



Πανεπιστήμιο Δυτικής Μακεδονίας
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Διπλωματική Εργασία

Σχεδιασμός και ανάπτυξη ιστοχώρου αναφορικά με την ανακάλυψη και την εξοικείωση του κοινού με ορόσημα τοπικής κουλτούρας ή παράδοσης.

Design and development of a website focused on affiliation and discovery of local hotspots hosting or organizing cultural events.

Λάμπρος Βαρνάβας

Επιβλέπων Καθηγητής: Δρ. Μηνάς Δασυγένης
Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

17 Μαρτίου 2020

Περίληψη

Τις τελευταίες δεκαετίες, ιδιαίτερα όμως στην σύγχρονη εποχή, το Διαδίκτυο είναι απαραίτητο και αναπόσπαστο κομμάτι της καθημερινότητας μας. Η χρήση του, αδιαμφισβήτητα, προσφέρει ποικίλα οφέλη σε οποιονδήποτε είναι πρόθυμος να το χρησιμοποιήσει με κατάλληλο τρόπο. Είναι κοινώς αποδεκτό, το γεγονός ότι το Διαδίκτυο αποτελεί την ισχυρότερη πηγή πληροφορίας και ένα από τα χρησιμότερα και πολυτιμότερα εργαλεία του είναι η άμεση και συνεχής δυνατότητα ενημέρωσης για όλους τους τομείς της ζωής μας.

Αντικείμενο της συγκεκριμένης διπλωματικής εργασίας είναι ακριβώς η προαναφερθείσα διαπίστωση. Συγκεκριμένα, ο κύριος σκοπός της είναι η ενημέρωση των πολιτών, μέσω του διαδικτύου, για τις κοινωνικές και πολιτιστικές εκδηλώσεις που πραγματοποιούνται σε συγκεκριμένες τοποθεσίες μιας περιοχής, καθώς και η δυνατότητα να δημιουργούν οι ίδιοι εκδηλώσεις.

Η παρούσα διπλωματική εργασία αφορά τον σχεδιασμό και την υλοποίηση ενός πληροφοριακού συστήματος διαχείρισης εκδηλώσεων. Το πληροφοριακό σύστημα αποτελείται από μια διαδικτυακή εφαρμογή, η οποία σχεδιάστηκε με κατάλληλο τρόπο, ώστε να καλύψει εξίσου την πιθανότητα χρήσης της από συσκευές με μικρότερα μεγέθη οθόνης από αυτές ενός Η/Υ. Ειδικότερα, ο εκάστοτε επισκέπτης έχει τη δυνατότητα να ενημερωθεί για τις εκδηλώσεις που έχουν καταχωρηθεί στην ιστοσελίδα, και κάνοντας εγγραφή στο σύστημα μπορεί να δημιουργήσει ο ίδιος μια εκδήλωση -να την επεξεργαστεί αφού την προσθέσει- αλλά και να μάθει περισσότερες πληροφορίες σχετικά με τις ήδη υπάρχουσες εκδηλώσεις. Στην κάθε εκδήλωση εμπεριέχονται όλες οι απαραίτητες πληροφορίες που χρειάζονται ώστε να προωθείται η σωστή ενημέρωση, όπως είναι η τοποθεσία (δηλαδή η συγκεκριμένη διεύθυνση), δίνοντας την δυνατότητα στον χρήστη μέσω ενός διαδικτυακού χάρτη να γνωρίζει το ακριβές σημείο της εκδήλωσης, την ημερομηνία έναρξης όπως και λήξης της, την ώρα έναρξης, την κατηγορία στην οποία κατατάσσεται, την πόλη και τον νομό που

πραγματοποιείται, φωτογραφίες και τίτλο που την αντιπροσωπεύουν, καθώς και επαρκή στοιχεία για το περιεχόμενο της. Επίσης, για τη διευκόλυνση του χρήστη προσφέρεται η δυνατότητα εύρεσης μιας εκδήλωσης. Συγκεκριμένα στην ιστοσελίδα δίνεται η επιλογή αναζήτησης της, με διάφορα φίλτρα όπως είναι η κατηγορία, η πόλη, ο νομός ή η ημερομηνία. Τέλος, ο χρήστης μπορεί να καταγγέλλει κάποια εκδήλωση που θεωρεί πως εμπεριέχει κακόβουλο περιεχόμενο στον δημιουργό της, να προσθέτει και να αφαιρεί σχόλια, όπως και να διαγράφει από το σύστημα και την ιστοσελίδα τις εκδηλώσεις που ο ίδιος δημιούργησε.

Περιβάλλον Ανάπτυξης Διαδικτυακής Εφαρμογής: Η διαδικτυακή εφαρμογή υλοποιήθηκε με τη χρήση σύγχρονων τεχνολογιών λογισμικού και προγραμματισμού διαδικτύου ανοιχτού κώδικα. Συγκεκριμένα έγινε χρήση των HTML5, CSS3, JavaScript (ES6, ES7, ES8, ES9, ES10, ES11), NPM, ReactJS, React Router (DOM), NodeJS, ExpressJS, Body Parser, MySQL, Sequelize, Nodemon, Webpack, Babel. Η ανάπτυξη της πραγματοποιήθηκε μέσω του Visual Studio Code μεταγλωττιστή.

Λέξεις κλειδιά: Πληροφοριακό Σύστημα, Διαδικτυακή Εφαρμογή, Προγραμματισμός Διαδικτύου, Διεπαφή Προγραμματισμού Εφαρμογών (API), Εκδηλώσεις.

Abstract

Over the last few decades, especially in the recent contemporary age, the internet has become a necessity and an integral part of our everyday lives. Anyone willing to make an effort to use it properly, will without a doubt, enjoy its various benefits. It's commonly acceptable, the fact that the internet constitutes the most powerful source of information, and one of its most useful and valuable tools is its ability to stay constantly updated.

Object of the particular thesis project is the ascertainment made above. More specifically, the thesis' main purpose is to keep people up to date about the social and cultural events that take place at a certain location/area, while also giving them the opportunity to create an event like that themselves.

The present thesis relates to the design and implementation of an information system for event management. The information system will consist of an online application, properly designed, to work both on a PC screen and on devices with smaller size screens. In detail, every visitor gets the chance to be informed about any event that has been registered on the website. By signing in to the system the user can get even more information on an event he may be interested in, and he is also able to create an event himself and edit/process it once he's done adding it to the events' list. Every event is consisted of all the necessary information needed in order to promote it and accurately inform the user about it, such as the area/region of the event, providing the user the ability to see through an online map the exact location of the event, the date and time the event starts and ends, the category it's classified in, the town and the prefecture it's taking place in, photos, a title, and a description that represents it. There also is a "search event" function. This means that into the website, the user takes the opportunity to search for events while filtering the options he gets, like the town, the prefecture, the category and the date it takes place in. Lastly, the user is able to, report any event that might include malicious content, add

and delete comments and also completely erase from the system and the website an event that he created.

Web Application Development Environment: The Web Application was implemented using the latest open source software technologies for Web development. In particular, the technologies that were used are HTML5, CSS3, JavaScript (ES6, ES7, ES8, ES9, ES10, ES11), NPM, ReactJS, React Router (DOM), NodeJS, ExpressJS, Body Parser, MySQL, Sequelize, Nodemon, Webpack, Babel. The development was accomplished through the Visual Studio Code compiler.

Keywords: Information System, Website Application, Web Development, Application Programming Interface (API), Events.

Δήλωση Πνευματικών Δικαιωμάτων

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο

”Σχεδιασμός και ανάπτυξη ιστοχώρου αναφορικά με την ανακάλυψη και την εξοικείωση του κοινού με ορόσημα τοπικής κουλτούρας ή παράδοσης.”

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.

Ευχαριστίες

Πρωτίστως, οφείλω το μεγαλύτερο ευχαριστώ στην οικογένεια μου για την αμέριστη ηθική, ψυχολογική και υλική υποστήριξη που μου παρείχε καθ'όλη την διάρκεια των σπουδών μου.

Επιπλέον, θα ήθελα να ευχαριστήσω όλους μου τους φίλους και περισσότερο αυτούς που στάθηκαν δίπλα μου. Ιδιαίτερα όμως, εκείνους τους οποίους μου πρόσφεραν ανιδιοτελώς την πολύτιμη βοήθεια τους.

Τέλος, εξίσου σημαντική και βαρύτιμη υπήρξε η συμβουλή, συμβολή και καθοδήγηση του επιβλέπων καθηγητή Δρ. Μηνά Δασυγένη στη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας.

Περιεχόμενα

1	Εισαγωγή	14
1.1	Περιγραφή σημερινής κατάστασης	14
1.2	Ιδέα και σκοπός υλοποίησης	15
1.3	Παρόμοια συστήματα	16
1.3.1	Premiere Events	16
1.3.2	ελculture	17
1.3.3	Discover Greece	17
1.3.4	Clubber	18
1.4	Συμπεράσματα και ανάλυση αντίστοιχων εφαρμογών	19
1.5	Σύνοψη διπλωματικής εργασίας	21
2	Θεωρητικό υπόβαθρο	22
2.1	Προγραμματισμός διαδικτύου	22
2.1.1	HTML	24
2.1.2	CSS	25
2.1.3	JavaScript	26
2.1.4	Database	27
2.2	Frameworks, platforms, libraries, tools and compilers	28
2.2.1	Node.js	29
2.2.2	NPM (Node Package Manager)	30
2.2.3	React	31
2.2.4	Express.js	33
2.2.5	Webpack	34
2.2.6	Babel	35
2.2.7	Visual Studio Code	35
2.3	Σύνοψη κεφαλαίου	36

3	Δομικά μέρη συστήματος	37
3.1	Απαιτήσεις συστήματος	37
3.2	Περιπτώσεις χρήσης (use cases)	39
3.3	Ανάλυση βάσης δεδομένων (Database)	42
3.3.1	Enhanced Entity – Relationship Diagram (EER model)	43
3.3.2	Πίνακας “events”	44
3.3.3	Πίνακας “users”	47
3.3.4	Πίνακας “categories”	48
3.3.5	Πίνακας “images”	49
3.3.6	Πίνακας “reports”	49
3.3.7	Πίνακας “comments”	50
3.3.8	Πίνακας “cities”	52
3.3.9	Πίνακας “prefectures”	52
3.4	Ασφάλεια συστήματος	53
3.5	Σύνοψη κεφαλαίου	56
4	Λειτουργικότητα συστήματος	58
4.1	Λειτουργίες διαδικτυακής εφαρμογής και διαπαφές χρήστη	58
4.1.1	Αρχική σελίδα διαδικτυακής εφαρμογής	59
4.1.2	Σελίδα εισόδου, εγγραφής και ανάκτησης κωδικού πρόσβασης	61
4.1.3	Σελίδα αναζήτησης	63
4.1.4	Σελίδα δημιουργίας εκδήλωσης εγγεγραμμένου χρήστη	65
4.1.5	Σελίδα Διαχείρισης εκδηλώσεων χρήστη	68
4.1.6	Σελίδα επιλεγμένης εκδήλωσης	72
4.1.7	Σελίδα προφίλ χρήστη	79
4.2	Προβολή διαδικτυακής εφαρμογής από μικρότερες οθόνες συσκευών	81
4.3	Ανάλυση σημαντικών σημείων κώδικα	81
4.3.1	Κομμάτια κώδικα στο Front-end	82
4.3.2	Κομμάτια κώδικα στο Back-end - RESTful API	85
4.4	Σύνοψη κεφαλαίου	90
5	Επίλογος	91
5.1	Ανακεφαλαίωση	91

5.1.1	Προβλήματα που προέκυψαν κατά την υλοποίηση	92
5.2	Μετρικά κώδικα συστήματος	93
5.2.1	Front-end	93
5.2.2	Back-end	93
5.3	Μοντέλο S.W.O.T	94
5.4	Μελλοντικές επεκτάσεις	95
5.5	Συμπεράσματα	96
5.6	Σύνοψη κεφαλαίου	96
	Παραρτήματα	97
	Α' Εγκατάσταση και χρήση εφαρμογής	98

Κατάλογος σχημάτων

1.1	Logo αντίστοιχης διαδικτυακής εφαρμογής.	16
1.2	Logo αντίστοιχης διαδικτυακής εφαρμογής.	17
1.3	Logo αντίστοιχης διαδικτυακής εφαρμογής.	17
1.4	Logo αντίστοιχης διαδικτυακής εφαρμογής.	18
2.1	Παράδειγμα σύνταξης CSS	26
2.2	Παράδειγμα σύνταξης ES5-ES6.	27
2.3	Παράδειγμα σύνταξης SQL.	29
2.4	Σύνταξη Node.js για έναν HTTP εξυπηρετητή.	30
2.5	Σύνταξη για εγκατάσταση npm/package.	31
2.6	Παράδειγμα σύνταξης JSX.	32
2.7	Εισαγωγή του Express.js στο Node.js και δημιουργία server.	33
2.8	Λειτουργίες του Webpack (workflow).	34
2.9	Λειτουργίες του Babel (workflow).	35
2.10	Λειτουργίες του Visual Studio Code (workflow).	36
3.1	EER diagram Events.	43
4.1	Μπάρα πλοήγησης μη εγγεγραμμένου χρήστη.	59
4.2	Μπάρα πλοήγησης εγγεγραμμένου χρήστη.	59
4.3	Υποσέλιδο διαδικτυακής εφαρμογής (footer).	59
4.4	Αρχική σελίδα πρώτου επισκέπτη.	60
4.5	Παραδείγματα περιεχομένου αρχικής σελίδας.	61
4.6	Σελίδα εγγραφής με κενές τιμές στα πεδία της φόρμας.	63
4.7	Σελίδα εγγραφής με μη έγκυρα στοιχεία στη φόρμα.	64
4.8	Σελίδα εγγραφής με δυο έγκυρα πεδία στη φόρμα.	65
4.9	Σελίδα εγγραφής με έγκυρα στοιχεία στη φόρμα.	66
4.10	Σελίδα εισόδου με κενά στοιχεία στη φόρμα.	67

4.11 Σελίδα εισόδου με μη έγκυρα στοιχεία στη φόρμα.	68
4.12 Σελίδα εισόδου με έγκυρα στοιχεία στη φόρμα.	69
4.13 Σελίδα ανάκτησης κωδικού πρόσβασης (εισαγωγή email).	69
4.14 Σελίδα αλλαγής κωδικού πρόσβασης (reset).	70
4.15 Σελίδα αναζήτησης.	70
4.16 Παράδειγμα μηνύματος σφάλματος στη σελίδα αναζήτησης	71
4.17 Σελίδα εκδηλώσεων που πραγματοποιούνται μια συγκεκριμένη ημε- ρομηνία.	71
4.18 Σελίδα δημιουργίας εκδήλωσης με κενές τιμές.	72
4.19 Σελίδα δημιουργίας εκδήλωσης με έγκυρες τιμές.	73
4.20 Μήνυμα σφάλματος στη σελίδα δημιουργίας εκδήλωσης για μη έγκυρη διεύθυνση.	73
4.21 Σελίδα δημιουργίας εκδήλωσης με κενή τιμή στο πεδίο επιλογής ει- κόνας.	74
4.22 Σελίδα εκδηλώσεων χρήστη (My events).	75
4.23 Κουμπί επεξεργασίας.	75
4.24 Σελίδα επεξεργασίας (Edit) εκδήλωσης.	76
4.25 Κουμπί διαγραφής.	76
4.26 Μήνυμα προειδοποίησης για την επιβεβαίωση της διαγραφής εκδήλω- σης.	76
4.27 Κουμπί προβολής τοποθεσίας στο χάρτη.	76
4.28 Προβολή τοποθεσίας εκδήλωσης στον χάρτη (Satellite).	77
4.29 Προβολή περιεχομένου επιλεγμένης εκδήλωσης.	77
4.30 Κουμπί αναφοράς.	77
4.31 Μήνυμα προειδοποίησης για την αναφορά εκδήλωσης.	78
4.32 Μήνυμα ειδοποίησης για εκδήλωση που έχει ήδη αναφερθεί από τον ίδιο χρήστη.	78
4.33 Έγκυρη προσθήκη σχολίου.	78
4.34 Παράδειγμα σχολίου όπως αυτό εμφανίζεται στη σελίδα.	79
4.35 Σελίδα προφίλ εγγεγραμμένου χρήστη.	80
4.36 Σελίδα αλλαγής κωδικού πρόσβασης (logged in).	80

4.37 Προβολή σελίδας “Authenticate” με κλειστό navigation bar από συσκευή μικρότερης οθόνης.	82
4.38 Προβολή σελίδας “Authenticate” με ανοιχτό navigation bar από συσκευή μικρότερης οθόνης.	83
4.39 Κώδικας χρήσης της βιβλιοθήκης “react-router-dom” για εγγεγραμμένο χρήστη.	83
4.40 Χρήση της συνάρτησης React.lazy για τη διαίρεση του κώδικα.	84
4.41 Παράδειγμα κώδικα για validation στον κωδικό πρόσβασης.	85
4.42 Κώδικας χρήσης του useState Hook στη React.	85
4.43 Κώδικας χρήσης του useEffect Hook στη React.	86
4.44 Κώδικας για τον ορισμό των αποδεκτών headers (middleware)	86
4.45 Κώδικας για validation στο sign up.	87
4.46 Κώδικας συνάρτησης getCoordsForAddress για τη σύνδεση με το API της Google.	88
4.47 Κώδικας αποστολής email στην περίπτωση επιτυχούς sign up.	88
4.48 Κώδικας κρυπτογράφησης password.	89
4.49 Κώδικας μεταφόρτωσης εικόνων.	89

Κατάλογος πινάκων

2.1	Παρουσίαση βασικών στοιχείων HTML.	25
3.1	Δομή του πίνακα “events”	44
3.2	Δομή του πίνακα “users”	47
3.3	Δομή του πίνακα “categories”	48
3.4	Δομή του πίνακα “images”	49
3.5	Δομή του πίνακα “reports”	50
3.6	Δομή του πίνακα “comments”	51
3.7	Δομή του πίνακα “cities”	52
3.8	Δομή του πίνακα “prefectures”	52
5.1	Πληροφορίες αρχείων στο front-end	93
5.2	Πληροφορίες αρχείων στο back-end	94

Κεφάλαιο 1

Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζεται αναλυτικά το αντικείμενο της διπλωματικής εργασίας. Αρχικά, αναλύεται η κατάσταση της σημερινής περιόδου και τα αντίστοιχα πλεονεκτήματα ή και μειονεκτήματα που προκύπτουν με τη χρήση της διαδικτυακής τεχνολογίας. Στη συνέχεια, γίνεται αναφορά στην ιδέα, συγχρόνως και στον σκοπό, που οδήγησαν στην ανάληψη και υλοποίηση της συγκεκριμένης διπλωματικής εργασίας. Επίσης, γίνεται λόγος για αντίστοιχες εφαρμογές περιλαμβάνοντας το τελικό συμπέρασμα συγκρίνοντας ξεχωριστά κάθε εφαρμογή. Καταληκτικά, συνοψίζεται η παρουσίαση της δομής της εργασίας.

1.1 Περιγραφή σημερινής κατάστασης

Δεν συμβαίνει πολύ συχνά μια εφεύρεση που ανέρχεται να αλλάζει την ανθρωπότητα για πάντα. Η πρόσβαση στο Διαδίκτυο έχει ανασχηματίσει σε μεγάλο βαθμό την ύπαρξη του ανθρώπου. Ο κόσμος δεν θα ήταν αυτό που έχει γίνει σήμερα χωρίς το Διαδίκτυο. Το Διαδίκτυο, απευθύνεται σε κάθε πτυχή του τρόπου με τον οποίο ζει, εργάζεται, κοινωνικοποιείται, ενημερώνεται, ψωνίζει ή παίζει ο άνθρωπος. Έχει φέρει επανάσταση, σχεδόν σε όλους τους κλάδους τεχνολογίας με σκοπό να συμβάλει στην βελτίωση της ποιότητας ζωής διευκολύνοντας την καθημερινότητα του. Ειδικότερα, τα τελευταία χρόνια, ένας από τους κλάδους όπου η τεχνολογία έχει κάνει ραγδαία βήματα εξέλιξης, είναι ο τομέας της ενημέρωσης. Λόγω των ανεξέλεγκτων και αυξανόμενων καθημερινών υποχρεώσεων, το οποίο έχει ως συμπέρασμα να προκαλείται έλλειψη χρόνου, καθώς και του μεγάλου όγκου πληροφοριών που δέχεται ο άνθρωπος, είναι σχεδόν ανέφικτο να μπορεί να ανταπεξέλθει σε αυτές. Έτσι, γεννάται η ανάγκη για τη δημιουργία προγραμμάτων και συσκευών, τα οποία

τόσο με την κατάλληλη υλοποίηση και προγραμματισμό από την πλευρά του δημιουργού, όσο και με την σωστή διαχείριση από πλευράς χρήστη, να είναι σε θέση να εξυπηρετούν και να προσδίδουν περισσότερη ευκολία στον άνθρωπο. Έχοντας, λοιπόν, τη δυνατότητα χρησιμοποιώντας την τεχνολογία και πιο συγκεκριμένα το Διαδίκτυο, να προσφέρει ή να δεχθεί βοήθεια ο άνθρωπος, η ποιότητα ζωής του θα βελτιώνεται μέρα με τη μέρα κάνοντας καλό τόσο στον ίδιο όσο και στις επερχόμενες γενιές.

1.2 Ιδέα και σκοπός υλοποίησης

Στην εποχή που διανύουμε, η τεχνολογία εξελίσσεται με ραγδαίους ρυθμούς με αρκετούς τομείς της πληροφορικής και των τηλεπικοινωνιών να πρωτοστατούν. Η αυξανόμενη βελτίωση του διαδικτύου, παρέχει υπηρεσίες τις οποίες είναι αναγκαίο να χρησιμοποιηθούν ορθά, ώστε να αποκομηθούν τα οφέλη τους. Έχοντας επίγνωση των εκάστοτε συνθηκών, και παίρνοντας υπόψη πληροφορίες από αντίστοιχες εφαρμογές στο Διαδίκτυο, προέκυψε η ιδέα για την ανάπτυξη μιας διαδικτυακής εφαρμογής, όπου ο κύριος στόχος της θα είναι η συγκέντρωση αλλά και η δημιουργία κοινωνικών και πολιτιστικών εκδηλώσεων, για την άμεση ενημέρωση των πολιτών. Είναι αδιαμφισβήτητο το γεγονός, πως τόσο κοινωνικές ομάδες όπως και επιχειρήσεις επιθυμούν να διαφημίσουν μια εκδήλωση που πρόκειται να δημιουργήσουν, αλλά και οι υπόλοιποι πολίτες που θέλουν να παραβρεθούν στη συγκεκριμένη εκδήλωση, λόγω προσωπικών προτιμήσεων όπως και υποχρεώσεων, να ενημερωθούν άμεσα για το ακριβές μέρος, τον χρόνο και το συγκεκριμένο διάστημα που θα λαμβάνει χώρα κάποια εκδήλωση. Υλοποιώντας, λοιπόν, και δημοσιεύοντας στο κοινό έναν κατάλληλα διαμορφωμένο ιστοχώρο με τις συγκεκριμένες προδιαγραφές, όπου οι ενδιαφερόμενοι μπορούν είτε να ενημερωθούν είτε να ενημερώσουν, ενδεχομένως βοηθά στην επίλυση των προβλημάτων που αναφέρθηκαν. Άρα, γνωρίζοντας ότι αν επισκεφθεί κάποιος ενδιαφερόμενος, την διεύθυνση της συγκεκριμένης ιστοσελίδας, σε οποιαδήποτε στιγμή της καθημερινότητας του, είτε είναι από ηλεκτρονικό υπολογιστή, είτε είναι από συσκευές με μικρότερο μέγεθος οθόνης (π.χ κινητά τηλέφωνα), έχοντας σύνδεση στο Διαδίκτυο, μπορεί να πάρει ή να δώσει όλες τις πληροφορίες που είναι απαραίτητες για να πραγματοποιηθεί μια εκδήλωση.

1.3 Παρόμοια συστήματα

Έπειτα από μια εκτενή έρευνα σε διάφορους διαδικτυακούς τόπους εκπαιδευτικού και τεχνολογικού περιεχομένου, συγκεντρώθηκαν κάποιες πανομοιότυπες εφαρμογές, που εξυπηρετούν τον σκοπό της συγκεκριμένης διπλωματικής εργασίας, και παρουσιάζονται παρακάτω. Ειδικότερα, μαζί με την παρουσίαση τους, γίνεται και μια περιληπτική αναφορά σχετικά με ποιες είναι οι περιπτώσεις που καλύπτουν. Στο τέλος, γίνεται σύγκριση όλων των εφαρμογών με την εφαρμογή που κατασκευάστηκε στην παρούσα διπλωματική εργασία, όπως και ποια είναι τα πλεονεκτήματα έναντι αυτών.

1.3.1 Premiere Events



Σχήμα 1.1: Logo αντίστοιχης διαδικτυακής εφαρμογής.

Μια πανομοιότυπη διαδικτυακή εφαρμογή είναι το Premiere Events [8]. Ο διαδικτυακός ιστοχώρος Premiere Events δίνει τη δυνατότητα στον επισκέπτη να αναζητήσει τις εκδηλώσεις που υπάρχουν στο σύστημα του επιλέγοντας την ημερομηνία, την κατηγορία ή την τοποθεσία που επιθυμεί. Επίσης, στη συγκεκριμένη εφαρμογή ο επισκέπτης μπορεί να δει τις προσεχείς εκδηλώσεις, καθώς και τα στοιχεία που τις απαρτίζουν. Στο σχήμα 1.1 παρουσιάζεται το λογότυπο της εφαρμογής.

Πιο αναλυτικά η εφαρμογή προσφέρει τις παρακάτω δυνατότητες:

- Εμφάνιση των καταχωρημένων εκδηλώσεων στο σύστημα
- Αναζήτηση μέσω ημερομηνίας, τοποθεσίας και κατηγορίας
- Εμφάνιση των λεπτομερειών των εκδηλώσεων, όπως η συγκεκριμένη τοποθεσία και διεύθυνση η οποία προβάλλεται και μέσω ενός χάρτη, οι ημερομηνίες και ώρα διεξαγωγής και η περιγραφή
- Εμφάνιση των προβολών της κάθε εκδήλωσης
- Ηλεκτρονική επικοινωνία (email) με τον διαχειριστή μέσω φόρμας

-
- Δημιουργία λογαριασμού και είσοδος στο σύστημα

1.3.2 ελculture



Σχήμα 1.2: Logo αντίστοιχης διαδικτυακής εφαρμογής.

Ακόμα μια παρόμοια διαδικτυακή εφαρμογή είναι η ελcultute [5]. Η ελculture πληροφορεί τον επισκέπτη προβάλλοντας τις καταχωρημένες εκδηλώσεις που περιέχει το σύστημα της. Ταξινομώντας τα είδη των εκδηλώσεων δίνει μια άμεση δυνατότητα στον επισκέπτη να αναζητήσει με βάση την κατηγορία ή την ημερομηνία εκδηλώσεις που τον αφορούν. Στο σχήμα 1.2 παρουσιάζεται το λογότυπο της εφαρμογής.

Πιο συγκεκριμένα αυτή η εφαρμογή δίνει τις ακόλουθες επιλογές:

- Εμφάνιση λίστας εκδηλώσεων ταξινομημένα ανά κατηγορία
- Προσθήκη εκδηλώσεων σε μια ατζέντα
- Χρήση της ιστοσελίδας μέσω κινητής εφαρμογής
- Ηλεκτρονική επικοινωνία με τον διαχειριστή (email) μέσω φόρμας

1.3.3 Discover Greece



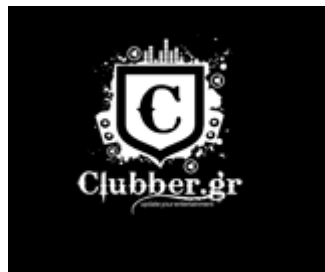
Σχήμα 1.3: Logo αντίστοιχης διαδικτυακής εφαρμογής.

Μια παρεμφερή διαδικτυακή εφαρμογή είναι η Discover Greece [6]. Η Discover Greece παρέχει τη δυνατότητα στον επισκέπτη να δει τη λίστα των εκδηλώσεων που υπάρχουν στο σύστημα της, καθώς και να αναζητήσει τον προορισμό, την κατηγορία και την ημερομηνία που αυτές πραγματοποιούνται. Εκτός από τις προαναφερθείσες δυνατότητες, η συγκεκριμένη ιστοσελίδα αναφέρεται κατά κύριο λόγο σε αυτούς οι οποίοι επιθυμούν να διοργανώσουν ένα ταξίδι στην Ελλάδα, διαθέτοντας έναν ταξιδιωτικό οδηγό και προτείνοντας στον επισκέπτη κάποια από τα αξιοθέατα της. Στο σχήμα 1.3 παρουσιάζεται το λογότυπο της εφαρμογής.

Αναλυτικότερα η εφαρμογή αυτή δίνει τις παρακάτω δυνατότητες:

- Εμφάνιση της λίστας των εκδηλώσεων που είναι καταχωρημένες στο σύστημα
- Αναζήτηση εκδηλώσεων μέσω φίλτρων όπως τίτλο, κατηγορία, προορισμό και ημερομηνία
- Ταξιδιωτικές ιδέες, όπως το που να πας, τι να κάνεις
- Εμφάνιση προτεινόμενων περιοχών επίσκεψης σε έναν χάρτη
- Κράτηση ξενοδοχείου διαμονής
- Μετάφραση της ιστοσελίδας σε διαφορετικές γλώσσες
- Ηλεκτρονική επικοινωνία με τον διαχειριστή (email) μέσω φόρμας

1.3.4 Clubber



Σχήμα 1.4: Logo αντίστοιχης διαδικτυακής εφαρμογής.

Μια άλλη εφαρμογή είναι η Clubber [3]. Η εφαρμογή Clubber αναφέρεται αποκλειστικά σε μουσικές εκδηλώσεις και πιο ειδικά στην ηλεκτρονική μουσική. Ο συγκεκριμένος ιστοχώρος, δίνει τη δυνατότητα στον επισκέπτη να ενημερωθεί και να

δει όλες τις εκδηλώσεις ηλεκτρονικής μουσικής που υφίστανται στο σύστημα της, σε μια λίστα που περιέχει την ημερομηνία, και την τοποθεσία. Επίσης, περιλαμβάνει μια λίστα με τους χώρους των εκδηλώσεων, όπως και μια λίστα των καλλιτεχνών που πρόκειται ή όχι, να διεξάγουν μια μουσική εκδήλωση. Στο σχήμα 1.4 παρουσιάζεται το λογότυπο της εφαρμογής.

Ειδικότερα η συγκεκριμένη διαδικτυακή εφαρμογή περιλαμβάνει:

- Εμφάνιση μιας λίστας με τις μουσικές εκδηλώσεις που είτε πραγματοποιήθηκαν είτε θα πραγματοποιηθούν.
- Εμφάνιση ενός χάρτη που υποδεικνύει τους χώρους που διεξάγονται οι μουσικές εκδηλώσεις.
- Εμφάνιση μιας λίστας καλλιτεχνών που δημιουργούν μια μουσική εκδήλωση
- Εμφάνιση μιας λίστας με τους χώρους όπου δημιουργούνται οι μουσικές εκδηλώσεις.
- Κράτηση ξενοδοχείου διαμονής
- Μετάφραση της ιστοσελίδας σε διαφορετικές γλώσσες
- Ηλεκτρονική επικοινωνία με τον διαχειριστή (email) μέσω φόρμας

1.4 Συμπεράσματα και ανάλυση αντίστοιχων εφαρμογών

Μελετώντας με προσοχή όλες τις προαναφερθείσες διαδικτυακές εφαρμογές ως προς τις δυνατότητες που προσφέρουν, διαπιστώθηκαν τα ακόλουθα συμπεράσματα:

- Όλες οι εφαρμογές είτε έχουν ταξινομήσει τις κατηγορίες των εκδηλώσεων είτε δίνουν τη δυνατότητα αναζήτησης των εκδηλώσεων με βάση την κατηγορία, την τοποθεσία ή την ημερομηνία, δίνοντας την δυνατότητα άμεσης εύρεσης εκδηλώσεων.
- Όλες οι εφαρμογές δίνουν τη δυνατότητα για επικοινωνία με τον διαχειριστή μέσω ηλεκτρονικής φόρμας.

