



Πανεπιστήμιο Δυτικής Αττικής,
Τμήμα Μηχανικών Πληροφορικής Και Υπολογιστών

Διπλωματική Εργασία
Ανάπτυξη τουριστικής εφαρμογής/ιστοσελίδας

Γιώργος Ι. Λυδιώτης
711161067

Εισηγητής: Δρ Αντώνιος Μπόγρης, Καθηγητής

Αθήνα, Ιούλιος 2024



University of West Attica,
Department of informatics and Computer Engineering

Diploma Thesis
Development of Tourist themed web application

Giorgos Lydiotis
711161067

Supervisor: Dr Antonios Bogris, Professor

Athens, July 2024



Πανεπιστήμιο Δυτικής Αττικής,
Τμήμα Μηχανικών Πληροφορικής Και Υπολογιστών

Διπλωματική Εργασία
Ανάπτυξη τουριστικής εφαρμογής/ιστοσελίδας

Εισηγητής:

Αντώνιος Μπόγρης, Καθηγητής

Εξεταστική Επιτροπή:

A/α	Όνομα Επώνυμο	Βαθμίδα/Ιδιότητα	Ψηφιακή Υπογραφή
1	Αντώνιος Μπόγρης	Καθηγητής	
2	Βασίλειος Μάμαλης	Καθηγητής	
3	Ιωάννα Καντζάβελου	Αναπληρώτρια Καθηγήτρια	

Ημερομηνία Εξέτασης: 24/07/2024

Δήλωση συγγραφέα Διπλωματικής Εργασίας

Ο κάτωθι υπογεγραμμένος Γεώργιος Λυδιώτης του Ιωάννη, με αριθμό μητρώου 711161067 φοιτητής του Τμήματος Μηχανικών Πληροφορικής και Υπολογιστών της Σχολής Μηχανικών του Πανεπιστημίου Δυτικής Αττικής, δηλώνω ότι:

«Βεβαιώνω ότι είμαι συγγραφέας αυτής της Διπλωματικής εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, οι όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε ακριβώς είτε παραφρασμένες, αναφέρονται στο σύνολό τους, με πλήρη αναφορά στους συγγραφείς, τον εκδοτικό οίκο ή το περιοδικό, συμπεριλαμβανομένων και των πηγών που ενδεχομένως χρησιμοποιήθηκαν από το διαδίκτυο. Ακόμη, βεβαιώνω ότι αυτή η εργασία έχει συγγραφεί από μένα αποκλειστικά και αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο δικής μου, όσο και του Ιδρύματος.

Παράβαση της ανωτέρω ακαδημαϊκής μου ευθύνης αποτελεί ουσιώδη λόγο για την ανάκληση του πτυχίου μου».

Ο Δηλών



Γεώργιος Λυδιώτης

Περίληψη

Το αντικείμενο της διπλωματικής είναι η δημιουργία μιας fullstack web εφαρμογής (ιστοσελίδας), χρησιμοποιώντας Angular για το front-end της ιστοσελίδας, Java Spring για το back-end και PostgreSQL για την βάση δεδομένων. Η θεματολογία είναι μια τουριστική εφαρμογή όπου ο χρήστης θα μπορεί να ανακαλύψει τουριστικές τοποθεσίες μέσω διαδραστικού χάρτη της Ελλάδος και να καταγράψει τα αξιοθέατα (πχ Μουσείο Ακρόπολης, Λευκός Πύργος) που έχει ή θα ήθελε επισκεφθεί, να τα αξιολογήσει, να αφήσει δημόσιο σχόλιο και να κάνει ιδιωτικές σημειώσεις, αλληλεπιδρώντας έτσι με αυτά. Ο χρήστης επίσης θα μπορεί να επιλέξει τι είδους αξιοθέατα θέλει να δει μέσω ειδικού φίλτρου, το οποίο θα ξεχωρίζει βάση τύπου, χρονικής περιόδου, υποκατηγορίας κ.α. Τέλος, στον χρήστη θα προτείνονται νέα αξιοθέατα βάση του ιστορικού του.

Λέξεις κλειδιά: Διαδραστικός χάρτης, fullstack, προτάσεις, αλληλεπίδραση με πληροφορία, φίλτρο

Abstract

The subject of the thesis is the creation of a fullstack web application (website), using Angular for the front-end of the website, Java Spring for the back-end and PostgreSQL for the database. The subject matter is a tourist application where the user can discover tourist sites through an interactive map of Greece and list the attractions (e.g. Acropolis Museum, White Tower) that he has or would like to visit, rate them, leave public comments and make private notes, thus interacting with them. The user will also be able to choose what kind of attractions he or she wants to see through a special filter, which will be sorted by type, time period, subcategory, etc. Finally, the user will be suggested new sites based on their history.

Keywords: Interactive map, fullstack, suggestions, information interaction, filter

Πίνακας περιεχομένων

Πίνακας περιεχομένων	9
Συνομογραφίες	12
Πίνακες	12
Εικόνες	13
Κεφάλαιο 1: Εισαγωγή	15
Κεφάλαιο 2: Ανασκόπηση τουριστικών εφαρμογών	17
2.1 Ανάλυση τουριστικών ιστοσελίδων	17
2.1.1 Ανάλυση visitgreece.com	17
2.1.2 Ανάλυση allovergreece.com	20
2.1.3 Ανάλυση tripadvisor.com	22
2.1.4 Ανάλυση τοπικών ιστοσελίδων & ιστοσελίδων του δημοσίου	25
2.2 Γενικές ιστοσελίδες	26
2.3 Περιγραφή προβλήματος και στόχοι	27
Κεφάλαιο 3: Μεθοδολογία συλλογής δεδομένων	29
3.1 Αξιοθέατα	29
3.1.1 Πρώτο Στάδιο: Εύρεση αξιοθέατου, τοποθεσίας και κύριας κατηγορίας	29
3.1.2 Δεύτερο Στάδιο: Εικόνα	30
3.1.3 Τρίτο Στάδιο: Περιγραφή	30
3.1.4 Τέταρτο Στάδιο: Δευτερεύων κατηγορίες και χρήσιμες ιστοσελίδες	31
3.2 Χάρτης της Ελλάδος	31
3.2.1 Πληροφορίες νομών της Ελλάδος	32
3.2.2 Ταμπέλες στον χάρτη	32
3.3 Κώδικας του scraping προγράμματος	34
Κεφάλαιο 4: Αρχιτεκτονική της Εφαρμογής	36
4.1 Κύριοι πίνακες	37
4.1.1 Place (Αξιοθέατο)	37
4.1.2 Username (Χρήστης)	37
4.1.3 Image (Εικόνα)	38
4.1.4 Comment (Σχόλιο/Σημείωση)	39
4.1.5 Link (Σύνδεσμος)	39
4.1.6 Prefecture (Νομός)	39
4.1.7 Subcategory (Υποκατηγορία)	40
4.2 Πίνακες ένα-προς-πολλά και πολλά-προς-πολλά	41
4.2.1 Prefecture_Location (Νομός κάθε τοποθεσίας)	41
4.2.2 Place_Subcategory (Υποκατηγορίες ανά αξιοθέατο)	41
4.2.3 Username_Place (Συσχέτιση αξιοθέατου με χρήστη)	42
4.2.4 Username_Comment (Συσχέτιση σχόλιου με χρήστη και αξιοθέατο)	42
Κεφάλαιο 5: Υλοποίηση Εφαρμογής	43
5.1 Διαδραστικός Χάρτης	47
5.2 Πάνελ με πληροφορίες	49
5.3 Δημιουργία λογαριασμού χρήστη (Sign up/Login/Logout)	51
5.4 Αλληλεπίδραση χρήστη με αξιοθέατα	52
5.4.1 Επίσκεψη, επιθυμία επίσκεψης, βαθμολόγηση, σήμανση ως αγαπημένο	52
5.4.2 Σχόλια χρήστη στα αξιοθέατα	53
5.5 Επιπλέον κουμπιά επάνω στον χάρτη	54
5.6 Λίστα όλων των αξιοθέατων	57
5.7 Πληροφορίες και στατιστικά χρήστη	57
5.8 Προτάσεις αξιοθέατων	59
5.9 Εναλλαγή φωτεινού θέματος και σκοτεινού θέματος	61
5.10 Αναζήτηση αξιοθέατου	64
Κεφάλαιο 6: Αποτέλεσμα και εγχειρίδιο χρήσης	65
6.1 Μπάρα μενού	65
6.2 Υποσελίδες	66

6.3 Πάνελ πληροφορίας	68
6.4 Σύνδεση και εγγραφή χρήστη	69
Κεφάλαιο 7: Συμπέρασμα και μελλοντικές επεκτάσεις	70
7.1 Περισσότερες γλώσσες διεπαφής	70
7.2 Περισσότερες πληροφορίες	71
7.3 Δυνατότητα συζήτησης και φιλτράρισμα σχολίων και χρηστών	71
7.4 Εφαρμογή κινητού και καλύτερη ανταποκρισιμότητα σε κάθετες οθόνες	71
Πηγές	73
Για την εκπόνηση του παρών εγγράφου	73
Για την εκπόνηση της εφαρμογής	73

Συνομογραφίες

ASCII: American Standard Code for Information Interchange
CLI: Command Line Interface
JSON: Javascript Simple Object Notation
QGIS: Quantum Geographic Information System
REST: Representational State Transfer
UI: User Interface
URL: Uniform Resource Locator
UUID: Universally Unique Identifier

Πίνακες

Πίνακας 3.1 : Κομμάτι κώδικα που δημιουργεί τον σύνδεσμο της τοποθεσίας του αξιοθέατου στο Google Maps.	31
Πίνακας 3.2 : Άρθρα που χρησιμοποιήθηκαν για τις περιγραφές των νομών	33
Πίνακας 3.3 : Ο κώδικας του αρχείου Main.java	34
Πίνακας 3.4 : Ο κώδικας του αρχείου StaticMethods.java	35
Πίνακας 4.1 : Το SQL Query που εκτελέστηκε για να δημιουργηθεί ο πίνακας Place_Subcategory	41
Πίνακας 5.1 : Περιγραφή των μεθόδων των Repository του backend ανά αρχείο	44
Πίνακας 5.2 : Περιγραφή των μεθόδων των Controller του backend ανά αρχείο	45
Πίνακας 5.3 : Περιγραφή component,model και service του frontend	46
Πίνακας 5.4 : Οι μέθοδοι initMap και addGeoJson, 2 από τις μεθόδους που χρησιμοποιούνται για την αρχικοποίηση του χάρτη.	48
Πίνακας 5.5 : Απόσπασμα κώδικα που καθορίζει το χρώμα φόντου των εικονιδίων που χρησιμοποιούνται ως ταμπέλες για τα αξιοθέατα. Αφορά μόνο τις κατηγορίες Μουσείο, Ορόσημο, Φυσικό Θέαμα και Μοναστήρι.	48
Πίνακας 5.6 : Παράδειγμα query που τρέχουν κατά την δημιουργία του πάνελ πληροφορίας	51
Πίνακας 5.7 : Μέθοδος getPlaceComments η οποία επιστρέφει όλα τα δημόσια σχόλια και όλα τα σχόλια του χρήστη και η μέθοδος getCommentsByPlaceId με το sql query.	53
Πίνακας 5.8 : Μέθοδος deleteComment για την διαγραφή σχολίου.	53
Πίνακας 5.9 : Μέθοδος addComment, για την δημιουργία του σχολίου. Πρέπει πρώτα να δημιουργηθεί το σχόλιο (μεταβλητή c) και έπειτα το κλειδί για την συσχέτιση του σχολίου με το αξιοθέατο και τον χρήστη (μεταβλητή uc).	54
Πίνακας 5.10 : Μέθοδος updateComment, δέχεται το ανανεωμένο σχόλιο, αλλάζει την ημερομηνία timeUpdated στην τωρινή και σώζει το σχόλιο στην βάση.	54
Πίνακας 5.11 : Η μέθοδος που καλείται κατά το πάτημα του κουμπιού 'Εμφάνιση ολόκληρου χάρτη'	54
Πίνακας 5.12 : Η CSS κλάση του κουμπιού 'Εμφάνιση ολόκληρου χάρτη'	55
Πίνακας 5.13 : Ανάκτηση όλων των αξιοθέατων τα οποία έχουν όλες τις υποκατηγορίες που έχει παραθέσει ο χρήστης.	56
Πίνακας 5.14 : Ανάκτηση περιληπτικής λίστας όλων των αξιοθέατων	57
Πίνακας 5.15 : Απόσπασμα κώδικα που εμφανίζει τις κύριες μεθόδους που χρησιμοποιούνται για την υλοποίηση των φίλτρων κύριων και δευτερεύων κατηγοριών. Εμφανίζονται οι μέθοδοι filterByType, filterCategory, filterBySubcategory & filterSubcategory	58
Πίνακας 5.16 : Απόσπασμα κώδικα που υπολογίζει τα στατιστικά βάση της λίστας που λαμβάνει από το backend	59
Πίνακας 5.17 : Το αρχείο .css του component user-places.	60
Πίνακας 5.18 : Τοποθεσία με τις περισσότερες επισκέψεις	60
Πίνακας 5.19 : Τύπος αξιοθέατου με τις περισσότερες επισκέψεις	60

Πίνακας 5.20 : Τοποθεσία με τα περισσότερα αγαπημένα αξιοθέατα	61
Πίνακας 5.21 : Τύπος αξιοθέατου με τα περισσότερα αγαπημένα αξιοθέατα	61
Πίνακας 5.22 : Πρόταση 10 κορυφαίων αξιοθέατων βάση του ιστορικού του χρήστη σε σχέση με τις βαθμολογίες των άλλων χρηστών	61
Πίνακας 5.23 : Πρόταση 10 τυχαίων τοποθεσιών τις οποίες δεν έχει επισκεφθεί ο χρήστης ..	61
Πίνακας 5.24 : Κώδικας του αρχείου theme-switcher.component.html	62
Πίνακας 5.25 : Η μέθοδος switchTheme() του αρχείου theme-switcher.component.ts	62
Πίνακας 5.26 : Κώδικας του αρχείου theme-switcher.component.css	63
Πίνακας 5.27 : Μέθοδος makeList που παράγει την λίστα αναζήτησης	64

