεχνικής υποστήριξης, καθώς και πιθανές αντιστάσεις υιοθέτησης από έμπειρους χρήστες που έχουν εθιστεί παλαιότερους τρόπους εργασίας.

Οι δυσκολίες που προέκυψαν κατά την υλοποίηση αποτέλεσαν χρήσιμα μαθήματα σχεδιασμού. Η ανάγκη για δυναμικό διαχωρισμό επικαλυπτόμενων συμβάντων στο ημερολόγιο αντιμετωπίστηκε με εξειδικευμένη λογική, ώστε να παρουσιαστεί καθαρή εικόνα σε περιπτώσεις αυξημένων συγκρούσεων. Η παραγωγή εκτυπώσιμης μορφής του προγράμματος οδήγησε σε πιο λιτή απόδοση του περιεχομένου, με στόχο την αξιοπιστία και την συμβατότητα. Και στις δύο περιπτώσεις, οι τεχνικοί συμβιβασμοί έγιναν με γνώμονα την προβλεψιμότητα της λειτουργίας και την απλότητα για τον τελικό χρήστη.

Τα μετρικά ανάπτυξης που συγκεντρώθηκαν παρέχουν μια ποσοτική αποτύπωση του φορτίου υλοποίησης και των επιλογών αρχιτεκτονικής. Η διάκριση μεταξύ του βασικού κώδικα της εφαρμογής και των βιβλιοθηκών υπογραμμίζει αφενός το μέγεθος της συνεισφοράς που αφορά την επιχειρησιακή λογική και την ασφάλεια, αφετέρου το εύρος των εξαρτήσεων που αξιοποιήθηκαν για να επιταχυνθεί η παράδοση μιας σταθερής λύσης. Η τεκμηρίωση και οι οδηγίες στο παράρτημα συμπληρώνουν το αποτύπωμα αυτό προσφέροντας δυνατότητα αναπαραγωγής και ευκολότερη συντήρησης.

Με βάση τα παραπάνω, το TMS μπορεί να χαρακτηριστεί πρωτότυπο προϊόν που καλύπτει τον πυρήνα της ανάγκης για οργάνωση ωρολογίου σε πανεπιστημιακό περιβάλλον και δημιουργεί σταθερό θεμέλιο για εξέλιξη. Τα επόμενα βήματα είναι σαφή. Πρώτον, ενίσχυση της λειτουργικότητας με αλγοριθμική υποβοήθηση στον προγραμματισμό και με πλήρη υποστήριξη εργαστηριακών υπομημάτων ως έννοιες πρώτης κλάσης. Δεύτερον, ολοκλήρωση με ιδρυματικά

συστήματα και υιοθέτηση ενιαίας ταυτοποίησης, ώστε η χρήση να γίνει απρόσκοπτη για όλα τα μέλη. Τρίτον, βελτιώσεις στην επικοινωνία με αποστολή ενημερώσεων εκτός συστήματος και με αναλυτικό ιστορικό ενεργειών που ενισχύει τη διαφάνεια. Τέταρτον, συνεχής έλεγχος απόδοσης και χρηστικότητας με βάση πραγματικά σενάρια χρήσης, με στόχο την ακόμη πιο γρήγορη και σταθερή εμπειρία.

Συνοψίζοντας, η εργασία παρέδωσε ένα λειτουργικό, επεκτάσιμο και ασφαλές πληροφορικό σύστημα, αποτύπωσε με σαφήνεια τα βήματα που απαιτούνται για την περαιτέρω ωρίμανσή του και τεκμηρίωσε τις σχεδιαστικές επιλογές που το καθιστούν κατάλληλο για υιοθέτηση από ακαδημαϊκά ιδρύματα. Το επόμενο διάστημα μπορεί να αφιερωθεί στην πιλοτική εφαρμογή σε πραγματικό περιβάλλον, στη συλλογή ανατροφοδότησης και στην σταδιακή ενσωμάτωση των προτεινόμενων επεκτάσεων, ώστε το TMS να εξελιχθεί σε σημείο αναφοράς για τη διαχείριση ωρολογίου προγράμματος στην τριτοβάθμια εκπαίδευση.

Παράρτημα Α – Οδηγοί Εγκατάστασης Προγραμματιστικών Εργαλείων

Α.1 Εγκατάσταση ΧΑΜΡΡ

Παρακάτω ακολουθεί λεπτομερής οδηγός εγκατάστασης και βασικής ρύθμισης του ΧΑΜΡΡ στα Windows, μαζί με την παρουσίαση εναλλακτικών τρόπων επίλυσης τυχόν συγκρούσεων θυρών.