-
- Εκτός από την εφαρμογή “Premiere Events”, καμία από τις υπόλοιπες δεν δίνει τη δυνατότητα στον επισκέπτη, να εισέλθει στο σύστημα τους δημιουργώντας λογαριασμό και να ανακτήσει τον κωδικό.
 - Ο μόνος τρόπος να προστεθεί μια εκδήλωση στο σύστημα των εφαρμογών αυτών είναι να έρθει ο ενδιαφερόμενος σε επικοινωνία με τον διαχειριστή ή υπεύθυνο της ιστοσελίδας μέσω της ηλεκτρονικής φόρμας που διαθέτουν, ώστε ύστερα από αμοιβαία συνεννόηση να καταχωρηθεί κάποια εκδήλωση στον ιστοχώρο.
 - Σε αντίθεση με τις υπόλοιπες, η εφαρμογή “Discover Greece” προσφέρει την προβολή και ενημέρωση των εκδηλώσεων ως μόνο ένα παρακλάδι της ιστοσελίδας. Η “Discover Greece” δηλαδή, δεν δίνει κεντρική προσοχή μόνο στο ζήτημα των εκδηλώσεων, αλλά προσφέρει απλή, σε γενικές γραμμές, εμπειρία για τον επισκέπτη.
 - Η εφαρμογή “Clubbers” επικεντρώνεται μόνο στις μουσικές εκδηλώσεις που αφορούν την ηλεκτρονική μουσική.
 - Η “elculture” εφαρμογή είναι η μόνη που παρέχει εκτός από την διαδικτυακή ιστοσελίδα, και εφαρμογή για κινητά τηλέφωνα (Android & iOS).

Κάνοντας τον απολογισμό της ανάλυσης των εφαρμογών που αναφέρθηκαν, αναμφίβολα, διαπιστώνεται ότι το πληροφοριακό σύστημα που αναπτύχθηκε στην παρούσα διπλωματική εργασία είναι επιδέξιο σύγκρισης με τα παραπάνω συστήματα, αφού καλύπτει δυναμικά ένα αρκετό ποσοστό ένα σύνολο των δυνατοτήτων που παρέχουν. Ειδικότερα, το εκάστοτε πληροφοριακό σύστημα της διπλωματικής είναι το μόνο το οποίο δίνει τη δυνατότητα στον επισκέπτη, αφού αποκτήσει λογαριασμό, να δημιουργήσει μια εκδήλωση, να την προσθέσει στο σύστημα και να την προβάλει άμεσα στην ιστοσελίδα. Επιπλέον, παρέχεται η δυνατότητα στον χρήστη να προσθέσει σχόλιο σε κάποια εκδήλωση, όπως και να την αναφέρει ως κακόβουλη στον διαχειριστή. Όλες οι εφαρμογές, έχουν ταξινομήσει τις εκδηλώσεις του συστήματος τους με βάση την κατηγορία και δίνουν την επιλογή φίλτρων για αναζήτηση τους, όπως για παράδειγμα ως προς την κατηγορία, την τοποθεσία ή την ημερομηνία. Τελευταίο, όμως εξίσου σημαντικό είναι η δυνατότητα που παρέχει η εφαρμογή

“elculture” για χρήση της μέσω εφαρμογής κινητού τηλεφώνου, προσδίδοντας μια φιλικότερη διεπαφή χρήστη και μια επιπλέον επιλογή για τους χρήστες αυτούς.

1.5 Σύνοψη διπλωματικής εργασίας

Ακολουθεί η ανάλυση της διαδικασίας σχεδίασης και υλοποίησης της διαδικτυακής εφαρμογής, η οποία αποτελείται από πέντε κεφάλαια.

Αυτό το κεφάλαιο αποτελεί την εισαγωγή στο αντικείμενο το οποίο πραγματεύεται η διπλωματική εργασία, όπως και τον λόγο για τον οποίο προέκυψε.

Το δεύτερο κεφάλαιο αναφέρεται σε όλες τις τεχνολογίες που χρησιμοποιήθηκαν και που ήταν απαραίτητες για την ολοκλήρωση της διαδικτυακής εφαρμογής.

Στο τρίτο κεφάλαιο γίνεται μια λεπτομερή καταγραφή των απαιτήσεων του συστήματος για τη σχεδίαση της βάσης δεδομένων και μια αναλυτική περιγραφή των οντοτήτων, των δεδομένων και των συσχετίσεων που αναπτύχθηκαν. Ακόμα, αναφέρονται οι τεχνικές ασφάλειας του συστήματος.

Στο τέταρτο κεφάλαιο παρουσιάζεται η ανάπτυξη του λογισμικού μέρους της διαδικτυακής εφαρμογής. Επίσης, προβάλλονται οι λειτουργίες και η διεπαφή χρήστη, σε μερικά στιγμιότυπα, σε πιθανές περιπτώσεις χρήσης του συστήματος.

Στο πέμπτο κεφάλαιο, γίνεται λόγος των συμπερασμάτων που προέκυψαν από την ανάπτυξη του συστήματος, καθώς και τα προβλήματα που προέκυψαν κατά τη διαδικασία ανάπτυξης και υλοποίησης όπως και οι ενδεχόμενες μελλοντικές επεκτάσεις και βελτιώσεις.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

Σε αυτό το κεφάλαιο, καταγράφονται οι τεχνολογίες που χρησιμοποιήθηκαν για την εκπόνηση της εργασίας. Αναλύονται και διατυπώνονται έννοιες που αναφέρονται στον προγραμματισμό διαδικτύου, όπως είναι οι γλώσσες προγραμματισμού και οι πλατφόρμες ανάπτυξης λογισμικού. Καθώς η ανάπτυξη ιστοσελίδων αποτελεί μια εκτεταμένη διαδικασία και δεδομένου του μεγάλου όγκου των τεχνολογιών που απαιτεί, θα δοθεί μεγαλύτερη βαρύτητα στην περιγραφή των τεχνολογιών που αφορούν τον σκελετό της εργασίας και λιγότερο σε αυτές που εφαρμόστηκαν σε μικρότερο βαθμό ως υποστηρικτικές. Παρακάτω λοιπόν, παρουσιάζονται οι έννοιες, οι μέθοδοι και οι τεχνολογίες για την ολοκλήρωση της εργασίας.

2.1 Προγραμματισμός διαδικτύου

Ο όρος προγραμματισμός διαδικτύου (web development) [16] αναφέρεται στην εργασία που αφορά την ανάπτυξη μιας ιστοσελίδας για το Διαδίκτυο (World Wide Web) ή ενός ενδοδικτύου, δηλαδή ενός ιδιωτικού δικτύου. Η ανάπτυξη ιστού μπορεί να κυμαίνεται από την κατασκευή μιας απλής μονής στατικής σελίδας, απλού κειμένου, μέχρι πολύπλοκες διαδικτυακές εφαρμογές διαδικτύου όπως ηλεκτρονικές επιχειρήσεις και υπηρεσίες κοινωνικών δικτύων. Μια πιο εκτεταμένη σειρά καθηκόντων στα οποίες αναφέρεται συνήθως η ανάπτυξη ενός ιστοχώρου μπορεί να περιλαμβάνει την τεχνική του ιστού (web engineering) [25], τον σχεδιασμό του (web design) [19], την ανάπτυξη περιεχομένου του ιστού, την σύνδεση και επικοινωνία μεταξύ του πελάτη και διακομιστή, τη διαμόρφωση ιστού και δικτύου και την ανάπτυξη ηλεκτρονικού εμπορίου. Σύμφωνα με τους επαγγελματίες του προγραμματισμού διαδικτύου, η ανάπτυξη και κατασκευή του ιστού συνήθως αναφέ-

ρεται στις κύριες πτυχές του σχεδιασμού ιστοσελίδων, και πιο συγκεκριμένα στην γραφή και κωδικοποίηση. Σε μεγάλες επιχειρήσεις και οργανισμούς, η κατασκευή ιστοσελίδων μπορεί να αποτελείται από εκατοντάδες άτομα που ειδικεύονται στον προγραμματισμό διαδικτύου. Αντίθετα, σε μικρότερες οργανώσεις ίσως απαιτείται μόνο ένα προγραμματιστή ο οποίος έχει μόνιμη θέση ή έναν συμβαλλόμενο προγραμματιστή και για δευτερεύουσες ανάγκες σχετικές με το web design όπως είναι ένας γραφίστας ή ένας τεχνικός συστημάτων πληροφορικής. Επιπρόσθετα, η ανάπτυξη ιστοχώρου ενδέχεται να απαιτεί συνεργασία μεταξύ όλων των σχετικών τμημάτων που χρειάζεται για την υλοποίηση του και όχι μόνο ενός τομέα ενός καθορισμένου τμήματος. Υπάρχουν τρία είδη εξειδίκευσης ενός προγραμματιστή διαδικτύου, ο front-end developer [13], ο back-end developer [23] και ο full-stack developer [15]. Οι front-end προγραμματιστές αναφέρονται σε αυτούς οι οποίοι είναι υπεύθυνοι για την εικόνα που βλέπει ο πελάτης (client), δηλαδή την συμπεριφορά και τα γραφικά που εκτελούνται στο πρόγραμμα περιήγησης χρηστών, ενώ οι back-end προγραμματιστές ασχολούνται με τους διακομιστές (web servers) [11], και ο full-stack developer γνωρίζει και υλοποιεί και τις δυο προηγούμενες τεχνικές. Ουσιαστικά, η ανάπτυξη ιστού είναι η δημιουργία και η συντήρηση ιστοσελίδων, δηλαδή είναι το έργο που συμβαίνει πίσω από τις σκηνές ώστε ένας ιστότοπος να προσφέρει μια απρόσκοπτη εμπειρία στον χρήστη και να αποδίδεται με ταχύτητα. Η ιεραρχία ανάπτυξης ακολουθεί τον προγραμματισμό ως προς την πλευρά του πελάτη (client-side programming) [2], τον προγραμματισμό από πλευράς διακομιστή (server-side programming) [9] και την τεχνολογία των βάσεων δεδομένων (database technology). Η κωδικοποίηση από πλευράς πελάτη χειρίζεται αυτό που βλέπει ο χρήστης, μέσω των κομματιών κώδικα (scripts) τα οποία είναι ενσωματωμένα και αλληλεπιδρούν με το HTML του ιστοτόπου, επιλέγοντας στοιχεία από αυτό και στη συνέχεια χειρίζεται αυτά τα στοιχεία ώστε να παρέχεται μια διαδραστική εμπειρία. Επίσης, τα scripts αλληλεπιδρούν και με αρχεία CSS τα οποία βοηθούν στον τρόπο εμφάνισης της σελίδας και τελικά επιστρέφει τα δεδομένα που έχουν περάσει από τον ιστότοπο με τρόπο που να είναι αναγνώσιμος από το πρόγραμμα περιήγησης (web browser). Η κωδικοποίηση από την πλευρά του διακομιστή είναι μια τεχνική η οποία περιλαμβάνει επίσης τη χρήση κομματιών κώδικα σε έναν διακομιστή ιστού τα οποία παράγουν μια απάντηση προσαρμοσμένη για το αίτημα κάθε χρήστη στον

ιστότοπο. Πιο συγκεκριμένα, διακρίνεται από τη δέσμη ενεργειών από την πλευρά του client, όπου τα ενσωματωμένα scripts, όπως η JavaScript, εκτελούνται σε ένα πρόγραμμα περιήγησης ιστού.

2.1.1 HTML

Η HTML (Hyper Text Markup Language) [24] είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων. Το Hyper Text είναι η μέθοδος με την οποία επιτρέπεται η μετακίνηση στον ιστό κάνοντας κλικ σε ένα ειδικό κείμενο που ονομάζεται υπερσύνδεσμος. Ο όρος Hyper (υπερβολικό) σημαίνει απλώς ότι δεν είναι γραμμικό, δηλαδή η μετάβαση σε οποιοδήποτε σημείο του διαδικτύου γίνεται οποτεδήποτε, δεν υπάρχει καμία σειρά για να γίνουν τα πράγματα. Markup (σήμανση) είναι αυτό που κάνουν οι ετικέτες (tags) HTML στο κείμενο μέσα σε αυτές, δηλαδή το επισημαίνουν ως ένα συγκεκριμένο τύπο κειμένου (συνηθίζεται η πλάγια γραφή). Η HTML είναι μια γλώσσα (Language) καθώς έχει λέξεις-κλειδιά και σύνταξη όπως κάθε άλλη γλώσσα. Η γραφή της είναι υπό μορφή στοιχείων τα οποία αποτελούνται από ετικέτες (tags), οι οποίες περικλείονται μέσα σε σύμβολα “< >”, (μεγαλύτερο από) και (μικρότερο από) μέσα στο περιεχόμενο της ιστοσελίδας. Η λειτουργία των ετικετών είναι συνήθως ανά ζεύγη, για παράδειγμα <p> ως ετικέτα έναρξης και </p> ως ετικέτα λήξης και ανάμεσα από αυτές τοποθετείται το κείμενο. Τα αρχεία HTML χρησιμοποιούνται από το πρόγραμμα περιήγησης, χωρίς όμως να τα εμφανίζει, ώστε να προβάλλει το περιεχόμενο της ιστοσελίδας για να είναι κατανοητό από τον άνθρωπο. Μέσω αυτής της γλώσσας ενσωματώνονται ποικίλα αντικείμενα, διαδραστικές φόρμες και εικόνες μέσα στη σελίδα, καθώς επίσης καθορίζονται δομικά στοιχεία για το κείμενο και ενσωματώνονται scripts σε γλώσσες όπως η JavaScript που επηρεάζουν την συμπεριφορά των ιστοσελίδων HTML και έτσι από στατικές γίνονται διαδραστικές. Η ύπαρξη της προήλθε από τον φυσικό Tim Berners-Lee το 1989 τον πρώτο web developer και εφευρέτη του WWW (World Wide Web) και αποτελεί τον σκελετό όλων των ιστοτόπων. Με την πάροδο του χρόνου, η HTML εξελίσσεται και εκδίδονται βελτιωμένες εκδόσεις ή εκδόσεις με πρόσθετο περιεχόμενο. Η HTML5 είναι η Πέμπτη και τελευταία -ως σήμερα- έκδοση της. Στον πίνακα 2.1 παρουσιάζονται οι βασικές ετικέτες (tags) της HTML, όπως και η περιγραφή τους.

Στοιχείο HTML	Περιγραφή στοιχείου
<html> ... </html>	Δηλώνει την ιστοσελίδα που θα γραφτεί σε HTML
<head> ... </head>	Οριοθετεί την κεφαλίδα της σελίδας
<title> ... </title>	Καθορίζει τον τίτλο (δεν εμφανίζεται στην σελίδα)
<body> ... </body>	Οριοθετεί το περιεχόμενο της σελίδας
<hn> ... </hn>	Οριοθετεί την επικεφαλίδα (n : αριθμός)
 ... 	Ορίζει το κείμενο ανάμεσα ως bold
<i> ... </i>	Ορίζει το κείμενο ανάμεσα ως italics
 ... 	Εμφανίζει μία ομαδοποιημένη λίστα αντικειμένων
<div> ... </div>	Ξεχωρίζει ή ομαδοποιεί στοιχεία σε ένα μέρος της σελίδας
<p> ... </p>	Ορίζει μια παράγραφο
	Εμφανίζει μια εικόνα
 ... 	Ορίζει έναν υπερσύνδεσμο

Πίνακας 2.1: Παρουσίαση βασικών στοιχείων HTML.

2.1.2 CSS

Η CSS (Cascading Style Sheets) [18] είναι μια γλώσσα εμφάνισης για το στυλ που χρησιμοποιείται για την περιγραφή της παρουσίας ενός εγγράφου γραμμένου σε μια γλώσσα σήμανσης, όπως η HTML. Αποτελεί τεχνολογία ακρογωνιαίου λίθου του World Wide Web, παράλληλα με την HTML και την JavaScript. Έχει σχεδιαστεί για να επιτρέπει τον διαχωρισμό της παρουσίασης και του περιεχομένου, συμπεριλαμβανομένης της διάταξης, των χρωμάτων και των γραμματοσειρών. Μέσω αυτού του διαχωρισμού βελτιώνεται η προσβασιμότητα του περιεχομένου, παρέχεται μεγαλύτερη ευελιξία και έλεγχος στις προδιαγραφές των χαρακτηριστικών παρουσίασης, επιτρέπεται σε πολλαπλές σελίδες να μοιράζονται τη μορφοποίηση καθορίζοντας το σχετικό style sheet σε ένα ξεχωριστό αρχείο CSS μειώνοντας έτσι την πολυπλοκότητα και την επανάληψη σε δομικό περιεχόμενο. Η παρουσίαση ενός εγγράφου σε έναν χρήστη σημαίνει μετατροπή του σε μια μορφή που μπορεί να χρησιμοποιήσει το κοινό. Τα προγράμματα περιήγησης έχουν σχεδιαστεί για να παρουσιάζουν οπτικά έγγραφα, για παράδειγμα σε μια οθόνη υπολογιστή, έναν προβολέα ή έναν εκτυπωτή. Η CCS διαθέτει επίσης κανόνες για εναλλακτική μορφοποίηση εάν το περιεχόμενο είναι προσβάσιμο σε μια κινητή συσκευή. Με τον καιρό η CSS εξελίσσεται όλο και περισσότερο επεκτείνοντας τις δυνατότητες της. Η CSS3 είναι η τελευταία έκδοση έως σήμερα και στοχεύει στην βέλτιστη παρουσίαση των αντικειμένων σε μια ιστοσελίδα. Ακόμα, η σύνταξη της CSS χρησιμοποιείται και από άλλες γλώσσες προγραμματισμού, όπως η SASS, η οποία με μερικές μικρές εξαιρέσεις είναι ένα

υπερσύνολο της CSS, δηλαδή όλα τα έγκυρα CSS είναι και έγκυρα SCSS και λόγω της ομοιότητας της με την CSS γίνεται ευκολότερη στη σύνταξη. Ουσιαστικά, η CSS χρησιμοποιώντας τα στοιχεία ή τις ετικέτες της HTML προσδίδει μια φιλική προς τον άνθρωπο παρουσίαση στον web browser. Στο σχήμα 2.1 διαφαίνεται ο κώδικας της.

```
body {  
  background-color: lightblue;  
}  
  
h1 {  
  color: white;  
  text-align: center;  
}  
  
p {  
  font-family: verdana;  
  font-size: 20px;  
}
```

Σχήμα 2.1: Παράδειγμα σύνταξης CSS.

2.1.3 JavaScript

Η JavaScript, Vanilla JavaScript, ECMAScript ή συντομογράφως JS [20], είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού υψηλού επιπέδου με σύντομη σύνταξη, δυναμική πληκτρολόγηση, πρωτότυπα με βάση τον προσανατολισμό των αντικειμένων (prototype-based object-orientation) και πρώτης κατηγορίας λειτουργίες/συναρτήσεις (first-class functions). Παράλληλα με την HTML και τη CSS, η JavaScript είναι μία από τις βασικές τεχνολογίες του World Wide Web. Ενεργοποιεί διαδραστικές ιστοσελίδες και αποτελεί ουσιαστικό μέρος των εφαρμογών ιστού. Χρησιμοποιείται από την πλειοψηφία των ιστοτόπων και τα πιο διαδεδομένα προγράμματα περιήγησης στο Διαδίκτυο διαθέτουν μια αποκλειστική μηχανή JavaScript, όπως η V8 JavaScript Engine του Google Chrome, για να την εκτελέσουν. Ως γλώσσα πολλαπλών παραδειγμάτων, η JavaScript υποστηρίζει ένα στυλ προγραμματισμού που βασίζεται σε γεγονότα, λειτουργικά (functional) και επιτακτικά (imperative). Διαθέτει APIs (Application Programming Interface) ώστε να εργάζεται με κείμενα, συστοιχίες, ημερομηνίες, κανονικές εκφράσεις και το DOM (Document Object Model), όμως δεν περιλαμβάνει I / O (Input / Output), όπως εγκαταστάσεις δικτύωσης, απο-

θήκευσης ή γραφικών. Είναι βασισμένη στο περιβάλλον υποδοχής στο οποίο είναι ενσωματωμένο για να παρέχει αυτά τα χαρακτηριστικά. Στην αρχή, εφαρμοζόταν μόνο από πλευράς πελάτη (client-side) στους web browsers, όμως πλέον οι μηχανές JavaScript ενσωματώνονται σε πολλούς άλλους τύπους λογισμικού, συμπεριλαμβανομένου του server-side στους διακομιστές διαδικτύου (web servers), στις βάσεις δεδομένων (databases), καθώς και σε προγράμματα εκτός δικτύου και σε περιβάλλοντα εκτέλεσης καθιστώντας την JavaScript διαθέσιμη για τη σύνταξη εφαρμογών για κινητά τηλέφωνα και εφαρμογές υπολογιστών. Θα πρέπει να επισημανθεί πως αν και υπάρχουν αρκετές ομοιότητες μεταξύ της JavaScript και της Java, με κύριες ομοιότητες το όνομα, την σύνταξη και τις αντίστοιχες τυποποιημένες βιβλιοθήκες, οι δυο γλώσσες είναι διακριτές και διαφέρουν σημαντικά στον σχεδιασμό. Η JavaScript επηρεάστηκε από γλώσσες προγραμματισμού όπως η Self και η Scheme. Εξίσου σημαντικό να αναφερθεί, είναι πως η μορφή σειριοποίησης JSON, που χρησιμοποιείται για την αποθήκευση δομών δεδομένων σε αρχεία ή για την μετάδοση τους σε δίκτυα, βασίζεται στη JavaScript και πως δημιουργήθηκε το 1995 από το πρόγραμμα περιήγησης ιστού της Netscape. Σε αυτό το έργο, έγινε χρήση πολλών εκδόσεων JavaScript, με κύριες τις ES5-ES6 (ECMAScript v5-6). Στο σχήμα 2.2 φαίνεται η σύνταξη της.

```
// ES5
const myArray=['tony','Sara','Said',5];

let Arr1= myArray.map(function(item){
    return item;
});
console.log(Arr1);//output (4) ["tony", "Sara", "Said", 5]

//ES6 Syntax

let Arr2 = myArray.map(item => item);
console.log(Arr2); //output (4) ["tony", "Sara", "Said", 5]
```

Σχήμα 2.2: Παράδειγμα σύνταξης ES5-ES6.

2.1.4 Database

Μια βάση δεδομένων [21] είναι μια οργανωμένη συλλογή δεδομένων, που αποθηκεύεται γενικά και προσπελάζεται ηλεκτρονικά από ένα σύστημα υπολογιστή. Σε

μεγάλες και πολύπλοκες βάσεις δεδομένων χρησιμοποιούνται τεχνικές σχεδιασμού και μοντελοποίησης. Το σύστημα διαχείρισης βάσεων δεδομένων DBMS (Database Management System) είναι το λογισμικό που αλληλεπιδρά με τους τελικούς χρήστες, τις εφαρμογές και την ίδια τη βάση δεδομένων για την καταγραφή και ανάλυση των δεδομένων. Το λογισμικό DBMS περιλαμβάνει επιπλέον τις βασικές διευκολύνσεις που παρέχονται για τη διαχείριση της βάσης δεδομένων. Γενικότερα, ο όρος “βάση δεδομένων” χρησιμοποιείται επίσης για να αναφέρεται σε οποιοδήποτε από τα DBMS, πιο συγκεκριμένα το σύστημα βάσης δεδομένων (database system) ή μια εφαρμογή (application) που σχετίζεται με τη βάση δεδομένων. Τα σύστημα διαχείρισης βάσεων δεδομένων ταξινομούνται σύμφωνα με τα μοντέλα βάσης δεδομένων που υποστηρίζουν. Οι σχεσιακές βάσεις δεδομένων (Relational Databases) κυριαρχούσαν ως προς τη χρήση τους από τους επιστήμονες υπολογιστών τη δεκαετία του '80. Αυτά τα μοντέλα κατηγοριοποιούν τα δεδομένα ως σειρές και στήλες πινάκων και η πλειονότητα των προγραμματιστών χρησιμοποιεί τη γλώσσα υπολογιστών SQL (Structured Query Language) για τη γραφή και την αναζήτηση δεδομένων. Με την εξέλιξη του χρόνου, στο προσκήνιο εμφανίστηκαν και οι μη σχεσιακές βάσεις δεδομένων (Non-Relational Databases), οι οποίες έγιναν δημοφιλείς, γνωστές ως “NoSQL” επειδή χρησιμοποιούν διαφορετικές γλώσσες ερωτημάτων. Ειδικότερα, μια βάση δεδομένων πέρα από την ικανότητα της να αποθηκεύει δεδομένα, παρέχει μέσω του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, δηλαδή τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων. Ένα σύστημα διαχείρισης δεδομένων ανοιχτού κώδικα (RDBMS) είναι η MySQL και χρησιμοποιείται από πολλούς δημοφιλείς ιστούς, όπως το Facebook και το YouTube. Παρέχει πρόσβαση πολλών χρηστών σε ένα σύνολο βάσεων δεδομένων “τρέχοντας” έναν εξυπηρετητή (server). Το σχήμα 2.3 προβάλλει τη σύνταξη της SQL.

2.2 Frameworks, platforms, libraries, tools and compilers

Η υποενότητα που ακολουθεί αναφέρεται στις δομές (frameworks), τις πλατφόρμες, τις βιβλιοθήκες τρίτων (third party libraries), τα εργαλεία (tools) και τους μεταγλωττιστές (compilers) που χρησιμοποιήθηκαν για την εκτέλεση του έργου της διπλωματικής εργασίας. Η ανάλυση και αναφορά τους θεωρείται αξιοσημείωτη, κα-


```
SELECT c.customer_id, c.full_name,
(
  SELECT COUNT(*)
  FROM subscriptions s
  WHERE s.customer_id = c.customer_id
  GROUP BY s.customer_id
) subscriptions_count
FROM customers c
```

Σχήμα 2.3: Παράδειγμα σύνταξης SQL.

θώς η επίτευξη του έργου πραγματοποιήθηκε με την συμβολή και συνεργασία όλων αυτών. Παρακάτω, παρατίθενται η αναφορά, ανάλυση και περιγραφή τους με την ίδια σημασία στη σημαντικότητα.

2.2.1 Node.js

Το Node.js [17] είναι μια πλατφόρμα ανάπτυξης λογισμικού ανοιχτού κώδικα που υποστηρίζεται σε πολλές πλατφόρμες, κυρίως διακομιστών, που εκτελεί κώδικα JavaScript έξω από ένα πρόγραμμα περιήγησης. Επιτρέπει στους προγραμματιστές να χρησιμοποιούν JavaScript για να γράφουν εργαλεία γραμμής εντολών (command line tools) και για scripts που εκτελούνται από την πλευρά του διακομιστή (server-side scripting) για την παραγωγή δυναμικού περιεχομένου ιστοσελίδας πριν από την αποστολή της σελίδας στο πρόγραμμα περιήγησης του χρήστη. Γενικότερα, το Node.js αντιπροσωπεύει το γνωστό “JavaScript everywhere”, δηλαδή χρήση της γλώσσας JavaScript σε όλο το περιβάλλον λειτουργίας, ενοποιώντας την ανάπτυξη εφαρμογών ιστού γύρω από μία μόνο γλώσσα προγραμματισμού, αντί για διαφορετικές γλώσσες για δέσμες ενεργειών διακομιστή και πελάτη (server-and client-side scripts). Η αρχιτεκτονική του Node.js βασίζεται σε γεγονότα (event-driven) και είναι ικανή για ασύγχρονα I / O. Λόγω των συγκεκριμένων επιλογών σχεδιασμού της, στόχος της είναι η βελτιστοποίηση της απόδοσης και κλιμάκωσης στις εφαρμογές ιστού με πολλές λειτουργίες εισόδου / εξόδου (input / output operations), καθώς και για εφαρμογές ιστού σε πραγματικό χρόνο (real-time Web applications). Το χαρακτηριστικό που ξεχωρίζει το Node είναι, όπως προαναφέρθηκε, η ασύγχρονη επικοινωνία μεταξύ των υπολογιστικών πόρων και αυτό γίνεται εφικτό με την χρήση γεγονότων (events) που προσφέρει η JavaScript, τα ονομαζόμενα callbacks. Ειδικότερα, όταν ένας web browser φορτώσει πλήρως ένα αρχείο, όπου ένας χρήστης πατάει κάποιο

κουπί, ολοκληρώνεται ένα αίτημα AJAX (Asynchronous JavaScript and XML), και τα γεγονότα αυτά καλούν ένα συγκεκριμένο callback, το οποίο με τη σειρά του επιτρέπει την ροή του κώδικα χωρίς να αφήνει ανενεργό τον επεξεργαστή προκειμένου να εκτελεστεί μια λειτουργία. Το Node.js αρχικά, γράφτηκε από τον Ryan Dahl το 2009 και η ανάπτυξη και συντήρηση του καθοδηγούνταν από τον Dahl και χρηματοδοτήθηκε από τον Joyent. Η έκδοση που χρησιμοποιήθηκε στο εκάστοτε έργο της διπλωματικής είναι η v.12.13.1 LTS. Στο σχήμα 2.4 παρουσιάζεται ένα χαρακτηριστικό παράδειγμα Node για έναν HTTP εξυπηρετητή.

```
var http = require('http');
http.createServer(function (req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain'});
  res.end('Hello World\n');
}).listen(1337, '127.0.0.1');
console.log('Server running at http://127.0.0.1:1337/');
```

Σχήμα 2.4: Σύνταξη Node.js για έναν HTTP εξυπηρετητή.

2.2.2 NPM (Node Package Manager)

Το NPM (Node Package Manager) [7] είναι ένας διαχειριστής πακέτων για τη γλώσσα προγραμματισμού JavaScript. Είναι ο προεπιλεγμένος διαχειριστής πακέτων για το περιβάλλον εκτέλεσης JavaScript Node.js (JavaScript runtime-environment). Αποτελείται από έναν client γραμμής εντολών, που ονομάζεται επίσης NPM, και μια ηλεκτρονική βάση δεδομένων τόσο δημόσιων όσο και ιδιωτικών (private paid-for) πακέτων. Τα αρχεία είναι προσβάσιμα από τον client και τα διαθέσιμα πακέτα μπορούν να αναζητηθούν μέσω της ιστοσελίδας του NPM, και υπεύθυνοι για αυτά είναι η NPM, Inc. Το NPM είναι γραμμένο εξ ολοκλήρου σε JavaScript και αναπτύχθηκε από τον Isaac Z. Schlueter με την έμπνευση από άλλα παρόμοια έργα όπως το PEAR (PHP). Ειδικότερα, είναι ένα online αποθετήριο για τη δημοσίευση έργων Node.js και ένα βοηθητικό πρόγραμμα γραμμής εντολών για αλληλεπίδραση με το εν λόγω αποθετήριο που βοηθά στην εγκατάσταση πακέτων, στη διαχείριση των εκδόσεων τους και στη διαχείριση εξάρτησης. Ένα πακέτο ουσιαστικά, είναι ένας φάκελος που περιέχει ένα πρόγραμμα το οποίο περιγράφεται από ένα αρχείο που ονομάζεται “package.json”. Το σχήμα 2.5 δείχνει μια σύνοψη των εντολών χρήσης

npm τόσο για την εγκατάσταση πακέτων όσο και την ενημέρωσή τους.

```
npm install (with no args, in package dir)
npm install [<@scope>/]<name>
npm install [<@scope>/]<name>@<tag>
npm install [<@scope>/]<name>@<version>
npm install [<@scope>/]<name>@<version range>
npm install <git-host>:<git-user>/<repo-name>
npm install <git repo url>
npm install <tarball file>
npm install <tarball url>
npm install <folder>
```

Σχήμα 2.5: Σύνταξη για εγκατάσταση npm/package.