Εικόνες

Εικόνα 2.1 : Δεδομένα επισκεψιμότητας από την ιστοσελίδα ahrefs.com, ανακτήθηκε Μάρτιο 2024	17
Εικόνα 2.2 : Βασική σελίδα visitgreece.com, ανακτήθηκε Μάρτιο 2024	18
Εικόνα 2.3 : Προορισμοί που εμφανίζονται μετά από πλοήγηση στην βασική σελίδα του visitgreece.com	19
Εικόνα 2.4 : Διαδραστικός χάρτης περιφερειών στην κύρια σελίδα του visitgreece.com	19
Εικόνα 2.5 : Διαδραστικός χάρτης διοικητικών περιφερειών της Ελλάδος του visitgreece.com	19
Εικόνα 2.6 : Διαδραστικός χάρτης γεωγραφικής διαίρεσης της Ελλάδος του visitgreece.com	20
Εικόνα 2.7 : Απόσπασμα της κύριας σελίδας της ιστοσελίδας allovergreece.com. Στην κορυφή με μαύρο φόντο οι κατηγορίες. Έπειτα από την επιλογή της κατηγορίας “Πολιτισμός” εμφανίζονται επιπλέον κατηγορίες με εικόνες από κάτω	20
Εικόνα 2.8 : Απόσπασμα της δομής της ιστοσελίδας με της κατηγορίες αξιοθέατων και προορισμών.	21
Εικόνα 2.9 : Περιγραφή και δείγμα αξιοθέατων από την κατηγορία “Αρχαιολογικοί χώροι” της ιστοσελίδας allovergreece.com, ανακτήθηκε Μάρτιο 2024	22
Εικόνα 2.10 : Ενσωματωμένος χάρτης google στην σελίδα του αξιοθέατου “Ακρόπολη” της ιστοσελίδας allovergreece.com	22
Εικόνα 2.11 : Στιγμιότυπο κατηγοριών των αξιοθέατων της Αθήνας στο tripadvisor.com	23
Εικόνα 2.12 : Στιγμιότυπο σελίδας του αξιοθέατου Ακρόπολη, τμήμα “Πληροφορίες”	23
Εικόνα 2.13 : Στιγμιότυπο σελίδας του αξιοθέατου Ακρόπολη, τμήμα “Η περιοχή”	23
Εικόνα 2.14 : Στιγμιότυπο σελίδας του αξιοθέατου Ακρόπολη, τμήμα “Κριτικές”	24
Εικόνα 2.15 : Χάρτης του tripadvisor.com και εικονίδια των ταμπελών του	24
Εικόνα 2.16 : Ο χάρτης της σελίδας odysseus.culture.gr, ανακτήθηκε Μάιο 2024	25
Εικόνα 2.17 : Πανόραμα αιθουσών του Εθνικού Ιστορικού Μουσείου	26
Εικόνα 2.18 : Το κουτί αλληλεπίδρασης χρήστη-ταινίας στην ιστοσελίδα letterboxd.com	27
Εικόνα 3.1 : Παράδειγμα μετατροπής πηγαίου URL σε URL που συμβατό με CORS (σε υπολογιστικό φύλλο)	30
Εικόνα 3.2 : Ο χάρτης με τα όρια των νομών της Ελλάδος, όπως φαίνεται στο QGIS	32
Εικόνα 3.3 : Ο χάρτης της Ελλάδος όπως φαίνεται με τις ταμπέλες στο QGIS	34
Εικόνα 4.1 : Το διάγραμμα της βάσης δεδομένων	37
Εικόνα 4.2 : Ο πίνακας Place αναλυτικά	37
Εικόνα 4.3 : Ο πίνακας Username αναλυτικά	38
Εικόνα 4.4 : Ο πίνακας Image αναλυτικά	38
Εικόνα 4.5 : Ο πίνακας Comment αναλυτικά	39
Εικόνα 4.6 : Ο πίνακας Link αναλυτικά	39
Εικόνα 4.7 : Ο πίνακας Prefecture αναλυτικά	40
Εικόνα 4.8 : Ο πίνακας Subcategory αναλυτικά	40
Εικόνα 4.9 : Ο πίνακας Prefecture_Location αναλυτικά	41
Εικόνα 4.10 : Ο πίνακας Place_Subcategory αναλυτικά	41
Εικόνα 4.11 : Ο πίνακας Username_Place αναλυτικά	42
Εικόνα 4.12 : Ο πίνακας Username_Comment αναλυτικά	42
Εικόνα 5.1 : Το πάνελ πληροφοριών αξιοθέατου	50
Εικόνα 5.2 : Παράδειγμα πάνελ πληροφοριών	51
Εικόνα 5.3 : Το κουμπί ‘Εμφάνιση ολόκληρου χάρτη’ με κόκκινη περιγράμμιση	55

Εικόνα 5.4 : Ο διακόπτης εναλλαγής θέματος, όταν είναι ανοιχτό το φωτεινό θέμα (πάνω) και όταν είναι ανοιχτό το σκοτεινό (κάτω).....	64
Εικόνα 6.1 : Κύρια Σελίδα.....	65
Εικόνα 6.2 : Επεξήγηση λειτουργιών μπάρας μενού.....	66
Εικόνα 6.3 : Η μπάρα μενού σε όλες της τις μορφές.....	66
Εικόνα 6.4 : Η λίστα των αξιοθέατων.....	67
Εικόνα 6.5 : Πληροφορίες Χρήστη.....	68
Εικόνα 6.6 : Το πάνελ πληροφορίας.....	69

Κεφάλαιο 1: Εισαγωγή

Ο σκοπός της διπλωματικής είναι η δημιουργία μιας fullstack web εφαρμογής (ιστοσελίδας). Η θεματολογία είναι τουριστική εφαρμογή όπου ο χρήστης θα μπορεί να ανακαλύψει τουριστικές τοποθεσίες μέσω διαδραστικού χάρτη της Ελλάδος και να καταγράψει τις τοποθεσίες (πχ Μουσείο Ακρόπολης, Λευκός Πύργος) που έχει ή θα ήθελε επισκεφθεί, να τις αξιολογήσει, να αφήσει δημόσιο σχόλιο και να κάνει ιδιωτικές σημειώσεις. Ο χρήστης επίσης θα μπορεί να επιλέξει τι είδους προορισμούς θέλει να δει μέσω ειδικού φίλτρου, το οποίο θα ξεχωρίζει βάση τύπου, χρονικής περιόδου, υποκατηγορίας και άλλων σχετικών λειτουργιών. Τέλος, στον χρήστη θα προτείνονται νέες τοποθεσίες βάση του ιστορικού του.

Μία τέτοια εφαρμογή είναι ιδιαίτερα σημαντική στην Ελλάδα, της οποίας η τουριστική βιομηχανία αποτελεί βασικό στοιχείο της οικονομικής δραστηριότητας της και δέχεται κάθε χρόνο δεκάδες εκατομμύρια τουρίστες από όλο το κόσμο. Έτσι, μία τέτοια εφαρμογή θα έχει ως σκοπό να διευκολύνει τους τουρίστες, εγχώριους αλλά και ξένους, να μάθουν σε έναν συγκεντρωμένο χώρο τα πολυπληθή τουριστικά αξιοθέατα που έχει να προσφέρει η χώρα μας, καθώς και μέσω της καταγραφής και της αξιολόγησης από άλλους χρήστες να μπορεί ο κάθε χρήστης να ξεχωρίσει ποια θα τους ενδιέφερε να επισκεφθούν.

Οι τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής διαχωρίζονται στα παρακάτω τρία μέρη: τεχνολογίες Frontend Development, τεχνολογίες Backend Development, τεχνολογίες βάσης δεδομένων & συλλογής δεδομένων.

Στο frontend κομμάτι χρησιμοποιήθηκε η Angular που είναι ένα δωρεάν και ανοιχτού κώδικα πλαίσιο εφαρμογών web μιας σελίδας(single page web framework) που βασίζεται στην TypeScript. Η Typescript είναι δωρεάν και ανοιχτού κώδικα γλώσσα προγραμματισμού υψηλού επιπέδου, η οποία προσθέτει στατικούς τύπους στην Javascript και αποτελεί υπερσύνολό της. Η Angular επιλέχθηκε αντί για άλλα αντίστοιχα πλαίσια εφαρμογών (React, Vue) κυρίως για την ενδημική της χρήση της Typescript, για την ισχυρή διεπαφή γραμμής εντολών (CLI) που κάνει την δημιουργία του προγράμματος ταχύτερη, και τέλος λόγω της πιο οργανωμένης σύστασης του προγράμματος με την χρήση των component. Για το κομμάτι του διαδραστικού χάρτη χρησιμοποιήθηκε η Javascript βιβλιοθήκη ανοιχτού κώδικα Leaflet.

Στο backend κομμάτι χρησιμοποιήθηκε το πλαίσιο εφαρμογών (framework) Spring της γλώσσας προγραμματισμού Java 17. Συγκεκριμένα χρησιμοποιήθηκαν οι παρακάτω εξαρτήσεις (dependencies): Spring Web, Lombok, JDBC API και PostgreSQL Driver. Ο σκοπός της χρήσης της Spring Web είναι η διευκόλυνση στην επικοινωνία μεταξύ της βάσης δεδομένων με το front end κομμάτι της εφαρμογής με την δημιουργία backend εφαρμογής που τηρεί τους αρχιτεκτονικούς περιορισμούς REST (RESTful Web Service).

Ως βάση δεδομένων χρησιμοποιήθηκε η σχεσιακή βάση PostgreSQL. Χρησιμοποιήθηκε σε συνδυασμό με το PostGis, το οποίο είναι δωρεάν και ανοιχτού κώδικα λογισμικό που επεκτείνει τις δυνατότητες της PostgreSQL προσθέτοντας υποστήριξη για αποθήκευση, ευρετηρίαση και αναζήτηση

γεωχωρικών δεδομένων. Για την συλλογή δεδομένων χρησιμοποιήθηκαν script σε αρχεία excel καθώς και πρόγραμμα στην Java για την απόξεση (scraping) της Βικιπαίδειας χρησιμοποιώντας την JWiki.¹ Τέλος για την δημιουργία του JSON αρχείου που διαβάζει στο frontend κομμάτι η Leaflet, χρησιμοποιήθηκε το πρόγραμμα QGIS, με μία εκ των χρήσεων του να είναι το ότι λαμβάνει δεδομένα από μία βάση δεδομένων - στην προκείμενη περίπτωση την PostgreSQL - και να τα μετατρέπει σε αρχείο GeoJSON το οποίο είναι μια μορφή JSON αρχείου που περιέχει γεωχωρικά δεδομένα.

¹ Βιβλιοθήκη για κλήση του API της Βικιπαίδειας [<https://github.com/fastily/jwiki>]

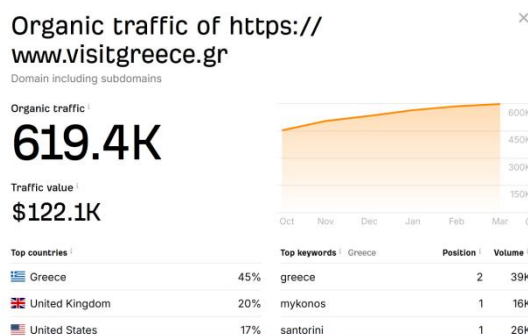
Κεφάλαιο 2: Ανασκόπηση τουριστικών εφαρμογών

Στο δεύτερο κεφάλαιο της παρούσας διπλωματικής εργασίας θα γίνει μία ανάλυση διαθέσιμων τουριστικών εφαρμογών και ιστοσελίδων της αγοράς και άλλων πηγών που θα μπορούσε ο μέσος άνθρωπος να χρησιμοποιήσει για να μάθει περισσότερες πληροφορίες για την Ελλάδα και τα αξιοθέατά της. Έτσι παρακάτω θα περιγραφούν μερικές αντίστοιχες εφαρμογές με τα θετικά τους αλλά και τα μειονεκτήματα ή ελλείψεις που εντοπίστηκαν. Από τουριστικές εφαρμογές και ιστοσελίδες ειδικές για την Ελλάδα, η διασημότερη είναι η [visitgreece.gr](https://www.visitgreece.gr). Το [allovergreece.com](https://www.allovergreece.com) είναι μια ιστοσελίδα η οποία έχει αξιοθέατα ανά κατηγορία και πολλές πληροφορίες που θα χρειαζόταν ένας τουρίστας για την διευκόλυνσή της διαβίωσης του στην Ελλάδα. Έπειτα υπάρχουν εφαρμογές παγκόσμιου βεληνεκούς όπως η [tripadvisor.com](https://www.tripadvisor.com) και γενικότερες ιστοσελίδες που δεν έχουν να κάνουν απαραίτητα με την τουριστική βιομηχανία αλλά είναι χρήσιμες στους τουρίστες όπως οι χάρτες της google και η βικιπαίδεια. Επίσης θα αναφερθούν και οι τοπικές ιστοσελίδες οι οποίες παρέχουν εκτενές πληροφορίες για κάθε νομό ή νησί, πχ [culture.lamia.gr](https://www.culture.lamia.gr), www.travelioannina.com, [confugreece.gr](https://www.confugreece.gr) και πολλές άλλες. Τέλος θα αναλυθεί μία εφαρμογή η οποία δεν έχει να κάνει με τον τουρισμό, η [letterboxd.com](https://www.letterboxd.com), στην οποία οι χρήστες μπορούν να αλληλεπιδράσουν με τα εγγραφές της βάσης δεδομένων της (ταινίες, ντοκιμαντέρ) καταγράφοντας αν τις έχουν δει ή θα ήθελαν μελλοντικά και αφήνοντας σχολιασμό και κριτική.

2.1 Ανάλυση τουριστικών ιστοσελίδων

2.1.1 Ανάλυση [visitgreece.com](https://www.visitgreece.com)

Η ιστοσελίδα [visitgreece.com](https://www.visitgreece.com) είναι η πιο διάσημη τουριστική ιστοσελίδα με αυστηρά ελληνικό περιεχόμενο, έχοντας 600 χιλιάδες επισκέψεις τον Μάρτη του 2024. Όπως φαίνεται στην **Εικόνα 2.1**, η ιστοσελίδα λαμβάνει πολλές επισκέψεις εντός των Ελληνικών συνόρων, αλλά η πλειοψηφία είναι από το εξωτερικό. Αυτό καθίσταται δυνατό καθώς η σελίδα έχει ελληνική και παράλληλα αγγλική μετάφραση. Καθώς είναι μια ιδιαίτερα μεγάλη ιστοσελίδα με δεκάδες υποδιευθύνσεις, θα δοθεί περισσότερη σημασία στην πλοήγηση της, στον διαδραστικό χάρτη της και στην αλληλεπίδραση των χρηστών με τα αξιοθέατα και τους προορισμούς της.

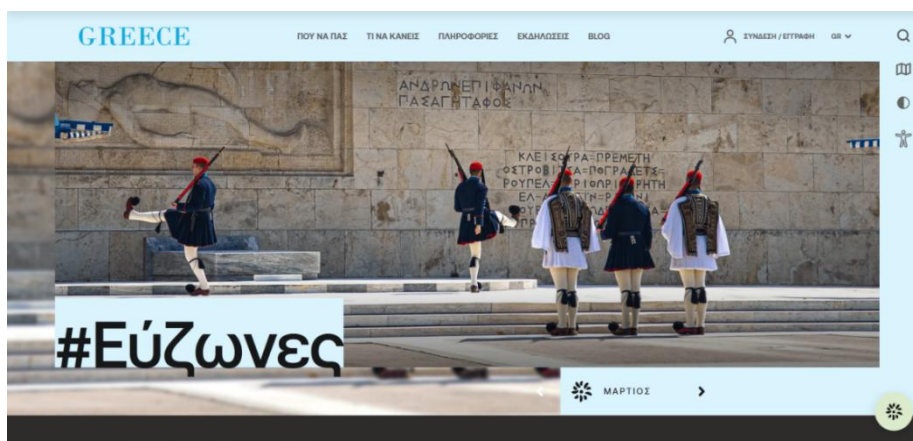


Εικόνα 2.1: Δεδομένα επισκεψιμότητας από την ιστοσελίδα [ahrefs.com](https://www.ahrefs.com), ανακτήθηκε Μάρτιο 2024

Κατά την πρώτη επίσκεψη στην σελίδα ο χρήστης καλωσορίζεται με μία μεγάλη εικόνα πολιτισμικού περιεχομένου, η οποία αλλάζει ανά τους μήνες (**Εικόνα 2.2**). Στην συνέχεια με την πλοήγηση της κύριας σελίδας ο χρήστης βλέπει απευθείας μερικά από τα αξιοθέατα που έχει να προσφέρει η Ελλάδα (**Εικόνα 2.3**) καθώς και προτάσεις δραστηριοτήτων, εκδηλώσεις, άρθρα που περιέχουν συνταγές και συνεντεύξεις, ενδιαφέροντα γεγονότα και πληροφορίες και τέλος έναν διαδραστικό χάρτη των περιφερειών της Ελλάδας με μέρη που μπορεί κανείς να επισκεφθεί σε αυτούς (**Εικόνα 2.4**) με έναν μεγαλύτερο χάρτη να βρίσκεται στην υποδιεύθυνση visitgreece.gr/el/map/ (**Εικόνες 2.5 & 2.6**). Συγκεκριμένα ο χάρτης της **Εικόνας 2.5** κάνει ανακατεύθυνση στην εκάστοτε τουριστική ιστοσελίδα της διοικητικής περιφέρειας που απεικονίζεται στον χάρτη. Ο χάρτης της **Εικόνας 2.6** από την άλλη αναδίδει ένα μικρό παράθυρο εντός σελίδας που παροτρύνει τον χρήστη να εξερευνήσει προορισμούς ή να δει τα τοπικά φαγητά και συνταγές τους. Πατώντας την εξερεύνηση προορισμών εμφανίζεται ένας τοπικός χάρτης με τις τοποθεσίες που μπορεί να επισκεφθεί κανείς εκεί, πχ στην Αττική εμφανίζει την Αθήνα, τον Πειραιά και την Ελευσίνα. Πατώντας σε οποιαδήποτε περιοχή εμφανίζεται μια περιγραφή της περιοχής και στο τέλος της περιγραφής υπάρχει το κουμπί “Ανακαλύψτε” το οποίο τελικά θα μας οδηγήσει σε μια ακόμη νέα υποδιεύθυνση, αυτή τη φορά της τοποθεσίας και ότι εκείνη έχει να προσφέρει.