Περιβάλλον εγκατάστασης:

1. **Λειτουργικό Σύστημα:** Windows 10 - 64 bit.
2. **Microsoft Visual C++ Redistributable 2017:**

Η έκδοση των Windows απαιτεί το Visual C++ 2017 Redistributable για να τρέξει σωστά ο Apache και η PHP

3. **Δικαιώματα Διαχειριστή:** Κατά την εγκατάσταση μπορεί να χρειαστεί άδεια διαχειριστή για εγγραφή στον φάκελο C:\xampp.

4. **Προσθήκη της PHP στο PATH (συναρτήσεις περιβάλλοντος):**

Θα γίνει ειδική αναφορά στην ενότητα «troubleshooting» του ΧΑΜΡΡ.

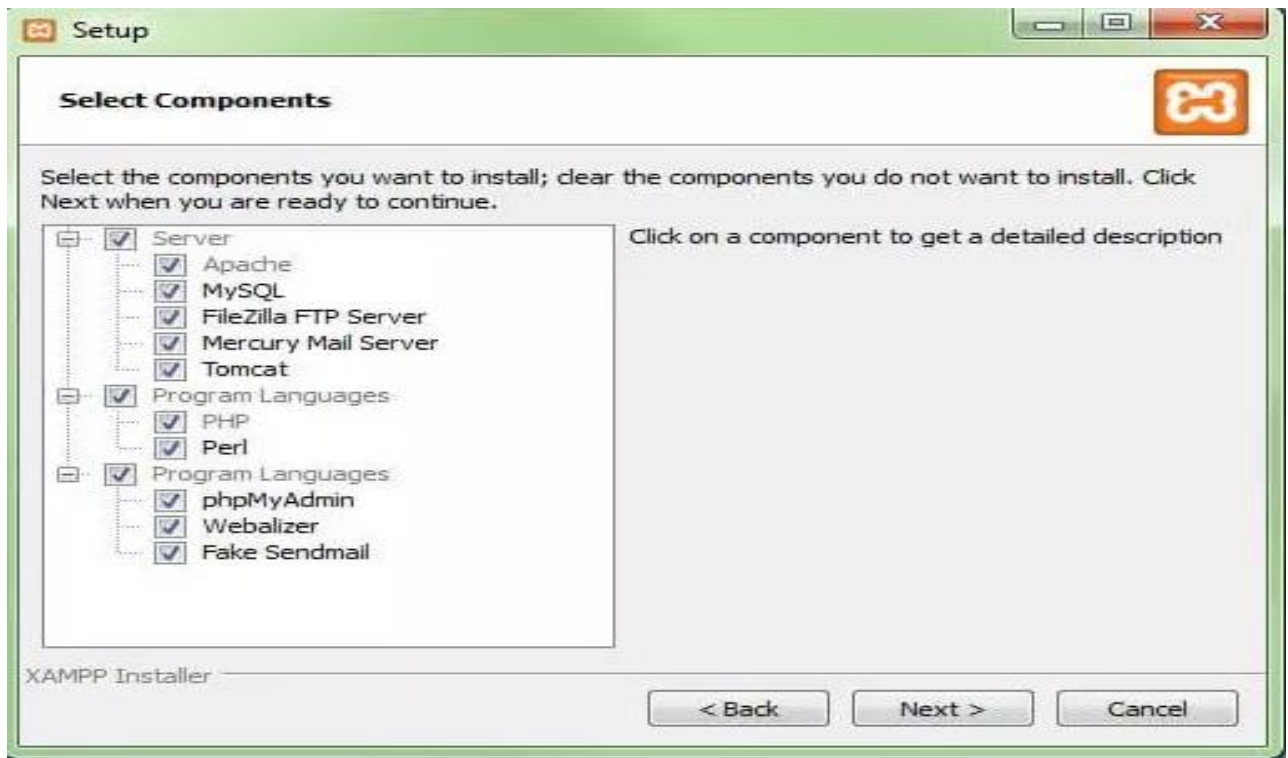
Χρησιμοποιήθηκε ο installer που παρέχεται από το apachefriends.org. Πριν την εκκίνηση του οδηγού εγκατάστασης μπορεί να ζητηθεί στον χρήστη να απενεργοποιήσει το UAC (User Account Control) όπως εμφανίζεται στην Εικόνα 47. Δεν συνέβη στη δική μου εγκατάσταση, αλλά δεν είναι επιβλαβές για τον υπολογιστή του χρήστη.