2.2.3 React

Η React, γνωστή και ως React.js ή ReactJS [26], είναι μια βιβλιοθήκη JavaScript για τη δημιουργία διεπαφών χρήστη. Δημιουργήθηκε και διατηρείται από την ομάδα του Facebook και από μια κοινότητα μεμονωμένων προγραμματιστών και εταιρειών. Χρησιμοποιείται ως βάση για την ανάπτυξη εφαρμογών μιας ιστοσελίδας ή κινητών τηλεφώνων, καθώς είναι βέλτιστη ως προς την εξαγωγή ταχέων μεταβαλλόμενων δεδομένων τα οποία πρέπει να καταγράφονται. Ωστόσο, η λήψη δεδομένων είναι μόνο η αρχή όσων συμβαίνουν σε μια ιστοσελίδα, για αυτό και οι σύνθετες εφαρμογές React συνήθως απαιτούν τη χρήση πρόσθετων βιβλιοθηκών, όπως React Router, για το state management, τη δρομολόγηση και την αλληλεπίδραση με ένα API. Ουσιαστικά, είναι το view layer (στρώμα προβολής) για εφαρμογές ιστού. Η React λειτουργεί με τη χρήση components (στοιχείων), δηλαδή μιας αυτοτελούς ενότητας που επιστρέφει κάποιο output. Γράφονται στοιχεία διεπαφής (interface elements), όπως ένα κουμπί ή ένα πεδίο εισαγωγής, ως ένα στοιχείο (component). Τα components μπορεί να είναι σύνθετα, και πιο συγκεκριμένα ένα component μπορεί να περιλαμβάνει ένα ή περισσότερα άλλα components στο output του. Σε γενικές γραμμές, προκειμένου να υλοποιηθεί μια εφαρμογή γράφονται στοιχεία React (React components) που αντιστοιχούν σε διάφορα στοιχεία διεπαφής, και στη συνέχεια οργανώνονται αυτά τα στοιχεία μέσα σε άλλα στοιχεία υψηλότερου επιπέδου που καθορίζουν τη δομή της εφαρμογής. Είναι σημαντικό να αναφερθεί, ότι κάθε στοιχείο σε μια εφαρμογή React

τηρεί αυστηρές αρχές διαχείρισης δεδομένων. Οι σύνθετες, διαδραστικές διεπαφές χρήστη συχνά περιλαμβάνουν σύνθετα δεδομένα και εφαρμογή state. Η επιφάνεια της React είναι περιορισμένη και έχει στόχο να δώσει τα εργαλεία για να μπορέσει να προβλεφθεί ο τρόπος εμφάνισης μια εφαρμογής με ένα δεδομένο σύνολο περιπτώσεων. Ειδικότερα, η React είναι ένα JavaScript framework. Η χρήση αυτού του framework είναι τόσο απλή όσο περιλαμβάνει ένα αρχείο JavaScript στην HTML και χρησιμοποιεί τις εξαγωγές της React σε μια εφαρμογή JavaScript. Σε αντίθεση με πολλούς προκατόχους, η React δε λειτουργεί άμεσα στο DOM (Document Object Model) του προγράμματος περιήγησης, αλλά σε ένα εικονικό DOM (virtual DOM). Αναλυτικότερα, αντί να χειρίζεται το έγγραφο σε ένα πρόγραμμα περιήγησης μετά από αλλαγές στα δεδομένα, που μπορεί να είναι αρκετά αργό, επιλύει τις αλλαγές σε ένα DOM που έχει δημιουργηθεί εξ ολοκλήρου στη μνήμη. Στη συνέχεια, αφού ενημερωθεί το εικονικό DOM, η React καθορίζει με έξυπνο τρόπο τις αλλαγές που πρέπει να κάνει στο DOM του πραγματικού προγράμματος περιήγησης. Εξίσου σημαντικό να αναφερθεί, είναι η συνεχόμενη εξέλιξη και ανανέωση της React από τους εμπειρότερους προγραμματιστές του κόσμου. Η έκδοση που ακολούθησε το συγκεκριμένο έργο είναι η v.16.12.0. Στο σχήμα 2.6 παρουσιάζεται η σύνταξη JSX (JavaScript XML) ενός component σε React.

```
class HelloMessage extends React.Component {
  render() {
    return (
      <div>
        Hello {this.props.name}
      </div>
    );
  }
}

ReactDOM.render(
  <HelloMessage name="Taylor" />,
  document.getElementById('hello-example')
);
```

Σχήμα 2.6: Παράδειγμα σύνταξης JSX.

2.2.4 Express.js

Το Express.js, ή απλά το Express, [14] είναι ένα Web application framework για το Node.js, το οποίο κυκλοφορεί ως ελεύθερο και ανοιχτό λογισμικό (open-source software) με την άδεια του MIT (MIT License - Massachusetts Institute of Technology). Έχει σχεδιαστεί για την κατασκευή εφαρμογών ιστού και API, έχει ονομαστεί ως το "de facto standard server framework" και επίσης, έχει χαρακτηριστεί ως "ένα γρήγορο και minimalist server-side Web framework για το Node.js. Γραμμένο σε γλώσσα JavaScript, το Express λειτουργεί ως ένα λεπτό στρώμα βασικών λειτουργιών σε εφαρμογές Web, σε αντίθεση με το Ruby framework για το Rails, δεν έχει καμία σχέση με την αντιστοίχιση αντικειμένων (object relational mapping). Το Express δεν είναι κατασκευασμένο γύρω από συγκεκριμένα συστατικά μέρη και δεν έχει καμία "γνώμη" σχετικά με τις τεχνολογίες που συνδέονται με αυτό. Προσπαθεί να θέσει τον έλεγχο στα χέρια του προγραμματιστή και να καταστήσει ευκολότερη την ανάπτυξη εφαρμογών Web για το Node.js. Αυτή η ελευθερία, σε συνδυασμό με το γρήγορο και καθαρό JavaScript περιβάλλον του Node, καθορίζει το Express έναν ισχυρό υποψήφιο για την γρήγορη ανάπτυξη και δημιουργία πρωτοτύπων, και ιδιαίτερα για επιχειρήσεις που θέλουν να κατασκευάσουν ένα προϊόν το συντομότερο δυνατό. Το περιβάλλον Node.js αποτελεί ένα βασικό κομμάτι του τι κάνει την Express τόσο εύκολη στην κατασκευή και την ανάπτυξη. Η εγκατάσταση του Express γίνεται μέσω του NPM. Στο σχήμα 2.7 γίνεται αντιληπτό ότι με μερικές γραμμές κώδικα και την Express βιβλιοθήκη, η δημιουργία ενός απλού server ο οποίος είναι ικανός να επικοινωνεί με Web browsers μέσω του localhost στο port 8000.

```
server.js
...
var express = require('express');
var app = express();

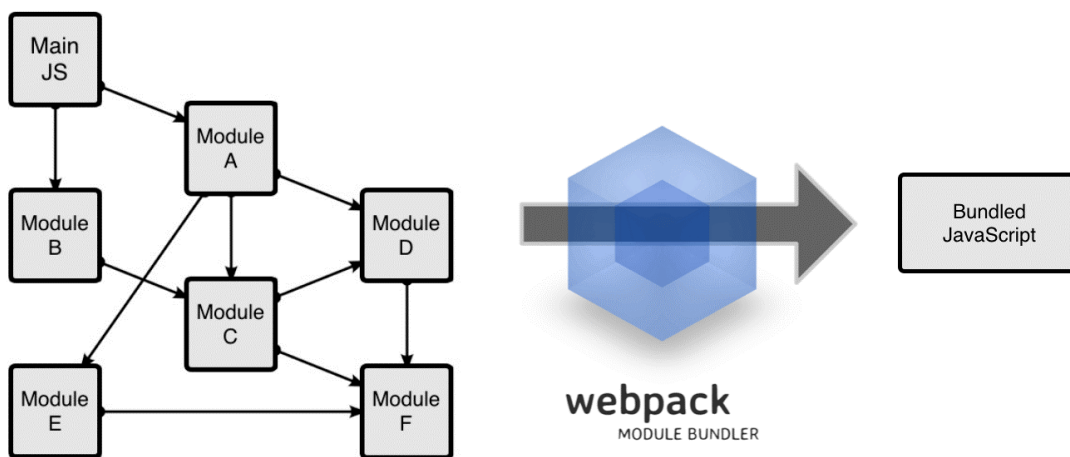
app.get('/', function(req, res){
    res.send('<h1>Hello World!</h1>');
});

app.listen(8000);
...
```

Σχήμα 2.7: Εισαγωγή του Express.js στο Node.js και δημιουργία server.

2.2.5 Webpack

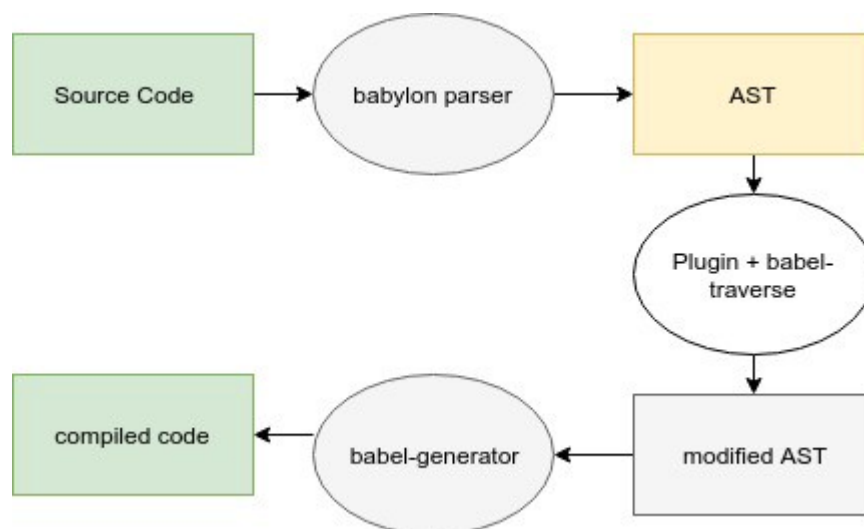
Το Webpack [12] είναι ένα πακέτο στοιχείων JavaScript ανοιχτού κώδικα (open-source JavaScript module bundler). Κυρίως είναι ένας συνδυασμός στοιχείων για JavaScript, αλλά μπορεί να μετασχηματίσει και στοιχεία front-end όπως HTML, CSS και εικόνες αν συμπεριληφθούν οι αντίστοιχοι φορτωτές. Το Webpack παίρνει τα στοιχεία με εξαρτήσεις (module with dependencies) και παράγει στατικά στοιχεία που αντιπροσωπεύουν αυτά τα στοιχεία, δηλαδή λαμβάνει τις εξαρτήσεις και δημιουργεί ένα γράφημα εξάρτησης που επιτρέπει τους προγραμματιστές να χρησιμοποιούν με αρθρωτή προσέγγιση για τους σκοπούς ανάπτυξης εφαρμογών ιστού. Μπορεί να χρησιμοποιηθεί από το command line ή να ρυθμιστεί χρησιμοποιώντας ένα αρχείο ρυθμίσεων που ονομάζεται “webpack.config.js”. Είναι εξαιρετικά επεκτάσιμο μέσω κανόνων που επιτρέπουν στους προγραμματιστές να γράφουν custom tasks που θέλουν να εκτελέσουν όταν συνδυάζουν τα αρχεία μαζί. Η χρήση του Webpack απαιτεί το Node.js. Η React όπως και η HMR (Hot Module Replacement) βοήθησαν στη διάδοση του Webpack και σε άλλα περιβάλλοντα, όπως η Ruby στο Rails. Έτσι, πλέον το Webpack υπάρχει προ εγκατεστημένο (pre-build) στα περισσότερα frameworks λόγω του μεγάλου μεγέθους των βιβλιοθηκών που χρησιμοποιούνται για την τελική εφαρμογή. Στο σχήμα 2.8 παρουσιάζονται οι λειτουργίες που εκτελούνται.



Σχήμα 2.8: Λειτουργίες του Webpack (workflow).

2.2.6 Babel

Η Babel [1] είναι ένας μεταγλωττιστής (compiler) ανοιχτού κώδικα JavaScript που χρησιμοποιείται κυρίως για την μετατροπή του κώδικα των εκδόσεων ECMAScript 2015 και άνω, σε μια συμβατή με το παρελθόν έκδοση της JavaScript, η οποία μπορεί να χρησιμοποιηθεί από παλαιότερους μηχανισμούς της. Είναι ένα δημοφιλές εργαλείο για τη χρήση των νέων δυνατοτήτων της γλώσσας προγραμματισμού JavaScript. Μέσω του Babel, οι προγραμματιστές μπορούν να χρησιμοποιήσουν νέες λειτουργίες γλώσσας JavaScript για να μετατρέψουν τον πηγαίο κώδικα σε εκδόσεις JavaScript που επεξεργάζονται οι εξελιγμένες μηχανές αναζήτησης. Παραδείγματα αυτών των νέων λειτουργιών είναι οι λειτουργίες βέλους, οι οποίες καθορίζονται στο ES6 και μετατρέπονται σε κανονικές δηλώσεις λειτουργίας, καθώς και η μη τυποποιημένη σύνταξη της JavaScript (JSX) μπορεί επίσης να μετασχηματιστεί. Το σχήμα 2.9 προβάλλει τον τρόπο με τον οποίο εκτελεί τις λειτουργίες.

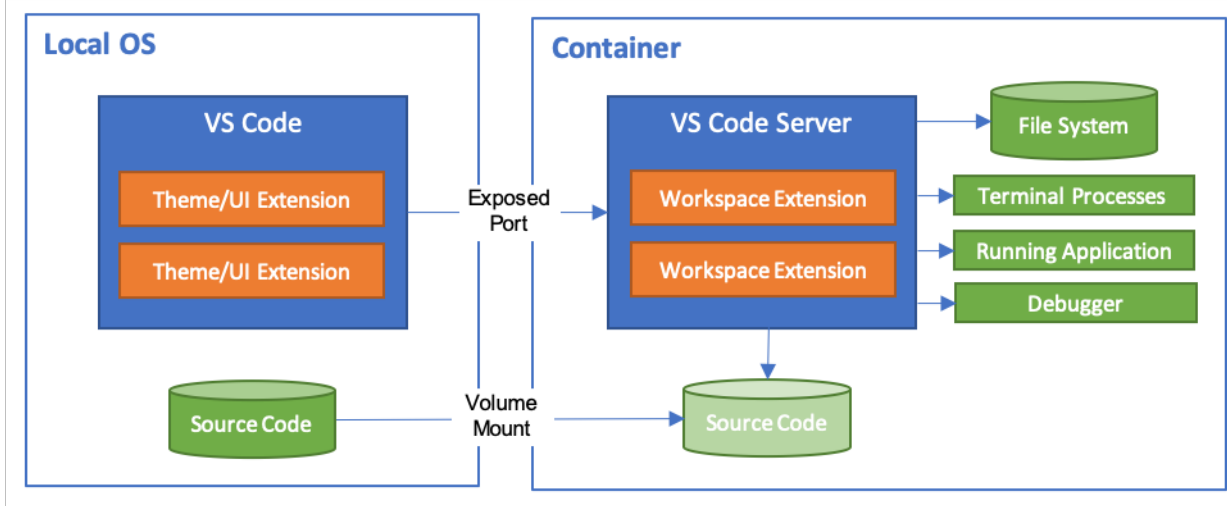


Σχήμα 2.9: Λειτουργίες του Babel (workflow).

2.2.7 Visual Studio Code

Το Visual Studio Code [10] είναι ένας επεξεργαστής πηγαίου κώδικα (source-code editor) που αναπτύχθηκε από την Microsoft για Windows, Linux και macOS. Περιλαμβάνει υποστήριξη για εντοπισμό σφαλμάτων (debugging), ενσωματωμένο έλεγχο Git και GitHub, επισήμανση σύνταξης, έξυπνη ολοκλήρωση κώδικα, snippets και αναδρομολόγηση κώδικα. Είναι ιδιαίτερα προσαρμόσιμο, επιτρέποντας στους

χρήστες να αλλάζουν το θέμα, να χρησιμοποιούν συντομεύσεις πληκτρολογίου, προ-σθήκη προτιμήσεων και να εγκαθιστούν επεκτάσεις που προσθέτουν πρόσθετες λει-τουργίες. Ο πηγαίος κώδικας είναι ελεύθερος και ανοιχτός κώδικας και κυκλοφορεί υπό την επιτρεπόμενη άδεια MIT License. Στο σχήμα 2.10 παρουσιάζονται οι λει-τουργίες του.



Σχήμα 2.10: Λειτουργίες του Visual Studio Code (workflow).

2.3 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό, αναλύθηκαν περιγραφικά και εικονικά οι κύριες τεχνολογίες που χρησιμοποιήθηκαν στη συγκεκριμένη διπλωματική εργασία. Η ανάλυση αυτή, αφορά τις σημαντικότερες τεχνολογίες που χρειάστηκαν τόσο για την υλοποίηση του front-end όσο και του back-end. Ειδικότερα, γίνεται μια θεωρητική εισαγωγή, αναφέροντας την ιστορική πτυχή των τεχνολογιών αυτών, αλλά και την εξέλιξη που είχαν.

Κεφάλαιο 3

Δομικά μέρη συστήματος

Αυτό το κεφάλαιο αναφέρεται σε όλα τα δομικά στοιχεία από τα οποία αποτελείται το εκάστοτε σύστημα του ιστοχώρου που αναπτύχθηκε. Τα συστατικά αυτά λοιπόν, που διαμόρφωσαν την δομή αυτού του έργου έχουν τοποθετηθεί και αναλυθεί με τον καλύτερο δυνατό τρόπο παρακάτω. Οι κατηγορίες που κατατάσσονται τα δομικά μέρη είναι οι απαιτήσεις που χρειάστηκε το σύστημα, οι περιπτώσεις χρήσης (use cases) και η βάση των δεδομένων.

3.1 Απαιτήσεις συστήματος

Ένας από τους σημαντικότερους συντελεστές, τόσο για την εκκίνηση αλλά και την πορεία κάθε επιτυχημένου έργου, είναι ο προσδιορισμός των απαιτήσεων που διαμορφώνει ένα σύστημα. Ειδικότερα, για το παρόν έργο, στόχος ήταν η ανάπτυξη καθώς και ο σχεδιασμός ενός ιστοτόπου. Προκειμένου να επιτευχθεί αυτό, έπρεπε να καθοριστούν με σαφήνεια τα προαπαιτούμενα στοιχεία του συστήματος. Παρακάτω παρατίθεται η διαίρεση όλων των απαιτήσεων η οποία ακολουθείται από μια εκτενή ανάλυση.

- **Πρόσβαση στο Διαδίκτυο.** Είναι πολύ σημαντικό να υπάρχει επικοινωνία σε όλα τα επίπεδα του συστήματος. Για να γίνει αυτό, απαραίτητη είναι η σύνδεση στο Διαδίκτυο. Χωρίς τη συγκεκριμένη προϋπόθεση το σύστημα δεν έχει καμία χρησιμότητα, αφού δεν θα είναι εφικτή η επικοινωνία του επισκέπτη και χρήστη με αυτό. Ο επισκέπτης είναι αδύνατο να ενημερωθεί για τις εκδηλώσεις και ο χρήστης χάνει τις πρόσθετες δυνατότητες που αποκτά κατά την είσοδο.

-
- **Το περιβάλλον χρήσης ακολουθεί φιλικά προς τον άνθρωπο πρότυπα σχεδίασης (User Interface friendly and responsive design).** Η διεπαφή σχεδιάστηκε με βάση τις πιο πρόσφατες έρευνες σχετικά με τα πρότυπα διεπαφής χρήστη (UI) και της εμπειρίας χρήστη (UX) με στόχο ένα περιβάλλον που είναι απλό και φιλικό προς τον χρήστη ώστε να καλύψει όλες τις ανάγκες των ανθρώπων και ιδιαίτερα αυτών που δεν έχουν αρκετές γνώσεις ή εμπειρία στην χρήση σύγχρονων συσκευών και Η/Υ. Διατηρεί σχεδόν σε όλες τις, διαφορετικών διαστάσεων, οθόνες συσκευών την ποιότητα και την εμφάνιση των λειτουργιών.
 - **Σωστή αρχιτεκτονική και καθαρός κώδικας.** Η κατανομή των συστατικών (components) του κώδικα και η συγγραφή του, ακολούθησε την ορθή πολιτική του software development για την ευκολότερη κατανόηση και τη δυνατότητα επέκτασης του υλικού. Αυτό βοήθησε στην καλύτερη διαχείριση του χρόνου υλοποίησης του προγραμματιστή, στο αναμενόμενο debugging, στην καταλληλότερη ανταπόκριση και αξιοποίηση του συστήματος καθώς και σε μια πιο θεμιτή ανάγνωση του κώδικα από πιθανούς μελλοντικούς συνεργάτες.
 - **Ενσωμάτωση ασφάλειας σε όλο το σύστημα.** Η τοποθέτηση –deploying– μιας εφαρμογής ή ιστοσελίδας στο Διαδίκτυο είναι μια πολύ σημαντική ευθύνη του developer. Για το λόγο αυτό, αλλά και επειδή πρέπει τα προσωπικά στοιχεία του κάθε χρήστη να κρατούνται ασφαλή, έχουν προστεθεί οι κατάλληλες μορφές ασφάλειας σε όλους τους τομείς του συστήματος, ώστε κάθε επικοινωνία που πραγματοποιείται μέσω της ιστοσελίδας, των χρηστών και των διαύλων να είναι ασφαλής.
 - **Η διαχείριση του συστήματος αποτελείται από δυο κατηγορίες χρηστών.** Ο ιστοχώρος προκειμένου να ενημερώνει όλους όσους τον επισκέπτονται, δίνει τις εξής δυνατότητες. Όσοι πληκτρολογούν την εκάστοτε διεύθυνση της ιστοσελίδας σε ένα πρόγραμμα περιήγησης, μπορούν να ενημερωθούν για όλες τις εκδηλώσεις που υπάρχουν στο σύστημα χωρίς να διαθέτουν λογαριασμό. Αντίθετα οι χρήστες της ιστοσελίδας, διαθέτουν κάποιες πρόσθετες δυνατότητες όπου τους επιτρέπεται η δημιουργία μιας εκδήλωσης, η μεταφόρτωση εικόνων, η προσθήκη και αφαίρεση σχολίου και η επιλογή για αναφορά αυτής

στον δημιουργό της.

3.2 Περιπτώσεις χρήσης (use cases)

Η ενότητα που ακολουθεί επεξηγεί με αναλυτικό τρόπο τις λειτουργίες που διαθέτουν όσοι επισκέπτονται τον ιστοχώρο. Ειδικότερα, το σύστημα που κατασκευάστηκε υποστηρίζει δυο κατηγορίες χρηστών. Αυτές αποτελούνται από τους επισκέπτες της ιστοσελίδας, οι οποίοι πρακτικά δεν είναι χρήστες και απλά βλέπουν το περιεχόμενο της και από τους χρήστες οι οποίοι έχοντας εγγραφεί και εισέλθει στο σύστημα αποκτούν ορισμένες δυνατότητες.

- Επισκέπτης

Use case 1: Εμφάνιση του περιεχομένου της ιστοσελίδας. Ο επισκέπτης όπως και ο χρήστης αφού εισέλθει στον ιστοχώρο, πληκτρολογώντας την αντίστοιχη διεύθυνση, έχει τη δυνατότητα να πληροφορηθεί για τις εκδηλώσεις που υπάρχουν στο σύστημα και να δει το περιεχόμενό τους. Δηλαδή, τις κατηγορίες των εκδηλώσεων, τις εικόνες τους, την τοποθεσία τους και την ημερομηνία που διεξάγονται. Επιπλέον, κάνοντας κλικ στο κουμπί “All Events”, δηλαδή στην προεπισκόπηση εκδηλώσεων που υπάρχει στην ιστοσελίδα, ο επισκέπτης βλέπει όλες τις εκδηλώσεις που υπάρχουν στο σύστημα.

Use case 2: Αναζήτηση εκδήλωσης. Ο επισκέπτης όπως αντίστοιχα και ο χρήστης μπορεί να αναζητήσει συγκεκριμένη εκδήλωση που πιθανόν τον ενδιαφέρει, χρησιμοποιώντας διάφορα φίλτρα ως προς την κατηγορία, τον νομό, την πόλη ή την ημερομηνία.

Use case 3: Εγγραφή και είσοδος στο σύστημα. Στην περίπτωση που ο επισκέπτης επιθυμεί να εγγραφεί και να γίνει μέλος του συστήματος, μπορεί να επιλέξει το πεδίο “Authenticate” που εμφανίζεται στην ιστοσελίδα. Εκεί, το εμφανίζονται δυο επιλογές, μπορεί να εγγραφεί στο σύστημα επιλέγοντας το κουμπί “Sign up”, αντίθετα, αν διαθέτει ήδη λογαριασμό, μπορεί να κάνει είσοδο επιλέγοντας το πεδίο “Log in“. Τα στοιχεία που απαιτούνται για την εγγραφή στο σύστημα είναι το όνομα (username), το ηλεκτρονικό ταχυδρομείο (email) και ο κωδικός (password).

- Χρήστης

Use case 1: Είσοδος στο σύστημα. Ο χρήστης μπορεί να εισέλθει στο σύστημα συμπληρώνοντας τα προσωπικά στοιχεία που επέλεξε κατά την εγγραφή του, συγκεκριμένα το email και το password, στην κατάλληλα διαμορφωμένα φόρμα εισόδου που εμφανίζεται κάνοντας κλικ στο πεδίο “Authenticate”. Η είσοδος θα πραγματοποιείται αν και μόνο αν ο συνδυασμός των στοιχείων στα πεδία είναι σωστός. Επιπρόσθετα, αφού ο χρήστης πλέον βρίσκεται εντός του συστήματος, θα μπορεί να δει τις πρόσθετες λειτουργίες που του δίνονται, δηλαδή θα εμφανίζονται στην οθόνη, στις αντίστοιχες σελίδες του συστήματος, οι επιπλέον δυνατότητες που έχει ως χρήστης του συστήματος. Τέλος, με την είσοδο του στο σύστημα, ο χρήστης ειδοποιείται με κατάλληλο email το οποίο αναφέρει ότι η εγγραφή του στο σύστημα ήταν επιτυχής, και το email αυτό αποστέλλεται στην προσωπική ηλεκτρονική διεύθυνση που εισήγαγε στην φόρμα εγγραφής.

Use case 2: Δημιουργία μιας εκδήλωσης. Επιλέγοντας το πεδίο “Create Event” που εμφανίζεται στο header των σελίδων, ο χρήστης έχει τη δυνατότητα να γίνει δημιουργός και να προσθέσει στο σύστημα μια εκδήλωση συμπληρώνοντας σε ένα διαμορφωμένο πλαίσιο μιας φόρμας τις πληροφορίες που απαιτούνται, δηλαδή τον τίτλο, την διεύθυνση, την ημερομηνία έναρξης, την ημερομηνία λήξης, την ώρα έναρξης, την περιγραφή, την πόλη, τον νομό, την κατηγορία και μεταφορτώνει τη φωτογραφία που θα αντιπροσωπεύει την εκδήλωση. Αναλυτικά, ο χρήστης / δημιουργός δίνει τον τίτλο και γράφει μια σύντομη περιγραφή, αποφασίζει το διάστημα που θα διαρκέσει μια εκδήλωση, δηλαδή ορίζει την ημερομηνία όπως και την ώρα έναρξης που αυτή θα πραγματοποιείται, επιλέγει την κατηγορία, τον νομό και την πόλη από μια λίστα που εμφανίζεται όταν κάνει κλικ στο αντίστοιχο κουμπί, η οποία εμφανίζει στην οθόνη όλες τις διαθέσιμες επιλογές που περιλαμβάνονται στο σύστημα, μεταφορτώνει όσες φωτογραφίες επιθυμεί καθώς και την συγκεκριμένη διεύθυνση που θα διεξαχθεί η εκδήλωση.

Use case 3: Διαχείριση εκδηλώσεων χρήστη. Ο χρήστης έχει τη δυνατότητα κάνοντας κλικ στο κουμπί “My Events” να βλέπει τις εκδηλώσεις που ο ίδιος δημιούργησε. Στη σελίδα αυτή, σε κάθε εκδήλωση ξεχωριστά, του δίνονται οι δυνατότητες για επεξεργασία και διαγραφή της εκδήλωσης καθώς και προβολή της τοποθεσίας που διεξάγεται στον χάρτη. Ειδικά, κάνοντας κλικ στο κουμπί επεξεργασίας “Edit” ο

διαχειριστής της εκάστοτε εκδήλωσης, μπορεί να κάνει μετατροπή ορισμένων πεδίων, όπως είναι ο τίτλος, ο διεύθυνση, οι ημερομηνίες έναρξης και λήξης, η ώρα έναρξης και η περιγραφή της. Η επεξεργασία των πεδίων είναι ευέλικτη, αφού μπορούν να μετατραπούν είτε όλα είτε κάθε ένα ξεχωριστά. Αλλαγή του πεδίου διεύθυνσης, σημαίνει αυτόματη αλλαγή και της τοποθεσίας της στο χάρτη. Επιπλέον, όπως υπάρχει το ενδεχόμενο να αλλάξουν κάποια από τα στοιχεία μιας εκδήλωσης για λόγους του δημιουργού, έτσι και σε περιπτώσεις όπου για παράδειγμα ακυρωθεί μια εκδήλωση αναπάντεχα, δίνεται η επιλογή αφαίρεσης της εκδήλωσης αυτής κάνοντας κλικ στο κουμπί διαγραφής “Delete”, και η εκδήλωση διαγράφεται από το σύστημα.

Use case 4: Διαχείριση πολυμεσικού περιεχομένου. Ο χρήστης προσθέτει μια ή περισσότερες εικόνες, χωρίς όριο στην ποσότητα, κάνοντας κλικ στο κουμπί “Pick Images“ που θα εμφανίζεται στην οθόνη, κατά τη διάρκεια της δημιουργίας μιας εκδήλωσης, ώστε να μεταφορτωθούν οι επιλεγμένες εικόνες στο σύστημα και έτσι οι εικόνες αυτές να εμφανίζονται στη συγκεκριμένη εκδήλωση που δημιούργησε, ώστε να είναι διαθέσιμες για προβολή και από τους υπόλοιπους χρήστες.

Use case 5: Προσθήκη σχολίου. Ο χρήστης μπορεί να προσθέσει πληκτρολογώντας στο αντίστοιχο πεδίο (comment text-area), που υπάρχει στη σελίδα της κάθε εκδήλωσης, το μήνυμα που θέλει να στείλει στο σύστημα ώστε να εμφανίζεται στη σελίδα της συγκεκριμένης εκδήλωσης, όπως επίσης έχει και την επιλογή να το αφαιρέσει στη συνέχεια κάνοντας κλικ στο κουμπί “Delete” που εμφανίζεται δίπλα από το σχόλιο και έτσι αφαιρεί το σχόλιο.

Use case 6: Διαδικτυακός χάρτης. Ο χρήστης κάνοντας κλικ στο κουμπί “View on map” που βρίσκεται σε κάθε εκδήλωση ξεχωριστά, μπορεί να δει στον ηλεκτρονικό χάρτη την ακριβή τοποθεσία που διεξάγεται η εκδήλωση που τον ενδιαφέρει. Ειδικότερα, αυτό επιτυγχάνεται, συμπληρώνοντας απλώς την επιθυμητή διεύθυνση στο πεδίο που του ζητείται στην φόρμα δημιουργίας εκδήλωσης.

Use case 7: Δυνατότητα αναφοράς. Στην περίπτωση που ο χρήστης θεωρεί πως το περιεχόμενο μιας εκδήλωσης δεν είναι ηθικό ή πως του προσβάλλει την αισθητική, έχει τη δυνατότητα κάνοντας κλικ στο κουμπί που εμφανίζεται στη σελίδα της εκδήλωσης, να αναφέρει την εκδήλωση αυτή. Ο χρήστης μπορεί να αναφέρει μόνο μια φορά την κάθε εκδήλωση. Στην περίπτωση που έχουν αναφέρει μια εκδήλωση,

περισσότεροι από έναν συγκεκριμένο αριθμό χρηστών, τότε ο δημιουργός της εκδήλωσης ειδοποιείται με κατάλληλο email, το οποίο τον ενημερώνει για την περίπτωση αυτή και του ζητά να αφαιρέσει αυτή την εκδήλωση από το σύστημα.