Η παραπάνω πλοήγηση της ιστοσελίδας είχε πολλά αλλά απλά βήματα, με την πληροφορία των αξιοθέατων όμως να κρύβεται πίσω από πολυάριθμες κινήσεις. Φυσικά για ένα site της κλίμακας του visitgreece.com, το οποίο δεν έχει ως πρωτεύων σκοπό μονάχα να αναδείξει τα αξιοθέατα της χώρας αλλά την κάθε λογής πολιτισμική κληρονομιά της, είναι κατανοητή η εκτενής διαμερισματοποίηση της πληροφορίας της.

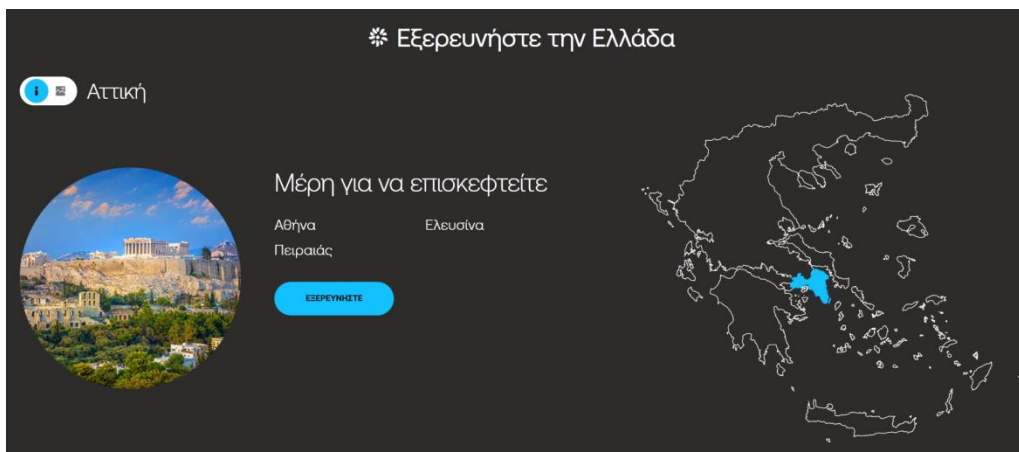
Όσον αφορά την αλληλεπίδραση του χρήστη με αυτήν την πληροφορία, η ιστοσελίδα δεν παρέχει κάτι τέτοιο για συγκεκριμένες τοποθεσίες. Αυτό που μπορεί να κάνει ένας χρήστης είναι να καταγράψει ένα πλάνο ταξιδιού, στο οποίο μπορεί να σημειώσει τους προορισμούς που θα ήθελε να επισκεφθεί στο ταξίδι του.



Εικόνα 2.2: Βασική σελίδα visitgreece.com, ανακτήθηκε Μάρτιο 2024



Εικόνα 2.3: Προορισμοί που εμφανίζονται μετά από πλοήγηση στην βασική σελίδα του visitgreece.com



Εικόνα 2.4: Διαδραστικός χάρτης περιφερειών στην κύρια σελίδα του visitgreece.com



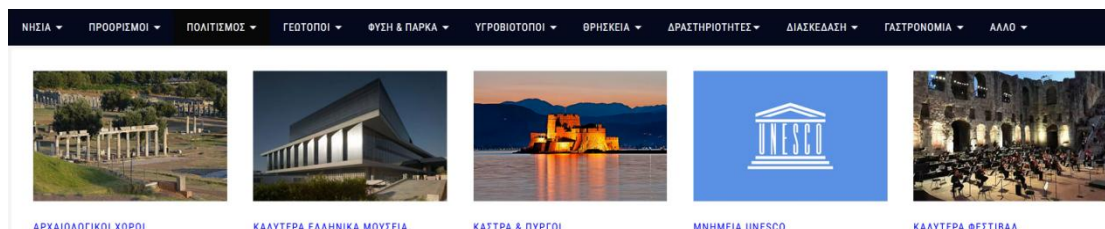
Εικόνα 2.5: Διαδραστικός χάρτης διοικητικών περιφερειών της Ελλάδος του visitgreece.com



Εικόνα 2.6: Διαδραστικός χάρτης γεωγραφικής διαίρεσης της Ελλάδος του visitgreece.com

2.1.2 Ανάλυση allogreece.com

Το allogreece.com είναι μια τουριστική ιστοσελίδα με επίκεντρο τους προορισμούς τους οποίους μπορεί να επισκεφθεί κανείς στην Ελλάδα, αλλά και πληροφορίες για δραστηριότητες και έθιμα. Ο λιτός σχεδιασμός της ιστοσελίδας κάνει την πλοήγηση της γρήγορη και η παρουσίαση της πληροφορίας εύκολα κατανοητή. Στην κύρια σελίδα υπάρχει η μπάρα της **Εικόνας 2.7**, η οποία δείχνει τις διάφορες κατηγορίες των αξιοθέατων. Σε αυτήν την εφαρμογή θα επικεντρωθεί ο τρόπος κατηγοριοποίησης των αξιοθέατων/προορισμών και η παρουσίαση αυτών.



Εικόνα 2.7: Απόσπασμα της κύριας σελίδας της ιστοσελίδας allogreece.com. Στην κορυφή με μαύρο φόντο οι κατηγορίες. Έπειτα από την επιλογή της κατηγορίας “Πολιτισμός” εμφανίζονται επιπλέον κατηγορίες με εικόνες από κάτω

Στην υποδιεύθυνση allogreece.com/Sitemap, βρίσκεται η δομή της ιστοσελίδας (**Εικόνα 2.8**) και μέσω αυτής μπορούμε να δούμε πιο αναλυτικά τον τρόπο που έχουν κατηγοριοποιηθεί τα δεδομένα της ιστοσελίδας. Για παράδειγμα, στην ενότητα “Πολιτισμός” υπάρχουν οι κατηγορίες (ετικέτες όπως αναφέρονται στην ιστοσελίδα): “Αρχαιολογικοί χώροι”, “Μουσεία”, “Κάστρα & Πύργοι” και “Καλύτερα Φεστιβάλ”. Αν επιλέξουμε οποιαδήποτε ετικέτα θα φορτώσει νέα σελίδα με μια λεπτομερείς περιγραφή αυτής της ετικέτας και από κάτω μία λίστα με τα αξιοθέατα με αυτήν την ετικέτα. Η **Εικόνα 2.9** δείχνει τι θα γινόταν αν επιλέγαμε την ετικέτα “Αρχαιολογικοί χώροι”. Η παρουσίαση των αξιοθέατων είναι λιτή και με μια ματιά παρέχει το όνομα του αξιοθέατου, μία εικόνα του, η κατηγορία του (διαφορετική από την ετικέτα), η τοποθεσία του, και ο αιώνας που δημιουργήθηκε, αν το αξιοθέατο

είναι φτιαγμένο από ανθρώπους. Επίσης, στην εικόνα 2.9 μπορούμε να παρατηρήσουμε ότι στην Ακρόπολη τονίζεται ότι είναι και μνημείο της UNESCO². Επιλέγοντας οποιοδήποτε αξιοθέατο φορτώνεται εκ νέου ιστοσελίδα που περιέχει αναλυτικές πληροφορίες για αυτό, όπως ιστορική περιγραφή, στοιχεία επικοινωνίας, διεύθυνση οδού, συντεταγμένες και έναν μικρό ενσωματωμένο χάρτη με την περιοχή γύρω από το αξιοθέατο (**Εικόνα 2.10**).

Όπως αναφέρθηκε και παραπάνω κάθε αξιοθέατο βρίσκεται σε μία ετικέτα και έχει μία κατηγορία, όπως φαίνεται και στην **Εικόνα 2.9**: Η “Ακρόπολη Αθηνών” βρίσκεται κάτω από την ετικέτα “Αρχαιολογικοί Χώροι” και έχει την κατηγορία “Ακροπόλεις”. Ενώ μπορούμε όμως να δούμε όλα τα αξιοθέατα που έχουν την ετικέτα “Αρχαιολογικοί Χώροι”, η κατηγορία “Ακρόπολη” και κάθε κατηγορία δεν φαίνεται να έχουν την αντίστοιχη αντιμετώπιση και δεν μπορούμε να δούμε πόσα αξιοθέατα έχουν αυτή την κατηγορία.

Όσον αφορά τον ενσωματωμένο χάρτη, είναι μια ωραία ιδέα καθώς όπως αναφέρθηκε και στην εισαγωγή του κεφαλαίου αυτού τα google maps είναι από τα πιο βασικά εργαλεία που χρησιμοποιεί ένας τουρίστας όταν επισκέπτεται μία ξένη χώρα. Μία επέκταση αυτού είναι ο σκοπός της παρούσας διπλωματικής, με την ενσωμάτωση ενός διαδραστικού χάρτη με σημάδια για κάθε αξιοθέατο αντί για έναν ενσωματωμένο χάρτη google της τριγύρω περιοχής.

Κατηγορίες

- | | | | | | |
|---------------------------|----------------------------|-------------------------------|-------------------------|-----------------------|------------------|
| > Νησιά | > Προορισμοί | > Πολιτισμός | > Γεύτοποι | > Φύση & Πάρκα | > Υδροβιότοποι |
| > Κυκλάδες | > Γραφικά Όρεινα Χωριά | > Αρχαιολογικοί Χώροι | > Σπήλαια | > Εθνικοί Δρυμοί | > Λίμνες |
| > Σποράδες | > Πόλεις | > Μουσεία | > Ηφαίστεια | > Εθνικά Πάρκα | > Λιμνοθάλασσες |
| > Ανατ. Αιγαίου | > Γραφικά Χωριά | > Κάστρα & Πύργοι | > Ιαματικές Πηγές | > Φυσικά Μνημεία | > Αλυκές |
| > Αργοσαρωνικός | > Εγκαταλειμμένοι Οικισμοί | > Καλύτερα Φεστιβάλ | > Φαράγγια | > Πάρκα | > Καταρράκτες |
| > Δωδεκάνησα | > Νηοί | | > Γεωπάρκα | > Δάση | |
| > Ιόνια | | | | | |
| > Κρήτη | | | | | |
| > Άλλα Νησιά | | | | | |
| > Νησιίδες | | | | | |
| > Θρησκεία | > Δραστηριότητες | > Διασκέδαση | > Τρόφιμα & Ποτά | > Περισσότερα | > Τέχνες |
| > Θρησκευτικοί Προορισμοί | > Αθλητικές Διοργανώσεις | > Καζίνο | > Εθνικά Προϊόντα | > Πέτρινα Γεφύρια | > Κινηματογράφος |
| > Μετέωρα | > ΓΚΟΛΦ | > Ζωολογικοί Κήποι & Ενυδρεία | > Φεστιβάλ Γαστρονομίας | > Γέφυρες στην Ελλάδα | |
| > Ιερές Μονές Αγίου Όρους | > Σημεία kitesurf | > Θεματικά Τραίνα | | > Φάρoi της Ελλάδας | |
| > Εντυπωσιακές Εκκλησίες | > Rafting | | | > Ναυάγια | |
| > Γάμος | > Μονοπάτια Πεζοπορίας | | | > Ανεμόμυλοι | |
| | > Χιονοδρομικά Κέντρα | | | > Ιαματικές Πηγές | |
| | | | | > Ξθιμα | |
| > Παραλίες | | | | | |
| > Κορυφαίες Παραλίες | | | | | |
| > Παραλίες Γυμναστών | | | | | |
| > Ράμπτα ΑμΕΑ | | | | | |

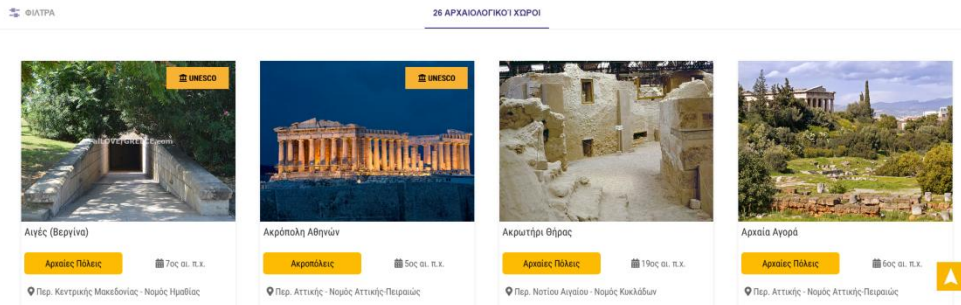
Εικόνα 2.8: Απόσπασμα της δομής της ιστοσελίδας με της κατηγορίες αξιοθέατων και προορισμών.

² Διεθνής οργανισμός των Ηνωμένων Εθνών ο οποίος απονέμει τον τίτλο Μνημείο Παγκόσμιας Κληρονομιάς σε μνημεία, κτίσματα ή τοποθεσίες, διακρίνοντας την ιστορική, πολιτιστική, καλλιτεχνική ή περιβαλλοντική σημασία τους παγκοσμίως.

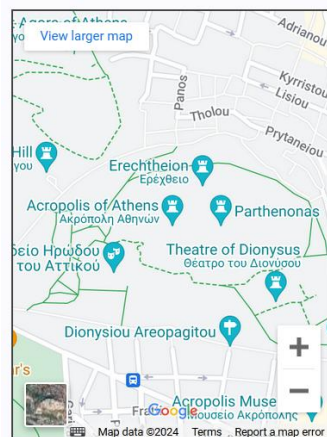
Αρχαιολογικοί Χώροι

...