Εικόνα 47 Απενεργοποίηση UAC πριν την εγκατάσταση

Πατώντας “next” ο χρήστης θα καταφθάσει στην οθόνη επιλογής των πακέτων του ΧΑΜΡΡ όπως στην Εικόνα 48 ώστε ο χρήστης να εγκαταστήσει τα features που επιθυμεί. Για την εργασία απαιτούνται και χρησιμοποιούνται μόνο τα:

- Apache
- PHP
- phpMyAdmin, MySQL



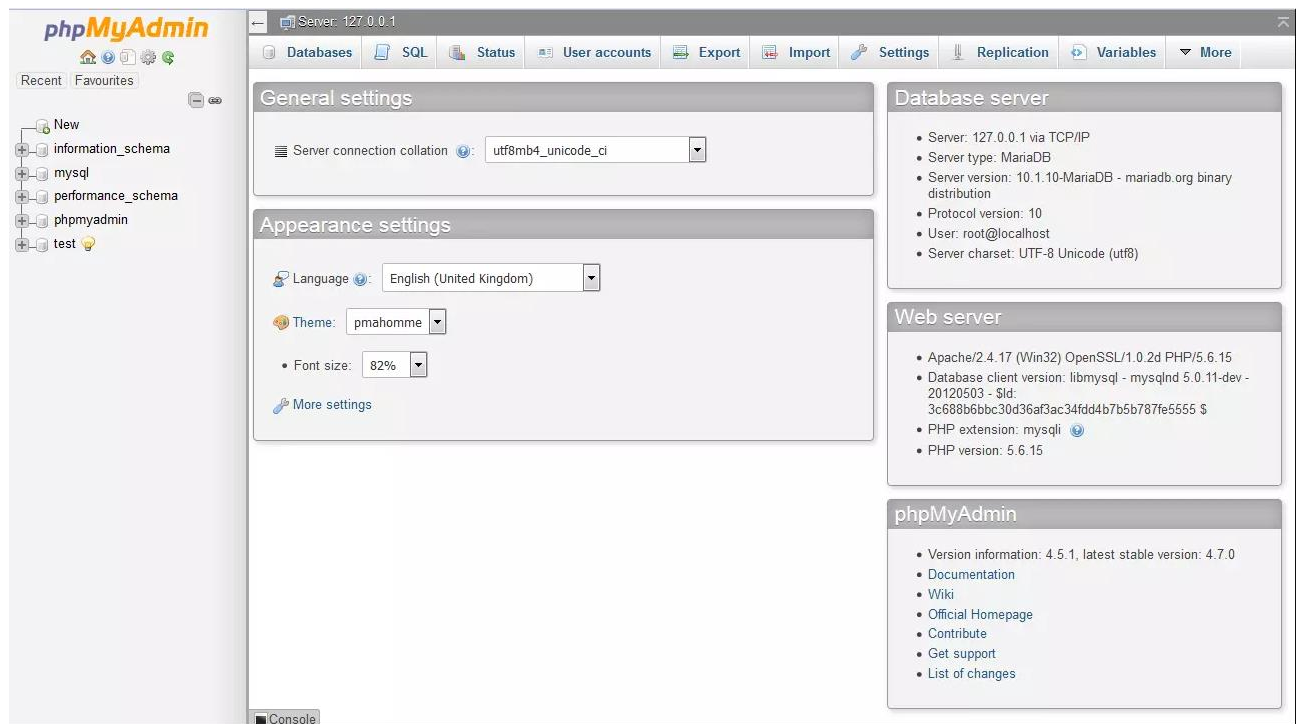
Εικόνα 48 Οθόνη επιλογής features του XAMPP

Αμέσως μετά επιλέγουμε τον φάκελο εγκατάστασης και η διαδικασία εγκατάστασης μπορεί να ξεκινήσει. Με την ολοκλήρωση της εγκατάστασης ο χρήστης μπορεί να ανοίξει το XAMPP και θα βρεθεί στο control panel του προγράμματος που φαίνεται στην Εικόνα 49.