Use case 8: Αλλαγή προσωπικών στοιχείων. Αν για οποιοδήποτε λόγο ο χρήστης αδυνατεί να εισέλθει στο σύστημα ή έχει ξεχάσει τον κωδικό πρόσβασης ή θέλει να τον αλλάξει, μπορεί από τη σελίδα όπου υπάρχουν οι φόρμες συμπλήρωσης εγγραφής ή εισόδου να επιλέξει κάνοντας κλικ στο κουμπί “Reset password” να εισάγει το νέο κωδικό που επιθυμεί για να έχει τη δυνατότητα να συνεχίσει να είναι χρήστης του συστήματος. Επιπλέον, του δίνεται η δυνατότητα αυτή και εφόσον έχει εισέλθει στον ιστοχώρο, από την σελίδα του προφίλ του. Τέλος, ο χρήστης ειδοποιείται με email όταν η τροποποίηση γίνει επιτυχής.

Use case 9: Έξοδος από το σύστημα. Υπάρχει πεδίο “Log out” στο header που εμφανίζεται σε όλες τις σελίδες, το οποίο αποσυνδέει τον χρήστη από τον ιστοχώρο. Έτσι, ο χρήστης χάνει τις πρόσθετες λειτουργίες που του παρείχε το σύστημα και γίνεται επισκέπτης.

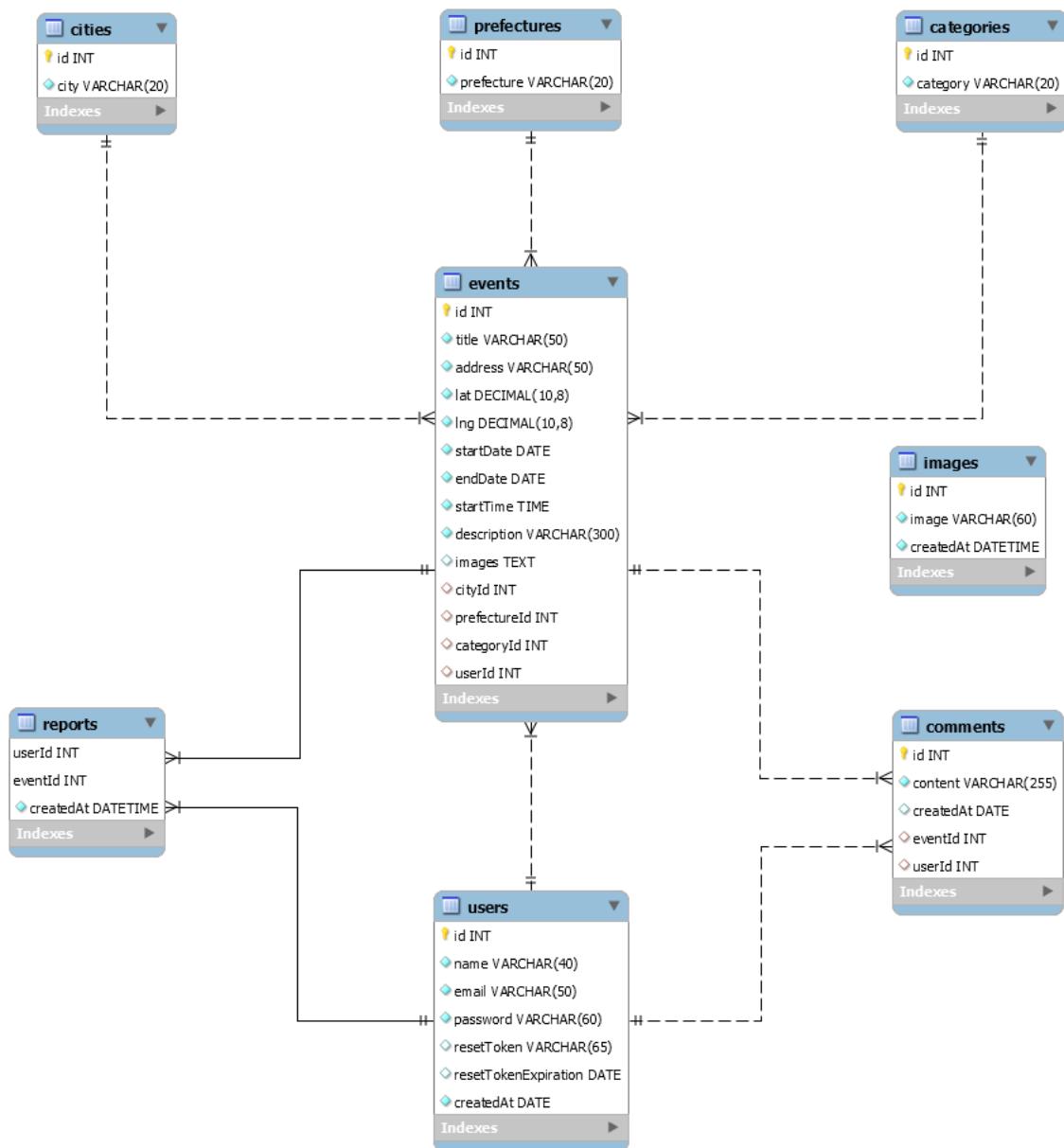
3.3 Ανάλυση βάσης δεδομένων (Database)

Η ενότητα που ακολουθεί αναφέρεται σε όλα τα δεδομένα που είναι απαραίτητα για την ορθή λειτουργία του συστήματος. Η εφαρμογή ενός συστήματος διαχείρισης δεδομένων προωθεί μια ολοκληρωμένη εικόνα των λειτουργιών ενός οργανισμού. Η χρήση αυτού του συστήματος αυξάνει την αποδοτικότητα των διεργασιών, μειώνει την πολυπλοκότητα των στοιχείων και δίνει την δυνατότητα στον προγραμματιστή να διαχειρίζεται με μεγαλύτερη ευκολία τα δεδομένα και τις πληροφορίες. Όταν η δημιουργία μιας βάσης δεδομένων αποτελεί αρχική προτεραιότητα σε ένα σύστημα, εξασφαλίζει ένα μεγάλο ποσοστό για την επιτυχία του. Προκειμένου να επιτευχθεί ο ιδανικός σχεδιασμός πρέπει να οριστεί με σαφήνεια η οργάνωση της βάσης ώστε να απλοποιηθεί η διαχείρισή της. Αυτό είναι εφικτό, με την κανονικοποίηση (normalization) της βάσης, χάρη στην οποία αποφεύγεται ο περιττός χρόνος κατανόησης της, και προσδίδει τη δυνατότητα για πιθανό μελλοντικό μετασχηματισμό καθώς και επέκτασης ή αναβάθμισης της. Παρακάτω παρουσιάζονται το διάγραμμα οντοτήτων και σχεσιακό διάγραμμα (relational diagram) και προβάλλονται όλοι οι

πίνακες που απαρτίζουν τη βάση δεδομένων.

3.3.1 Enhanced Entity – Relationship Diagram (EER model)

Το μοντέλο ενισχυμένης οντότητας – σχέσης, στην επιστήμη των υπολογιστών είναι ένα υψηλού επιπέδου ή εννοιολογικής ενσωμάτωσης μοντέλο δεδομένων που χρησιμοποιείται για τον σχεδιασμό βάσεων δεδομένων. Ακολουθεί το σχήμα 3.1 που αντιπροσωπεύει την βάση δεδομένων του συστήματος, παρουσιάζοντας όλους τους πίνακες με τα πεδία τους, τις εξαρτήσεις μεταξύ των πινάκων και τους τύπους των δεδομένων.



Σχήμα 3.1: EER diagram Events.

3.3.2 Πίνακας “events”

Η στήλη “Όνομα” υποδεικνύει τις ονομασίες των πεδίων, η στήλη “Τύπος” τους τύπους των γνωρισμάτων, η στήλη “Κενό” είναι ο περιορισμός “Null” και μπορεί να είναι είτε «Ναι» για όσα πεδία δέχονται κενή τιμή είτε «Όχι» για αυτά που πρέπει να δέχονται μια τιμή. Ακόμα, η στήλη “Πρόσθετα” είναι οι επιπλέον επιλογές που μπορούν να έχουν τα πεδία και τέλος η στήλη “Προεπιλογή” δηλώνει μια τιμή, ένα χαρακτήρα ή ένα σύμβολο που μπορούν να έχουν ως προεπιλογή. Στον πίνακα 3.1 παρουσιάζεται η δομή του πίνακα “events”. Ο συγκεκριμένος πίνακας περιέχει τα βασικά στοιχεία των εκδηλώσεων.

Όνομα	Τύπος	Κενό	Πρόσθετα	Προεπιλογή
<u>id</u>	INT	Όχι	Auto Increment	Καμία
title	VARCHAR(50)	Όχι	-	Καμία
address	VARCHAR(50)	Όχι	-	Καμία
lat	DECIMAL(10,8)	Όχι	-	Καμία
lng	DECIMAL(10,8)	Όχι	-	Καμία
startDate	DATE	Όχι	-	Καμία
endDate	DATE	Όχι	-	Καμία
startTime	TIME	Όχι	-	Καμία
description	VARCHAR(300)	Όχι	-	Καμία
images	TEXT	Όχι	-	Καμία
<u>userId</u>	INT	Όχι	-	Καμία
<u>cityId</u>	INT	Όχι	-	Καμία
<u>prefectureId</u>	INT	Όχι	-	Καμία
<u>categoryId</u>	INT	Όχι	-	Καμία

Πίνακας 3.1: Δομή του πίνακα “events”

Ανάλυση των στηλών του πίνακα “events”:

- **id:** Δηλώνει το αναγνωριστικό της κάθε εκδήλωσης και είναι το πρωτεύον κλειδί (Primary Key) του πίνακα. Ο τύπος που έχει το συγκεκριμένο γνώρισμα έχει καθοριστεί ως ακέραιος (INT) με σκοπό να καλύψει μεγάλες τιμές για

αρκετό μελλοντικό διάστημα. Η τιμή αυξάνεται αυτόματα (Auto Increment) με κάθε νέα δημιουργία ενός event. Φυσικά δεν μπορεί να είναι μηδενική τιμή (Not Null).

- **title:** Δηλώνει την ονομασία της εκδήλωσης με τύπο γνωρίσματος συμβολοσειράς (VARCHAR) έως 50 χαρακτήρες. Το όνομα το συμπληρώνει ο χρήστης / δημιουργός της εκδήλωσης. Δεν μπορεί να έχει κενή τιμή (Not Null).
- **address:** Δηλώνει την διεύθυνση που αντιστοιχεί στις εκδηλώσεις. Ο τύπος γνωρίσματος είναι συμβολοσειρά (VARCHAR) έως 50 χαρακτήρες. Η τιμή δεν μπορεί να είναι κενή (Not Null).
- **lat:** Δηλώνει το γεωγραφικό πλάτος, δηλαδή τον γεωγραφικό συντελεστή που προσδιορίζει μια θέση ενός σημείου στην επιφάνεια της Γης. Όπως είναι γνωστό, μια τιμή γεωγραφικού πλάτους μπορεί να είναι μεταξύ -90 και +90 βαθμούς. Για να επιτευχθεί ακρίβεια σε ένα μέτρο, απαιτούνται έξι δεκαδικά ψηφία, τα οποία είναι επαρκή για τις περισσότερες περιπτώσεις. Χρησιμοποιώντας όμως οκτώ μέρη, υπάρχει μεγαλύτερη ακρίβεια. Άρα, ο τύπος γνωρίσματος είναι δεκαδικός αριθμός DECIMAL(10,8). Δεν μπορεί να έχει κενή τιμή (Not Null).
- **lng:** Δηλώνει το γεωγραφικό μήκος, δηλαδή τον γεωγραφικό συντελεστή που προσδιορίζει μια θέση ενός σημείου στην επιφάνεια της Γης. Αντίθετα με το γεωγραφικό πλάτος, η τιμή του μπορεί να είναι μεταξύ -180 και +180 μοίρες. Άρα, ο τύπος γνωρίσματος είναι δεκαδικός αριθμός DECIMAL(10,8). Δεν μπορεί να έχει κενή τιμή (Not Null).
- **startDate:** Δηλώνει την ημερομηνία έναρξης μιας εκδήλωσης. Ο τύπος γνωρίσματος είναι ημερομηνία (DATE). Η τιμή δεν μπορεί να είναι κενή (Not Null).
- **endDate:** Δηλώνει την ημερομηνία λήξης μιας εκδήλωσης. Ο τύπος γνωρίσματος είναι ημερομηνία (DATE). Η τιμή δεν μπορεί να είναι κενή (Not Null).
- **startTime:** Δηλώνει την ώρα έναρξης μιας εκδήλωσης. Ο τύπος γνωρίσματος είναι ώρα (TIME). Η τιμή δεν μπορεί να είναι κενή (Not Null).

-
- **description:** Δηλώνει την περιγραφή που εκπροσωπεί την εκδήλωση. Ο τύπος γνωρίσματος είναι συμβολοσειρά (VARCHAR) έως 300 χαρακτήρες. Δεν μπορεί να είναι κενή η τιμή (Not Null).
 - **images:** Δηλώνει το Array που περιλαμβάνει όλες τις εικόνες που κάνει upload ο χρήστης / δημιουργός. Ο τύπος γνωρίσματος είναι TEXT διότι το συγκεκριμένο πεδίο έχει τροποποιηθεί έτσι ώστε να δέχεται μεγάλο κείμενο, δηλαδή πολλά hashed ονόματα εικόνων. Δεν μπορεί να έχει κενή τιμή (Not Null).
 - **userId:** Δηλώνει το αναγνωριστικό που αντιστοιχεί σε ένα χρήστη καθώς και τον συγκεκριμένο δημιουργό της αντίστοιχης εκδήλωσης. Αποτελεί το πρωτεύον κλειδί (Primary Key) του πίνακα “users” και ξένο κλειδί (Foreign Key) στον πίνακα “events”. Ο τύπος γνωρίσματος είναι ακέραιος αριθμός (INT) και δεν μπορεί να έχει μηδενική τιμή (Not Null).
 - **cityId:** Δηλώνει το αναγνωριστικό που αντιστοιχεί σε ένα χρήστη καθώς και την πόλη της αντίστοιχης εκδήλωσης. Αποτελεί το πρωτεύον κλειδί (Primary Key) του πίνακα “cities” και ξένο κλειδί (Foreign Key) στον πίνακα “events”. Ο τύπος γνωρίσματος είναι ακέραιος αριθμός (INT) και δεν μπορεί να έχει μηδενική τιμή (Not Null).
 - **prefectureId:** Δηλώνει το αναγνωριστικό που αντιστοιχεί σε ένα χρήστη καθώς και τον νομό της αντίστοιχης εκδήλωσης. Αποτελεί το πρωτεύον κλειδί (Primary Key) του πίνακα “prefectures” και ξένο κλειδί (Foreign Key) στον πίνακα “events”. Ο τύπος γνωρίσματος είναι ακέραιος αριθμός (INT) και δεν μπορεί να έχει μηδενική τιμή (Not Null).
 - **categoryId:** Δηλώνει το αναγνωριστικό που αντιστοιχεί σε μια κατηγορία καθώς και την κατηγορία στην οποία ανήκει η εκδήλωση. Αποτελεί το πρωτεύον κλειδί (Primary Key) του πίνακα “categories” και ξένο κλειδί (Foreign Key) στον πίνακα “events”. Ο τύπος γνωρίσματος είναι ακέραιος αριθμός (INT) και δεν μπορεί να έχει μηδενική τιμή (Not Null).

3.3.3 Πίνακας “users”

Ο πίνακας 3.2 προβάλλει την δομή του πίνακα “users”. Στον πίνακα περιέχονται τα στοιχεία που εκπροσωπούν τον χρήστη της ιστοσελίδας ή τον δημιουργό μιας εκδήλωσης.

Όνομα	Τύπος	Κενό	Πρόσθετα	Προεπιλογή
<u>id</u>	INT	Όχι	Auto Increment	Καμία
name	VARCHAR(40)	Όχι	-	Καμία
email	VARCHAR(50)	Όχι	Unique	Καμία
password	VARCHAR(60)	Όχι	-	Καμία
resetToken	VARCHAR(65)	Ναι	-	NULL
resetTokenExpiration	DATE	Ναι	-	NULL
createdAt	DATE	Όχι	-	NOW()

Πίνακας 3.2: Δομή του πίνακα “users”

Ανάλυση των στηλών του πίνακα “users”:

- **id:** Δηλώνει το αναγνωριστικό των χρηστών και είναι το πρωτεύον κλειδί (Primary Key) του πίνακα. Ο τύπος που έχει το συγκεκριμένο γνώρισμα έχει καθοριστεί ως ακέραιος αριθμός (INT). Δεν μπορεί να είναι μηδενική τιμή (Not Null).
- **name:** Δηλώνει το όνομα και το επίθετο του χρήστη της ιστοσελίδας. Ο τύπος γνωρίσματος είναι συμβολοσειρά (VARCHAR) έως 40 χαρακτήρες και το συμπληρώνει ο επισκέπτης στην φόρμα εγγραφής ώστε να γίνει χρήστης. Δεν μπορεί να είναι κενή η τιμή (Not Null).
- **email:** Δηλώνει την προσωπική ηλεκτρονική διεύθυνση ταχυδρομείου του χρήστη. Ο τύπος γνωρίσματος είναι συμβολοσειρά (VARCHAR) έως 50 χαρακτήρες και την προσθέτει κατά την εγγραφή στο σύστημα. Το email είναι μοναδικό. Δεν μπορεί να έχει κενή τιμή (Not Null).
- **password:** Δηλώνει τον προσωπικό κωδικό που αντιστοιχεί στον χρήστη και αποτελεί ένα από τα ευαίσθητα στοιχεία του. Ο τύπος γνωρίσματος είναι

συμβολοσειρά (VARCHAR) έως 60 χαρακτήρες, λόγω της κρυπτογράφησης που δέχονται οι χαρακτήρες για την ασφαλέστερη αποθήκευση του. Δεν μπορεί να έχει κενή τιμή (Not Null).

- **resetToken:** Δηλώνει το προσωπικό token, δηλαδή, έναν κρυπτογραφημένο κωδικό με 32 τυχαία bytes, που δημιουργείται όταν ο χρήστης επιθυμεί να ανακτήσει τον κωδικό πρόσβασης του. Ο τύπος γνωρίσματος είναι VARCHAR έως 65 χαρακτήρες αφού οι τυχαίοι χαρακτήρες δεν ξεπερνούν τον αριθμό αυτό. Μπορεί να έχει κενή τιμή, με προεπιλογή NULL.
- **resetTokenExpiration:** Δηλώνει την ημερομηνία κατά την οποία δημιουργήθηκε το token. Το πεδίο αυτό είναι απαραίτητο για λόγους ασφάλειας του συστήματος, ώστε το token αξιοποιηθεί σε συγκεκριμένο χρονικό διάστημα. Ο τύπος γνωρίσματος είναι ημερομηνία DATE. Μπορεί να έχει κενή τιμή, με προεπιλογή NULL.
- **createdAt:** Δηλώνει την ώρα και την ημερομηνία κατά τους οποίους εγγράφηκε ο χρήστης στο σύστημα. Ο τύπος γνωρίσματος είναι DATE ώστε να αποθηκεύεται η ημερομηνία της δημιουργίας. Η αποθήκευση τους γίνεται αυτόματα τη στιγμή εκείνη αφού έχει οριστεί η συνάρτηση “NOW()” ως προεπιλογή. Δεν μπορεί να έχει κενή τιμή (Not Null).

3.3.4 Πίνακας “categories”

Στον πίνακα 3.3 φαίνεται η δομή του πίνακα “categories”. Σε αυτόν τον πίνακα αποθηκεύεται η κατηγορία που αντιστοιχεί ένα event.

Όνομα	Τύπος	Κενό	Πρόσθετα	Προεπιλογή
<u>id</u>	INT	Όχι	Auto Increment	Καμία
category	VARCHAR(20)	Όχι	Unique	Καμία

Πίνακας 3.3: Δομή του πίνακα “categories”

- **id:** Δηλώνει το αναγνωριστικό των κατηγοριών και είναι το πρωτεύον κλειδί (Primary Key) του πίνακα. Ο τύπος που έχει αυτό το γνώρισμα έχει καθοριστεί ως ακέραιος αριθμός (INT) με σκοπό να καλύψει μεγάλες τιμές στο μέλλον. Δεν μπορεί να είναι κενή τιμή (Not Null).

- **category:** Δηλώνει την κατηγορία που αντιπροσωπεύει μια εκδήλωση. Ο τύπος γνωρίσματος έχει οριστεί ως συμβολοσειρά (VARCHAR) έως 20 χαρακτήρες. Η καταχώρηση της είναι μοναδική. Δεν μπορεί να είναι κενή τιμή (Not Null).

3.3.5 Πίνακας “images”

Ο πίνακας “images” εκπροσωπεί τις εικόνες που έχει η κάθε εκδήλωση. Στον πίνακα 3.4 παρουσιάζεται η δομή του.

Όνομα	Τύπος	Κενό	Πρόσθετα	Προεπιλογή
<u>id</u>	INT	Όχι	Auto Increment	Καμία
image	VARCHAR(60)	Όχι	-	Καμία
createdAt	DATETIME	Όχι	-	NOW()

Πίνακας 3.4: Δομή του πίνακα “images”

Ανάλυση της δομής του πίνακα “images”:

- **id:** Δηλώνει το αναγνωριστικό των εικόνων και είναι το πρωτεύον κλειδί (Primary Key) του πίνακα. Ο τύπος που έχει αυτό το γνώρισμα έχει καθοριστεί ως ακέραιος αριθμός (INT) με σκοπό να καλύψει μεγάλες τιμές στο μέλλον. Δεν μπορεί να είναι κενή τιμή (Not Null).
- **image:** Δηλώνει το hashed όνομα (url) του αρχείου εικόνας που κάνει upload ο χρήστης / δημιουργός. Ο τύπος γνωρίσματος είναι συμβολοσειρά (VARCHAR) έως 60 χαρακτήρες. Δεν μπορεί να έχει κενή τιμή (Not Null).
- **createdAt:** Δηλώνει την ώρα και την ημερομηνία κατά τους οποίους προστέθηκε μια εικόνα στο σύστημα. Ο τύπος γνωρίσματος είναι DATETIME ώστε να αποθηκεύεται η ημερομηνία και ώρα της προσθήκης. Η αποθήκευση τους γίνεται αυτόματα τη στιγμή εκείνη αφού έχει οριστεί η συνάρτηση “NOW()” ως προεπιλογή. Δεν μπορεί να έχει κενή τιμή (Not Null).

3.3.6 Πίνακας “reports”

Στον πίνακα “reports” αποθηκεύονται οι αναφορές που γίνονται στις εκδηλώσεις. Ο πίνακας 3.5 που ακολουθεί προβάλλει τη δομή του.

Όνομα	Τύπος	Κενό	Πρόσθετα	Προεπιλογή
createdAt	DATETIME	Όχι	-	NOW()
<u>userId</u>	INT	Όχι	-	Καμία
<u>eventId</u>	INT	Όχι	-	Καμία

Πίνακας 3.5: Δομή του πίνακα “reports”

Ανάλυση της δομής του πίνακα “reports”:

- **createdAt:** Δηλώνει τον χρόνο και την ημερομηνία που κάποιος χρήστης εισήλθε στο σύστημα. Ο τύπος γνωρίσματος είναι DATETIME. Η εισαγωγή γίνεται αυτόματα στο σύστημα λόγω του ότι έχει προεπιλεχθεί ή συνάρτηση “NOW()”. Δε μπορεί να έχει κενή τιμή (Not Null).
- **eventId:** Δηλώνει το αναγνωριστικό των εκδηλώσεων. Είναι το πρωτεύον κλειδί (Primary Key) του πίνακα “events” και ταυτόχρονα πρωτεύον κλειδί (Primary Key) και του πίνακα “reports”. Επίσης, είναι ξένο κλειδί (Foreign Key) στον πίνακα “reports”. Δεν μπορεί να έχει κενή τιμή (Not Null).
- **userId:** Δηλώνει το αναγνωριστικό που αντιστοιχεί σε ένα χρήστη. Αποτελεί το πρωτεύον κλειδί (Primary Key) του πίνακα users και ταυτόχρονα και πρωτεύον κλειδί (Primary Key) του πίνακα “reports”. Επίσης, είναι ξένο κλειδί (Foreign Key) στον πίνακα “reports”. Ο τύπος γνωρίσματος είναι ακέραιος αριθμός (INT) και δεν μπορεί να έχει κενή τιμή (Not Null).

Ο λόγος για τον οποίο το “eventId” και το “userId” είναι και πρωτεύον και ξένο κλειδί στον ίδιο πίνακα, είναι διότι ο χρήστης μπορεί να αναφέρει μόνο μια φορά μια εκδήλωση.

3.3.7 Πίνακας “comments”

Ο πίνακας “comments” αποθηκεύει τα σχόλια των χρηστών στις εκδηλώσεις. Στον πίνακα 3.6 παρατηρείται η δομή του.

Όνομα	Τύπος	Κενό	Πρόσθετα	Προεπιλογή
id	INT	Όχι	Auto Increment	Καμία
content	VARCHAR(255)	Όχι	-	Καμία
createdAt	DATETIME	Όχι	-	NOW()
<u>userId</u>	INT	Όχι	-	Καμία
<u>eventId</u>	INT	Όχι	-	Καμία

Πίνακας 3.6: Δομή του πίνακα “comments”

Ανάλυση της δομής του πίνακα “comments”:

- **id:** Δηλώνει το αναγνωριστικό των σχολίων και είναι το πρωτεύον κλειδί (Primary Key) του πίνακα. Ο τύπος που έχει αυτό το γνώρισμα έχει καθοριστεί ως ακέραιος αριθμός (INT). Η τιμή αυξάνεται αυτόματα. Δεν μπορεί να είναι κενή τιμή (Not Null).
- **content:** Δηλώνει το κείμενο του σχολίου που έχει προσθέσει ο χρήστης σε κάποια εκδήλωση. Ο τύπος γνωρίσματος είναι συμβολοσειρά VARCHAR έως 255 χαρακτήρες δίνοντας αρκετό περιθώριο στον χρήστη να τοποθετήσει τη γνώμη του για ένα event. Η τιμή του πεδίου δεν μπορεί να έχει κενή τιμή (Not Null).
- **createdAt:** Δηλώνει την ακριβή ημερομηνία καθώς και την ώρα που ο χρήστης / δημιουργός δημοσίευσε το προσωπικό του σχόλιο στην αντίστοιχη εκδήλωση. Ο τύπος γνωρίσματος του είναι ημερομηνία και ώρα (TIMESTAMP) με την προεπιλογή της συνάρτησης “NOW()” για την αυτόματη αποθήκευση των εκάστοτε πληροφοριών ημέρας και ώρας. Δεν θα είναι κενή η τιμή (Not Null).
- **eventId:** Δηλώνει το αναγνωριστικό των εκδηλώσεων και την συγκεκριμένη εκδήλωση που αντιστοιχεί το σχόλιο. Είναι το πρωτεύον κλειδί (Primary Key) του πίνακα “events” και ξένο κλειδί (Foreign Key) στον πίνακα “comments”. Δεν μπορεί να έχει κενή τιμή (Not Null).
- **userId:** Δηλώνει το αναγνωριστικό που αντιστοιχεί σε ένα χρήστη και τον συγκεκριμένο χρήστη που δημοσίευσε το σχόλιο. Αποτελεί το πρωτεύον κλειδί (Primary Key) του πίνακα users και ξένο κλειδί (Foreign Key) στον πίνακα

“comments”. Ο τύπος γνωρίσματος είναι ακέραιος αριθμός (INT) και δεν μπορεί να έχει κενή τιμή (Not Null).

3.3.8 Πίνακας “cities”

Ο πίνακας “cities” αποθηκεύει τις πόλεις όπου πραγματοποιούνται οι εκδηλώσεις. Παρακάτω στον πίνακα 3.7 παρουσιάζεται η δομή του.

Όνομα	Τύπος	Κενό	Πρόσθετα	Προεπιλογή
<u>id</u>	INT	Όχι	Auto Increment	Καμία
city	VARCHAR(20)	Όχι	Unique	Καμία

Πίνακας 3.7: Δομή του πίνακα “cities”

Ανάλυση της δομής του πίνακα “cities”:

- **id**: Δηλώνει το αναγνωριστικό των πόλεων και είναι το πρωτεύον κλειδί (Primary Key) του πίνακα. Ο τύπος που έχει αυτό το γνώρισμα έχει καθοριστεί ως ακέραιος αριθμός (INT). Η τιμή αυξάνεται αυτόματα. Δεν μπορεί να είναι κενή τιμή (Not Null).
- **city**: Δηλώνει την πόλη που γίνεται η εκδήλωση. Ο τύπος γνωρίσματος είναι συμβολοσειρά (VARCHAR) έως 20 χαρακτήρες και τον διαλέγει ο χρήστης / δημιουργός του event. Το όνομα της είναι μοναδικό. Δεν μπορεί να είναι κενή τιμή (Not Null).

3.3.9 Πίνακας “prefectures”

Στον πίνακα “prefectures” γίνεται αποθήκευση των νομών που γίνεται ένα event. Ακολουθεί η δομή του πίνακα αυτού στον πίνακα 2.8.

Όνομα	Τύπος	Κενό	Πρόσθετα	Προεπιλογή
id	INT	Όχι	Auto Increment	Καμία
prefecture	VARCHAR(20)	Όχι	Unique	Καμία

Πίνακας 3.8: Δομή του πίνακα “prefectures”

-
- **id:** Δηλώνει το αναγνωριστικό των νομών και είναι το πρωτεύον κλειδί (Primary Key) του πίνακα. Ο τύπος που έχει αυτό το γνώρισμα έχει καθοριστεί ως ακέραιος αριθμός (INT). Η τιμή αυξάνεται αυτόματα. Δεν μπορεί να είναι κενή τιμή (Not Null).
 - **prefecture:** Δηλώνει τον νομό στον οποίο πραγματοποιείται η εκδήλωση. Ο τύπος γνώρισματος είναι συμβολοσειρά (VARCHAR) έως 20 χαρακτήρες και τον επιλέγει ο χρήστης / δημιουργός της εκδήλωσης. Το όνομα του είναι μοναδικό. Δεν μπορεί να είναι κενή τιμή (Not Null).