Μια χώρα, ένας απέραντος αρχαιολογικός χώρος. Με 5.000 χρόνια ιστορίας κάθε μέρος είναι ένα ταξίδι στο παρελθόν και την ιστορία. Αρχαιολογικοί χώροι όπου άνθισε ο δικαίος πολιτισμός αποκόλλει κομμάτια της ανθρωπότητας. Αρχαίοι ναοί και θέατρα, στήλες, βουκαυτήρια και θαυμάσιες σφραγίδες ταξιδεύουν τον επισκέπτη σε μια άλλη εποχή. Χώροι με ανεκμνηστικά αξία, ημερία της παγκόσμιας κληρονομιάς που ανεβάζουν τον μέγεθος του ελληνικού πολιτισμού και πνεύματος. Σε κάθε γωνιά της χώρας υπάρχει ένας αρχαιολογικός χώρος που προκαλεί δέος και αναρωτιέται. Αναμφίβολα, η Ελλάδα είναι η χώρα με τους περισσότερους αρχαιολογικούς χώρους παγκοσμίως. Εκκαθαρισμένα επισκέπτες από όλα τα μέρη και πλάτη της γηραιάς ημερίας κάθε χρόνο στη πατρίδα μας για να θαυμάσουν τους αρχαιολογικούς θαυμάσιους και να μνησθούν στην ελληνική ιστορία και πολιτισμό. Παλλοί από τους αρχαιολογικούς χώρους συμπεριλαμβάνονται στον κατάλογο ημερησίων παγκόσμιας πολιτιστικής κληρονομιάς της UNESCO. Ας κάνουμε λοιπόν μία περιήγηση στους σημαντικότερους αρχαιολογικούς χώρους της Ελλάδας για να γνωρίσουμε την πλούσια ιστορία τους.



Εικόνα 2.9: Περιγραφή και δείγμα αξιοθεάτων από την κατηγορία “Αρχαιολογικοί χώροι” της ιστοσελίδας allovergreece.com, ανακτήθηκε Μάρτιο 2024



GPS Συντεταγμένες 37.97153, 23.72574

Εικόνα 2.10: Ενσωματωμένος χάρτης google στην σελίδα του αξιοθέατου “Ακρόπολη” της ιστοσελίδας allovergreece.com

2.1.3 Ανάλυση tripadvisor.com

Το tripadvisor.com είναι μια τουριστική ιστοσελίδα κριτικής και βαθμολογίας αξιοθεάτων, δραστηριοτήτων, ξενοδοχείων και εστιατορίων για πολλές χώρες. Η παρούσα ανάλυση θα έχει ως επίκεντρο τα αξιοθέατα της Ελλάδος και πως αναπαριστώνται πληροφορίες για αυτά. Αρχικά, η ιεραρχία των αξιοθεάτων έχει ως εξής: Ήπειρος - Χώρα - Νομός/Περιφέρεια - Πόλη - Αξιοθέατο. Όπως φαίνεται στην **Εικόνα 2.11** το κάθε αξιοθέατο μπορεί να φιλτραριστεί και βάση κατηγορίας, και μπορεί να έχει παραπάνω από μία κατηγορία. Η σελίδα του κάθε αξιοθέατου χωρίζεται σε τρία τμήματα: *Πληροφορίες*, *Η περιοχή*, *Κριτικές* όπως φαίνονται στις **Εικόνες 2.12, 2.13, 2.14** αντίστοιχα. Στο τμήμα “Πληροφορίες” υπάρχουν πληροφορίες για το πόσες κριτικές και τον μέσο όρο του, για την δημοφιλία του σε σχέση με άλλες δραστηριότητες στην πόλη του, τις κατηγορίες του, τις ώρες λειτουργίας του και φωτογραφίες για αυτό, αναρτημένες από τους χρήστες της ιστοσελίδας. Τέλος, πατώντας το κουμπί καρδιά πάνω αριστερά, ο χρήστης μπορεί να

χαρακτηρίσει το αξιοθέατο ως αγαπημένο. Στο τμήμα “Η περιοχή” εμφανίζονται οδηγίες για το πως θα φτάσει κανείς εκεί, χρήσιμοι σύνδεσμοι, μερικά εστιατόρια και αξιοθέατα κοντά, και ένα στιγμιότυπο χάρτη της τριγύρω περιοχής. Τέλος, στο τμήμα “Κριτικές” εμφανίζονται στατιστικά των κριτικών, έναν πλαίσιο αναζήτησής τους και φυσικά οι ίδιες οι κριτικές.

Εξερευνήστε δημοφιλείς εμπειρίες

Δείτε τι αρέσει σε άλλους ταξιδιώτες να κάνουν, με βάση τις αξιολογήσεις και τον αριθμό των κρατήσεων.

Προβολή όλων

- Αρχαία ερείπια (51)
- Ιστορικές τοποθεσίες (49)
- Κήποι (4)
- Θεματικά πάρκα (3)
- Υπαίθριες και λαϊκές αγορές (6)
- Μουσεία ιστορίας (22)
- Γειτονίες (9)
- Παραλίες (7)
- Μνημεία και σημεία ενδιαφέροντος (100)
- Μουσεία τέχνης (23)
- Ιστορικές περιοχές για πεζοπορία (28)
- Εξειδικευμένα μουσεία (40)
- Θέατρα (36)
- Μνημεία και αγάλματα (67)

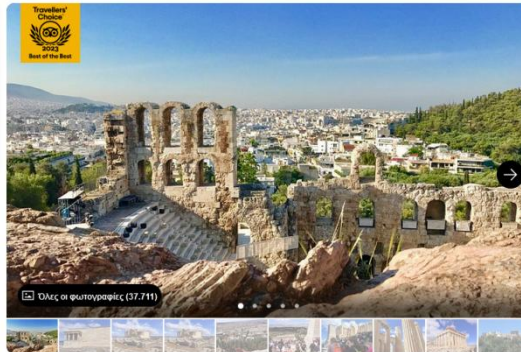
Εικόνα 2.11: Στιγμιότυπο κατηγοριών των αξιοθέατων της Αθήνας στο tripadvisor.com

Ακρόπολη

37.107 κριτικές • No 3 από 586 δραστηριότητες σε Αθήνα • Ιστορικές τοποθεσίες • Αρχαία Ερείπια
Ανοιχτό τώρα • 8:00 πμ - 8:00 μμ [Γράψτε μια κριτική](#)

Πληροφορίες

- Διάρκεια: < 1 ώρα
- Προτείνετε αλλαγές για να βελτιώσουμε τι δείχνουμε.
[Βελτίωση αυτής της καταχώρισης](#)



Τα έσοδα επηρεάζουν τις εμπειρίες που εμφανίζονται στην παρούσα σελίδα. [Μάθετε περισσότερα.](#)

Εικόνα 2.12: Στιγμιότυπο σελίδας του αξιοθέατου Ακρόπολη, τμήμα “Πληροφορίες”

Η περιοχή

Via Dionysiou Δεσφραφίου Str., Αθήνα 105 58 Ελλάδα

Πώς θα φτάσετε εκεί
Ακρόπολη • 2 λεπτά περπάτημα
Μοναστηράκι • 7 λεπτά περπάτημα

Απευθυνθείτε απευθείας

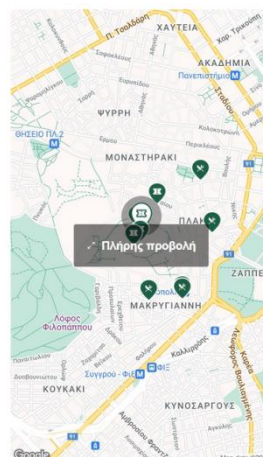
[Επισκεφθείτε τον ιστότοπο](#) * [Κλήση](#)

Τα καλύτερα εκεί κοντά

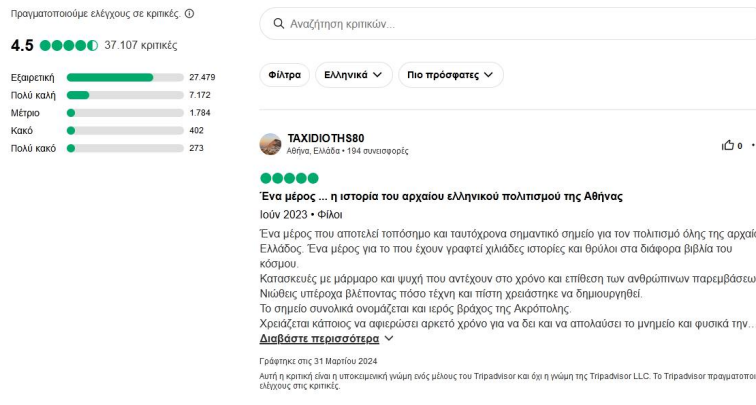
Εστιατόρια
3.612 εντός 5 χλμ.

Αξιοθέατα
606 εντός 10 χλμ.

- | | |
|--|--|
| Elaea Dine & Wine
4.445
0,4 χλμ • €€ - €€€ • Θαλασσινά • Μεσογειακή... | Παρθενώνας
17.561
39 μ • Ιστορικές τοποθεσίες • Αρχαία Ερείπια... |
| Liondi Traditional Greek...
5.223
0,5 χλμ • €€ - €€€ • Μεσογειακή • Ελληνική... | Ερέχθειον
2.855
56 μ • Αρχαία Ερείπια • Ορόσημο και Σημ... |
| Συμπόσιο Resto Bar
171
0,5 χλμ • €€ - €€€ • Μεσογειακή • Ελληνική... | Ναός της Αθηνάς Νίκης
1.282
77 μ • Αρχαία Ερείπια • Μνημεία και Αγάλ... |
- [Προβολή όλων](#) [Προβολή όλων](#)

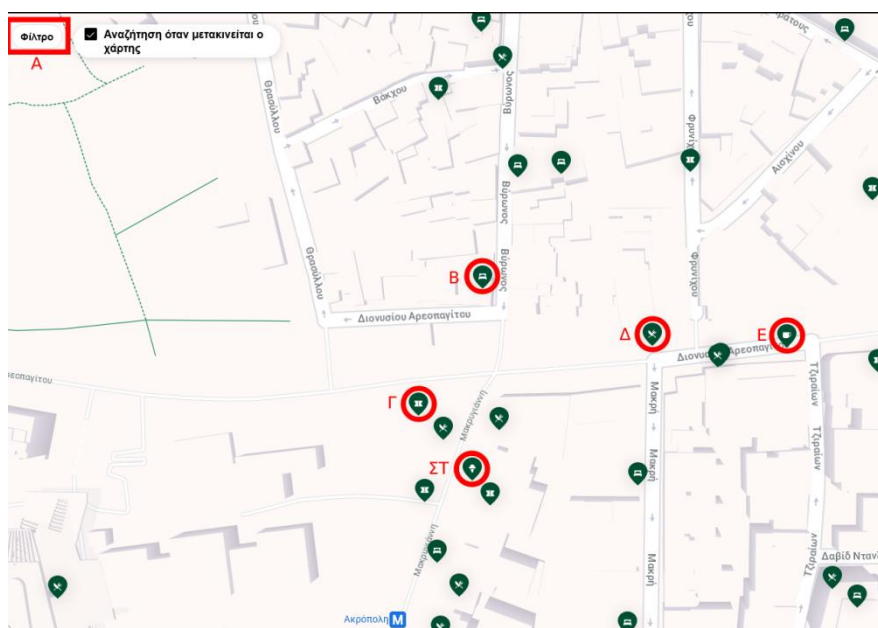


Εικόνα 2.13: Στιγμιότυπο σελίδας του αξιοθέατου Ακρόπολη, τμήμα “Η περιοχή”



Εικόνα 2.14: Στιγμιότυπο σελίδας του αξιοθέατου Ακρόπολη, τμήμα “Κριτικές”

Η ιστοσελίδα παρέχει συνοπτικές αλλά εύστοχες πληροφορίες για τα αξιοθέατα, όμως η δύναμή της βρίσκεται στην αλληλεπίδραση των χρηστών με αυτά, οι οποίοι μπορούν να παρέχουν εκτεταμένη κριτική για αυτό καθώς και αξιολογήσουν και τις κριτικές των υπόλοιπων χρηστών ως χρήσιμες, και τέλος να επισημάνουν ως αγαπημένο το αξιοθέατο και να εμφανίζεται ως πιο δημοφιλές. Εκεί που υστερεί αρκετά είναι στις ιστορικές πληροφορίες του αξιοθέατου, οι οποίες δεν υπάρχουν στην ίδια σελίδα, η οποία παρέχει μόνο σύνδεσμο για τις πληροφορίες στο τμήμα “Η περιοχή”. Όσον αφορά τον χάρτη, τον οποίο μπορούμε να επισκεφθούμε πατώντας “Πλήρης προβολή”, οι ταμπέλες από είναι χωρισμένες σε τρία είδη, όπως φαίνεται στο φίλτρο: Ξενοδοχεία, Δραστηριότητες και Εστιατόρια. Οι δραστηριότητες, ανεξαρτήτως είδους/κατηγορίας τους (Μουσεία, Αγάλματα, Πλατείες, Εκκλησίες κλπ) έχουν το ίδιο εικονίδιο. Παρομοίως τα ξενοδοχεία έχουν επίσης ένα εικονίδιο. Τα εστιατόρια φαίνεται να έχουν πιο εξειδικευμένα εικονίδια, με διαφορετικά για τα εστιατόρια, καφετέριες, παγωτατζίδικα κλπ. Στην **Εικόνα 2.15** εμφανίζεται στιγμιότυπο του μεγεθυμένου χάρτη, με τα τονισμένα με κόκκινο κύκλο να είναι Α) το κουμπί φίλτρου και τα εικονίδια Β) ξενοδοχείου Γ) δραστηριότητας Δ) εστιατορίου Ε) καφετέριας ΣΤ) παγωτατζίδικου.

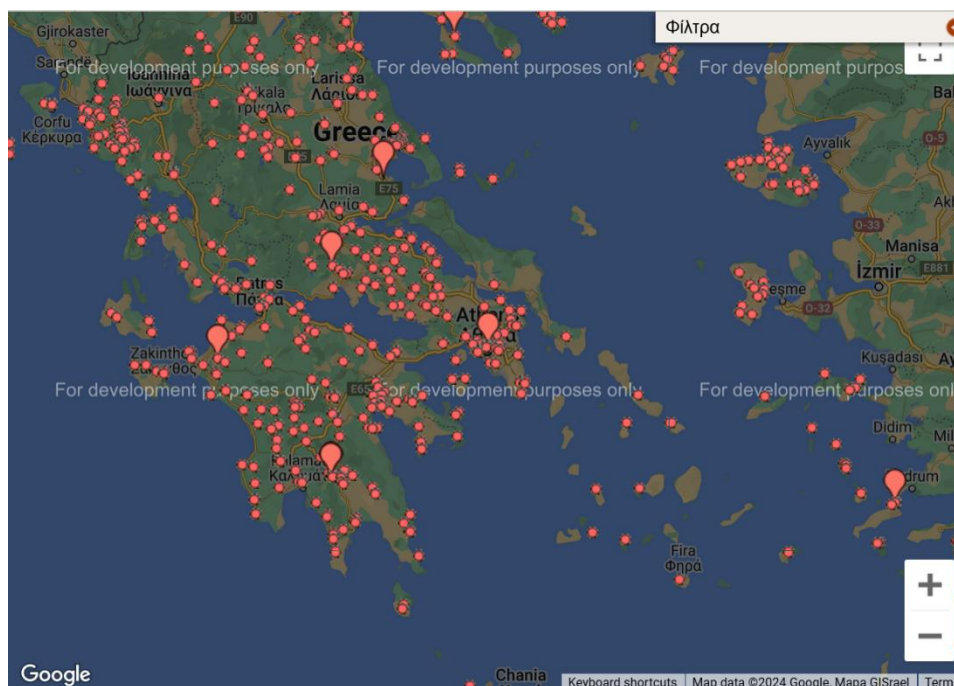


Εικόνα 2.15: Χάρτης του tripadvisor.com και εικονίδια των ταμπελών του

2.1.4 Ανάλυση τοπικών ιστοσελίδων & ιστοσελίδων του δημοσίου

Σχεδόν κάθε περιφέρεια, δήμος, ακόμα και πόλη στην Ελλάδα έχουν μια τοπική ιστοσελίδα με πληροφορίες για δραστηριότητες και αξιοθέατα σε αυτά. Επίσης, αρκετά μουσεία έχουν και εκείνα ιστοσελίδα, με άφθονες πληροφορίες για τα κειμήλια, πίνακες και αγάλματα που στεγάζονται σε αυτό. Σε αυτό το κεφάλαιο θα αναλυθούν οι παρακάτω τρεις ιστοσελίδες: odysseus.culture.gr που είναι η ιστοσελίδα του υπουργείου πολιτισμού και αθλητισμού της Ελλάδος, culture.lamia.gr/el που είναι η ιστοσελίδα του Δήμου Λαμίας, και nhmuseum.gr που είναι η ιστοσελίδα του εθνικού ιστορικού μουσείου.