Εικόνα 49 XAMPP control panel

Αν η διαδικασία ολοκληρωθεί με επιτυχία, ο χρήστης μπορεί από το panel να ξεκινήσει τον Apache server και την MySQL. Ο πηγαίος φάκελος του διακομιστή Apache λέγεται «htdocs» και βρίσκεται στον φάκελο του XAMPP. Για να έχει πρόσβαση τοπικά ο χρήστης στο phpMyAdmin που φαίνεται στην Εικόνα 50, αρκεί να γράψει στον browser της επιλογής του, τη διεύθυνση « localhost/phpmyadmin/». Επίσης, με την ίδια ακριβώς λογική μπορεί να έχει πρόσβαση στα αρχεία της εφαρμογής του παραθέτοντας το τοπικό μονοπάτι στον φάκελο της εφαρμογής (π.χ localhost/myApp/).[24]



Εικόνα 50 phpMyAdmin

A.2 Εγκατάσταση Visual studio Code και επεκτάσεων

Περιβάλλον εγκατάστασης:

1. **Λειτουργικό Σύστημα:** Windows 10 - 64 bit.
2. **Σύνδεση στο Διαδίκτυο:** Για τη λήψη των αρχείων εγκατάστασης και των αυτόματων ενημερώσεων.

Εγκατάσταση:

- Ακολουθείτε το [σύνδεσμο εγκατάστασης Visual Studio Code](#).
- Επιλέξτε την έκδοση **Windows** → **User Installer (.msi)** για εύκολη εγκατάσταση χωρίς δικαιώματα συστήματος ή **System Installer (.msi)** για εγκατάσταση που αφορά όλους τους χρήστες του υπολογιστή.
 - Ο Setup Wizard της εφαρμογής είναι απλός και η μόνη επιλογή που χρειάζεται να κάνει ο χρήστης είναι να επιλέξει ως φάκελο εγκατάστασης είτε τον **C:\Program Files\Microsoft VS Code** ή **C:\Users\<Username>\AppData\Local\Programs\Microsoft VS Code** που είναι και η προεπιλογή.

A.3 Εγκατάσταση διαχειριστή εξαρτήσεων Composer

Περιβάλλον εγκατάστασης:

1. **Λειτουργικό Σύστημα:** Windows 10 - 64 bit.
2. **Σύνδεση στο Διαδίκτυο:** Για τη λήψη των αρχείων εγκατάστασης και των αυτόματων ενημερώσεων.
3. **PHP 7.2+:** Πρέπει να υπάρχει εγκατεστημένη και διαθέσιμη στο PATH η php.exe. Ελέγξτε με εντολή στο terminal «php -v»
4. **Δικαιώματα διαχειριστή στο σύστημα φακέλων Program Files:** Για τη ρύθμιση μεταβλητών περιβάλλοντος.
5. **Visual Studio Code:** Για τη διαχείριση μέσω integrated terminal.

Βήματα Εγκατάστασης:

1. Κατεβάστε τον Installer από το επίσημο [Composer Download Page](#).
2. Εκτελούμε το Composer-Setup.exe και ακολουθούμε το οδηγό εγκατάστασης.
3. Ο installer εντοπίζει αυτόματα το μονοπάτι της PHP, αλλά σε περίπτωση που δεν γίνει, το μονοπάτι βρίσκεται στον φάκελο του XAMPP που περιέχει την PHP.
4. **Σημαντικό:** Πολλά από τα λάθη που μπορεί να προκύψουν μπορεί να έχουν να κάνουν με τη μη πρόσθεση του Composer στην «\$PATH» μεταβλητή περιβάλλοντος των Windows.

Βασικές εντολές διαχείρισης Composer:

Στο τερματικό του visual studio, αφού μπούμε στον φάκελο που βρίσκεται το project μας, οι παρακάτω εντολές που περιλαμβάνονται στον Πίνακα 26 είναι αυτές που θα χρησιμοποιηθούν κατά κόρο για τη διαχείριση των εξαρτήσεων στο project.[25]