3.4 Ασφάλεια συστήματος

Η διαδικτυακή ασφάλεια (Cyber-security) [22], για την οποία αναφέρονται όλοι ως έναν από τους ζωτικής σημασίας παράγοντες για ένα σύστημα, έχει επιστήσει ολοένα και μεγαλύτερη την προσοχή την τελευταία δεκαετία, καθώς οι περισσότεροι κλάδοι συνειδητοποίησαν πόσο σημαντικό είναι να υπάρχει ισχυρή προστασία των υπολογιστικών τους συστημάτων. Στο όχι και τόσο μακρινό παρελθόν, σε αρκετές επιχειρήσεις, ακόμα και σε γνωστές πανίσχυρες εταιρίες, έγιναν παραβιάσεις ασφαλείας όπου τα προσωπικά δεδομένα των χρηστών διέρρευσαν και διακυβεύτηκαν εκατομμύρια λογαριασμοί χρηστών. Αυτό συνέβη λόγω της αδύναμης διαχείρισης των κωδικών πρόσβασης από την πλευρά των επιχειρήσεων. Από τις καθιερωμένες πολυεθνικές μέχρι τις νεοσύστατες επιχειρήσεις, όλοι έχουν την υποχρέωση να αποθηκεύουν τα διαπιστευτήρια των χρηστών με ασφάλεια. Στην κορυφή της λίστας των ζητημάτων ασφαλείας βρίσκονται οι κωδικοί πρόσβασης για τους οποίους αρκετοί πιστεύουν πως είναι ο ασθενέστερος κρίκος στην ασφάλεια του διαδικτύου. Βέβαια, ακολουθεί η ασφάλεια για τις «ενέσεις» (injections), και πιο συγκεκριμένα “SQL Injections” οι οποίες θεωρούνται οι πιο συνηθισμένες τεχνικές επίθεσης τα τελευταία χρόνια και οδηγούν στις μεγαλύτερες απώλειες. Αν λοιπόν, ο προγραμματιστής καταφέρει να εμποδίσει αυτές τις «ενέσεις», τότε έχει καλύψει ένα μεγάλο κομμάτι βελτίωσης στην ασφάλεια του συστήματος. Εξίσου σημαντικό ρόλο για την ασφάλεια του συστήματος, διαδραματίζουν τα frameworks. Όσο περισσότερες απαιτήσεις έχει μια εφαρμογή, τόσο περισσότερες βιβλιοθήκες τρίτων (third party libraries) αναζητά ο κάθε προγραμματιστής για την διευκόλυνση του

development, και αυτό δεν είναι κακό. Η ιδέα του “NPM” (Node Package Manager) είναι να μοιράζεται ο κώδικας σε αυτούς που θέλουν να τον χρησιμοποιήσουν. Εδώ όμως προκαλείται το πρόβλημα της αμφιβολίας από την μεριά του προγραμματιστή, ο οποίος δεν είναι βέβαιος αν ο αντίστοιχος κώδικας είναι αξιόπιστος, αφού δεν αποτελεί προσωπική του δημιουργία και γνωρίζει πως οποιοσδήποτε μπορεί να κάνει upload όποιο πακέτο θέλει. Συνεπώς, πρέπει πάντα να γίνεται έλεγχος των πακέτων ώστε να θεωρείται έμπιστη η χρήση τους. Γενικότερα όμως, ευτυχώς για τους προγραμματιστές υπάρχουν πολλές και μεγάλες βιβλιοθήκες πακέτων που μπορούν να εμπιστευτούν, όπως είναι η React, διότι αναπτύσσονται από μεγάλες εταιρίες και χρησιμοποιούνται από την πλειοψηφία τους. Ένα ακόμα σοβαρό ζήτημα, είναι η ασφαλής διαδικτυακή σύνδεση που είναι ανάλογης σημαντικότητας και πρέπει να εφαρμόζεται σε κάθε σύστημα που συλλέγει ευαίσθητες πληροφορίες. Με λίγα λόγια, επιτακτική ανάγκη κάθε ιστοσελίδας είναι να υπάρχει HTTPS (Hypertext Transfer Protocol Secure) παντού. Αυτό σημαίνει όταν στέλνεται μια πληροφορία από τον client στο σύστημα, η οποία είναι απλά κείμενο, κάποιος μπορεί να την παρακολουθεί και να τη δει. Οπότε ασφαρίζοντας τη σύνδεση γίνεται ανέφικτο για κάποιον να δει ποιο είναι το output των δεδομένων που στέλνονται μέσω των καλωδίων (wires). Τελικά, αξίζει να αναφερθεί πως υπάρχουν κι άλλες πρακτικές επίθεσης και πως θα πάντα θα υπάρχουν, και ένα σύστημα δεν μπορεί ποτέ στο 100% να είναι ασφαλές, όπως ομοίως πάντα θα εφαρμόζονται οι βέλτιστες πρακτικές για την ασφάλεια του. Παρακάτω αναλύονται οι τεχνικές ασφάλειας που αναφέρθηκαν και που ακολούθησε το συγκεκριμένο σύστημα.

- **Κρυπτογράφηση των κωδικών πρόσβασης (passwords) που επιλέγουν οι χρήστες κατά την εγγραφή τους στο σύστημα.:** Ένας επισκέπτης της ιστοσελίδας, προκειμένου να γίνει χρήστης της, πρέπει πρώτα να συμπληρώσει στην φόρμα εγγραφής (registration form) τα στοιχεία του. Εκεί, του ζητείται να πληκτρολογήσει τον κωδικό, που θα συμπληρώνει κάθε φορά που επιθυμεί να εισέρχεται στο σύστημα. Για να μπορέσει ο κωδικός του να είναι ασφαλής κατά την μεταφορά του στο Διαδίκτυο, έχει εφαρμοστεί η συνάρτηση κρυπτογράφησης “bcrypt” ένα από τα δημοφιλέστερα πακέτα (NPM package). Εκτός από την ενσωμάτωση ενός salt για την προστασία από επιθέσεις rainbow table, το “bcrypt” είναι μια προσαρμοστική συνάρτηση, έτσι ώστε με την πάροδο του

χρόνου, ο αριθμός επανεξέτασης να αυξάνεται και να γίνεται πιο αργός παραμένοντας ανθεκτικός σε επιθέσεις αναζήτησης με αυξανόμενη υπολογιστική ισχύ. Ο λόγος που επιλέχθηκε το συγκεκριμένο πακέτο είναι λόγω της μεγάλης αξιοπιστίας που φέρει με τα χρόνια καθώς το επιλέγουν κατά κόρον πολλοί προγραμματιστές, και χαίρει μεγάλης εκτίμησης από τα web communities. Επίσης, είναι μια εξαιρετική επιλογή για έναν node.js server αφού προσφέρει εύκολο documentation.

- **Προστασία από την τεχνική επίθεσης “ενέσεων” (SQL injections):** Ο όρος injections σημαίνει να γράφεις κώδικα (injecting code) μέσα σε ένα άλλο κομμάτι κώδικα. Αυτή είναι μια αρχή ιδέας, το γεγονός δηλαδή ότι μπορεί να τοποθετείται ένα κομμάτι κώδικα από κάποιον που θέλει να αλλάξει ή ίσως να κάνει κακό στο σύστημα. Η πιο συνηθισμένη επίθεση, η οποία ονομάζεται “Sequel Injection”, στοχεύει στην αλλαγή των εντολών της βάσης δεδομένων με σκοπό να επεξεργαστεί τα στοιχεία της ή ακόμα και να τη διαγράψει. Προκειμένου να αποτραπεί αυτό, θα πρέπει ο προγραμματιστής να είναι σίγουρος πως οτιδήποτε ο χρήστης συμπληρώνει στα inputs της ιστοσελίδας, πρόκειται για κάτι το οποίο το πρόγραμμα περιμένει. Κάνοντας χρήση λοιπόν, αυτής της πρακτικής, και τοποθετώντας ORMS (Object Relational Mappers) στο σύστημα και συγκεκριμένα τη βιβλιοθήκη “Sequelize” (Sequelize ORM), αποφεύγεται το ζήτημα των επιθέσεων μέσω της SQL στη βάση δεδομένων.
- **Χρήση του NPM (Node Package Manager).** Ο διαχειριστής πακέτων NPM ο οποίος χρησιμοποιείται στο εκάστοτε σύστημα, σε μια από της τελευταίες του αναβάθμισης, και συγκεκριμένα μετά την έκδοση 6 (version 6 or higher), έχει προσθέσει ασφάλεια ως ένα built-in feature το οποίο εγκαθίσταται αυτόματα κατά την διαδικασία του download των πακέτων. Έτσι, λήφθηκαν αυτόματες ειδοποιήσεις σχετικά με τα τρωτά σημεία ασφάλειας στα πακέτα που έχουν εγκατασταθεί στο σύστημα. Άρα, έχουν ελεγχθεί όλα τα έτοιμα πακέτα που χρησιμοποιήθηκαν από βιβλιοθήκες τρίτων.
- **Πιστοποίηση Secure Sockets Layer / Transport Layer Security (SSL/TLS Certificate).** Κάθε ιστοσελίδα που περιλαμβάνει φόρμες συμπλήρωσης προσωπικών στοιχείων (form data) θα πρέπει να χρησιμοποιεί HTTPS (Hypertext

Transfer Protocol Secure). Όταν γίνεται απλά χρήση HTTP οι πληροφορίες που στέλνονται από τον client στον server είναι απλό κείμενο και αυτό δεν πρέπει να συμβαίνει, διότι ενδέχεται να παρακολουθείται από κάποιον attacker που μπορεί να υποκλέψει τα δεδομένα του χρήστη. Έτσι, το συγκεκριμένο σύστημα έχει πάρει τις πιστοποιήσεις των SSL/TLS πρωτοκόλλων, τα οποία δημιουργούν ένα τούνελ στη μεταφορά των στοιχείων που ασφαλίζει και κρυπτογραφεί τις πληροφορίες μεταξύ του client και του server, γνωστοποιώντας ότι ολόκληρο το σύστημα είναι υπό την επίβλεψη μόνο του προγραμματιστή.

- **Κατάλληλο validation σε όλες τις περιπτώσεις εισαγωγής και αποστολής δεδομένων.** Είναι απαραίτητη προϋπόθεση η ύπαρξη validation, δηλαδή έλεγχου των δεδομένων που εισάγονται και αποστέλλονται στον server από τον client. Για τον λόγο αυτό, έχει τοποθετηθεί κατάλληλος έλεγχος τόσο στο front-end όσο και στο back-end του συγκεκριμένου συστήματος.
- **Δημιουργία token και τοποθέτηση εξουσιοδότησης χρήστη (authorization).** Για την μεγιστοποίηση της ασφάλειας των συστημάτων, απαραίτητος είναι ο έλεγχος της αξιοπιστίας του χρήστη. Κάθε σύστημα θα πρέπει να γνωρίζει κατά πόσο ο χρήστης που εισήλθε στο σύστημα έχει την εξουσιοδότηση να εκτελέσει ενέργειες που παίρνουν, προσθέτουν ή διαγράφουν δεδομένα από αυτό. Για τον λόγο αυτό, έχει τοποθετηθεί στο συγκεκριμένο σύστημα, στο back-end, ένα middleware πριν από τα routes τα οποία πρέπει να προστατευτούν και στο front-end η κατάλληλη υλοποίηση ώστε να είναι εφικτή η αναγνώριση του χρήστη. Έτσι, το σύστημα καλύπτει περιπτώσεις όπου κάποιο επιτηδευμένο πρόγραμμα είναι ικανό να στέλνει requests στο σύστημα παρακάπτοντας για παράδειγμα την ασφάλεια ενός περιηγητή.

3.5 Σύνοψη κεφαλαίου

Στο κεφάλαιο αυτό, αποσαφηνίστηκε και περιγράφηκε ο τρόπος ανάπτυξης του συστήματος. Αρχικά, διατυπώθηκαν οι απαιτήσεις του συστήματος και οι περιπτώσεις χρήσης (use cases) των χρηστών και στη συνέχεια αναλύθηκε η βάση δεδομένων τόσο σχηματικά όσο και περιγραφικά. Τέλος, παρουσιάστηκε μια γενική εικόνα για την ασφάλεια των συστημάτων, καταλήγοντας στις τεχνικές ασφάλειας που ακο-

λούθησε το συγκεκριμένο σύστημα.

Κεφάλαιο 4

Λειτουργικότητα συστήματος

Το κεφάλαιο που ακολουθεί, παρουσιάζει τη λειτουργικότητα του συστήματος και αποτελεί το κεφάλαιο με το οποίο ολοκληρώνεται η συγκεκριμένη διπλωματική εργασία. Ειδικότερα, γίνεται ανάλυση των λειτουργιών καθώς και των διεπαφών χρηστών (UI) και διαδικτυακής εφαρμογής όπως και οι περιπτώσεις χρήσης (use cases) τόσο για τους επισκέπτες όσο και για τους χρήστες. Τέλος, συνοφίζονται όλα όσα αναφέρθηκαν στο κεφάλαιο αυτό.

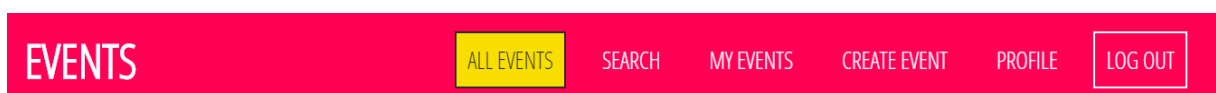
4.1 Λειτουργίες διαδικτυακής εφαρμογής και διαπαφές χρήστη

Μια επιτυχημένη διαδικτυακή εφαρμογή καθορίζεται από τον σωστό σχεδιασμό της διεπαφής χρήστη και αποτελεί εξέχουσα θέση στην ιεραρχία ενός συστήματος. Η διεπαφή χρήστη (UI) επικεντρώνεται στην αισθητική και παράλληλα στην μεγιστοποίηση της ανταπόκρισης, της αποδοτικότητας και της προσβασιμότητας ενός διαδικτυακού τόπου. Η δυσχρηστία σε μια εφαρμογή μπορεί να οδηγήσει στην απομάκρυνση των επισκεπτών από αυτή, πράγμα που ο προγραμματιστής πρέπει πάντα να λαμβάνει υπόψιν του. Ειδικότερα, για την εκκίνηση μιας μικρής ιστοσελίδας, η σημασία του UI καθίσταται ακόμη πιο κρίσιμη, καθώς η πρώτη εντύπωση θεωρείται αρκετά σημαντική. Εξίσου σημαντική, είναι και η παροχή όλων των απαραίτητων λειτουργιών, δίνοντας τη δυνατότητα στον χρήστη να περιηγηθεί κάνοντας πολύπλευρη χρήση του ιστοχώρου, μέσω ενός ικανοποιητικού εύρους λειτουργιών. Επιπλέον, πριν γίνει η ανάλυση των λειτουργιών και των διεπαφών χρήστη, επιτακτική ανάγκη είναι να αναφερθούν κάποια από τα στοιχεία του συστήματος ώστε να αποφευχθεί η αναπαραγωγή τους σε κάθε πτυχή της ανάλυσης ξεχωριστά. Ένα από τα στοιχεία αυτά, είναι το κομμάτι της επικύρωσης των δε-

δομένων (data validation). Τα δεδομένα που εισάγονται στις φόρμες συμπλήρωσης στοιχείων επικυρώνονται τόσο στο front-end (React) όσο και στο back-end (Node server), σε όλη την διαδικτυακή εφαρμογή. Αυτό πραγματοποιήθηκε για να αποτραπεί η παρέλευση δεδομένων που δεν έχουν ελέγξει, διότι υπάρχουν λογισμικά που δημιουργούν http requests, όπως είναι για παράδειγμα το postman, τα οποία παρακάμπτουν το front-end κομμάτι και μπορούν να το διαπεράσουν. Για το λόγο αυτό, όπως προαναφέρθηκε, έχει χρησιμοποιηθεί validation και στον server μέσω της γνωστής βιβλιοθήκης “express-validator”, έτσι ώστε σε τέτοιες περιπτώσεις να επιστρέφει κατάλληλο μήνυμα για το είδος εισαγωγής των δεδομένων. Επίσης, κάνοντας χρήση των React Hooks, παραμένει ο φόρτος των εργασιών του server σε αρκετά χαμηλό επίπεδο. Τέλος, πριν αρχίσει η ανάλυση των λειτουργιών, παρουσιάζονται στο σχήμα 4.1 και σχήμα 4.2 η μπάρα πλοήγησης (navigation bar), τόσο για τον επισκέπτη όσο και για τον χρήστη, καθώς και στο σχήμα 4.3 το υποσέλιδο της διαδικτυακής εφαρμογής (footer).



Σχήμα 4.1: Μπάρα πλοήγησης μη εγγεγραμμένου χρήστη.



Σχήμα 4.2: Μπάρα πλοήγησης εγγεγραμμένου χρήστη.

Copyright © 2020 Events. Developed by Lampros Varnavas | Supervised by Minas Dasygenis

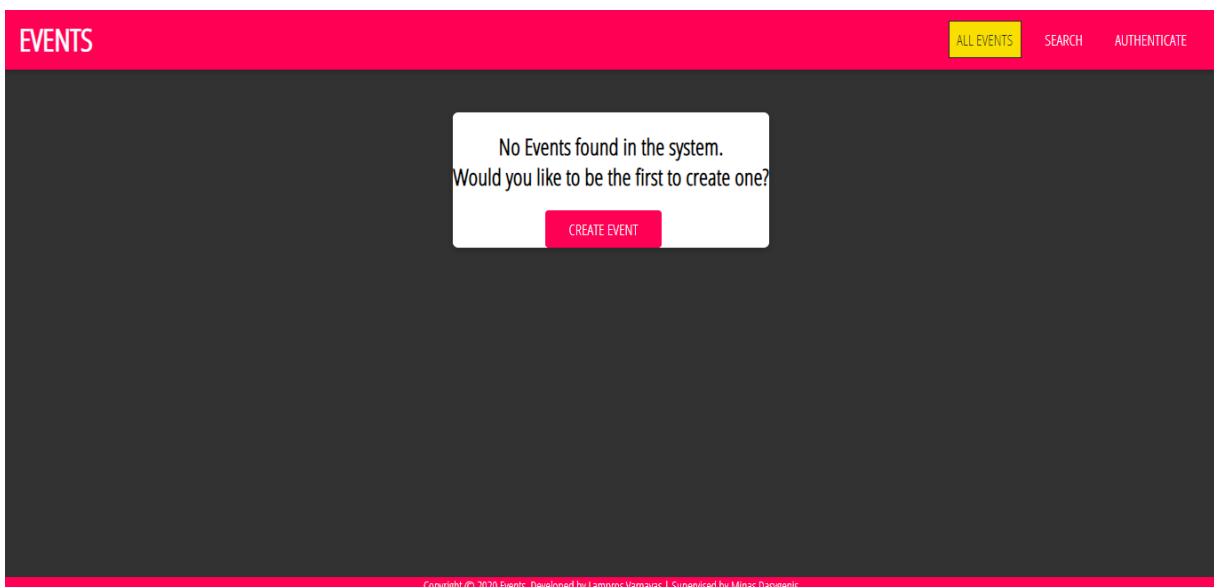
Σχήμα 4.3: Υποσέλιδο διαδικτυακής εφαρμογής (footer).

4.1.1 Αρχική σελίδα διαδικτυακής εφαρμογής

Σε αυτή την υποενότητα γίνεται ανάλυση της πρώτης επαφής ενός επισκέπτη με τον ιστοχώρο. Πιο συγκεκριμένα, ακολουθεί παρουσίαση της αρχικής σελίδας της διαδικτυακής εφαρμογής, τόσο για κάποιον ο οποίος επισκέπτεται την ιστοσελίδα χωρίς να διαθέτει λογαριασμό, όσο και για τα μέλη της. Επιπλέον, αναφέρονται οι περιπτώσεις όπου ο επισκέπτης είναι ο πρώτος που επισκέφθηκε τον ιστοχώρο μετά

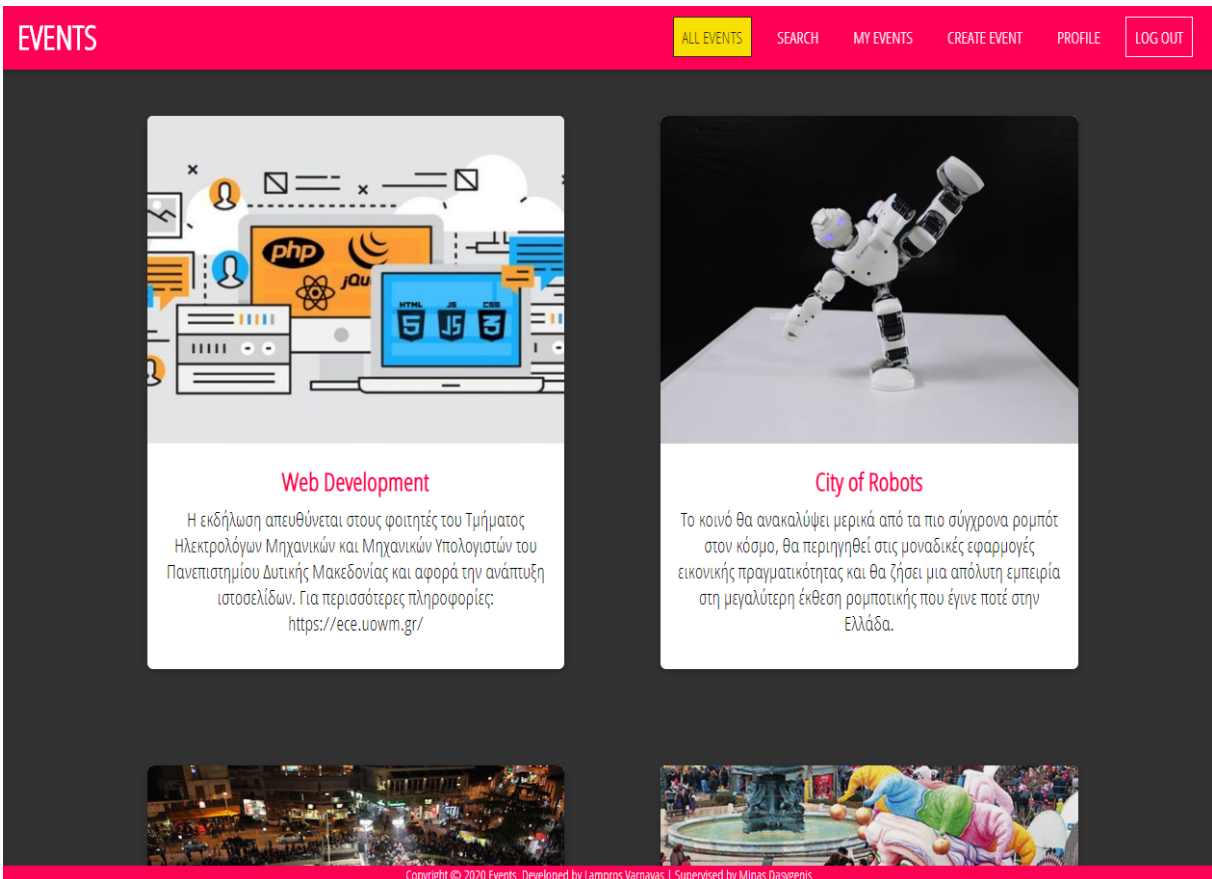
την μεταφόρτωση του online (deployment), αφού δεν θα υπάρχει κάποια εκδήλωση στο σύστημα, καθώς και όπου οι χρήστες έχουν δημιουργήσει κάποιες εκδηλώσεις και άρα υπάρχουν ήδη καταχωρημένες εκδηλώσεις.

Στην περίπτωση, λοιπόν, που μετά με το deployment της εφαρμογής κάποιος επισκεφθεί την ηλεκτρονική διεύθυνση της, θα εμφανίζεται στην αρχική σελίδα του ιστοχώρου κατάλληλο μήνυμα όπου θα τον ενημερώνει πως δεν υπάρχουν καταχωρημένες εκδηλώσεις, προωθώντας τον επισκέπτη να γίνει ο πρώτος δημιουργός μιας εκδήλωσης μέσω του αντίστοιχου κουμπιού “Create Event” που εμφανίζεται κάτω από το μήνυμα και θα τον ανακατευθύνει στην φόρμα εγγραφής αν κάνει κλικ σε αυτό. Άρα λοιπόν, θα εμφανίζεται όπως φαίνεται στο σχήμα 4.4 παρακάτω.



Σχήμα 4.4: Αρχική σελίδα πρώτου επισκέπτη.

Το σχήμα 4.5 παρουσιάζει την περίπτωση που υπάρχουν ήδη καταχωρημένες εκδηλώσεις στο σύστημα ο επισκέπτης θα βλέπει τις εικόνες, τον τίτλο και την περιγραφή των εκδηλώσεων, δηλαδή κάτι αντίστοιχο όπως στην ακόλουθη εικόνα. Οι φωτογραφίες που απεικονίζονται σε κάθε μια από τις εκδηλώσεις, εναλλάσσονται δυναμικά κάθε 5 δευτερόλεπτα. Συγκεκριμένα, έχει τοποθετηθεί αυτόματο image slider με τη χρήση των βιβλιοθηκών “react-slick” και “slick-carousel” ώστε να είναι ορατές οι φωτογραφίες που επέλεξε ο δημιουργός για την κάθε εκδήλωση ξεχωριστά, κατά την περιήγηση του στην αρχική σελίδα του ιστοχώρου.



Σχήμα 4.5: Παραδείγματα περιεχομένου αρχικής σελίδας.

4.1.2 Σελίδα εισόδου, εγγραφής και ανάκτησης κωδικού πρόσβασης

Η σελίδα "Authenticate" αποτελεί τη σημαντικότερη σελίδα του συστήματος, αφού μέσω αυτής γίνεται δυνατή η είσοδος στο σύστημα και θα δωθεί ιδιαίτερη σημασία στις διεπαφές του χρήστη, αναλύοντας όλες τις περιπτώσεις. Στη σελίδα αυτή, ο επισκέπτης μπορεί να κάνει εγγραφή (sign up), συμπληρώνοντας το όνομα του, το ηλεκτρονικό ταχυδρομείο (email) του και τον κωδικό (password) που επιθυμεί, πληρώνοντας όμως συγκεκριμένα κριτήρια του validation, να εισέλθει (log in) στο σύστημα, συμπληρώνοντας το email και το password που χρησιμοποίησε κατά την εγγραφή και να ανακτήσει τον κωδικό πρόσβασης σε περίπτωση που δεν τον θυμάται, πληκτρολογώντας το email του. Η σελίδα αυτή, περιλαμβάνει τις επιλογές του "Log in", "Sign up" και "Reset" κάνοντας εναλλαγή φόρμας (switch form) με κλικ στο κουμπί "Log in", "Sign up" ή "Reset" αντίστοιχα. Πιο συγκεκριμένα, ο επισκέπτης που θέλει να εγγραφεί θα πρέπει να συμπληρώσει σε κάθε ένα από τα πεδία έγκυρα στοιχεία, αλλιώς το πεδίο δεν γίνεται έγκυρο για αποστολή στον server, και εμφανίζεται κατάλληλο μήνυμα validation. Όπως έχει προαναφερθεί, έχει τοποθε-

τηθεί κατάλληλο validation σε όλο το front-end. Το βασικότερο χαρακτηριστικό που περιλαμβάνει το validation είναι ότι κανένα από τα πεδία δεν μπορεί να είναι κενό. Στην περίπτωση, λοιπόν, του πεδίου "Name" οι χαρακτήρες θα πρέπει να είναι ίσοι ή περισσότεροι από έναν. Το πεδίο του ηλεκτρονικού ταχυδρομείου (email) είναι απαραίτητο να είναι της μορφής "test@test.com", όπου επίσης εμφανίζεται κατάλληλο μήνυμα. Τέλος, στην περίπτωση του κωδικού (password) είναι υποχρεωτική η επιλογή περισσότερων από 6 χαρακτήρες, καθώς σε αντίθετη περίπτωση εμφανίζεται κατάλληλο μήνυμα εξίσου με τα προηγούμενα. Είναι πολύ σημαντικό να αναφερθεί πως ο κωδικός που εισάγει ο χρήστης ως προσωπικό κωδικό πρόσβασης στον ιστοχώρο, κρυπτογραφείται αυτόματα. Αν και μόνο αν, τα πεδία πληρούν το τοποθετημένο validation, τότε το κουμπί "Sign up" θα αλλάξει κατάσταση και από απενεργοποιημένο (disabled) θα γίνει ενεργοποιημένο (enabled). Κάνοντας κλικ στο κουμπί "Reset" ο χρήστης μεταβαίνει στη σελίδα ανάκτησης, όπου του ζητείται να εισάγει το email στο οποίο θα λάβει ένα μήνυμα email μέσω του οποίου δίνεται ένα συγκεκριμένο link στο οποίο μπορεί να εισάγει έναν καινούριο κωδικό πρόσβασης. Παρακάτω παρουσιάζονται οι περιπτώσεις που αναφέρθηκαν.

Το σχήμα 4.6 προβάλλει την περίπτωση όπου τα πεδία εγγραφής στη φόρμα είναι κενά.

Στο σχήμα 4.7 παρουσιάζεται η περίπτωση όπου τα πεδία εγγραφής δεν πληρούν το συγκεκριμένο validation που έχει τοποθετηθεί.

Στο σχήμα 4.8 φαίνεται η περίπτωση όπου για παράδειγμα τα δυο από τα τρία πεδία της φόρμας είναι valid, όμως για να είναι εφικτό να γίνει η εγγραφή και να ενεργοποιηθεί το κουμπί "Sign up" είναι απαραίτητο να συμπληρωθούν όλα τα πεδία.

Στο σχήμα 4.9 παρουσιάζεται η περίπτωση όπου και τα τρία πεδία είναι valid και το κουμπί "Sign up" ενεργοποιείται.

Στο σχήμα 4.10 και 4.11 εμφανίζονται οι περιπτώσεις όπου στη φόρμα εισόδου τα πεδία είναι είτε κενά είτε δεν ακολουθούν το validation.

Στο σχήμα 4.12 προβάλλεται η περίπτωση όπου είναι valid και τα δυο πεδία και ενεργοποιείται το κουμπί εισόδου "Log in".

Στο σχήμα 4.13 γίνεται γνωστό ότι στην περίπτωση που ο χρήστης πληκτρολογήσει στη φόρμα εισαγωγής email, ένα έγκυρο email τότε ειδοποιείται ότι θα λάβει

The screenshot shows a 'Login Required' form with three input fields: 'Your Name', 'E-Mail', and 'Password'. Each field has a red border and a red error message below it. The 'Your Name' field has the message 'Please enter a name.' The 'E-Mail' field has the message 'Please enter a valid email address.' The 'Password' field has the message 'Please enter a valid password, at least 6 characters.' Below the fields are three buttons: 'SIGNUP' (disabled), 'LOGIN', and 'RESET PASSWORD'. The form is centered on a dark background. At the bottom of the page, there is a copyright notice: 'Copyright © 2020 Events. Developed by Lampros Varnavas | Supervised by Minas Dasygenis'.

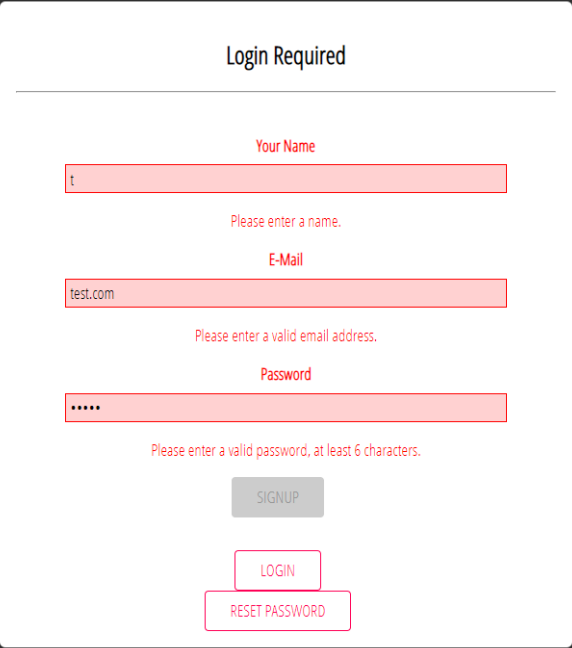
Σχήμα 4.6: Σελίδα εγγραφής με κενές τιμές στα πεδία της φόρμας.

ένα μήνυμα email από το οποίο μπορεί να ανακτήσει τον κωδικό πρόσβασης.

Τέλος, στο σχήμα 4.14 φαίνεται η σελίδα αλλαγής του κωδικού πρόσβασης, με τη φόρμα συμπλήρωσης του νέου κωδικού πρόσβασης. Ο χρήστης πρέπει να εισάγει δυο φορές τον ίδιο κωδικό πρόσβασης, για λόγους ασφάλειας, ώστε να εμφανιστεί το κουμπί "Update Password" και να είναι εφικτή η αλλαγή του. Όταν ο χρήστης κάνει κλικ στο κουμπί "Update Password" ειδοποιείται πάλι με κατάλληλο email ότι η αλλαγή του κωδικού πρόσβασης ήταν επιτυχής.

4.1.3 Σελίδα αναζήτησης

Στη συγκεκριμένη σελίδα τόσο ο επισκέπτης όσο και ο χρήστης του ιστοχώρου, έχει τη δυνατότητα να αναζητήσει εκδηλώσεις που τον ενδιαφέρουν, οι οποίες είναι ταξινομημένες με βάση την πόλη, τον νομό ή την ημερομηνία έναρξης που πραγματοποιείται αλλά και με την κατηγορία στην οποία κατατάσσεται. Είναι εξίσου σημαντικό να αναφερθεί, ότι τα πεδία σε κάθε μια από τις περιπτώσεις, είναι απαραίτητο να μην είναι κενά. Συνεπώς, έχει τοποθετηθεί αντίστοιχο validation και



The screenshot shows a 'Login Required' form with three input fields, each with a red border and a red error message below it. The first field is labeled 'Your Name' and contains the letter 't', with the error message 'Please enter a name.' The second field is labeled 'E-Mail' and contains 'test.com', with the error message 'Please enter a valid email address.' The third field is labeled 'Password' and contains six dots, with the error message 'Please enter a valid password, at least 6 characters.' Below the fields are three buttons: 'SIGNUP' (disabled), 'LOGIN', and 'RESET PASSWORD'. At the bottom of the page, there is a copyright notice: 'Copyright © 2020 Events. Developed by Lampros Varnavas | Supervised by Minas Dasgenis'.