Η odysseus.culture.gr κατέχει την πιο εκτενή βιβλιοθήκη με πληροφορίες για τα διάφορα αξιοθέατα όλης της Ελλάδος καθώς επίσης και έναν διαδραστικό χάρτη. Η πληροφορία όμως είναι σχετικά δύσκολο να ανακτηθεί. Υπάρχει λειτουργία εύρεσης συγκεκριμένου αξιοθέατου και λειτουργία προχωρημένης αναζήτησης³, η οποία είναι ωστόσο δυσέυρετη αλλά παρέχει την δυνατότητα αναζήτησης ανά κατηγορία (Μουσείο, Μνημείο, Αρχαιολογικός Χώρος κλπ). Ο χάρτης της σελίδας χρησιμοποιεί το [google maps](https://www.google.com/maps), και, καθώς δεν είναι δωρεάν κώδικας ανοικτής πηγής, φαίνεται να δημιουργεί προβλήματα κατά την απεικόνιση του χάρτη όπως φαίνεται στην **Εικόνα 2.16**. Επίσης, κατά την επιλογή κάποιας ταμπέλας εμφανίζεται ένα αναδυόμενο παράθυρο με τον σύνδεσμο του αξιοθέατου, πράγμα που θα κάνει τον χρήστη να κάνει κλικ μια περιττή δεύτερη φορά.



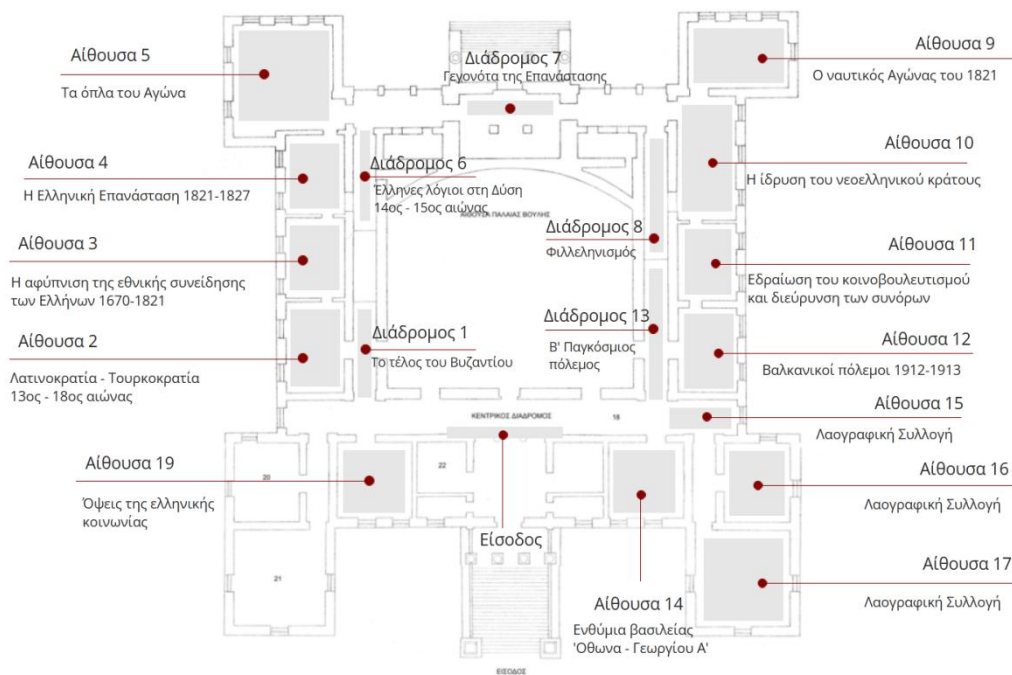
Εικόνα 2.16: Ο χάρτης της σελίδας odysseus.culture.gr, ανακτήθηκε Μάιο 2024

Η culture.lamia.gr είναι η επίσημη ιστοσελίδα του Δήμου Λαμίας και δημιουργήθηκε με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης όπως αναφέρεται και στην ίδια την ιστοσελίδα. Είναι διαθέσιμη σε τρεις γλώσσες, τα ελληνικά, τα αγγλικά και τα γερμανικά. Περιέχει 44 σημεία

³ <http://odysseus.culture.gr/a/6/ga60.jsp>

ενδιαφέροντος, πανοραμικές και εναέριες φωτογραφίες και βίντεο. Για κάθε σημείο ενδιαφέροντος υπάρχουν σχετικές ιστορικές και βοηθητικές πληροφορίες, καθώς και ένας χάρτης που υποδεικνύει την τοποθεσία του. Επιπλέον, υπάρχουν αναφορές για διαδραστικό χάρτη, αλλά κατά την διάρκεια της συγγραφής της παρούσας διπλωματικής δεν ήταν λειτουργικός.

Η nhmuseum.gr είναι μια μοντέρνα ιστοσελίδα του εθνικού ιστορικού μουσείου, που περιέχει πληροφορίες για την μόνιμη έκθεση και περιοδικές εκθέσεις του παρελθόντος. Στην σελίδα της μόνιμης έκθεσης υπάρχει το διαδραστικό πανόραμα με τις αίθουσες του μουσείου (**Εικόνα 2.17**), τις οποίες αν επιλέξει ο χρήστης θα πάει στην σελίδα με εκτεταμένες πληροφορίες και γκαλερί από φωτογραφίες.



Εικόνα 2.17: Πανόραμα αιθουσών του Εθνικού Ιστορικού Μουσείου

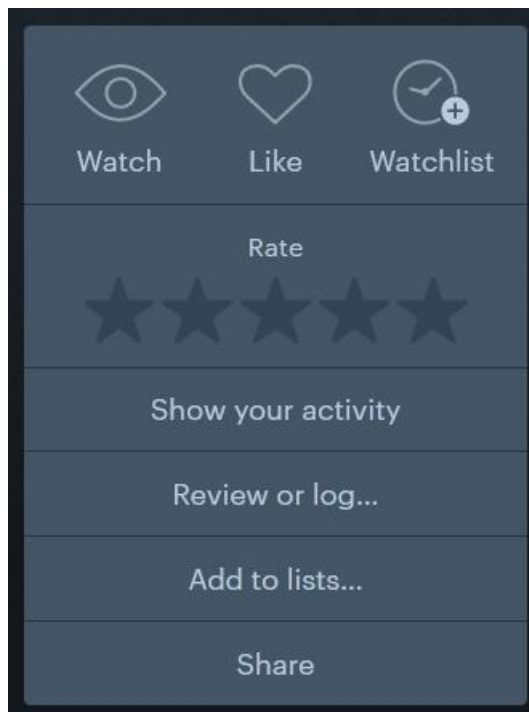
2.2 Γενικές ιστοσελίδες

Σε αυτό το κεφάλαιο θα γίνει μια συνοπτική αναφορά σε γενικές ιστοσελίδες, οι οποίες δεν έχουν ως πρωταρχικό σκοπό την βοήθεια ενός τουρίστα, τους χάρτες της google και της βικιπαίδειας. Επίσης θα γίνει αναφορά στην ιστοσελίδα letterboxd.com και στον τρόπο αλληλεπίδρασης χρήστη πληροφορίας, από όπου η τελική εφαρμογή που αναπτύχθηκε πήρε έμπνευση.

Οι χάρτες της google είναι πλέον ένα αναπόσπαστο κομμάτι της καθημερινής ζωής καθώς παρέχουν χρήσιμες οδηγίες και πληροφορίες για οποιοδήποτε προορισμό μας. Έτσι, όταν ένας τουρίστας επισκέπτεται μία χώρα το πιο πιθανό είναι να συμβουλευτεί την εφαρμογή αυτή. Εκεί μπορεί να αναζητήσει συγκεκριμένα αξιοθέατα ή να πλοηγηθεί στον χάρτη και να ψάξει για αυτά και να λάβει οδηγίες για να το πως να τα επισκεφθεί, τις ώρες λειτουργίας τους, φωτογραφίες τους, να συμβουλευτεί κριτικές άλλων χρηστών ή να ανεβάσει δική του.

Όσον αφορά την βικιπαίδεια, είναι η μεγαλύτερη εγκυκλοπαίδεια και περιέχει άρθρα για αξιοθέατα σε εκατοντάδες γλώσσες. Έτσι, ένας τουρίστας ανεξαρτήτως χώρας προέλευσης μπορεί να μάθει πληροφορίες για σχεδόν οποιοδήποτε αξιοθέατο στην μητρική του γλώσσα.

Το letterboxd.com είναι μία εφαρμογή κοινωνικής καταλογογράφησης για ταινίες⁴. Στην **Εικόνα 2.18** εμφανίζεται παράδειγμα του UI αυτής της αλληλεπίδρασης μεταξύ του χρήστη και μιας ταινίας. Κατά σειρά από αριστερά προς τα δεξιά ο χρήστης μπορεί: να δηλώσει ότι έχει δει μια ταινία, να επισημάνει ότι του άρεσε, να δηλώσει ότι επιθυμεί να την παρακολουθήσει μελλοντικά, να την βαθμολογήσει μεταξύ μισού αστεριού και πέντε αστεριών, να δει το ιστορικό του με αυτή τη ταινία, να αφήσει κριτική και να καταγράψει την ημερομηνία που την είδε/ξαναείδε, να την προσθέσει σε λίστα, να διαμοιράσει τον σύνδεσμο που οδηγεί σε αυτήν. Οι πρώτες τέσσερις λειτουργίες, αλλά αντί για ταινίες, για τα αξιοθέατα της Ελλάδος, ήταν κομμάτι της έμπνευσης να δημιουργηθεί η εφαρμογή της παρούσας διπλωματικής.



Εικόνα 2.18: Το κουτί αλληλεπίδρασης χρήστη-ταινίας στην ιστοσελίδα letterboxd.com

2.3 Περιγραφή προβλήματος και στόχοι

Σε αυτό το κεφάλαιο θα εξηγηθεί το πρόβλημα το οποίο υπάρχει στις εφαρμογές τουρισμού, το οποίο είχε ως στόχο να λύσει η εφαρμογή που αναπτύχθηκε. Το πρόβλημα αρχικά αναγνωρίστηκε κατά την πλοήγηση των χαρτών της google και μετέπειτα κατά την αναζήτηση πληροφοριών για τα πολλά αξιοθέατα της Ελλάδας. Όπως αναφέρθηκε στα προηγούμενα υπό κεφάλαια αυτού του κεφαλαίου, οι τουριστικές ιστοσελίδες παρέχουν άφθονη πληροφορία, δυστυχώς όμως όχι επικεντρωμένη. Έτσι, ο χρήστης πιθανώς να πρέπει να επισκέπτεται μία σελίδα για τις πολιτισμικές πληροφορίες, και

⁴ <https://en.wikipedia.org/wiki/Letterboxd>

την ιστορία του αξιοθέατου, σε μία άλλη για τις κριτικές από το κοινό που την επισκέφθηκε και σε μία τρίτη για τις οδηγίες. Τυπικά ένας χρήστης θα πρέπει να πλοηγηθεί το Google Maps για να βρει αξιοθέατα να επισκεφθεί και για τις κριτικές του, πιθανώς έπειτα στο visitgreece.com ή την βικιπαίδεια για να βρει πληροφορίες για αυτό, και τέλος αν δε καλύφθηκε από τις κριτικές του google maps να επισκεφθεί το tripadvisor.com. Επιπλέον, για λιγότερο γνωστά αξιοθέατα θα χρειαζόταν να επισκεφθεί και μία τοπική ιστοσελίδα για περισσότερες πληροφορίες. Επιπλέον, καμία από τις τουριστικές ιστοσελίδες που αναλύθηκαν παραπάνω δεν διαθέτει ικανοποιητική αλληλεπίδραση μεταξύ πληροφορίας και χρήστη, με το κοντινότερο σε κάτι τέτοιο να είναι η κριτική και η βαθμολογία που παρέχει το tripadvisor.com.

Ο στόχος της παρούσας διπλωματικής είναι η δημιουργία ενός διαδραστικού χάρτη της Ελλάδος, ο οποίος θα αναπαριστά τις τοποθεσίες των αξιοθέατων στην επικράτεια της. Έτσι θα μπορεί ο οποιοσδήποτε χρήστης να λάβει πληροφορίες για κάποιο αξιοθέατο έχοντας στο νου του που βρίσκεται αυτό το αξιοθέατο γεωχωρικά και σε σχέση με άλλα αξιοθέατα. Όπως αναφέρθηκε σε πολλά διαστήματα σε αυτό το κεφάλαιο, οι πολυπληθείς τουριστικές ιστοσελίδες/εφαρμογές δεν καλύπτουν επαρκώς τον παραπάνω στόχο. Πέρα από αυτό, η αλληλεπίδραση μεταξύ χρήστη και αξιοθέατο είναι επίσης ένας στόχος της εφαρμογής της παρούσας διπλωματικής, με τον χρήστη να μπορεί να δηλώσει ότι έχει επισκεφθεί ή θα ήθελε μελλοντικά να επισκεφθεί αξιοθέατα και να τα δηλώσει ως αγαπημένα. Επίσης όπως είναι εφικτό και σε άλλες ιστοσελίδες, να μπορεί αν τα βαθμολογήσει και να αφήσει κριτική για αυτά, καθώς επίσης να κάνει και μία ιδιωτική σημείωση. Στο τρίτο, τέταρτο και πέμπτο κεφάλαιο εξηγείται ο τρόπος με τον οποίο δημιουργήθηκε η εφαρμογή για να επιτύχει αυτόν τον σκοπό και στο έβδομο κεφάλαιο αναφέρονται οι περιορισμοί που δημιουργήθηκαν κατά την εκπόνησή της, καθώς και ενδεικνύονται μελλοντικές επεκτάσεις βάση αυτών.

Κεφάλαιο 3: Μεθοδολογία συλλογής δεδομένων

Μία εφαρμογή χρειάζεται και τα αντίστοιχα δεδομένα έτσι ώστε οι χρήστες της να μπορούν να την χρησιμοποιούν για αρκετό καιρό αλλά και για μπορούν να βγάλουν αξία από αυτήν την χρήση. Έτσι, σε μια εφαρμογή τουριστικού περιεχομένου με σκοπό την καταγραφή των αξιοθέατων της Ελλάδας χρειάζεται δεδομένα για κάθε νομό και νησί της, δεδομένα διαφορετικής φύσης (κατηγορίας) όπως μουσεία, φυσικά μέρη κλπ. Επίσης θα πρέπει ο χρήστης να μπορεί να ξεχωρίζει μεταξύ των διαφορετικών κατηγοριών και τόπων των αξιοθέατων. Τέλος, καθώς το διαδίκτυο είναι ένα κυρίως οπτικό μέσο, η εμφάνιση φωτογραφιών για το κάθε αξιοθέατο είναι ένα σημαντικό μέρος που θα ξεχώριζε την εφαρμογή από ένα απλό ευρετήριο και θα ήταν κοντινότερο σε ένα τουριστικό φυλλάδιο. Με την ίδια λογική, μια συνοπτική περιγραφή για κάθε αξιοθέατο παροτρύνει τον χρήστη να μάθει περισσότερα για το μέρος, δίνοντας αξία ακόμα και στον χρήστη ο οποίος δεν σκοπεύει να επισκεφθεί το μέρος.