Πίνακας 26 Εντολές Composer

Εντολή Composer	Περιγραφή
composer init	Δημιουργεί το αρχείο composer.json με έναν βήμα-βήμα οδηγό για όνομα, περιγραφή, εξαρτήσεις κ.λπ.
composer require <vendor/package>	Προσθέτει και εγκαθιστά το πακέτο <vendor/package>, ενημερώνοντας αυτόματα composer.json και composer.lock.
composer install	Εγκαθιστά όλες τις εξαρτήσεις όπως ορίζονται στο composer.lock, εξασφαλίζοντας αναπαραγωγιμότητα των εκδόσεων.
composer update	Ενημερώνει όλες τις εξαρτήσεις σύμφωνα με τις εκδόσεις που καθορίζονται στο composer.json και γράφει τα νέα κλειδωμένα δεδομένα.
composer remove <vendor/package>	Αφαιρεί το πακέτο <vendor/package> από το composer.json και διαγράφει τον αντίστοιχο φάκελο από το vendor/.
composer dump-autoload	Αναδημιουργεί το autoloader (π.χ. autoload.php, class maps) μετά από αλλαγές στο namespace ή νέους φακέλους στον κώδικα.
composer self-update	Αναβαθμίζει το ίδιο το Composer στην πιο πρόσφατη διαθέσιμη έκδοση LTS.
composer show --installed	Εμφανίζει λίστα με όλα τα πακέτα που έχουν ήδη εγκατασταθεί στο τρέχον έργο.
composer outdated	Δείχνει ποια από τα εγκατεστημένα πακέτα έχουν διαθέσιμες νεότερες εκδόσεις.
composer licenses	Εμφανίζει τις άδειες (licenses) όλων των εγκατεστημένων πακέτων.
composer audit	Ελέγχει τις εγκατεστημένες εξαρτήσεις για γνωστά ευάλωτα σημεία ασφαλείας.

A.4 Εγκατάσταση περιβάλλοντος διαχείρισης πακέτων Node.js

Περιβάλλον εγκατάστασης:

- **Λειτουργικό Σύστημα:** Windows 10 - 64 bit.

- **Σύνδεση στο Διαδίκτυο:** Για τη λήψη των αρχείων εγκατάστασης και των αυτόματων ενημερώσεων.

Βήματα Εγκατάστασης:

1. Ανοίξτε τον browser και πλοηγηθείτε στην επίσημη σελίδα του Node.js: [Node.js official page](https://nodejs.org/en)
2. Επιλέξτε το κουμπί εγκατάστασης και διαλέξτε την τελευταία Long Term Support(LTS) εκδοχή και κατεβάστε τον Installer.
3. Εμφανίζεται ο οδηγός εγκατάστασης (Setup Wizard). Πατήστε Next >.
4. Επιλέξτε τα στοιχεία που θέλετε να εγκαταστήσετε (προεπιλογή: Node.js runtime, npm package manager, Online documentation shortcuts) και πατήστε Next >.
5. Πατήστε Εγκατάσταση.

Για έλεγχο εγκατάστασης του Node και του npm package manager μπορείτε στο τερματικό να γράψετε:

- `node -v`
- `npm -v`

Εάν οι παραπάνω εντολές δεν αναγνωρίζονται, πιθανώς ο φάκελος εγκατάστασης του Node.js δεν έχει προστεθεί αυτόματα στο PATH.

- **Επίλυση:**
 1. Κάντε δεξί κλικ στο εικονίδιο των Windows (Start) → **System** → **Advanced system settings** → καρτέλα **Environment Variables**.
 2. Κάτω από τις **System variables**, επιλέξτε Path και πατήστε **Edit**.
 3. Πατήστε **New** και προσθέστε τη διαδρομή: `C:\Program Files\nodejs\`
 4. Πατήστε **OK** μέχρι να κλείσουν όλα τα παράθυρα ρυθμίσεων.
 5. Ανοίξτε ξανά το τερματικό και ελέγξτε με `node -v` και `npm -v`.

A4.1 Δημιουργία & Διαχείριση Έργου Node.js στο Visual Studio Code

- Αφού επιλέξτε το project σας και ανοίξετε τον φάκελό του στο Visual Studio Code, ανοίγετε το τερματικό και εκτελείτε την `npm init`.
- Θα σας ζητηθεί να συμπληρώσετε με prompts στο τερματικό κάποια στοιχεία που αφορούν το project σας όπως name, version, git repository και άλλα. Μπορείτε επίσης για γρήγορη δημιουργία να χρησιμοποιήσετε και το όρισμα `npm init -y`.
- Η εκτέλεση της εντολής δημιουργεί ένα αρχείο που ονομάζεται «package.json». Μπορεί ο χρήστης να προσθέσει αν θέλει αντικείμενο json “scripts” που μπορεί να ορίσει τεστ περιβάλλον, development περιβάλλον και αρχικό σημείο εκτέλεσης όπως φαίνεται παρακάτω:

```
"scripts": {
  "start": "node index.js",
  "dev": "nodemon index.js",
  "test": "echo \"No tests specified\" && exit 1"
},
```

- Πρέπει να ακολουθήσει η εγκατάσταση του περιβάλλοντος npm με την εντολή: `npm install --save-dev nodemon`
- Για εκκίνηση του package manager: `npm start`. Αν όλα πήγαν ορθά, θα δείτε μήνυμα π.χ «Server listening on port 3000»