Σχήμα 4.7: Σελίδα εγγραφής με μη έγκυρα στοιχεία στη φόρμα.

σε αυτή τη σελίδα, για κενές τιμές. Παρακάτω προβάλλονται οι περιπτώσεις που προαναφέρθηκαν.

Στο σχήμα 4.15 παρουσιάζεται η περίπτωση αρχικής κατάστασης όπου ο επισκέπτης ή χρήστης του ιστοτόπου επιλέγει να μεταβεί στη σελίδα αναζήτησης.

Στο σχήμα 4.16 εμφανίζεται η περίπτωση όπου έχει επιλεγεί κάποιο από τα κουμπιά “Search”. Παραδείγματος χάρη έχει επιλεγεί το κουμπί “Search” στο πλαίσιο της πόλης, το οποίο είτε είναι κενό είτε έχει επιλεγεί κάποια πόλη στην οποία δεν διεξάγεται κάποιο Event. Έτσι, εμφανίζεται στην οθόνη κατάλληλο μήνυμα, το οποίο στέλνεται από τον server αφού ψάχνοντας στη βάση δεν έχει βρεθεί εκδήλωση στη συγκεκριμένη πόλη, και ενημερώνει τον επισκέπτη ή χρήστη ότι δεν ήταν δυνατή η αυτή η επιλογή εύρεσης. Αξίζει να αναφερθεί ότι θα εμφανιστεί επίσης κατάλληλο μήνυμα και για τις υπόλοιπες περιπτώσεις αντίστοιχα.

Τέλος, στην περίπτωση όπου επιλεγεί, για παράδειγμα μια συγκεκριμένη ημερομηνία, τότε εμφανίζονται όλες οι εκδηλώσεις οι οποίες διεξάγονται τότε. Ένα τυχαίο παράδειγμα, φαίνεται στην εικόνα που ακολουθεί, όπου ένας χρήστης έχει επιλέξει

Copyright © 2020 Events. Developed by Lampros Varnavas | Supervised by Minas Dasggenis

Σχήμα 4.8: Σελίδα εγγραφής με δυο έγκυρα πεδία στη φόρμα.

μια ημερομηνία, κάνει κλικ στο κουμπί “Search” και ανακατευθύνεται στη σελίδα όπου εμφανίζεται κατάλληλο μήνυμα με την ημερομηνία που διάλεξε, και εμφανίζονται οι εκδηλώσεις που έχουν καταχωρηθεί στο σύστημα για αυτή την ημερομηνία. Αυτή η περίπτωση προβάλλεται στο σχήμα 4.17. Επίσης και στις υπόλοιπες περιπτώσεις αναζήτησης, εμφανίζονται επίσης οι αντίστοιχες σελίδες αυτές με παρόμοιο τρόπο.

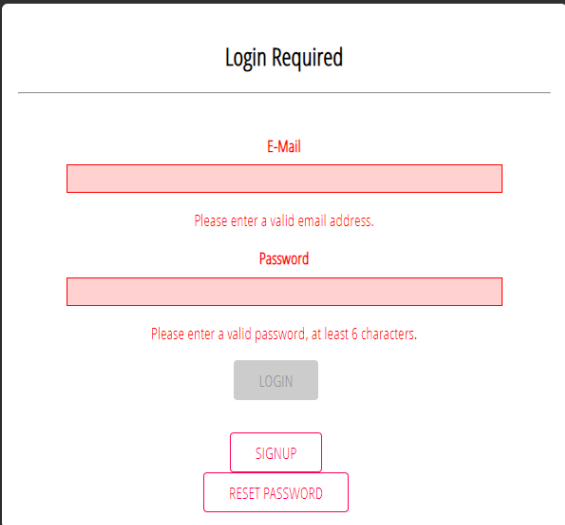
4.1.4 Σελίδα δημιουργίας εκδήλωσης εγγεγραμμένου χρήστη

Σε αυτή τη σελίδα, ο χρήστης έχει την δυνατότητα να δημιουργήσει μια εκδήλωση. Τα απαραίτητα πεδία που πρέπει να συμπληρωθούν είναι ο τίτλος, η διεύθυνση, η ημερομηνία έναρξης και λήξης, η ώρα έναρξης, η πόλη, ο νομός, η κατηγορία, η περιγραφή και οι φωτογραφίες που θα αποτελούν τα προσωπικά στοιχεία μιας εκδήλωσης. Στα πεδία αυτά, έχει τοποθετηθεί κατάλληλο validation για κάθε πεδίο ξεχωριστά. Αναλυτικότερα, κανένα από τα πεδία δεν μπορεί να είναι κενό και όλα τα πεδία πρέπει να συμπληρωθούν ώστε να ενεργοποιηθεί το κουμπί “Create Event”.

The image shows a web form titled "Login Required" centered on a dark background. The form has a white background and contains three input fields: "Your Name" with the value "Λάμπρος Βαρνάβας", "E-Mail" with the value "test@test.com", and "Password" with masked characters ".....". Below the fields are three buttons: a red "SIGNUP" button, a white "LOGIN" button with a red border, and a white "RESET PASSWORD" button with a red border. At the bottom of the form area, there is a small copyright notice: "Copyright © 2020 Events. Developed by Lampros Varnavas | Supervised by Minas Dasyverdis".

Σχήμα 4.9: Σελίδα εγγραφής με έγκυρα στοιχεία στη φόρμα.

Το πεδίο “Title” θα πρέπει να αποτελείται τουλάχιστον 3 χαρακτήρες. Το πεδίο “Address” είναι αναγκαίο να περιλαμβάνει υπαρκτή διεύθυνση έτσι ώστε να γίνει η μετατροπή της διεύθυνσης που εισήγαγε ο χρήστης στις αντίστοιχες συντεταγμένες που θα καθορίζουν και θα εμφανίζουν την ακριβή τοποθεσία της εκδήλωσης στον χάρτη. Στο πεδίο “Start Date” θα πρέπει να γίνει επιλογή ημερομηνίας έναρξης από το ημερολόγιο που εμφανίζεται όταν επιλεχθεί το βέλος που εμφανίζεται στο input του πεδίου ή να πληκτρολογηθεί η επιθυμητή ημερομηνία, αλλά για λόγους εγκυρότητας, συνίσταται η επιλογή του ημερολογίου, και επίσης την ίδια λογική ακολουθούν και τα πεδία “End Date” και “Start Time”. Στις περιπτώσεις επιλογής πόλης, νομού και κατηγορίας η διαδικασία είναι παρόμοια με την διαφορά ότι ο χρήστης πρέπει να τα επιλέξει από μια συγκεκριμένη λίστα που έχει οριστεί από τον προγραμματιστή. Το πεδίο “Description” ακολουθεί το validation της απαραίτητης συμπλήρωσης τουλάχιστον έξι χαρακτήρων για να είναι έγκυρο. Τέλος, το πεδίο μεταφόρτωσης εικόνων “Pick Images” γίνεται έγκυρο με την επιλογή τουλάχιστον μιας εικόνας. Παρακάτω παρουσιάζεται η φόρμα εισαγωγής στοιχείων εκδήλωσης



The screenshot shows a login form titled "Login Required" centered on a dark background. The form has a white background and contains the following elements from top to bottom: a red heading "Login Required", a red label "E-Mail" above an empty text input field, a red error message "Please enter a valid email address.", a red label "Password" above an empty text input field, a red error message "Please enter a valid password, at least 6 characters.", a grey "LOGIN" button, a red "SIGNUP" button, and a red "RESET PASSWORD" button. At the bottom of the page, a red footer contains the text: "Copyright © 2020 Events. Developed by Lampros Varnavas | Supervised by Minas Dasygenis".

Σχήμα 4.10: Σελίδα εισόδου με κενά στοιχεία στη φόρμα.

στη σελίδα “Create Event” όπως και οι περιπτώσεις διεπαφής χρήστη.

Στο σχήμα 4.18 φαίνεται η φόρμα δημιουργίας εκδήλωσης και ένα παράδειγμα μηνύματος για μη έγκυρη τιμή στο πεδίο του τίτλου. Αντίστοιχα μηνύματα υπάρχουν και για τα υπόλοιπα πεδία.

Στο σχήμα 4.19 παρουσιάζεται η περίπτωση όπου ο χρήστης έχει συμπληρώσει σωστά όλα τα πεδία για τη δημιουργία μιας εκδήλωσης.

Στο σχήμα 4.20 εμφανίζεται η περίπτωση όπου τα όλα τα πεδία είναι έγκυρα, αλλά στο πεδίο της διεύθυνσης έχει τοποθετηθεί διεύθυνση η οποία δεν υφίσταται, όπως για παράδειγμα η τοποθεσία “τεστ”, και εμφανίζεται κατάλληλο μήνυμα από τον server ότι δεν βρέθηκε τοποθεσία για την συγκεκριμένη διεύθυνση.

Στο σχήμα 4.21 φαίνεται η περίπτωση όπου έχουν συμπληρωθεί όλα τα πεδία και δεν έχει γίνει για παράδειγμα η επιλογή εικόνας. Το κουμπί “Create Event” δεν ενεργοποιείται. Αντίστοιχα, και για κάποιο από τα υπόλοιπα πεδία. Όπως αναφέρθηκε και προηγουμένως, είναι απαραίτητη η συμπλήρωση όλων των πεδίων ώστε να είναι εφικτή η δημιουργία μιας εκδήλωσης για να αποφεύγεται αρχικά η έλλειψη

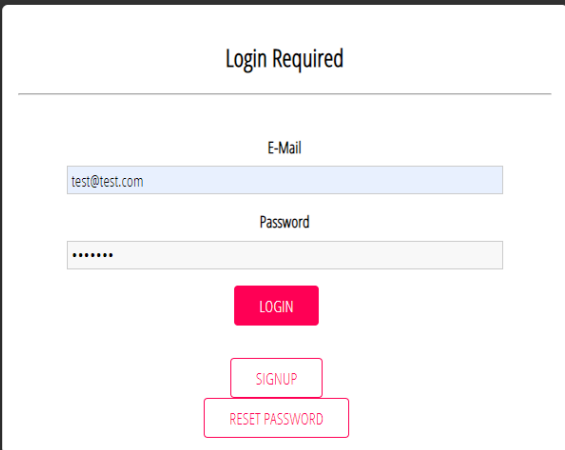
Copyright © 2020 Events. Developed by Lampros Varnavas | Supervised by Minas Dasygenis

Σχήμα 4.11: Σελίδα εισόδου με μη έγκυρα στοιχεία στη φόρμα.

πληροφοριών και τελικά η αποθήκευση κενών στοιχείων στη βάση το οποίο βέβαια δεν μπορεί να συμβεί εκ των πραγμάτων λόγω της λογικής που ακολουθεί η βάση δεδομένων για τα δεδομένα που δέχεται.

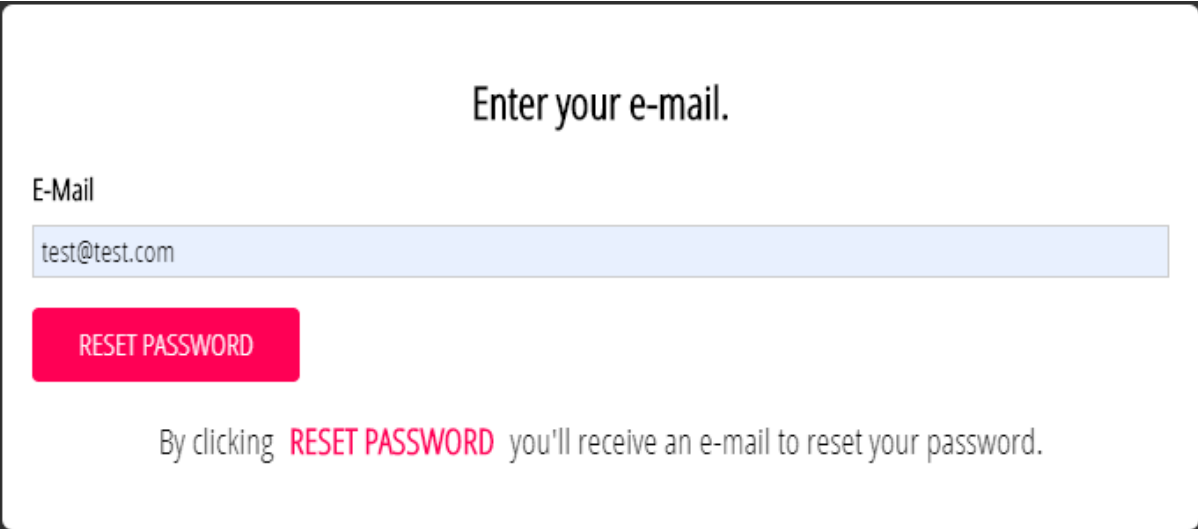
4.1.5 Σελίδα Διαχείρισης εκδηλώσεων χρήστη

Στη σελίδα αυτή ο χρήστης που έχει δημιουργήσει εκδηλώσεις, επιλέγοντας από το navigation bar το πεδίο “My Events” μεταφέρεται στην σελίδα όπου εμφανίζονται οι εκδηλώσεις που του ανήκουν όπως και ένα μήνυμα το οποίο αναγράφει τον αριθμό των εκδηλώσεων που έχει δημιουργήσει. Σε κάθε μια από τις εκδηλώσεις, υπάρχουν τρεις επιλογές. Οι επιλογές αυτές, δίνονται στον χρήστη κάνοντας κλικ στο αντίστοιχο κουμπί. Συγκεκριμένα, υπάρχουν τα κουμπιά για επεξεργασία (Edit), για διαγραφή (Delete) και για προβολή τοποθεσίας στον χάρτη (View on map). Όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο, η επιλογή επεξεργασίας προσφέρει τη δυνατότητα μετατροπής ορισμένων πεδίων, όπως είναι ο τίτλος, η διεύθυνση, οι ημερομηνίες έναρξης και λήξης, η ώρα έναρξης και η περιγραφή.



Copyright © 2020 Events. Developed by Lampros Varnavas | Supervised by Minas Dasigenis

Σχήμα 4.12: Σελίδα εισόδου με έγκυρα στοιχεία στη φόρμα.



By clicking **RESET PASSWORD** you'll receive an e-mail to reset your password.

Σχήμα 4.13: Σελίδα ανάκτησης κωδικού πρόσβασης (εισαγωγή email).

Αλλάζοντας τη διεύθυνση, αυτόματα αλλάζει και η τοποθεσία της εκδήλωσης που εμφανίζεται στον χάρτη. Τα πεδία επιδέχονται πολλαπλή μετατροπή, δηλαδή αλλαγή είτε μερικών είτε όλων των στοιχείων που προαναφέρθηκαν. Στην επιλογή της διαγραφής, όταν γίνεται κλικ στο κουμπί “Delete” εμφανίζεται προειδοποιητικό μήνυμα στην οθόνη, ρωτώντας τον χρήστη εάν είναι σίγουρος για την επιλογή της

Enter your new password.

Password

.....

Re-type the new password

.....

UPDATE PASSWORD

Σχήμα 4.14: Σελίδα αλλαγής κωδικού πρόσβασης (reset).

EVENTS

ALL EVENTS SEARCH AUTHENTICATE

Search for Events choosing a city from the list.

Pick city

SEARCH

Search for Events choosing a prefecture from the list.

Pick prefecture

SEARCH

Search for Events choosing a category from the list.

Pick category

SEARCH

Search for Events choosing a date from the calendar.

Pick starting date

mm/dd/yyyy

SEARCH

Copyright © 2020 Events. Developed by Lampros Varnavas | Supervised by Minas Dotsis

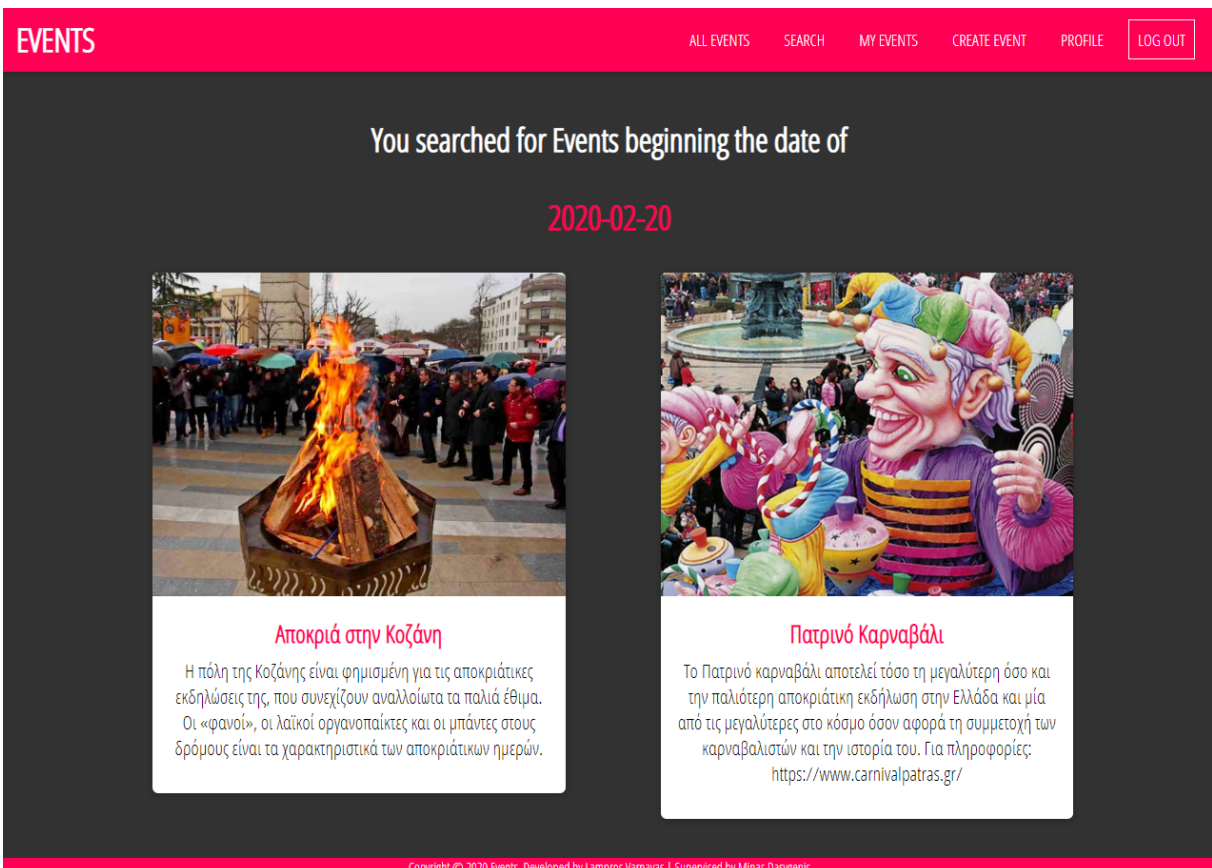
Σχήμα 4.15: Σελίδα αναζήτησης.

αφαίρεσης της εκδήλωσης από το σύστημα, καθώς δεν είναι εφικτή η αναίρεση της. Παρακάτω παρουσιάζονται οι επιλογές και οι δυνατότητες που αναφέρθηκαν.

Στο σχήμα 4.22 γίνεται προβολή του περιεχομένου της σελίδας “My Events” ενός χρήστη που έχει δημιουργήσει εκδήλωση.



Σχήμα 4.16: Παράδειγμα μηνύματος σφάλματος στη σελίδα αναζήτησης.



Σχήμα 4.17: Σελίδα εκδηλώσεων που πραγματοποιούνται μια συγκεκριμένη ημερομηνία.

Κάνοντας κλικ στο κουμπί “Edit” όπως αυτό προβάλλεται στο σχήμα 4.23, ο χρήστης μεταβαίνει στη σελίδα επεξεργασίας εκδήλωσης, όπου εμφανίζεται μια φόρμα με τα πεδία που επιδέχονται μετατροπή. Επιπλέον, μπορεί να βλέπει τα στοιχεία που ήδη έχει επιλέξει προηγουμένως. Αφού, λοιπόν, ο χρήστης κάνει τις αλλαγές που επιθυμεί, επιλέγει το κουμπί “Update Event” που υπάρχει στο τέλος της φόρμας επεξεργασίας. Αυτή η περίπτωση φαίνεται στο σχήμα 4.24.

Περίπτωση όπου ο χρήστης επιλέγει το κουμπί “Delete”, όπως αυτό παρουσιάζεται στο σχήμα 4.25, και εμφανίζεται, όπως στο σχήμα 4.26, το μήνυμα προειδο-

The screenshot shows a web interface for creating an event. At the top, there is a red navigation bar with the word 'EVENTS' on the left and several links: 'ALL EVENTS', 'SEARCH', 'MY EVENTS', 'CREATE EVENT', 'PROFILE', and 'LOG OUT'. Below the navigation bar, the main content area has a dark background. A white form is centered on the page with the title 'Insert the information to each field to create a new Event'. The form contains the following fields:

- Title:** A text input field with a red border and a red error message below it: 'Please enter a valid title (at least 3 characters)'.
- Address:** A text input field with a placeholder 'π.χ Παλατιού 10, 50100'.
- Start Date:** A date input field with a placeholder 'mm/dd/yyyy'.
- End Date:** A date input field with a placeholder 'mm/dd/yyyy'.
- Start Time:** A time input field with a placeholder '--:--'.
- City:** A dropdown menu.
- Prefecture:** A dropdown menu.
- Category:** A dropdown menu.
- Description:** A text area with a placeholder 'π.χ Η εκδήλωση...'.
- Image:** A placeholder box with the text 'Pick an image.' and a red 'Pick Images' button below it. A red error message below the button says 'Please provide at least one image.'.
- Submit:** A grey 'ADD EVENT' button at the bottom of the form.

 At the very bottom of the page, there is a small copyright notice: 'Copyright © 2020 Events. Developed by Lampira Varnas | Supervised by Mino Dazgimis'.

Σχήμα 4.18: Σελίδα δημιουργίας εκδήλωσης με κενές τιμές.

ποίησης και τα αντίστοιχα κουμπιά για την επιβεβαίωση αυτής της πράξης “Delete” και “Cancel” για την ακύρωση της.

Αν ο χρήστης επιλέξει το κουμπί “View on map” όπως αυτό προβάλλεται στο σχήμα 4.27, βλέπει στον χάρτη την τοποθεσία της διεύθυνσης που εισήγαγε στο πεδίο “Address” στη δημιουργία της εκδήλωσης. Ο χάρτης που εμφανίζεται στο σχήμα 4.28 είναι API της Google, και δίνει δυνατότητες όπως (zoom in, zoom out, street view, full screen, simple map, satellite).

4.1.6 Σελίδα επιλεγμένης εκδήλωσης

Στη συγκεκριμένη σελίδα ο χρήστης μεταφέρεται όταν από τη λίστα των εκδηλώσεων που υπάρχουν, επιλέξει, κάνοντας κλικ, την εκδήλωση που τον ενδιαφέρει. Εκεί, εμφανίζονται όλες οι πληροφορίες της εκδήλωσης, καθώς και οι δυνατότητες που έχει. Αναλυτικά, υπάρχουν όλα τα στοιχεία για την εκδήλωση, δηλαδή όσα είναι απαραίτητα κατά την δημιουργία της, όπως επίσης και το όνομα του χρήστη που την δημιούργησε. Επιπλέον, οι επιλογές που έχει ο χρήστης σε αυτή τη σελίδα

Title
Web Development

Address
Καραμανλή και Λυγερής, Κοζάνη

Start Date
04/27/2020

End Date
04/27/2020


Start Time
06:00 PM

City
ΚΟΖΑΝΗ

Prefecture
ΚΟΖΑΝΗΣ

Category
ΕΠΙΣΤΗΜΗ

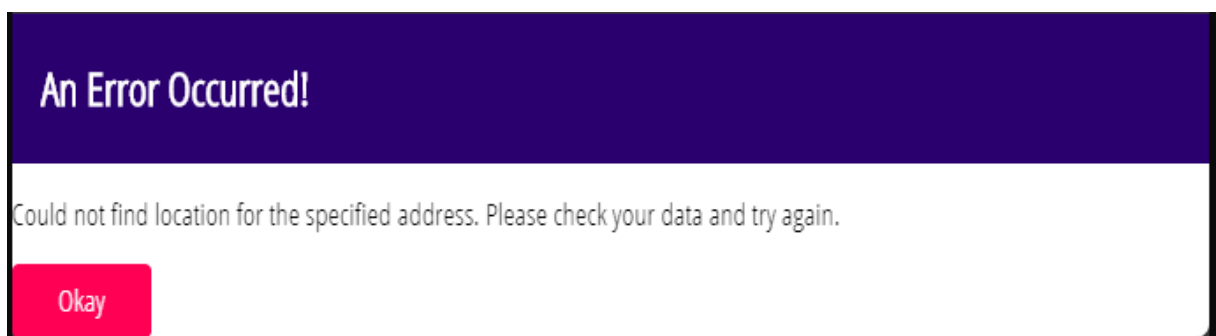
Description
Η εκδήλωση απευθύνεται στους φοιτητές του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας και αφορά την ανάπτυξη ιστοσελίδων. Για περισσότερες πληροφορίες: <https://ece.uowm.gr/>



Pick Images

ADD EVENT

Σχήμα 4.19: Σελίδα δημιουργίας εκδήλωσης με έγκυρες τιμές.



Σχήμα 4.20: Μήνυμα σφάλματος στη σελίδα δημιουργίας εκδήλωσης για μη έγκυρη διεύθυνση.

είναι η προβολή της τοποθεσίας στο χάρτη, με το ίδιο κουμπί “View on map” ο χρήστης βλέπει την τοποθεσία που διεξάγεται η εκδήλωση, η επιλογή για αναφορά

The image shows a web form for creating an event. The fields are as follows:

- Title:** Web Development
- Address:** Καραμανλή και Λυγερός, Κοζάνη
- Start Date:** 04/27/2020
- End Date:** 04/27/2020
- Start Time:** 06:00 PM
- City:** ΚΟΖΑΝΗ
- Prefecture:** ΚΟΖΑΝΗΣ
- Category:** ΕΠΙΣΤΗΜΗ
- Description:** Η εκδήλωση απευθύνεται στους φοιτητές του Τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας και αφορά την ανάπτυξη ιστοσελίδων. Για περισσότερες πληροφορίες: <https://ece.uowm.gr/>

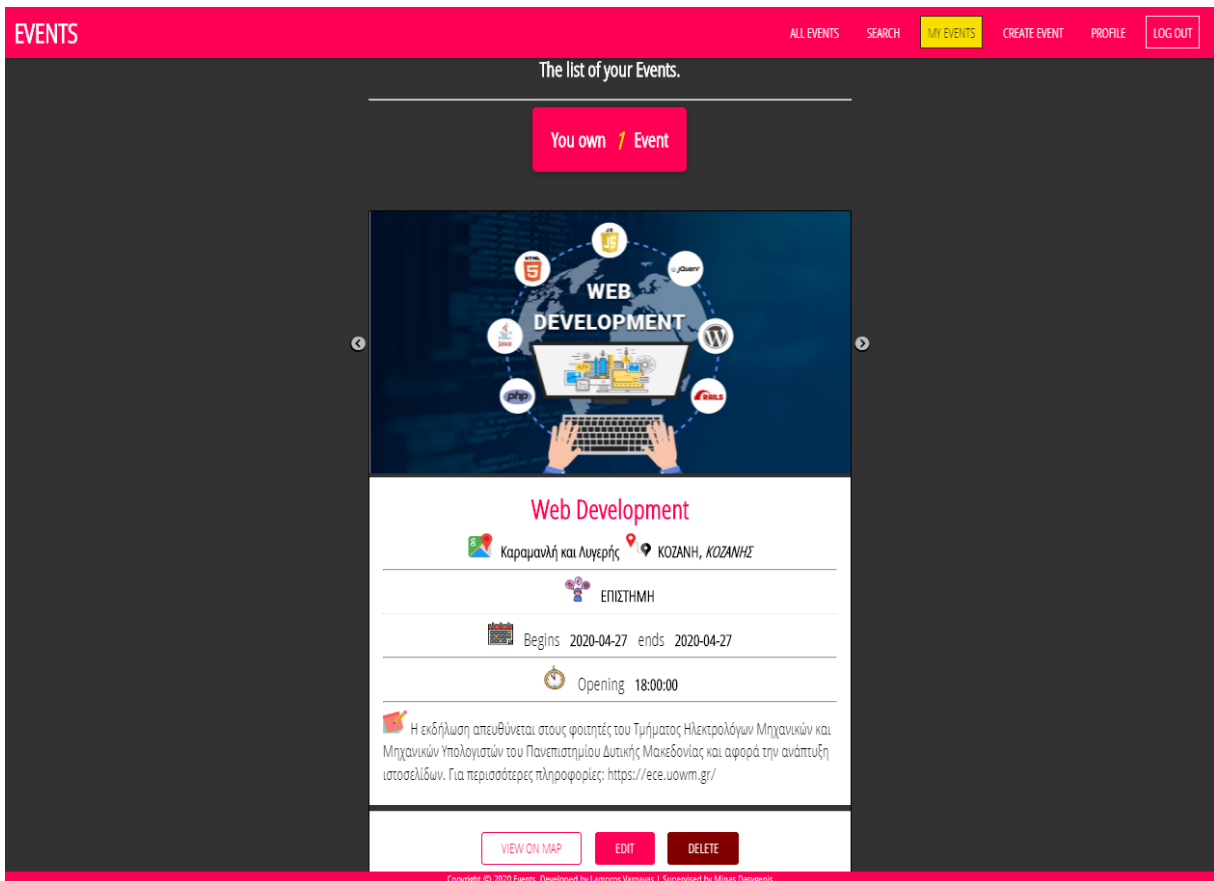
Below the description is a placeholder for an image with the text "Pick an image." and a red "Pick Images" button. At the bottom, there is a note "Please provide at least one image." and a grey "ADD EVENT" button.

Σχήμα 4.21: Σελίδα δημιουργίας εκδήλωσης με κενή τιμή στο πεδίο επιλογής εικόνας.

της εκδήλωσης στον δημιουργό της, πατώντας το κουμπί “Report”, και η προσθήκη σχολίου στην περιοχή σύνταξης κειμένου (text-area) που βρίσκεται στο κάτω μέρος της σελίδας, επιλέγοντας το κουμπί “Add Comment”, για αυτή την εκδήλωση. Τέλος, οι χρήστες που προσθέτουν σχόλια, μπορούν να αφαιρέσουν τα σχόλια που οι ίδιοι έκαναν, αν επιλέξουν το κουμπί “Delete” που εμφανίζεται μαζί με το συγκεκριμένο σχόλιο. Παρακάτω παρουσιάζονται όλα όσα αναφέρθηκαν.

Στο σχήμα 4.29 γίνεται προβολή του περιεχομένου της συγκεκριμένης σελίδας στην εφαρμογή, όταν μεταβαίνει ο χρήστης κάνοντας κλικ στην εκδήλωση.

Στην περίπτωση όπου κάποιος χρήστης επισκεφθεί τη σελίδα μιας συγκεκριμέ-



Σχήμα 4.22: Σελίδα εκδηλώσεων χρήστη (My events).