Για την συλλογή των εικόνων και των συντεταγμένων χρησιμοποιήθηκε το Google Maps, με χειροκίνητη εύρεση των αξιοθέατων ανά νομό, πόλη και νησί. Κατά κανόνα επιλέχθηκαν όσα αξιοθέατα είχαν τουλάχιστον 10 κριτικές στο Google, με την εξαίρεση τοποθεσιών που είχαν ελάχιστα ή κανένα αξιοθέατο. Τα δεδομένα αποθηκεύονταν σε ένα υπολογιστικό φύλλο προτού εισαχθούν στην βάση δεδομένων.

3.1 Αξιοθέατα

Για κάθε αξιοθέατο έπρεπε να συλλεχθούν τα παρακάτω δεδομένα: όνομα, τοποθεσία, κύρια και δευτερεύοντες κατηγορίες, γεωγραφικές συντεταγμένες, περιγραφή, εικόνα Η χειροκίνητη εύρεση των δεδομένων από το Google Maps είχε τρία στάδια, με το πρώτο στάδιο να είναι το κυρίως στάδιο που δημιουργεί την εγγραφή στην βάση δεδομένων, και τα υπόλοιπα να εμπλουτίζουν την εγγραφή με επιπλέον χρήσιμες πληροφορίες.

3.1.1 Πρώτο Στάδιο: Εύρεση αξιοθέατου, τοποθεσίας και κύριας κατηγορίας

Τα σημαντικότερα δεδομένα πρέπει να έχει ένα αξιοθέατο για να συμπεριληφθεί στην βάση δεδομένων είναι το όνομα, την τοποθεσία και τις γεωγραφικές συντεταγμένες, και την κύρια κατηγορία του. Αρχικά, με την περιήγηση του χάρτη γινόταν η επιλογή του αξιοθέατος και συνεπώς και η εύρεση του ονόματος του και της τοποθεσίας του. Η εύρεση των συντεταγμένων, έτσι ώστε το αξιοθέατο να εμφανίζεται στην σωστή τοποθεσία στον διαδραστικό χάρτη, επιτεύχθηκε με την ενσωματωμένη λειτουργία στο δεξί “κλικ” που εμφανίζει ως πρώτη επιλογή το γεωγραφικό μήκος και πλάτος της τοποθεσίας όπου έγινε το “κλικ”. Οι κύριες κατηγορίες των αξιοθέατων είναι οι: Μουσείο (Museum), Ορόσημο (Landmark), Φυσικό Θέαμα (Natural Sight), Μοναστήρι (Monastery), Ζωολογικός Κήπος (Zoo), Πάρκο (Park). Συγκεκριμένα ως Μουσείο και Πάρκο χαρακτηρίστηκαν τα μουσεία και τα

πάρκα οποιουδήποτε είδους. Ως ορόσημο χαρακτηρίστηκαν τα αξιοθέατα τα οποία είχαν δημιουργηθεί από τον άνθρωπο όπως γέφυρες και φάροι ενώ ως φυσικό θέαμα οι σπηλιές και οι καταρράκτες. Ως μοναστήρι χαρακτηρίστηκε οποιοσδήποτε ορθόδοξος ναός, με τους τεμένους να είναι ως ορόσημα σε δική τους υποκατηγορία Τέμενος (Mosque). Τέλος, ως ζωολογικοί κήποι χαρακτηρίστηκαν και τα ενυδρεία και τα ερπετάρια.

3.1.2 Δεύτερο Στάδιο: Εικόνα

Η εύρεση της εικόνας επιτεύχθηκε με δύο τρόπους. Ο πρώτος ήταν η επιλογή της εικόνας η οποία υπήρχε στην σελίδα της Βικιπαίδειας του εκάστοτε αξιοθέατου. Ο δεύτερος χρειάστηκε λόγω του ότι η πλειοψηφία των αξιοθέατων δεν έχουν λήμμα στην Βικιπαίδεια, και έτσι χρειάστηκε να βρεθεί το URL από το Google Maps. Αυτό επιτεύχθηκε με την αντιγραφή του URL μιας εικόνας που υπήρχε στην συγκεκριμένη τοποθεσία για το αξιοθέατο στο Google Maps, και την μετατροπή του σε URL το οποία μπορεί να χρησιμοποιηθεί από μία ιστοσελίδα. Αυτό έγινε με την χρήση του παρακάτω κώδικα κελιού του excel ο οποίος χωρίστηκε για λόγους σαφήνειας σε τρία σκέλη.

Αρχικά το URL το οποίο αντιγράφηκε και είναι φανερό στους χρήστες του Google Maps είναι της μορφής (1) στην **Εικόνα 3.1**. Με την χρήση του `=RIGHT(F1;LEN(F1)-SEARCH("https: ";F1;2)+1)` (υποθέτοντας ότι το παραπάνω URL (1) βρίσκεται στο κελί F1), εξάγεται το πραγματικό URL της φωτογραφίας και μετατρέπεται στην μορφή (2) της **Εικόνας 3.1**. Έπειτα με την χρήση του `=LEFT(E1;SEARCH("%3D";E1)-1)` (υποθέτοντας ότι το παραπάνω URL (2) βρίσκεται στο κελί E1), εξάγεται το σκέλος του URL μέχρι το λεκτικό %3D - το οποίο στη κωδικοποίηση ASCII είναι ο χαρακτήρας "=" - και μετατρέπεται στην μορφή (3) της **Εικόνας 3.1**. Τέλος με την χρήση του `=SUBSTITUTE(D1;"%2F";"/")` αντικαθίσταται το λεκτικό "%2F" με τον χαρακτήρα "/" λόγω της κωδικοποίησης ASCII που έχει το URL. Η τελική μορφή του URL της φωτογραφίας η οποία θα προστεθεί στον αντίστοιχο πίνακα της βάσης δεδομένων είναι η μορφή (4) της **Εικόνας 3.1**.

1. Αρχικό URL	2. Πρώτη Τροποποίηση	3. Δεύτερη Τροποποίηση	4. Τρίτη τροποποίηση και τελικό URL
https://www.google.com/maps/place/P arthenon/@37.9715285,23.7267166,3a, 75y,90t/data=!3m8!1e2!3m6!1sAF1Qip MQ_urwoVK84i4xjmau3l6L860bZmmP8 F19rjz!2e10!3e12!6shhttps:%2F%2Fh5.g oogleusercontent.com%2Fp%2FAF1Qip MQ_urwoVK84i4xjmau3l6L860bZmmP8 F19rjz%3Dw114-h86-k- no!7i12000!8i9000!4m7!3m6!1s0x14a1 bd19ca39ee61:0x1b3fa079b878a218!8 m2!3d37.9715285!4d23.7267166!10e5! 16zL20vMDV4N2I?entry=ttu#	https:%2F%2Fh5.googleusercontent.co m%2Fp%2FAF1QipMQ_urwoVK84i4xj mau3l6L860bZmmP8F19rjz%3Dw114-h86- k- no!7i12000!8i9000!4m7!3m6!1s0x14a1 bd19ca39ee61:0x1b3fa079b878a218! m2!3d37.9715285!4d23.7267166!10e5! 16zL20vMDV4N2I?entry=ttu#	https:%2F%2Fh5.googleusercontent.co m%2Fp%2FAF1QipMQ_urwoVK84i4xj mau3l6L860bZmmP8F19rjz	https://lh5.googleusercontent.com/p/A F1QipMQ_urwoVK84i4xjmau3l6L860bZ mmP8F19rjz

Εικόνα 3.1: Παράδειγμα μετατροπής πηγαίου URL σε URL που συμβατό με CORS (σε υπολογιστικό φύλλο)

3.1.3 Τρίτο Στάδιο: Περιγραφή

Για την περιγραφή του εκάστοτε αξιοθέατου έγινε απόξεση (scraping) της Βικιπαίδειας με την δημιουργία προγράμματος στην γλώσσα προγραμματισμού Java και την χρήση της βιβλιοθήκης JWiki, η οποία βοηθάει στην αλληλεπίδραση μεταξύ της Java και της Βικιπαίδειας. Το πρόγραμμα διάβαζε ένα αρχείο με τα ονόματα των αξιοθέατων και για κάθε

αξιοθέατο, αν έβρισκε άρθρο στην Βικιπαίδεια, επέστρεφε μια συνοπτική περιγραφή του. Καθώς η πλειοψηφία των αξιοθέατων δεν έχουν λήμμα στην Βικιπαίδεια, η εύρεση περιγραφής με αυτόν τον τρόπο ήταν ημιτελής.

3.1.4 Τέταρτο Στάδιο: Δευτερεύων κατηγορίες και χρήσιμες ιστοσελίδες.

Η καταγραφή των δευτερεύων κατηγοριών έγινε εμπειρικά, με την βοήθεια της περιγραφής του κάθε αξιοθέατου και με περαιτέρω αναζήτησης πληροφοριών στο διαδίκτυο.

Οι χρήσιμες ιστοσελίδες για κάθε αξιοθέατο είναι για παράδειγμα οι επίσημες ιστοσελίδες του αξιοθέατου ή σελίδα της περιφέρειας, νομού ή δήμου στους οποίους βρίσκεται το αξιοθέατο και έχουν πληροφορίες για αυτό. Όλα αυτά θα έπρεπε να συλλεχθούν χειροκίνητα. Επίσης στις χρήσιμες ιστοσελίδες είναι και η ανακατεύθυνση προς στο Google Maps, έτσι ώστε να μπορεί ο χρήστης να χρησιμοποιήσει τις διάφορες υπηρεσίες που προσφέρει η ιστοσελίδα όπως πχ οδηγίες προς το αξιοθέατο. Αυτό το URL δημιουργείται για κάθε αξιοθέατο αυτόματα με τον κώδικα που φαίνεται στον **Πίνακα 3.1**. Η `placeTemp` είναι η μεταβλητή με τα δεδομένα που λαμβάνει το frontend από το `get query` προς το backend, `placeTemp.coordinates[0]` είναι το γεωγραφικό μήκος και `placeTemp.coordinates[1]` είναι το γεωγραφικό πλάτος.

Πίνακας 3.1: Κομμάτι κώδικα που δημιουργεί τον σύνδεσμο της τοποθεσίας του αξιοθέατου στο Google Maps.

```
let mapsLink = `https://www.google.com/maps/search/?api=1&query=${placeTemp.coordinates[1]},${placeTemp.coordinates[0]}`
```

3.2 Χάρτης της Ελλάδος

Ο χάρτης της Ελλάδος ανακτήθηκε από την κρατική ιστοσελίδα `geodata.gov.gr`⁵, και είναι συγκεκριμένα ο χάρτης των ορίων των νομών της Ελλάδος (ΟΚΧΕ). Η μεταφόρτωση περιείχε αρχείο μορφής `.shp` (Shapefile), η οποία είναι ειδική μορφή γεωχωρικών διανυσματικών δεδομένων για λογισμικό συστημάτων γεωγραφικών πληροφοριών (GIS). Έτσι τοποθετήθηκε στο λογισμικό QGIS για την τροποποίησή του έτσι ώστε να φτάσει στην επιθυμητή μορφή που θα είναι συμβατή με το `leaflet` στο frontend κομμάτι. Αρχικά χρειάστηκε να γίνει αλλαγή του CRS από `GGRS87 / Greek Grid` σε `EPSG:3857 - WGS 84 / Pseudo-Mercator`. Έπειτα έγινε εξαγωγή του αρχείου σε μορφή `.geojson` και τέλος μετατροπή του σε `.json`.

Η χρήση του χάρτη είναι πρακτική αλλά και αισθητική. Αρχικά βοηθάει τον χρήστη να ξεχωρίσει την Ελλάδα από τον χάρτη του κόσμου στο υπόβαθρο, προσφέροντας έντονα όρια στα σύνορα της χώρας και μεταξύ των νομών της. Επίσης, κατά την χρήση της ιστοσελίδας ο χρήστης μπορεί να επιλέξει οποιοδήποτε νομό για να μάθει περισσότερες πληροφορίες για αυτόν καθώς και να δει όλα τα αξιοθέατα που βρίσκονται σε αυτόν.

⁵ <https://geodata.gov.gr/el/dataset/oria-nomon-okkhe>. ανακτήθηκε Σεπτέμβριο 2023



Εικόνα 3.2: Ο χάρτης με τα όρια των νομών της Ελλάδος, όπως φαίνεται στο QGIS

3.2.1 Πληροφορίες νομών της Ελλάδος

Οι πληροφορίες για τους νομούς συλλέχθηκαν μέσω άρθρων της βικιπαίδειας όπως φαίνεται στον **Πίνακα 3.2** με την χρήση του προγράμματος που αναλύεται στην υπό ενότητα 3.3 για όσες βρέθηκαν κατάλληλα λύματα και χειροκίνητα για όσες όχι, πάλι από την βικιπαίδεια.

3.2.2 Ταμπέλες στον χάρτη

Για να μπορεί ο χρήστης να ξεχωρίσει το κάθε νησί και τον κάθε νομό, δημιουργήθηκαν ταμπέλες με τις ονομασίες του εκάστοτε νησιού και νομού τις οποίες μπορεί να ενεργοποιήσει/απενεργοποιήσει από το παράθυρο του φίλτρου (περισσότερα για το φίλτρο στο κεφάλαιο 5.5).

Δημιουργήθηκαν με την βοήθεια του QGIS και του PostgreSQL. Αρχικά εισάγονται οι πληροφορίες για την κάθε τοποθεσία στον πίνακα `Prefecture_Location` (βλέπε Κεφάλαιο 4). Στην συνέχεια εισάγεται ο πίνακας ως επίπεδο στο έργο του QGIS, και έτσι μπορούμε να δούμε με γραφικό τρόπο τα δεδομένα μας για να ελέγξουμε αν οι τοποθεσίες των ταμπελών είναι οι επιθυμητές. Τελικά εξάγουμε το τελικό `.geojson` αρχείο με τις γεωχωρικές πληροφορίες των όλων ταμπελών και το μετατρέπουμε σε μορφή `.json`, το οποίο αρχείο θα είναι και αυτό που θα χρησιμοποιηθεί στο frontend κομμάτι.