Βιβλιογραφία

- [1] 'Πρόβλημα P=NP', *Βικιπαίδεια*. 15 Ιούλιος 2023. Ημερομηνία πρόσβασης: 8 Σεπτέμβριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://el.wikipedia.org/w/index.php?title=%CE%A0%CF%81%CF%8C%CE%B2%CE%BB%CE%B7%CE%BC%CE%B1_P%3DNP&oldid=10137864
- [2] 'FET - Free Timetabling Software'. Ημερομηνία πρόσβασης: 24 Αύγουστος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://lalescu.ro/liviu/fet/>
- [3] 'UniTime | University Timetabling'. Ημερομηνία πρόσβασης: 24 Αύγουστος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://www.unitime.org/>
- [4] 'Home', Untis. Ημερομηνία πρόσβασης: 24 Αύγουστος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://www.untis.at/en>
- [5] George Papakis, 'DESIGN & IMPLEMENTATION OF AN INFORMATION SYSTEM OF A TIMETABLE MANAGEMENT PROGRAM'.
- [6] 'HTML: HyperText Markup Language | MDN'. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [7] 'Introduction to HTML'. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.w3schools.com/html/html_intro.asp
- [8] 'CSS: Cascading Style Sheets | MDN'. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [9] 'CSS Selectors'. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.w3schools.com/css/css_selectors.asp
- [10] 'PHP Introduction'. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.w3schools.com/php/php_intro.asp
- [11] 'Relational database', *Wikipedia*. 31 Μάιος 2025. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://en.wikipedia.org/w/index.php?title=Relational_database&oldid=1293220490
- [12] 'Database normalization', *Wikipedia*. 14 Μάιος 2025. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://en.wikipedia.org/w/index.php?title=Database_normalization&oldid=1290441274
- [13] 'What is AJAX'. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://www.w3schools.com/whatis/whatis_ajax.asp
- [14] 'PHP Manual: PDO Functions'. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://www.nusphere.com/kb/phpmanual/ref.pdo.htm>
- [15] A. C. Codex, 'A Comprehensive Guide to the PHP PDO Library for Database Access | Reintech media'. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://reintech.io/blog/a-comprehensive-guide-to-php-pdo-library-for-database-access>
- [16] 'PHP: Transactions and auto-commit - Manual'. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://www.php.net/manual/en/pdo.transactions.php>
- [17] 'Node.js', *Wikipedia*. 22 Ιούνιος 2025. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://en.wikipedia.org/w/index.php?title=Node.js&oldid=1296888691>
- [18] 'Basic usage - Composer'. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε

ψηφιακή μορφή]. Διαθέσιμο στο: <https://getcomposer.org/doc/01-basic-usage.md>

[19] D. C. Science, 'Understanding Composer Autoloading in PHP', *dailycomputerscience*. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://www.dailycomputerscience.com/post/understanding-composer-autoloading-in-php>

[20] Adityasinghwhit, 'How to run your project using xampp', *Medium*. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://medium.com/@adityasinghw84it/how-to-run-your-project-using-xampp-3a1c16a1fe0f>

[21] 'bcrypt', *Wikipedia*. 5 Ιούλιος 2025. Ημερομηνία πρόσβασης: 1 Σεπτέμβριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://en.wikipedia.org/w/index.php?title=Bcrypt&oldid=1298892562>

[22] 'Cross-site scripting', *Wikipedia*. 15 Αύγουστος 2025. Ημερομηνία πρόσβασης: 1 Σεπτέμβριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: https://en.wikipedia.org/w/index.php?title=Cross-site_scripting&oldid=1306074893#Safely_validating_untrusted_HTML_input

[23] 'CLOC - Count number of lines of code in file', *GeeksforGeeks*. Ημερομηνία πρόσβασης: 14 Σεπτέμβριος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://www.geeksforgeeks.org/linux-unix/cloc-count-number-of-lines-of-code-in-file/>

[24] 'XAMPP tutorial: installation and first steps', *IONOS Digital Guide*. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://www.ionos.com/digitalguide/server/tools/xampp-tutorial-create-your-own-local-test-server/>

[25] 'Basic usage - Composer'. Ημερομηνία πρόσβασης: 3 Ιούλιος 2025. [Έκδοση σε ψηφιακή μορφή]. Διαθέσιμο στο: <https://getcomposer.org/doc/01-basic-usage.md>