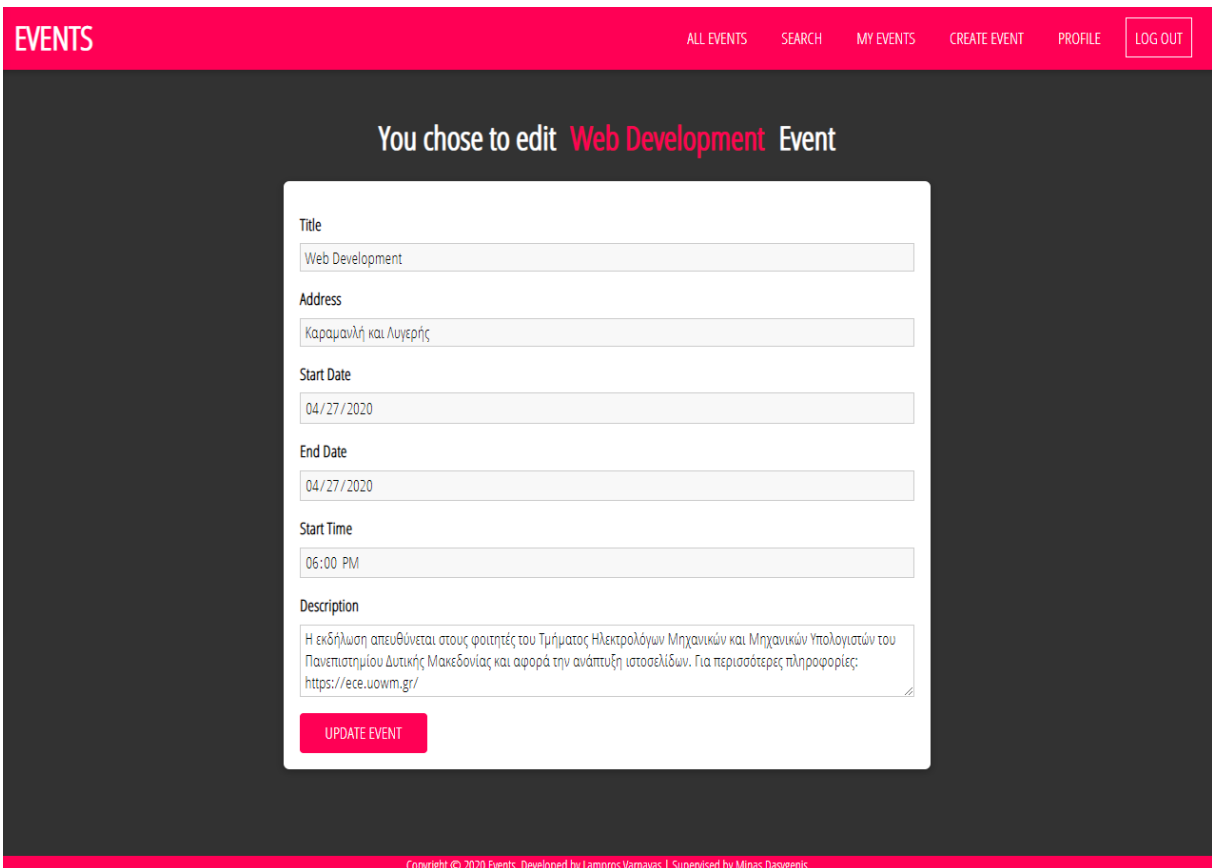


Σχήμα 4.23: Κουμπί επεξεργασίας.

νης εκδήλωσης και λόγω του περιεχομένου της, πιστεύει ότι περιέχει κακόβουλο υλικό, μπορεί να επιλέξει να αναφέρει αυτή την εκδήλωση στον δημιουργό της κάνοντας κλικ στο κουμπί “Report” όπως αυτό εμφανίζεται στο σχήμα 4.30.

Όταν γίνεται κλικ στο κουμπί αναφοράς “Report”, όπως και στην περίπτωση του κουμπιού διαγραφής “Delete”, εμφανίζεται μήνυμα που προειδοποιεί τον χρήστη για αυτή την ενέργεια, και τον ρωτάει αν είναι σίγουρος να προχωρήσει πατώντας “Report” ή να ακυρώσει πατώντας “Cancel”, καθώς αυτή η πράξη δεν μπορεί να αναιρεθεί, όπως φαίνεται στο σχήμα 4.31. Ο χρήστης μπορεί να επιλέξει να αναφέρει μια εκδήλωση μόνο μια φορά. Αν ξανά προσπαθήσει να το κάνει, τότε εμφανίζεται κατάλληλο μήνυμα πως έχει ήδη προσθέσει την αναφορά του για την εκδήλωση αυτή, όπως φαίνεται στο σχήμα 4.32.

Όπως ήδη αναφέρθηκε, ο χρήστης έχει την επιλογή να προσθέσει κάποιο σχόλιο



Σχήμα 4.24: Σελίδα επεξεργασίας (Edit) εκδήλωσης.



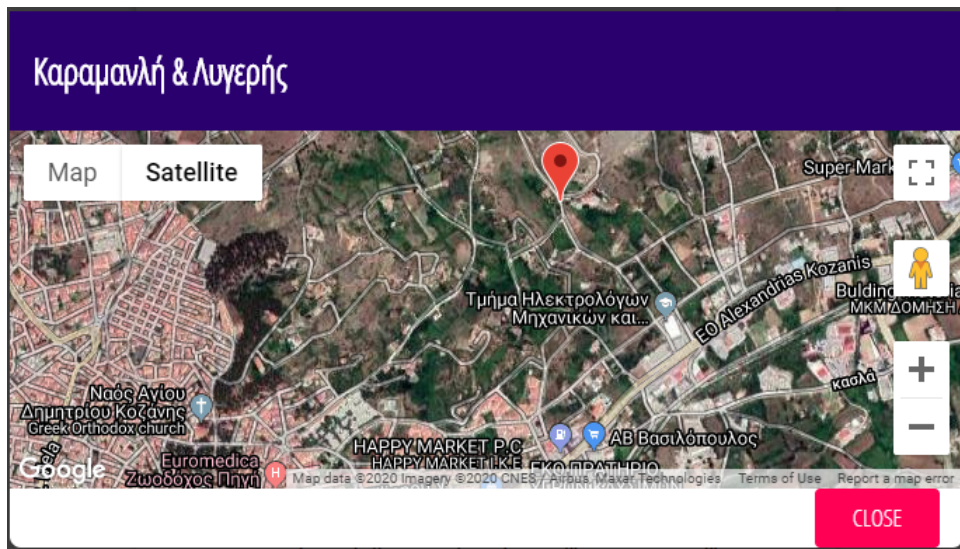
Σχήμα 4.25: Κουμπί διαγραφής.



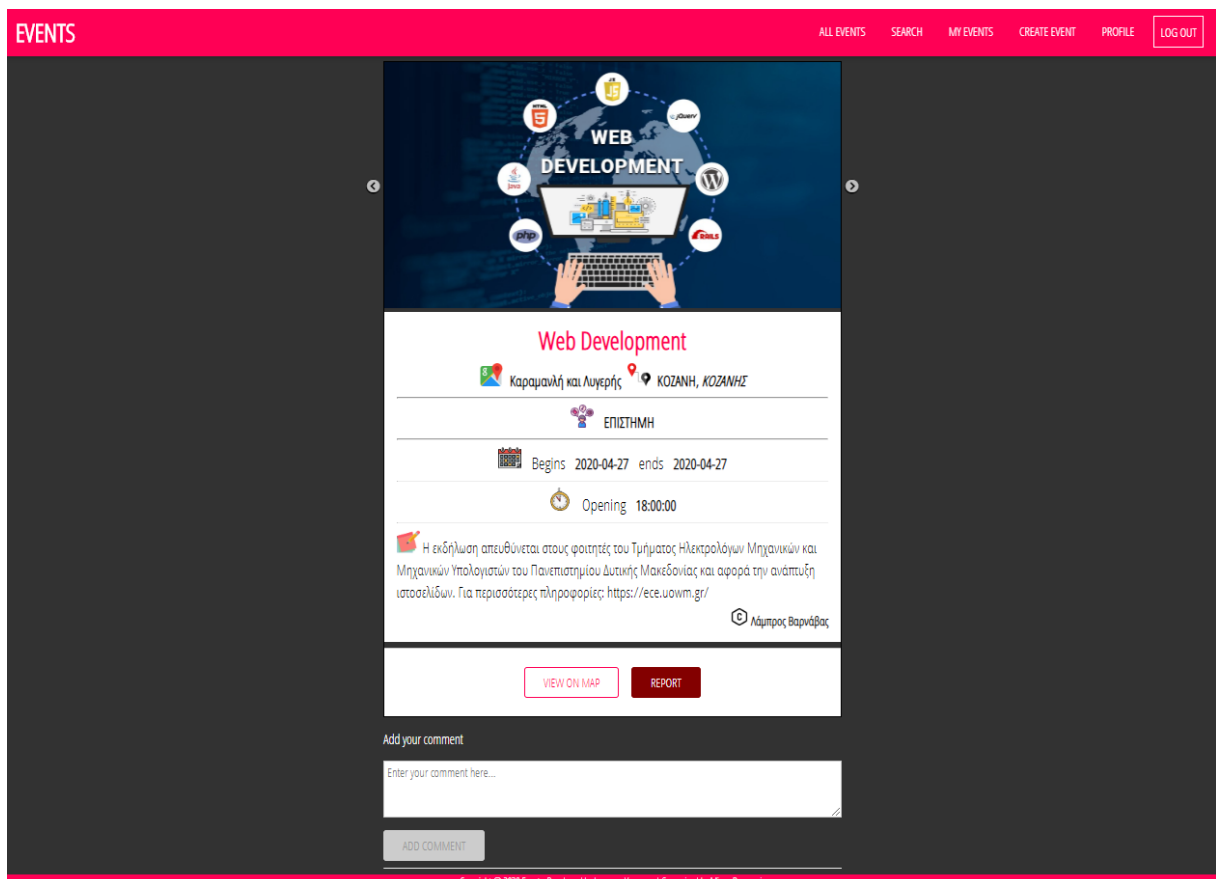
Σχήμα 4.26: Μήνυμα προειδοποίησης για την επιβεβαίωση της διαγραφής εκδήλωσης.



Σχήμα 4.27: Κουμπί προβολής τοποθεσίας στο χάρτη.



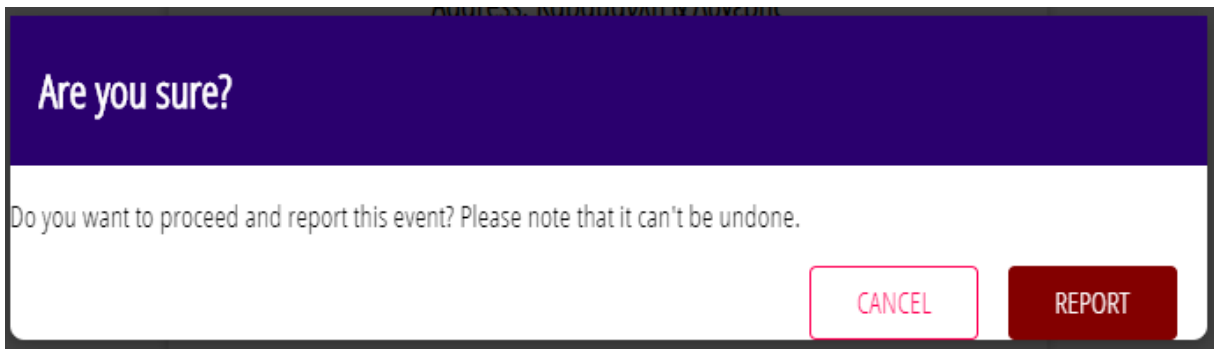
Σχήμα 4.28: Προβολή τοποθεσίας εκδήλωσης στον χάρτη (Satellite).



Σχήμα 4.29: Προβολή περιεχομένου επιλεγμένης εκδήλωσης.



Σχήμα 4.30: Κουμπί αναφοράς.



Σχήμα 4.31: Μήνυμα προειδοποίησης για την αναφορά εκδήλωσης.



Σχήμα 4.32: Μήνυμα ειδοποίησης για εκδήλωση που έχει ήδη αναφερθεί από τον ίδιο χρήστη.

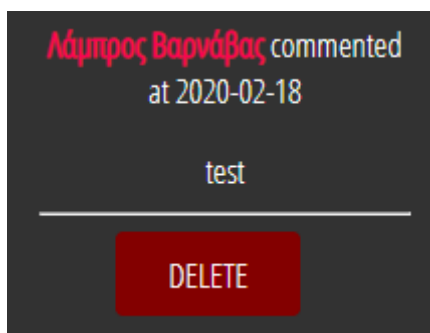
για μια εκδήλωση. Αυτό είναι εφικτό, πληκτρολογώντας στην περιοχή κειμένου σχολίου τον προσωπικό του λόγο, όμως μόνο εφόσον συμπληρώσει τουλάχιστον τρεις χαρακτήρες για να αποφεύγονται μικρά και ανούσια σχόλια. Άρα, κι αυτό το πεδίο περιλαμβάνει validation καθώς ο χρήστης δεν μπορεί να στείλει κενή τιμή, και προειδοποιείται με αντίστοιχο μήνυμα. Το κουμπί “Add Comment” παραμένει απενεργοποιημένο μέχρι να είναι σωστό το validation. Η περίπτωση αυτή φαίνεται στο σχήμα 4.33.



Σχήμα 4.33: Έγκυρη προσθήκη σχολίου.

Αφού γίνει η προσθήκη κάποιου σχολίου, τότε το σχόλιο εμφανίζεται στη συγκεκριμένη σελίδα της εκδήλωσης, στο κάτω μέρος της. Ο χρήστης που έγραψε αυτό

το σχόλιο, μπορεί να το αφαιρέσει πατώντας το “Delete” κουμπί. Για τη διαγραφή του σχολίου, εμφανίζεται επίσης προειδοποιητικό μήνυμα στην οθόνη, για το εάν ο χρήστης θέλει να προχωρήσει σε αυτή την ενέργεια επιλέγοντας “Delete” ή “Cancel” για ακύρωση. Επιπλέον, εκτός από το περιεχόμενο του σχολίου, δηλαδή το κείμενο, εμφανίζεται το όνομα του χρήστη που έκανε το σχόλιο, καθώς και η ημερομηνία την οποία προστέθηκε, όπως φαίνεται στο παράδειγμα του σχήματος 4.34.



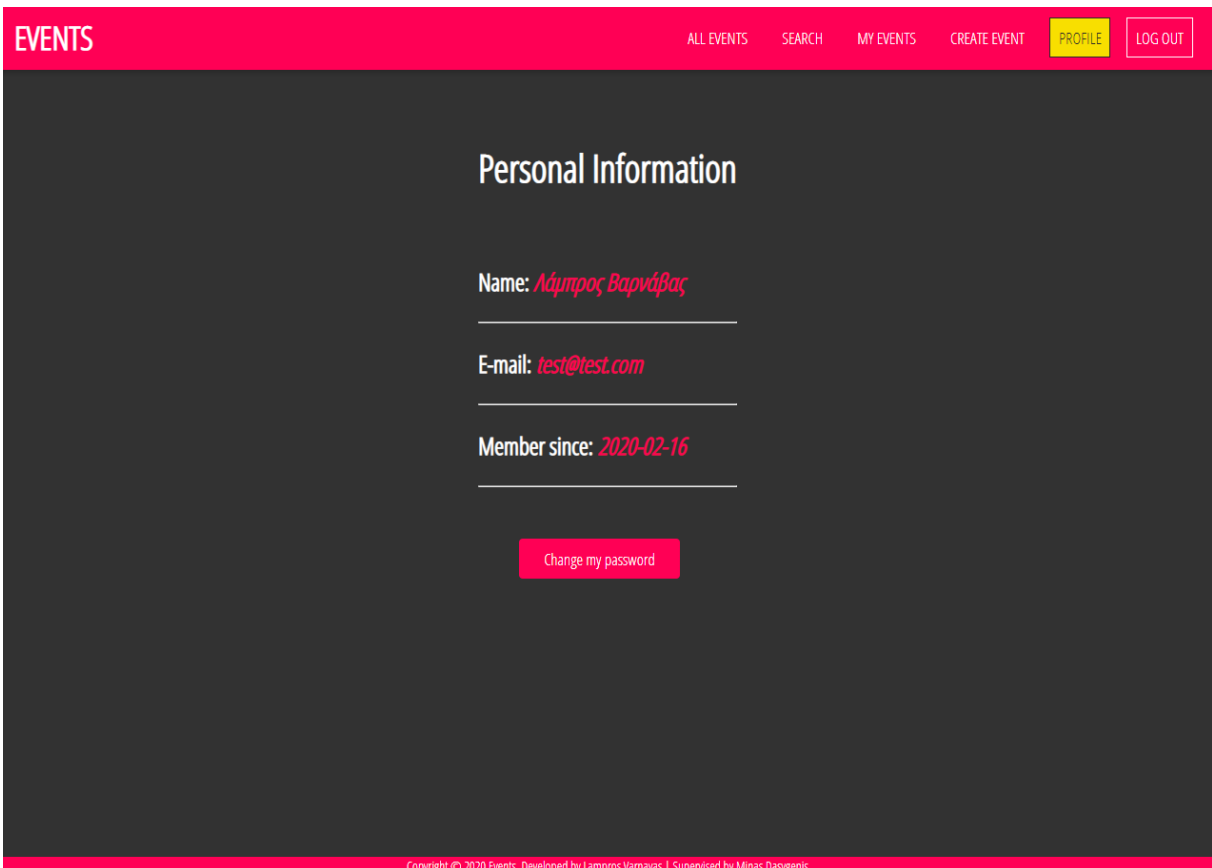
Σχήμα 4.34: Παράδειγμα σχολίου όπως αυτό εμφανίζεται στη σελίδα.

Η διαδικασία της διαγραφής σχολίου, ακολουθεί την ίδια λογική όπως αυτή παρουσιάζεται στο σχήμα 4.25 και 4.26.

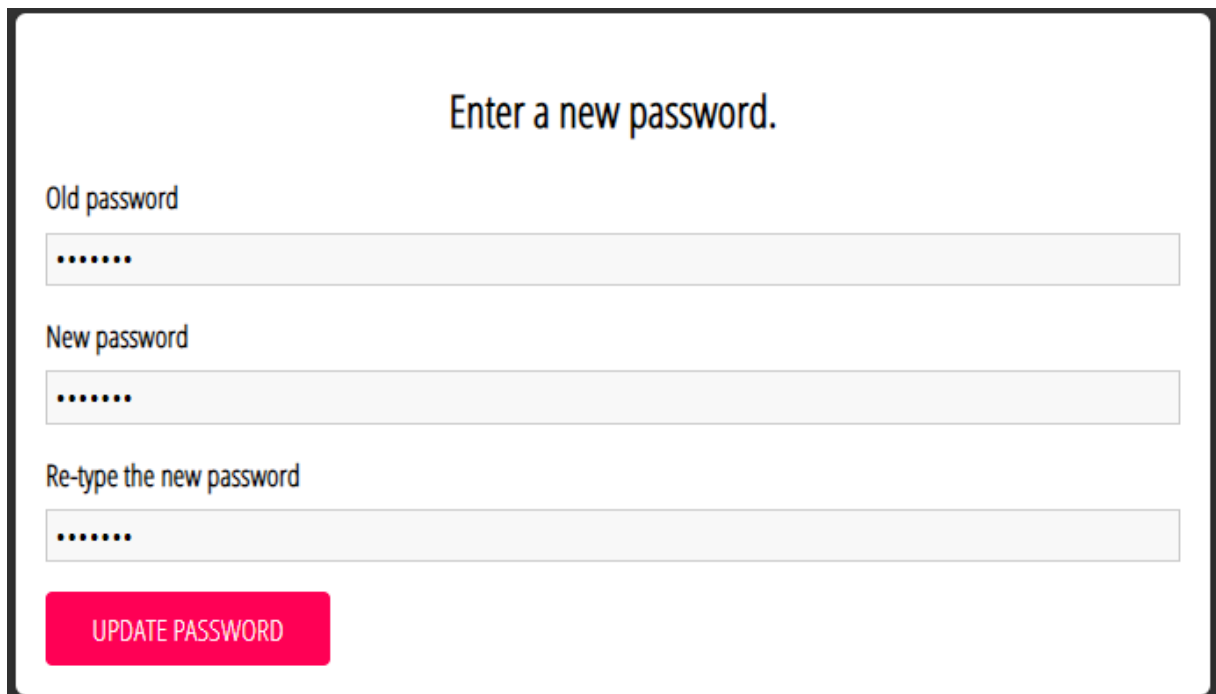
Είναι σημαντικό να αναφερθεί σε αυτό το σημείο, πως το κουμπί “Delete” εμφανίζεται μόνο στα σχόλια τα οποία έγραψαν οι συγκεκριμένοι χρήστες.

4.1.7 Σελίδα προφίλ χρήστη

Στη σελίδα του προφίλ χρήστη, εμφανίζονται κάποια από τα προσωπικά στοιχεία του χρήστη, όπως είναι το όνομα, το email του και η ημερομηνία που έγινε χρήστης του ιστοχώρου. Επιπλέον, σε αυτή τη σελίδα δίνεται η δυνατότητα στον χρήστη να αλλάξει τον κωδικό πρόσβασης του στον ιστοχώρο, κάνοντας κλικ στο κουμπί “Change my password” που εμφανίζεται. Έπειτα, ο χρήστης μεταβαίνει στη σελίδα αλλαγής του κωδικού πρόσβασης, όπου του ζητείται να εισάγει τον παλιό κωδικό και έναν καινούριο κωδικό δύο φορές για λόγους ασφαλείας. Αν και μόνο αν και τα τρία πεδία είναι έγκυρα, τότε εμφανίζεται το κουμπί “Update Password” και μπορεί να γίνει η αλλαγή του κωδικού πρόσβασης. Σε αντίθετη περίπτωση, εμφανίζεται κατάλληλο μήνυμα. Είναι σημαντικό να σημειωθεί, πως και στη συγκεκριμένη περίπτωση το password του χρήστη ακολουθεί τη λογική της διαδικασίας εγγραφής, όπου γίνεται η κρυπτογράφηση του. Το σχήμα 4.35 δείχνει το περιεχόμενο της σελίδας “Profile”, και το σχήμα 4.36 τη φόρμα αλλαγής password.



Σχήμα 4.35: Σελίδα προφίλ εγγεγραμμένου χρήστη.



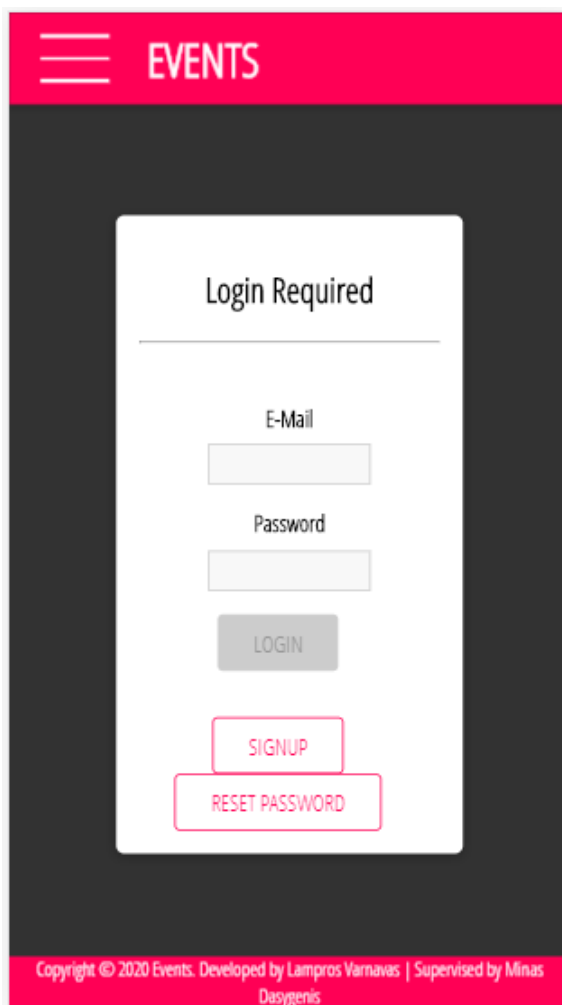
Σχήμα 4.36: Σελίδα αλλαγής κωδικού πρόσβασης (logged in).

4.2 Προβολή διαδικτυακής εφαρμογής από μικρότερες οθόνες συσκευών

Σε αυτή την ενότητα, παρουσιάζεται το ενδεχόμενο πρόσβασης στην ιστοσελίδα από συσκευές που το μέγεθος της οθόνης τους είναι μικρότερο από ένα τυπικό μέγεθος μιας οθόνης ενός Η/Υ. Είναι πολύ συχνό, το φαινόμενο, εν έτη 2020, να γίνεται χρήση του διαδικτύου από κινητές συσκευές και tablet. Για το λόγο αυτό, αλλά και για να έχει ο χρήστης καλύτερη και αποδοτικότερη περιήγηση στην συγκεκριμένη διαδικτυακή εφαρμογή, ήταν επιτακτική ανάγκη να καλυφθεί η περίπτωση αυτή, η οποία είναι γνωστή ως “responsive design” για mobile view. Ένα από τα σημαντικά κομμάτια αυτής της υλοποίησης, είναι η αναδιαμόρφωση της μπάρας πλοήγησης (navigation bar), όπου υπάρχουν τα βασικά πεδία επιλογής για την πλοήγηση στην εφαρμογή. Έτσι, υλοποιήθηκε με τρόπο τέτοιο, κάνοντας χρήση της βιβλιοθήκης “React Transition Group”, ώστε να μετασχηματίζεται η μπάρα αυτή σε διαστάσεις μικρότερες από (768 x 657) και να εμφανίζεται στην επάνω αριστερή άκρη της οθόνης ένα drop down μενού με τις διαθέσιμες επιλογές πλοήγησης. Επίσης, για διαστάσεις που αφορούν την πλειοψηφία των συσκευών προσαρμόζεται και το περιεχόμενο της εφαρμογής. Στο σχήμα 4.37 και 4.38, παρουσιάζονται οι περιπτώσεις αυτές.

4.3 Ανάλυση σημαντικών σημείων κώδικα

Η ενότητα αυτή, πραγματεύεται τα σημεία αυτά του κώδικα, τόσο στο front-end όσο και στο back-end, που καθόρισαν την ορθή και επιτυχή λειτουργία του συστήματος. Αρχικά, παρουσιάζονται και αναλύονται τα κομμάτια κώδικα στο front-end και στη συνέχεια ακολουθούν αυτά στο RESTful API (back-end) της διαδικτυακής εφαρμογής με επιλεκτική αναφορά και στιγμιότυπα. Τα κομμάτια αυτά είναι γραμμένα στη γλώσσα προγραμματισμού JavaScript, με βασική τη χρήση της βιβλιοθήκης React και των Components που περιλαμβάνει, καθώς επίσης και χρησιμοποιώντας την βιβλιοθήκη Expressjs στο Nodejs του back-end server.

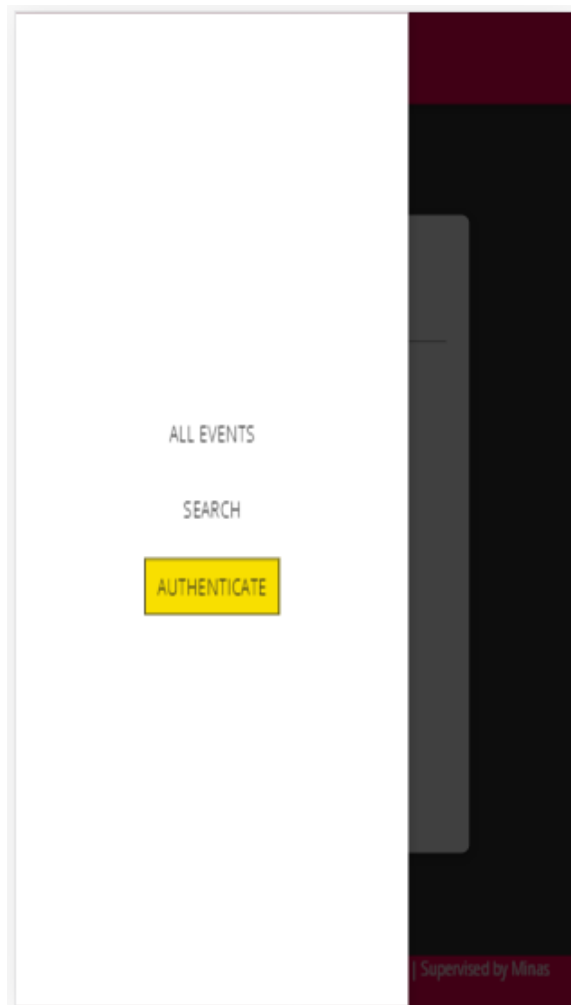


Σχήμα 4.37: Προβολή σελίδας “Authenticate” με κλειστό navigation bar από συσκευή μικρότερης οθόνης.

4.3.1 Κομμάτια κώδικα στο Front-end

React router DOM

Καθώς η διαδικτυακή εφαρμογή έχει ακολουθήσει το πρότυπο SPA (single-page application), η καλύτερη επιλογή για την δυνατότητα πλοήγησης των χρηστών μεταξύ των σελίδων ήταν η χρήση της βιβλιοθήκης “react-router-dom”, η οποία βέβαια δεν ανήκει στην οικογένεια της React, αλλά λόγω της ευχρηστίας και αξιοπιστίας της επιλέγεται κατά κόρον από την πλειοψηφία των προγραμματιστών. Στο σχήμα 4.39, προβάλλεται η δομή των routes της εφαρμογής, που αφορούν τις σελίδες του εγγεγραμμένου χρήστη, και η σύνταξη της γίνεται με ασύγχρονο τρόπο. Αντίστοιχη λογική ακολουθεί και η δομή των routes που αφορούν μη εγγεγραμμένο χρήστη.



Σχήμα 4.38: Προβολή σελίδας “Authenticate” με ανοιχτό navigation bar από συσκευή μικρότερης οθόνης.

```
if (token) {  
  routes = (  
    <Switch>  
      <Route path="/" exact> <Events/> </Route>  
      <Route path="/search" exact> <Filters/> </Route>  
      <Route path="/profile/:userId" exact> <User/> </Route>  
      <Route path="/user/:userId" exact> <UserEvents/> </Route>  
      <Route path="/specific/:eventId" exact> <SpecificEvent/> </Route>  
      <Route path="/update/:userId" exact> <UpdateUser/> </Route>  
      <Route path="/events/new" exact> <NewEvent/> </Route>  
      <Route path="/edit/:eventId" exact> <UpdateEvent/> </Route>  
      <Route path="/city/:cityId" exact> <CityEvents/> </Route>  
      <Route path="/prefecture/:prefectureId" exact> <PrefectureEvents/> </Route>  
      <Route path="/category/:categoryId" exact> <CategoryEvents/> </Route>  
      <Route path="/startdate/:date" exact> <StartDateEvents/> </Route>  
      <Redirect to="/" />  
    </Switch>  
  );  
};
```

Σχήμα 4.39: Κώδικας χρήσης της βιβλιοθήκης “react-router-dom” για εγγεγραμμένο χρήστη.

Code Splitting

Στη συγκεκριμένη διαδικτυακή εφαρμογή δεν είναι απαραίτητο να τρέχουν όλα τα κομμάτια κώδικα (components) από την αρχή. Αν ένας χρήστης επισκεφθεί την ιστοσελίδα μπορεί ή όχι να εισέλθει αυτόματα. Είτε είναι αυτή η περίπτωση είτε όχι, μερικές σελίδες του ιστοχώρου δεν έχουν προσεγγιστεί, οπότε δεν είναι απαραίτητο να τρέξουν τα συγκεκριμένα components, δηλαδή δεν χρειάζεται να γίνει download του κώδικα για αυτά τα routes. Έτσι, ο αρχικός χρόνος για τη φόρτωση των σελίδων της εφαρμογής μειώνεται, επειδή κατεβαίνει λιγότερος κώδικας εξ αρχής και έτσι η αποδοτικότητα του συστήματος αυξάνεται. Αυτό επιτυγχάνεται με την build-in συνάρτηση `React.lazy` και το `Suspense` component όπως φαίνεται στο σχήμα 4.40.

```
const NewEvent = React.lazy(() => import('./events/pages/NewEvent'));
const UserEvents = React.lazy(() => import('./events/pages/UserEvents'));
const UpdateEvent = React.lazy(() => import('./events/pages/UpdateEvent'));
```

Σχήμα 4.40: Χρήση της συνάρτησης `React.lazy` για τη διαίρεση του κώδικα.