Πίνακας 3.2: Άρθρα που χρησιμοποιήθηκαν για τις περιγραφές των νομών

Pieria	https://en.wikipedia.org/wiki/Pieria_(regional_unit)
West Attica	https://en.wikipedia.org/wiki/West_Attica
Ioannina	https://en.wikipedia.org/wiki/Ioannina_(regional_unit)
Pella	https://en.wikipedia.org/wiki/Pella_(regional_unit)
Evrytania	https://en.wikipedia.org/wiki/Evrytania
Phocis	https://en.wikipedia.org/wiki/Phocis
Aetolia-Acarnania	https://en.wikipedia.org/wiki/Aetolia-Acarnania
Larissa	https://en.wikipedia.org/wiki/Larissa_(regional_unit)
Laconia	https://en.wikipedia.org/wiki/Laconia
Heraklion	https://en.wikipedia.org/wiki/Heraklion_(regional_unit)
Evros	https://en.wikipedia.org/wiki/Evros_(regional_unit)
Samos	https://en.wikipedia.org/wiki/Samos & https://en.wikipedia.org/wiki/Ikaria
Piraeus	https://en.wikipedia.org/wiki/Piraeus_Prefecture
Chalkidiki	https://en.wikipedia.org/wiki/Chalkidiki
Serres	https://en.wikipedia.org/wiki/Serres_(regional_unit)
Kozani	https://en.wikipedia.org/wiki/Kozani_(regional_unit)
Imathia	https://en.wikipedia.org/wiki/Imathia
Phthiotis	https://en.wikipedia.org/wiki/Phthiotis
Chios	https://en.wikipedia.org/wiki/Chios_(regional_unit)
Achaea	https://en.wikipedia.org/wiki/Achaea
Florina	https://en.wikipedia.org/wiki/Florina_(regional_unit)
Preveza	https://en.wikipedia.org/wiki/Preveza_(regional_unit)
Euboea	https://en.wikipedia.org/wiki/Euboea_(regional_unit)
Grevena	https://en.wikipedia.org/wiki/Grevena_(regional_unit)
Athens	https://en.wikipedia.org/wiki/Central_Athens_(regional_unit)
East Attica	https://en.wikipedia.org/wiki/East_Attica
Cephalonia	https://en.wikipedia.org/wiki/Cephalonia_Prefecture & https://en.wikipedia.org/wiki/Ithaca_(island)
Rhodope	https://en.wikipedia.org/wiki/Rhodope_(regional_unit)
Corinthia	https://en.wikipedia.org/wiki/Corinthia
Chania	https://en.wikipedia.org/wiki/Chania_(regional_unit)
Lesbos	https://en.wikipedia.org/wiki/Lesbos
Arcadia	https://en.wikipedia.org/wiki/Arcadia_(regional_unit)
Kastoria	https://en.wikipedia.org/wiki/Kastoria_(regional_unit)
Zakynthos	https://en.wikipedia.org/wiki/Zakynthos
Drama	https://en.wikipedia.org/wiki/Drama_Prefecture
Cyclades	https://en.wikipedia.org/w/index.php?title=Cyclades_Prefecture
Lefkada	https://en.wikipedia.org/wiki/Lefkada_(regional_unit)
Boeotia	https://en.wikipedia.org/wiki/Boeotia
Lasithi	https://en.wikipedia.org/wiki/Lasithi
Rethymno	https://en.wikipedia.org/wiki/Rethymno_(regional_unit)
Magnesia	https://en.wikipedia.org/wiki/Magnesia_(regional_unit)
Argolis	https://en.wikipedia.org/wiki/Argolis
Kavala	https://en.wikipedia.org/wiki/Kavala_(regional_unit) & https://en.wikipedia.org/wiki/Thasos
Elis	https://en.wikipedia.org/wiki/Elis
Karditsa	https://en.wikipedia.org/wiki/Karditsa_(regional_unit)
Arta	https://en.wikipedia.org/wiki/Arta_(regional_unit)
Mount Athos	https://en.wikipedia.org/wiki/Mount_Athos
Thessaloniki	https://en.wikipedia.org/wiki/Thessaloniki_(regional_unit)
Xanthi	https://en.wikipedia.org/wiki/Xanthi_(regional_unit)
Dodecanese	https://en.wikipedia.org/wiki/Dodecanese
Kilkis	https://en.wikipedia.org/wiki/Kilkis_(regional_unit)
Messenia	https://en.wikipedia.org/wiki/Messenia
Thesprotia	https://en.wikipedia.org/wiki/Thesprotia
Trikala	https://en.wikipedia.org/wiki/Trikala_Prefecture
Corfu	https://en.wikipedia.org/wiki/Corfu_Prefecture



Εικόνα 3.3: Ο χάρτης της Ελλάδος όπως φαίνεται με τις ταμπέλες στο QGIS

3.3 Κώδικας του scrapping προγράμματος

Το πρόγραμμα Java δημιουργήθηκε για να κάνει scrapping της Βικιπαίδειας για τις περιγραφές των αξιοθέατων και των νομών. Για αυτόν τον σκοπό χρησιμοποιήθηκε η βιβλιοθήκη Jwiki. Το πρόγραμμα διαβάζει από το αρχείο file.txt τα αξιοθέατα και τους νομούς προς αναζήτηση και γράφει ανά γραμμή στο description.txt την εισαγωγική παράγραφο της σελίδας της βικιπαίδειας με το αντίστοιχο τίτλο, αν βρεθεί αυτή. Στους παρακάτω Πίνακες 3.3 & 3.4 δίνεται ο κώδικας του προγράμματος.

Πίνακας 3.3: Ο κώδικας του αρχείου Main.java

```
import io.github.fastily.jwiki.core.Wiki;
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        Wiki wiki = new Wiki.Builder().build();
        ArrayList<String> arr = StaticMethods.read("file.txt");
        for (String place : arr) {
            StaticMethods.write("description.txt", wiki, place );
        }
    }
}
```

Πίνακας 3.4: Ο κώδικας του αρχείου StaticMethods.java

```
import io.github.fastily.jwiki.core.Wiki;

import java.io.*;
import java.util.ArrayList;

public class StaticMethods {
    public static ArrayList<String> read(String location) {
        ArrayList<String> list = new ArrayList<>();
        try {
            File file = new File(Main.class.getResource(location).toURI());
            BufferedReader br = new BufferedReader(new FileReader(file));

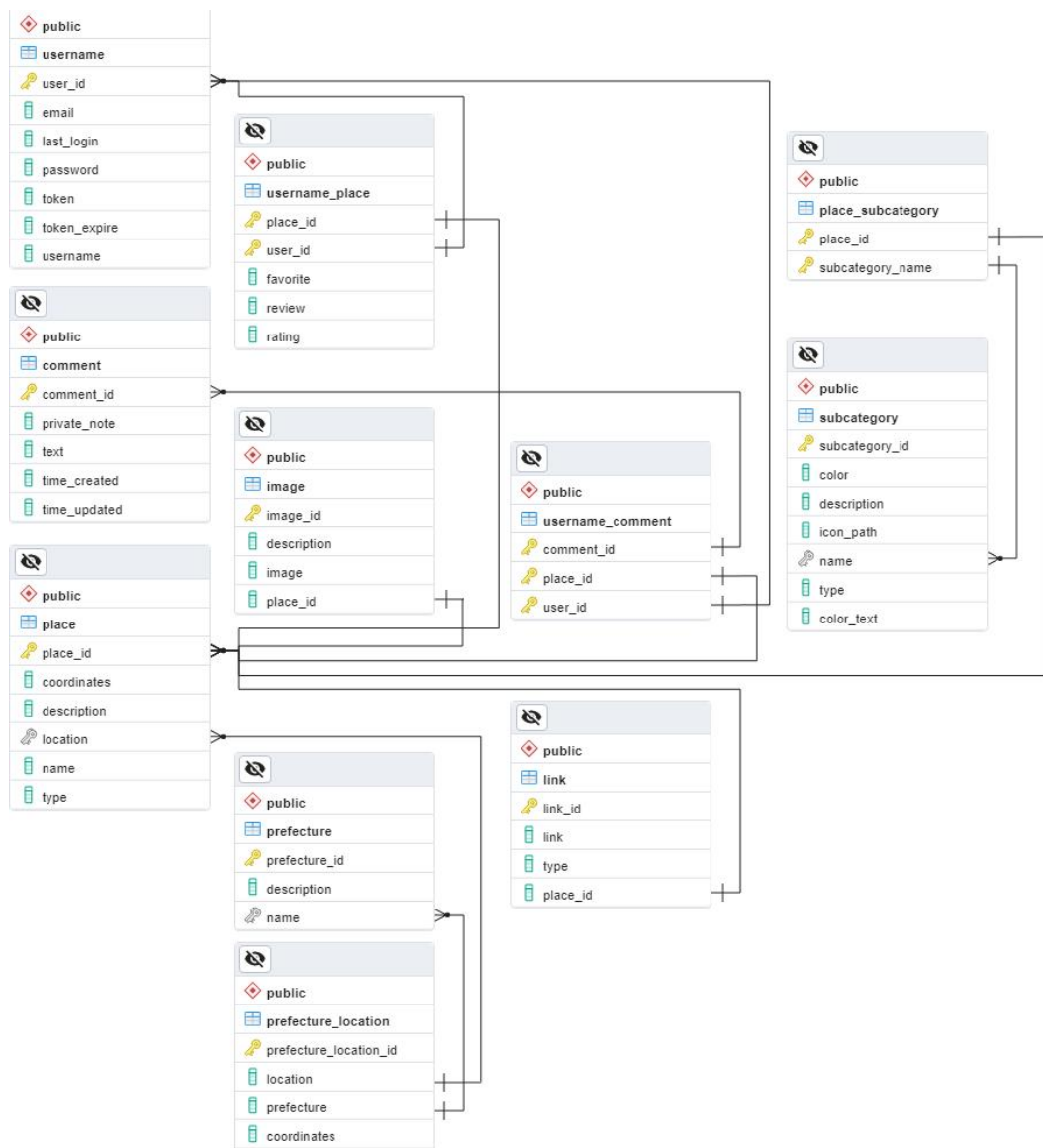
            String line;
            while ((line = br.readLine()) != null)
                list.add(line);
            br.close();
        } catch (Exception exc) {
            System.out.println(exc.getMessage());
        }

        return list;
    }

    public static void write(String location, Wiki wiki, String place) {
        String text = wiki.getTextExtract(place);
        try {
            FileWriter file = new FileWriter(location,true);
            BufferedWriter bw = new BufferedWriter(file);
            bw.write("\n");
            if (text!=null)
                bw.write("[ "+place+" ] " + text);
            else
                bw.write("");
            bw.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Κεφάλαιο 4: Αρχιτεκτονική της Εφαρμογής

Στο τέταρτο κεφάλαιο της παρούσας διπλωματικής εργασίας θα αναλυθεί ο τρόπος αρχιτεκτονικής της βάσης δεδομένων της εφαρμογής και τους τρόπους που δημιουργήθηκαν οι πίνακές της. Για την δημιουργία της βάσης δεδομένων επιλέχθηκε η PostgreSQL (χρησιμοποιήθηκε η έκδοση 15.1), αρχικά λόγο του ότι είναι λογισμικό ανοιχτού κώδικα και κυρίως λόγο των δυνατοτήτων της βιβλιοθήκης PostGis, επίσης ανοικτού κώδικα, η οποία προσθέτει υποστήριξη για αποθήκευση, ευρετηρίαση και αναζήτηση γεωχωρικών δεδομένων. Κατά τον σχεδιασμό της εφαρμογής χρειάστηκε να ληφθούν υπ' όψη οι πίνακες οι οποίοι θα χρησιμοποιούνταν καθώς και η συσχέτιση μεταξύ τους. Συνολικά δημιουργήθηκαν έντεκα πίνακες, με τέσσερις να είναι σχεσιακοί πίνακες για τις many-to-many σχέσεις μεταξύ άλλων πινάκων. Οι περισσότεροι πίνακες δημιουργήθηκαν αυτόματα μέσω του Hibernate, ενός εργαλείου αντιστοίχισης αντικειμένου (Object-Relational Mapping) της Java Spring.



Εικόνα 4.1: Το διάγραμμα της βάσης δεδομένων

4.1 Κύριοι πίνακες

4.1.1 Place (Αξιοθέατο)

Ο πίνακας place περιέχει τις κύριες πληροφορίες για το κάθε αξιοθέατο. Δημιουργήθηκε αυτόματα μέσω του Hibernate. Το πεδίο coordinates τύπου geometry αναφέρει τις συντεταγμένες του αξιοθέατου οι οποίες αξιοποιούνται με την χρήση QGIS για να εμφανιστούν στον διαδραστικό χάρτη στο front end κομμάτι της εφαρμογής (Βλέπε **Κεφάλαιο 5.1**). Τα πεδία name, description, location και type αντιστοιχούν στο όνομα, περιγραφή, τοποθεσία και βασική κατηγορία του αξιοθέατου. Το πεδίο location αντιστοιχεί στον πίνακα prefecture_location με many-to-one σχέση. Το place_id που είναι το primary key του πίνακα αποτελεί foreign key με one-to-many σχέση στους πίνακες Image, Link, many-to-many στους πίνακες Username_Place και Place_Subcategory. Στον πίνακα Username_Comment όσον αφορά το user_id έχει many-to-many σχέση (πολλοί χρήστες μπορούν να αφήσουν σχόλιο σε πολλά αξιοθέατα και πολλά αξιοθέατα μπορούν να έχουν σχόλιο από πολλούς χρήστες) και όσον αφορά το comment_id έχει one-to-many σχέση (ένα αξιοθέατο μπορεί να έχει πολλά σχόλια αλλά ένα σχόλιο μπορεί να είναι μόνο σε ένα αξιοθέατο).

Field Name	Data Type	Constraints
place_id	character varying (255)	Primary Key
coordinates	geometry	
description	text	
location	character varying (255)	Foreign Key
name	character varying (255)	
type	character varying (255)	

Εικόνα 4.2: Ο πίνακας Place αναλυτικά.

4.1.2 Username (Χρήστης)

Ο πίνακας username περιέχει πληροφορίες για τον κάθε χρήστη. Δημιουργήθηκε αυτόματα μέσω του Hibernate. Στην δημιουργία μιας εγγραφής στον πίνακα (δηλαδή την εγγραφή του κάθε χρήστη στην ιστοσελίδα) χρειάζεται ένας τρόπος το πεδίο user_id να είναι μοναδικό και όχι σειριακό, καθώς υπάρχει η πιθανότητα δεκάδες χρήστες να κάνουν εγγραφή ταυτόχρονα, με αποτέλεσμα να χρειαστεί πιθανώς η δημιουργία μια σειράς προτεραιότητας. Για την αποφυγή αυτού, επιλέχθηκε ο τύπος του πεδίου να είναι UUID και κατά την εγγραφή να δημιουργείται από έναν generator ο οποίος επιλέγει ένα UUID μεταξύ 2^{122} , που κάνει την πιθανότητα να υπάρχουν διπλοεγγραφές πολύ κοντά στο μηδέν⁶. Τα πεδία username, email, password

⁶ <https://fusionauth.io/dev-tools/uuid-generator>, Why generate UUID version 4? (Ανακτήθηκε Μάρτιο 2024)

και last_login αντιστοιχούν στο όνομα του χρήστη, το email του, τον κωδικό του και την ημερομηνία που συνδέθηκε τελευταία φορά στην εφαρμογή. Τα πεδία token και token_expire αντιστοιχούν στο token του χρήστη το οποίο δημιουργείται έτσι ώστε ο χρήστης να μην χρειάζεται να εισάγει συνεχώς τον κωδικό και το email του (Βλέπε **Κεφάλαιο 5.3**) και την ημερομηνία την οποία λήγει αυτό το token. Το user_id που είναι το primary key του πίνακα αποτελεί foreign key με one-to-many σχέση στο πίνακα Username_Place και στον πίνακα Username_Comment όσον αφορά το place_id έχει many-to-many σχέση (πολλοί χρήστες μπορούν να αφήσουν σχόλιο σε πολλά αξιοθέατα και πολλά αξιοθέατα μπορούν να έχουν σχόλιο από πολλούς χρήστες) και όσον αφορά το comment_id έχει one-to-many σχέση (ένας χρήστης μπορεί να έχει πολλά σχόλια).

username
user_id uuid
email character varying(255)
last_login timestamp with time zone(6)
password character varying(255)
token character varying(255)
token_expire timestamp with time zone(6)
username character varying(255)

Εικόνα 4.3: Ο πίνακας Username αναλυτικά

4.1.3 Image (Εικόνα)

Ο πίνακας Image περιέχει τις εικόνες των αξιοθέατων. Το πεδίο image_id είναι το primary key του πίνακα, το image περιέχει το σύνδεσμο της εικόνας και το description τυχών πληροφορίες για την εικόνα. Το πεδίο place_id είναι foreign key και αντιστοιχεί στον πίνακα Place με σχέση many-to-one, εφόσον στον πίνακα μπορούν να υπάρχουν πολλές εγγραφές με όμοιο place_id αλλά με μοναδικό image_id (πολλές εικόνες ανά ένα αξιοθέατο).

image
image_id character varying(255)
description character varying(255)
image text
place_id character varying(255)

Εικόνα 4.4: Ο πίνακας Image αναλυτικά

4.1.4 Comment (Σχόλιο/Σημείωση)

Ο πίνακας Comment περιέχει πληροφορίες για τα δημόσια σχόλια και ιδιωτικές σημειώσεις του χρήστη στο κάθε αξιοθέατο. Δημιουργήθηκε αυτόματα μέσω του Hibernate. Το πεδίο comment_id είναι το primary key του πίνακα και είναι τύπου UUID για τον ίδιο λόγο που είναι και το user_id στον πίνακα Username παραπάνω. Το text περιέχει το κείμενο του σχολίου, το private_note καθορίζει εάν το σχόλιο είναι ιδιωτικό (TRUE) ή δημόσιο (FALSE). Τα δύο πεδία τύπου timestamp time_created και time_updated αντιστοιχούν στην ημερομηνία δημιουργίας του σχολίου (που μένει σταθερή) και την ημερομηνία ανανέωσης του σχολίου (που αλλάζει κάθε φορά που γίνεται ανανέωση του σχολίου). Το comment_id αποτελεί foreign key στον πίνακα Username_Comment με σχέση one-to-one.

comment	
comment_id	uuid
private_note	boolean
text	text
time_created	timestamp without time zone(6)
time_updated	timestamp without time zone(6)

Εικόνα 4.5: Ο πίνακας Comment αναλυτικά

4.1.5 Link (Σύνδεσμος)

Ο πίνακας Link περιέχει τους συνδέσμους του κάθε αξιοθέατου. Δημιουργήθηκε αυτόματα μέσω του Hibernate. Το πεδίο link_id είναι το primary key του πίνακα, η μεταβλητή link περιέχει τον σύνδεσμο και η μεταβλητή type περιέχει τον τύπο του συνδέσμου και καθορίζει την εικόνα που θα εμφανίζεται στην θέση του. Το πεδίο place_id είναι foreign key και αντιστοιχεί στον πίνακα Place με σχέση many-to-one, όπως και στον πίνακα Image παραπάνω.

link	
link_id	character varying(255)
link	character varying(255)
type	character varying(255)
place_id	character varying(255)

Εικόνα 4.6: Ο πίνακας Link αναλυτικά

4.1.6 Prefecture (Νομός)

Ο πίνακας Prefecture περιέχει πληροφορίες για τους νομούς της Ελλάδας. Δημιουργήθηκε αυτόματα μέσω του Hibernate. Τα πεδία name και description

αντιστοιχούν στο όνομα του νομού και μια συνοπτική περιγραφή του. Το name είναι foreign key και έχει σχέση one-to-many με την μεταβλητή prefecture του πίνακα Prefecture_Location.

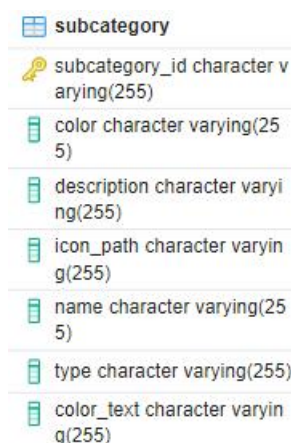


prefecture
prefecture_id
description
name

Εικόνα 4.7: Ο πίνακας Prefecture αναλυτικά

4.1.7 Subcategory (Υποκατηγορία)

Ο πίνακας Subcategory περιέχει πληροφορίες για τις δευτερεύων κατηγορίες (υποκατηγορίες) των αξιοθέατων. Τα πεδία name, description και type αντιστοιχούν στην ονομασία της υποκατηγορίας, στην περιγραφή της και στον τύπο της υποκατηγορίας. Ο τύπος της υποκατηγορίας μπορεί να είναι οποιοσδήποτε εκ των 9 παρακάτω: Park (Πάρκο), Gallery (Γκαλερί), Structure (Κτήριο), Nature (Φύση), Era (Έπος), Ship (Πλοίο), Point of Interest (Σημείο ενδιαφέροντος), Museum (Μουσείο), People (Άνθρωποι). Τα πεδία color και color_text αναφέρουν το χρώμα - του γεμίσματος και του κειμένου αντίστοιχα - της κάψουλας της υποκατηγορίας έτσι όπως φαίνεται στο κομμάτι του front end (Βλέπε Εικόνα 5.2 δ). Το πεδίο icon_path αντιστοιχεί στο μονοπάτι της εικόνας της κάψουλας. Το πεδίο name είναι foreign key και αντιστοιχεί στο πεδίο subcategory_name του πίνακα Place_Subcategory με σχέση many-to-many, με πολλές υποκατηγορίες να μπορούν να είναι σε πολλά αξιοθέατα. Στην τελική κατάσταση της εφαρμογής, τα πεδία description και icon_path δεν χρησιμοποιούνται πια, αλλά σε μελλοντική επέκταση της εφαρμογής θα μπορούσαν να φανούν χρήσιμα.



subcategory
subcategory_id character varying(255)
color character varying(255)
description character varying(255)
icon_path character varying(255)
name character varying(255)
type character varying(255)
color_text character varying(255)

Εικόνα 4.8: Ο πίνακας Subcategory αναλυτικά

4.2 Πίνακες ένα-προς-πολλά και πολλά-προς-πολλά

4.2.1 Prefecture_Location (Νομός κάθε τοποθεσίας)

Ο πίνακας Prefecture_Location περιέχει την συσχέτιση του πίνακα Prefecture (πεδίο Prefecture) με το κάθε location του πίνακα Place (πεδίο location). Το πεδίο coordinates αντιστοιχεί στις συντεταγμένες της συγκεκριμένης τοποθεσίας (location) και χρησιμοποιείται στο front end κομμάτι για τη δημιουργία ταμπελών των νησιών και των νομών⁷. Το πεδίο type μπορεί να πάρει μία εκ των παρακάτω 3 τιμών: Island (Νησί), Land (Νομός), Other (Δεν έχει coordinates).

Column Name	Data Type
prefecture_location_id	character varying(255)
location	character varying(255)
prefecture	character varying(255)
coordinates	geometry
type	character varying(255)

Εικόνα 4.9: Ο πίνακας Prefecture_Location αναλυτικά

4.2.2 Place_Subcategory (Υποκατηγορίες ανά αξιοθέατο)

Ο πίνακας Place_Subcategory περιέχει την many-to-many συσχέτιση του πίνακα Place (place_id) με τον Subcategory (Subcategory_Name). Δημιουργήθηκε χειροκίνητα χρησιμοποιώντας query του Πίνακα 4.1. Τα πεδία place_id και subcategory_name είναι το primary key του πίνακα.

Πίνακας 4.1: Το SQL Query που εκτελέστηκε για να δημιουργηθεί ο πίνακας Place_Subcategory