Validation

Όπως έγινε γνωστό στην ενότητα 4.2 από τις περιπτώσεις διεπαφής χρήστη, στα πεδία που εισάγονται δεδομένα έχει τοποθετηθεί αντίστοιχο validation. Είναι αναγκαίο λοιπόν, να παρουσιαστεί ένα παράδειγμα κώδικα που εκτελεί τουλάχιστον μια από τις περιπτώσεις αυτές στο front-end. Στο σχήμα 4.41 προβάλλεται η περίπτωση της εισαγωγής password. Αυτό, γίνεται αντιληπτό από το στοιχείο `validators`, το οποίο δέχεται τον validator για ελάχιστους χαρακτήρες και ελέγχει ότι το πεδίο πρέπει να συμπληρωθεί με τουλάχιστον 6 χαρακτήρες.

React Hooks

Τα Hooks είναι μια καινούρια προσθήκη στην έκδοση 16.8 της React και επιτρέπει τη χρήση του state και άλλων χαρακτηριστικών της React χωρίς να χρειάζεται να γραφτεί ένα class. Ουσιαστικά, τα Hooks είναι λειτουργίες που επιτρέπουν τη σύνδεση μέσα στο state της React και στα lifecycle features της από components συ-

```

<Input
  element="input"
  id="password"
  type="password"
  label="Password"
  validators={[VALIDATOR_MINLENGTH(6)]}
  errorText="Please enter a valid password, at least 6 characters."
  onInput={inputHandler}
/>

```

Σχήμα 4.41: Παράδειγμα κώδικα για validation στον κωδικό πρόσβασης.

ναρτήσεων (function components) που δημιουργούνται στον κώδικα. Στην εφαρμογή έχει γίνει χρήση τόσο των βασικών όσο και των επιπρόσθετων Hooks της React, αλλά και custom Hooks. Είναι σημαντικό, να γίνει τουλάχιστον μια αναφορά χρήσης των βασικών Hooks όπως είναι το useState και το useEffect. Ειδικότερα, παρουσιάζεται στο σχήμα 4.42 ένα παράδειγμα χρήσης του useState Hook για την αρχική κατάσταση του token το οποίο στέλνεται στο front-end αφού δημιουργηθεί στο back-end με τη χρήση της βιβλιοθήκης "jsonwebtoken", του user id και της αρχικής ημερομηνίας λήξης του token από τη συνάρτηση "App" που υπάρχει σε αυτό το component ("App" component).

```

const App = () => {
  const [token, setToken] = useState(false);
  const [userId, setUserId] = useState(false);
  const [tokenExpirationDate, setTokenExpirationDate] = useState();

```

Σχήμα 4.42: Κώδικας χρήσης του useState Hook στη React.

Επίσης, στο σχήμα 4.43 φαίνεται ένα παράδειγμα χρήσης του useEffect Hook στο οποίο γίνεται η διαχείριση της αυτόματης εξόδου του χρήστη από το σύστημα μετά από συγκεκριμένο χρονικό διάστημα, δηλαδή όταν το token λήξει. Η λογική αυτή έχει τοποθετηθεί για λόγους ασφάλειας.

4.3.2 Κομμάτια κώδικα στο Back-end - RESTful API

CORS policy

Η γενική ιδέα για το CORS (Cross Origin Resource Sharing) [4] είναι ότι οι πόροι

```

useEffect(() => {
  if (token && tokenExpirationDate) {
    const remainingTime = tokenExpirationDate.getTime() - new Date().getTime();
    logOutTimer = setTimeout(logout, remainingTime)
  } else {
    clearTimeout(logOutTimer);
  }
}, [token, logout, tokenExpirationDate]);

```

Σχήμα 4.43: Κώδικας χρήσης του useEffect Hook στη React.

(resources) σε έναν server μπορούν να ζητηθούν μόνο από requests που έρχονται από τον ίδιο server. Έτσι, κατά την υλοποίηση της εφαρμογής, η οποία έγινε σε localhost αλλά σε διαφορετικά domains για το front-end και το back-end, για να είναι εφικτό να αποσταλούν requests από το front-end μέσω του browser στο back-end έπρεπε να οριστούν στο back-end συγκεκριμένα headers στα responses που στέλνονται πίσω στον client ώστε ο browser να αναγνωρίζει αυτά τα headers και να επιτρέψει την πρόσβαση και να μην εμφανίσει το security error. Οπότε, στο back-end δημιουργήθηκε ένα middleware πριν φτάσουν τα requests στα συγκεκριμένα routes της εφαρμογής, δηλαδή πριν από αυτά τα middlewares ώστε να προσθέτονται τα headers σε όλα τα responses. Επιπλέον, αυτά τα headers είναι απαραίτητα όταν η εφαρμογή τοποθετείται on-line σε δύο διαφορετικούς servers, έναν για το front-end και έναν για τον back-end. Στο σχήμα 4.44 προβάλλεται ο αντίστοιχος κώδικας για την επιτυχή υλοποίηση των παραπάνω.

```

app.use((req, res, next) => {
  res.setHeader('Access-Control-Allow-Origin', '*');
  res.setHeader(
    'Access-Control-Allow-Headers',
    'Origin, X-Requested-With, Content-Type, Accept, Authorization'
  );
  res.setHeader('Access-Control-Allow-Methods', 'GET, POST, PATCH, DELETE');
  next();
});

```

Σχήμα 4.44: Κώδικας για τον ορισμό των αποδεκτών headers (middleware).

Validation

Όπως έχει ήδη αναφερθεί σε προηγούμενη ενότητα, έχει τοποθετηθεί κατάλληλο validation τόσο στο front-end όσο και στο back-end. Χρησιμοποιώντας, λοιπόν, μια

από της πιο γνωστές βιβλιοθήκες της Expressjs για validation στο Nodejs, και συγκεκριμένα την Express Validator, σφραγίζεται το validation και από πλευράς back-end. Στο σχήμα 4.45 παρουσιάζει τη χρήση της συγκεκριμένης βιβλιοθήκης για ένα από τα endpoints, ωστόσο παρόμοια λογική έχει ακολουθηθεί και για τα υπόλοιπα routes που είναι αναγκαίο να έχουν validation.

```
router.post(  
  '/signup',  
  [  
    check('name').not().isEmpty(),  
    check('email')  
      .normalizeEmail() // Test@test.com => test@test.com  
      .isEmail(),  
    check('password').isLength({min: 6})  
  ],  
  usersControllers.signup);
```

Σχήμα 4.45: Κώδικας για validation στο sign up.

Google Maps API

Για να είναι εφικτή η χρήση αλλά και η προβολή του χάρτη στην εφαρμογή, έγινε σύνδεση του back-end server με ένα API της Google με τη χρήση της βιβλιοθήκης “axios” το οποίο μετατρέπει τη διεύθυνση σε συντεταγμένες (latitude και longitude), βρίσκει και προβάλλει αυτή τη διεύθυνση σε έναν διαδικτυακό χάρτη. Στο σχήμα 4.46 προβάλλεται ο κώδικας της συνάρτησης getCoordsForAddress.

Send emails

Σε μερικές διεπαφές χρήστη, γίνεται αποστολή email από τον server της εφαρμογής στην ηλεκτρονική διεύθυνση του με τη χρήση των βιβλιοθηκών “nodemailer” και “nodemailer-sendgrid-transport”. Έτσι, ο χρήστης λαμβάνει αυτά τα emails αποκτώντας την απαραίτητη ενημέρωση. Το περιεχόμενο του σχήματος 4.47 αφορά μια από τις περιπτώσεις που γίνεται η αποστολή. Ωστόσο, πρέπει να σημειωθεί πως ακολουθείται πανομοιότυπη διαδικασία και για τις υπόλοιπες περιπτώσεις.

Password hashing

```

const axios = require('axios');

const HttpError = require('../models/http-error');

const API_KEY = ██████████;

async function getCoordsForAddress(address) {
  const response = await axios.get(
    `https://maps.googleapis.com/maps/api/geocode/json?address=${encodeURIComponent(
      address
    )}&key=${API_KEY}`
  );

  const data = response.data;

  if (!data || data.status === 'ZERO_RESULTS') {
    const error = new HttpError(
      'Could not find location for the specified address.', 422);
    throw error;
  }

  const coordinates = data.results[0].geometry.location;

  return coordinates;
}

module.exports = getCoordsForAddress;

```

Σχήμα 4.46: Κώδικας συνάρτησης getCoordsForAddress για τη σύνδεση με το API της Google.

```

transporter.sendMail({
  to: email,
  from: 'events@app.com',
  subject: 'Sign up succeeded!',
  html: '<h1>You successfully signed up!</h1>'
})

```

Σχήμα 4.47: Κώδικας αποστολής email στην περίπτωση επιτυχούς sign up.

Αναμφίβολα σημαντική είναι η αναφορά της κρυπτογράφησης στη συγκεκριμένη εφαρμογή. Ο χρήστης, κατά την εγγραφή του στο σύστημα, συμπληρώνει τα προσωπικά του στοιχεία και μαζί τον κωδικό πρόσβασης. Ο κωδικός πρόσβασης, είναι αναγκαίο να παραμένει μυστικός τόσο από τους υπόλοιπους χρήστες, όσο και από τους πιθανούς attackers. Για αυτούς τους λόγους, έχει χρησιμοποιηθεί μια από της γνωστότερες βιβλιοθήκες κρυπτογράφησης χαρακτήρων μέσω ενός αξιόπιστου αλγορίθμου που προσφέρει η “bcrypt”. Το κομμάτι κώδικα που περιλαμβάνει το σχήμα 4.48, δείχνει τον τρόπο με τον οποίο γίνεται η κρυπτογράφηση του password.

Images upload

Τελευταία αλλά εξίσου σημαντική, είναι η αναφορά της μεταφόρτωσης εικόνων στο σύστημα. Στη φόρμα δημιουργίας μιας εκδήλωσης, ο χρήστης πρέπει να κά-

```

let hashedPassword;
try {
  hashedPassword = await bcrypt.hash(password, 12);
} catch (err) {
  const error = new HttpError('Could not create user, please try again.', 500);
  return next(error);
}

```

Σχήμα 4.48: Κώδικας κρυπτογράφησης password.

νει upload μια ή περισσότερες φωτογραφίες οι οποίες θα προβάλλονται μαζί με τις υπόλοιπες πληροφορίες μιας εκδήλωσης. Για την αποφυγή μη αποδεκτών αρχείων αλλά και για την αποδοτικότερη λειτουργία του συστήματος, έχουν οριστεί συγκεκριμένοι τύπου αποδεκτών εικόνων αλλά και συγκεκριμένο όριο στο μέγεθος αυτών με τη χρήση της βιβλιοθήκης “Multer”. Συγκεκριμένα, ο χρήστης είναι δυνατόν να ανεβάσει μόνο αρχεία των επεκτάσεων png, jpeg, jpg, και το όριο μεγέθους στα 500000 bytes, δηλαδή 500 kilobytes, όπως αυτό διαπιστώνεται στο σχήμα 4.49. Πρέπει να αναφερθεί επίσης ότι, χρησιμοποιώντας τη βιβλιοθήκη “uuid” γίνεται hash των ονομάτων των εικόνων.

```

const multer = require('multer');
const uuid = require('uuid/v1');

const MIME_TYPE_MAP = {
  'image/png': 'png',
  'image/jpeg': 'jpeg',
  'image/jpg': 'jpg'
};

const fileUpload = multer({
  limits: 500000,
  storage: multer.diskStorage({
    destination: (req, file, cb) => {
      cb(null, 'uploads/images');
    },
    filename: (req, file, cb) => {
      const ext = MIME_TYPE_MAP[file.mimetype];
      cb(null, uuid() + '.' + ext);
    }
  }),
  fileFilter: (req, file, cb) => {
    const isValid = !!MIME_TYPE_MAP[file.mimetype];
    let error = isValid ? null : new Error('Invalid mime type!');
    cb(error, isValid);
  }
});

module.exports = fileUpload;

```

Σχήμα 4.49: Κώδικας μεταφόρτωσης εικόνων.

4.4 Σύνοψη κεφαλαίου

Στο παρόν κεφάλαιο, προτάσσεται η ανάλυση του τρόπου λειτουργίας της διαδικτυακής εφαρμογής. Αρχικά, έγινε λόγος τόσο περιγραφικά όσο και εικονικά για το γραφικό περιβάλλον της εφαρμογής, όπως και όλες οι ενέργειες που μπορούν να εκτελέσουν οι χρήστες σε αυτή. Καταληκτικά, έγινε περιγραφή μερικών σημαντικών κομματιών του κώδικα στο front-end και back-end με την προβολή του στα σημεία αυτά.

Κεφάλαιο 5

Επίλογος

Το κεφάλαιο αυτό αποτελεί μια σύνοψη της διαδικτυακής εφαρμογής που υλοποιήθηκε και σχεδιάστηκε στην συγκεκριμένη διπλωματική εργασία. Αρχικά, γίνεται μια ανακεφαλαίωση των όσων παρουσιάστηκαν. Επιπλέον, γίνεται μια αναφορά στα συμπεράσματα που διεξήχθησαν στο πλαίσιο της υλοποίησης, όπως και τα μετρικά του συστήματος. Και τέλος, ακολουθεί μια σύντομη περιγραφή μελλοντικών επεκτάσεων για τη βελτιστοποίηση του πληροφοριακού συστήματος, καθώς και τα προβλήματα που προέκυψαν στη διαδικασία της υλοποίησης και του σχεδιασμού του.

5.1 Ανακεφαλαίωση

Συγκεφαλαιώνοντας, γίνεται αντιληπτό πως το παρόν έργο μπορεί να τοποθετηθεί ως μια νεοσύστατη και πρωτότυπη ιδέα, αφού συνδυάζει σύγχρονες και ποικίλες τεχνολογίες για την άμεση εύρεση και ενημέρωση διαφόρων εκδηλώσεων αλλά και την ευκαιρία δημιουργίας αυτών από χρήστες του συστήματος. Επίσης, συνδυάζει την αλληλεπίδραση μεταξύ των χρηστών, αφού κάθε χρήστης που είναι εγγεγραμμένος στο σύστημα, αποκτά την δυνατότητα να αφήνει σχόλια στις εκδηλώσεις αλλά και να βλέπει το όνομα του δημιουργού της. Επιπλέον, δόθηκε έμφαση στις πτυχές της διεπαφής χρήστη με τη διαδικτυακή εφαρμογή καθιστώντας τη προσιτή ως προς το κοινό και τον χρήστη, όπως και προσδίδοντας ευχρηστία και αποδοτικότητα σε όλο το εύρος λειτουργιών αυτής. Για την ανάπτυξη της εφαρμογής, έγινε χρήση σύγχρονων τεχνολογιών με βασικές τις HTML5, CSS3, JavaScript, React, Nodejs, Express αλλά και μια σειρά επιπρόσθετων frameworks. Η βάση δεδομένων που επιλέχθηκε είναι η MySQL και περιλαμβάνει οχτώ πίνακες. Η σύνδεση μεταξύ του back-end

server και της βάσης δεδομένων, γίνεται με τη χρήση του Sequelize ORM, κατάλληλο για την ασφάλεια του συστήματος. Η σύνδεση μεταξύ του front-end και του back-end ακολουθεί τη διαδικασία παραγωγής tokens. Για τη μεγιστοποίηση της ασφάλειας έχει τοποθετηθεί κρυπτογράφηση στους κωδικούς πρόσβασης και κατάλληλος έλεγχος για την προσπέλαση των δεδομένων κατά την επικοινωνία των servers, όπως είναι η εισαγωγή δεδομένων στις φόρμες συμπλήρωσης, και τέλος https σύνδεση. Για όλες τις περιπτώσεις επικοινωνίας, γίνεται αποστολή και αποδοχή αρχείων JSON. Συνοψίζοντας, η επιλογή εύκολης εύρεσης και ενημέρωσης για τα δρώμενα της κοινωνίας όπως και η παροχή υπηρεσιών και δυνατοτήτων, αποτελούν το κύριο συστατικό για την πρόοδο και εξέλιξη της υπάρχουσας κατάστασης.

5.1.1 Προβλήματα που προέκυψαν κατά την υλοποίηση

Ένα από τα βασικά προβλήματα στην υλοποίηση του παρόν έργου ήταν η ανάγκη για τη χρήση και την αξιοποίηση των πιο σύγχρονων και νεωτεριστικών τεχνολογιών της σημερινής εποχής, αφού πλέον εφαρμόζονται από την πλειοψηφία των προγραμματιστών τόσο σε ατομικό όσο και σε επιχειρησιακό επίπεδο. Βέβαια, λόγω της πρόσφατης εφαρμογής τους, παρόλο που προσφέρουν περισσότερα στο κομμάτι της υλοποίησης και διευκολύνουν αρκετά τον προγραμματιστή, δεν διατίθενται όσες πηγές πληροφορίες υπάρχουν για τις παλαιότερες τεχνολογίες. Επίσης, το πλήθος των συσκευών που υπάρχουν και που μπορούν να συνδεθούν στο Διαδίκτυο, δημιούργησε την απαίτηση για αποκριτικό σχεδιασμό (responsive design). Η επίλυση αυτού του προβλήματος χρειάστηκε αρκετές προσπάθειες και χρόνο. Το σημαντικότερο και δυσκολότερο πρόβλημα όμως, προέκυψε στο κομμάτι της μεταφόρτωσης των εικόνων. Ειδικά, επειδή οι εικόνες είναι δυαδικά δεδομένα (binary data), η διαχείριση τους καθίσταται αρκετά δύσκολη και διαφορετική, πόσο μάλλον όταν μια από τις απαιτήσεις του συστήματος ήταν να μην υπάρχει περιορισμός στην ποσότητα επιλογής τους. Η χρήση της βάσης δεδομένων MySQL, κατέστησε ακόμα ένα εμπόδιο στην επίλυση του προβλήματος, αφού η συγκεκριμένη βάση δεδομένων δεν δέχεται ως data type πίνακα (Array), και έτσι αφού δεν ήταν εφικτή η αλλαγή της βάσης δεδομένων στο εκάστοτε σημείο της υλοποίησης, ήταν επιτακτική η ανάγκη να βρεθεί μια εξαιρετικά δύσκολη λύση που θα έκανε το έργο να ολοκληρωθεί. Επιπλέον, λόγω των πολλών διαφορετικών επεκτάσεων αλλά και της υψηλής

ποιότητας ενός αρχείου εικόνας έπρεπε να οριστεί συγκεκριμένη έγκριση και όριο μεγέθους για την αποδοχή αυτών, όπως και για να δοθεί η κατάλληλη εμπειρία στο χρήστη τοποθετήθηκε τόσο η πολλαπλή προεπισκόπηση τους κατά το upload όσο και το δυναμικό αυτόματο και εναλλασσόμενο display στις σελίδες της εφαρμογής. Άρα, λοιπόν, γίνεται αντιληπτό πως για την αξιοποίηση, το συγχρονισμό και την εναρμόνιση των τεχνολογιών που χρησιμοποιήθηκαν για την πραγματοποίηση του συγκεκριμένου έργου της παρούσας διπλωματικής εργασίας, ανέκυψαν διάφορων μορφών προβλήματα τα οποία φυσικά και επιλύθηκαν.

5.2 Μετρικά κώδικα συστήματος

Στην ενότητα αυτή, γίνεται αναφορά των στοιχείων προγραμματισμού της διαδικτυακής εφαρμογής, δηλαδή τα αρχεία κώδικα που είναι απαραίτητα για τη λειτουργία ολόκληρου του συστήματος. Συγκεκριμένα, αναφέρονται τα αρχεία πηγαίου κώδικα, τόσο στο front-end όσο και στο back-end.

5.2.1 Front-end

Για την επιτυχή ολοκλήρωση και λειτουργία του front-end του παρόν έργου δημιουργήθηκαν συνολικά 87 αρχεία κώδικα. Γίνεται μια πιο αναλυτική αναφορά στον πίνακα 5.1 που ακολουθεί.

Γλώσσα	Αρχεία	Αριθμός Γραμμών
JavaScript - JSX	57	3700
CSS	24	700
Other	6	120

Πίνακας 5.1: Πληροφορίες αρχείων στο front-end

5.2.2 Back-end

Αντίστοιχα, για την υλοποίηση του back-end αυτού του έργου δημιουργήθηκαν 18 αρχεία κώδικα. Παρουσιάζονται αναλυτικά στον πίνακα 5.2 παρακάτω.

Γλώσσα	Αρχεία	Αριθμός Γραμμών
JavaScript	18	1400

Πίνακας 5.2: Πληροφορίες αρχείων στο back-end

5.3 Μοντέλο S.W.O.T

Το μοντέλο S.W.O.T (Strengths, Weaknesses, Opportunities, Threats) όπως αναφέρεται και στις έννοιες που το αντιπροσωπεύει, αποτελεί ένα στρατηγικού σχεδιασμού εργαλείο, με το οποίο είναι εφικτό να διαπιστώσουμε κάποιες πληροφορίες όπως είναι τα δυνατά σημεία, τις αδυναμίες, τις ευκαιρίες και τις απειλές αντίστοιχα, του συστήματος που υλοποιήθηκε.

Δυνατά σημεία - Strengths

- Προάγει την άμεση ανακάλυψη και ενημέρωση για εκδηλώσεις που διεξάγονται σε διάφορες τοποθεσίες.
- Πρωτότυπη ιδέα ως προς τις δυνατότητες που παρέχονται στο χρήστη του συστήματος.
- Προσβάσιμο και εύχρηστο για τις περισσότερες συσκευές που διαθέτουν μικρότερο μέγεθος οθόνης από αυτές των τυπικών Η/Υ.
- Είναι εντελώς δωρεάν σε όλες τις λειτουργίες του. Δεν απαιτεί κόστος για τη χρήση του.
- Δεν εμπεριέχει διαφημίσεις και pop ups που προκαλούν μια δυσάρεστη αίσθηση στην περιήγηση.

Αδυναμίες (Weaknesses)

- Είναι απαραίτητη η σύνδεση στο Διαδίκτυο.

Ευκαιρίες (Opportunities)

- Το σύστημα δημιουργεί ευκαιρίες ως προς την ενημέρωση και επικοινωνία των χρηστών.

Απειλές (Threats)

- Πιθανότητα απόκτησης μεγάλου όγκου πληροφοριών.

5.4 Μελλοντικές επεκτάσεις

Ένα από τα βασικότερα χαρακτηριστικά των συστημάτων είναι ότι επιδέχονται διαρκή βελτίωση, αφού λόγω των συνεχών απαιτήσεων που δημιουργούνται από τους χρήστες, οι προσθήκες σε ένα σύστημα μπορεί να είναι απεριόριστες. Έτσι, λοιπόν, και το συγκεκριμένο σύστημα μπορεί να εξελιχθεί και να επεκταθεί. Ενδεικτικές μελλοντικές επεκτάσεις που μπορούν να συμβάλλουν στην βελτίωση του συστήματος αναφέρονται παρακάτω.

- **Προσθήκη αξιολόγησης.** Η προσθήκη αξιολόγησης στη διαδικτυακή εφαρμογή θα μπορούσε να βελτιώσει σημαντικά την ποιότητα της εφαρμογής, αφού οι χρήστες θα είχαν τη δυνατότητα να βαθμολογήσουν ή να δηλώσουν πως τους αρέσει κάποια εκδήλωση.
- **Μεταφόρτωση βίντεο.** Αρκετά σημαντική βελτίωση στο εκάστοτε σύστημα, μπορεί να θεωρηθεί η προσθήκη επιλογής μεταφόρτωσης αρχείων βίντεο για την καλύτερη προώθηση και ενημέρωση του περιεχομένου μιας εκδήλωσης.
- **Δημόσιο προφίλ χρηστών και άμεση επικοινωνία μεταξύ αυτών (chat).** Μια σημαντική προσθήκη στο σύστημα, θα μπορούσε να ήταν η δυνατότητα να μπορεί ο κάθε χρήστης να βλέπει τις πληροφορίες ενός άλλου χρήστη καθώς και η επιλογή να μπορεί να επικοινωνήσει με αυτόν πιο άμεσα μέσω chat και όχι μόνο μέσω των comments σε κάποια εκδήλωση.
- **Ζωντανή μετάδοση (Live streaming).** Μια εξίσου σημαντική επέκταση στο σύστημα, θα ήταν να δίνεται η δυνατότητα στο χρήστη να μεταδίδει ζωντανά μέσω κάποια συσκευής συνδεδεμένης στο Διαδίκτυο, την υπάρχουσα κατάσταση μιας εκδήλωσης που πραγματοποιείται εκείνη τη στιγμή, ώστε να μπορεί κάποιος ενδιαφερόμενος να βλέπει κάποια στιγμιότυπα της.

5.5 Συμπεράσματα

Καταλήγοντας στο σημείο αυτό της διπλωματικής εργασίας, μπορεί να αντιληφθεί κανείς πως τα συμπεράσματα που προκύπτουν είναι ποικίλα και πολυδιάστατα. Το μόνο σίγουρο είναι πως εξάγονται αρκετά οφέλη από το συγκεκριμένο έργο. Συνοψίζοντας, το παρόν σύστημα είναι ικανό να ενισχύσει την ενημέρωση, ανακάλυψη και συλλογή εκδηλώσεων που πραγματοποιούνται σε διάφορες τοποθεσίες, διευκολύνοντας το κοινό να κερδίσει χρόνο, αφού όπως είναι γνωστό λόγω των ραγδαίων και διαρκών εξελίξεων περιορίζεται ολοένα και περισσότερο σημαντικά.

5.6 Σύνοψη κεφαλαίου

Το κεφάλαιο αυτό, αποτελεί το τελικό κεφάλαιο για την ολοκλήρωση της συγκεκριμένης διπλωματικής εργασίας. Αρχικά, έγινε μια ανακεφαλαίωση του περιεχομένου του παρόν έργου, και στη συνέχεια έγινε αναφορά και ανάλυση των σημαντικών προβλημάτων που δημιουργήθηκαν κατά την υλοποίηση της εφαρμογής, υπογραμμίστηκαν τα μετρικά κώδικα του συστήματος και παρουσιάστηκε το μοντέλο S.W.O.T. Τέλος, έγινε λόγος για τα συμπεράσματα που προέκυψαν.

Παραρτήματα

Παράρτημα Α΄

Εγκατάσταση και χρήση εφαρμογής

Στο συγκεκριμένο σημείο, δίνονται αναλυτικές οδηγίες που πρέπει να ακολουθήσει κανείς, ώστε να εγκαταστήσει την διαδικτυακή εφαρμογή σε ένα σύστημα (localhost) και να τη χρησιμοποιήσει.

Οδηγίες εγκατάστασης συστήματος σε τοπικό υπολογιστή (localhost)

Η εγκατάσταση του συστήματος απαιτεί την ακολουθία των παρακάτω βημάτων. Αρχικά, απαραίτητη προϋπόθεση είναι η εγκατάσταση του Node.js από την επίσημη ιστοσελίδα <https://nodejs.org/en/> η οποία προσφέρει δυνατότητες download για διάφορα λειτουργικά συστήματα. Το Node.js εγκαθιστά αυτόματα και το NPM (Node Package Manager) και αυτό μπορεί να επαληθευτεί ανοίγοντας ένα παράθυρο τερματικού και πληκτρολογώντας τις εντολές `"npm -v"` και `"node -v"`, όπου θα πρέπει να εμφανίζονται οι αντίστοιχες πρόσφατες εκδόσεις. Επίσης, απαραίτητη είναι και η εγκατάσταση της βάσης δεδομένων MySQL. Στη συνέχεια, πρέπει να γίνει το download του συγκεκριμένου project, δηλαδή το API της εφαρμογής καθώς και το front-end, από το github. Το repository του back-end/api, βρίσκεται στην ηλεκτρονική διεύθυνση <https://github.com/devarnavas/events-api.git> και του front-end στην <https://github.com/devarnavas/events.git>. Όταν ολοκληρωθούν τα downloads των αρχείων από τα repositories πρέπει να προσαρμοστούν οι αντίστοιχες ρυθμίσεις. Πρωτίστως, αυτό σημαίνει ότι πρέπει να εγκατασταθούν τα node modules και οι πρόσφατες εκδόσεις όλων των απαιτούμενων αρχείων τόσο για το back-end όσο και για το front-end, το οποίο επιτυγχάνεται κάνοντας πλοήγηση μέσω του τερματικού στα αντίστοιχα repositories και τρέχοντας την εντολή `"npm install"`. Επιπλέον, στο back-end στο αρχείο `"nodemon.json"` πρέπει να τοποθετηθούν οι πληροφορίες που

απαιτούνται, δηλαδή να συμπληρωθούν τα στοιχεία για τη σύνδεση με τη βάση δεδομένων και τα APIs. Στο front-end στο αρχείο ".env" είναι, επίσης, αναγκαίο να γίνουν οι αντίστοιχες τροποποιήσεις. Εφόσον γίνουν αυτές οι αλλαγές, πρέπει να τρέξει η εντολή "npm start" αρχικά για το back-end και έπειτα για το front-end. Αυτόματα, η διαδικτυακή εφαρμογή θα ανοίξει σε μια νέα καρτέλα στον προκαθορισμένο browser του συστήματος.

Βιβλιογραφία

- [1] Babel. <https://babeljs.io/docs/en/>. [Accessed February-2020].
- [2] Client side programming. https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview. [Accessed February-2020].
- [3] Clubber. <https://www.clubber.gr/>. [Accessed February-2020].
- [4] CORS policy. <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>. [Accessed February-2020].
- [5] ελculture. <https://www.elculture.gr/>. [Accessed February-2020].
- [6] Discover Greece. <https://www.discovergreece.com/>. [Accessed February-2020].
- [7] NPM Node Package Manager. <https://docs.npmjs.com/about-npm/>. [Accessed February-2020].
- [8] Premiere-events. <https://premiere-events.gr/>. [Accessed February-2020].
- [9] Server side programming. https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Introduction. [Accessed February-2020].
- [10] Visual Studio Code. <https://code.visualstudio.com/docs/editor/whyvscode>. [Accessed February-2020].
- [11] Web server. https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server. [Accessed February-2020].
- [12] Webpack. <https://www.freecodecamp.org/news/an-intro-to-webpack-what-it-is-and-how-to-use-it-8304ecdc3c6> [Accessed February-2020].
- [13] C. Aquino and T. Gandee. *Front-End Web Development: The Big Nerd Ranch Guide*. Big Nerd Ranch Guides. Pearson Education, 2016.
- [14] R. Archer. *Express.js: Web App Development With Node.js Framework*. CreateSpace Independent Publishing Platform, 2016.
- [15] G. Blokdyk. *Full Stack JavaScript A Complete Guide - 2020 Edition*. 5STARCOOKS, 2019.
- [16] G. Blokdyk. *Web Development Team A Complete Guide - 2020 Edition*. Emereo Pty Limited, 2019.

-
- [17] V. Bojinov, D. Herron, and D. Resende. *Node.js Complete Reference Guide: Discover a more sustainable way of writing software with high levels of reusability and collaboration using Node.js*. Packt Publishing, 2018.
- [18] D. Cederholm and J. Zeldman. *CSS3 for Web Designers*. Book Apart. A Book Apart, 2014.
- [19] Against The Clock. *Web Design 2020: The Professional Portfolio*. Ellenn Behoriam, 2019.
- [20] Douglas Crockford. *JavaScript: The Good Parts*. OReilly, 2008.
- [21] C.J. Date. *An Introduction to Database Systems*. Pearson Education, 2006.
- [22] K. Geers. *Strategic Cyber Security*. NATO Cooperative Cyber Defence Centre of Excellence, 2011.
- [23] W.B. Publishing. *Back End Engineer 2020 Weekly and Monthly Planner: 2020 Planner Monthly Weekly Inspirational Quotes To Do List to Jot Down Work Personal Office Stuffs Keep Tracking Things Motivations Notebook*. Independently Published, 2019.
- [24] B. Sanders. *Smashing HTML5*. Smashing magazine book series. Wiley, 2010.
- [25] W. Suh. *Web Engineering: Principles and Techniques*. IGI Global research collection. Idea Group Pub., 2005.
- [26] R. Wieruch. *The Road to Learn React: Your Journey to Master Plain Yet Pragmatic React. Js*. CreateSpace Independent Publishing Platform, 2018.