```
CREATE TABLE Place_Subcategory (
  place_id VARCHAR(255),
  subcategory_name VARCHAR(255),
  PRIMARY KEY (place_id, subcategory_name)
)
```

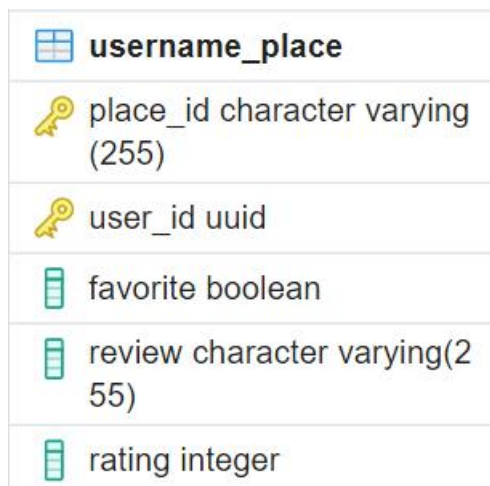
Column Name	Data Type
place_id	character varying(255)
subcategory_name	character varying(255)

Εικόνα 4.10: Ο πίνακας Place_Subcategory αναλυτικά

⁷ Κεφάλαιο 4.2.2

4.2.3 Username_Place (Συσχέτιση αξιοθέατου με χρήστη)

Ο πίνακας Username_Place αφορά τα αξιοθέατα τα οποία έχει επισκεφθεί ο χρήστης. Περιέχει την many-to-many συσχέτιση των πινάκων Username (user_id) και Place (place_id).



Column Name	Data Type
place_id	character varying (255)
user_id	uuid
favorite	boolean
review	character varying(255)
rating	integer

Εικόνα 4.11: Ο πίνακας Username_Place αναλυτικά

4.2.4 Username_Comment (Συσχέτιση σχόλιου με χρήστη και αξιοθέατο)

Ο πίνακας Username_Comment αφορά τα σχόλια που έχει το κάθε αξιοθέατο και τους χρήστες που τα έχουν δημιουργήσει. Δημιουργήθηκε αυτόματα από το Hibernate. Περιέχει την many-to-many συσχέτιση των πινάκων Username (user_id), Place (place_id) και Comment (comment_id).



Column Name	Data Type
comment_id	uuid
place_id	character varying (255)
user_id	uuid

Εικόνα 4.12: Ο πίνακας Username_Comment αναλυτικά

Κεφάλαιο 5: Υλοποίηση Εφαρμογής

Στο έκτο κεφάλαιο θα αναλυθεί ο τρόπος υλοποίησης της εφαρμογής στο backend Java Spring και στο frontend Angular κομμάτι, θα παρουσιαστούν στιγμιότυπα κώδικα για τις λειτουργίες που απαιτείται και θα περιγραφεί ο τρόπος επικοινωνίας μεταξύ frontend και backend. Στο backend κομμάτι χρησιμοποιήθηκε η Java 17 με την Spring 3.0.6 και τα dependencies Lombok, Spring Web, JDBC API, PostgreSQL Driver, Hibernate για την διευκόλυνση της επικοινωνίας του backend με την βάση δεδομένων και το frontend, και τα JTS Core & Jackson Datatype JTS για τον χειρισμό των γεωχωρικών μεταβλητών και δεδομένων. Στο frontend κομμάτι χρησιμοποιήθηκε Angular 14.2.9, Typescript 4.7.2 και το dependency leaflet 1.9.4.

Ο κώδικας του backend είναι χωρισμένος σε 3 κομμάτια: Model, Repository, Controller. Σαν model θεωρούνται όλες οι πρωτογενής κλάσεις που περιγράφουν το κάθε πίνακα ή projection, και οι serializable κλάσεις οι οποίες χρησιμοποιούνται ως κλειδιά σε ορισμένους πίνακες. Δημιουργήθηκαν 11 μοντέλα: (*Comment, Image, Link, Place, PlaceSubcategory, Prefecture, PrefectureLocation, Subcategory, Username, UsernameComment και UsernamePlace*) καθώς και τα παρακάτω 3 κλειδιά: (*PlaceSubcategoryKey, UsernameCommentKey και UserPlaceKey*). Ως repository χαρακτηρίζονται οι διεπαφές (τύπος Interface) που κάνουν extend την κλάση *org.springframework.data.repository.CrudRepository*. Τα repository μπορούν να χειριστούν την βάση δεδομένων απευθείας με μεθόδους όπως η *getAll()* που επιστρέφουν όλες τις εγγραφές του εκάστοτε πίνακα, με μεθόδους που δημιουργούνται αυτόματα βάση των στηλών του πίνακα όπως η *getNameByPlaceId()* αλλά και με προσαρμοσμένα native SQL query. Δημιουργήθηκαν τα παρακάτω 7 repository: *UsernameCommentRepository, PlaceRepository, UsernamePlaceRepository, SubcategoryRepository, CommentRepository, PrefectureRepository και UserRepository* για το μοντέλο Username. Τέλος, τα controller είναι οι κλάσεις που λαμβάνουν τα REST request από το frontend και αντίστοιχα επιστρέφουν τα κατάλληλα δεδομένα. Για να δημιουργηθούν αυτές χρειάζεται υποχρεωτικά το annotation *@RestController* της Spring Web καθώς και προστέθηκαν επίσης τα annotation *@CrossOrigin* για την σωστή συμπεριφορά έναντι CORS request και *@RequestMapping* για την καλύτερη οργάνωση των request path. Δημιουργήθηκαν συνολικά 3 controller (*PlaceController, UserController και UsernamePlaceController*). Στα παρακάτω υπο-κεφάλαια θα παρουσιαστούν στιγμιότυπα κώδικα για ορισμένα model και για ορισμένες μεθόδους των repository και controller και στους **Πίνακες 5.1 & 5.2** θα παρουσιαστούν περιληπτικά όλες οι μέθοδοι τους αντίστοιχα.

Ο κώδικας του frontend είναι χωρισμένος στα εξής κομμάτια: Component, Service, Model & Module. Τα component αποτελούνται από τεσσάρων τύπων αρχεία: τα .html για την σήμανση των τμημάτων της σελίδας, τα .css για την μορφοποίηση της σελίδας και .ts για τον typescript κώδικα. Ο τέταρτος τύπος αρχείων είναι ο .spec.ts που είναι ο κώδικας typescript για testing αλλά δεν χρησιμοποιήθηκε κατά την εκπόνηση της εφαρμογής. Τα

υπόλοιπα είναι της μορφής .ts. Τα service είχαν ως λειτουργικότητα την επικοινωνία του κώδικα των component με τον backend κομμάτι μέσω REST request, τα model είναι οι κλάσεις των αντικειμένων (Πίνακας 5.3)

Πίνακας 5.1: Περιγραφή των μεθόδων των Repository του backend ανά αρχείο

Όνομα αρχείου	Μέθοδος & Περιγραφή
CommentRepository	getCommentsByPlaceId(place_id,user_id) <i>Ανάκτηση σημειώσεων αξιοθέατου</i>
	getByCommentIdMy(comment_id) <i>Ανάκτηση σχόλιου</i>
PlaceRepository	getAll() <i>Ανάκτηση όλων των αξιοθέατων</i>
PrefectureRepository	getName(name) <i>Ανάκτηση νομών βάση ονόματος τοποθεσίας</i>
	getPrefecturePlacesByName(name) <i>Ανάκτηση αξιοθέατων βάση ονόματος τοποθεσίας</i>
SubcategoryRepository	getAllSubcategoryNames() <i>Ανάκτηση των ονομάτων όλων των υποκατηγοριών</i>
	getSubcategories(place_id) <i>Ανάκτηση υποκατηγοριών του αξιοθέατου</i>
	getSubcategoryByName(name) <i>Ανάκτηση υποκατηγορίας βάση ονόματος</i>
	getPlacesBySubcategories(subcategories[],count) <i>Ανάκτηση λίστας αξιοθέατων βάση την υποκατηγορία τους</i>
UserCommentRepository	getByCommentId(comment_id) <i>Ανάκτηση σχέσης χρήστη-σημείωσης βάση comment id</i>
UserPlaceRepository	getUserAndPlace(user,place) <i>Ανάκτηση σχέσης χρήστη-αξιοθέατου</i>
	getUserIdAndPlaceId(user_id,place_id) <i>Ανάκτηση σχέσης χρήστη-αξιοθέατου</i>
	getTotalVisits(place_id) <i>Ανάκτηση συνολικών επισκέψεων του αξιοθέατου</i>
	getTotalWishlist(place_id) <i>Ανάκτηση συνολικών επιθυμιών επίσκεψης του αξιοθέατου</i>
	getAverageRating(place_id) <i>Ανάκτηση μέσου όρου βαθμολογιών αξιοθέατου</i>
	getPlacesByUser(user_id) <i>Ανάκτηση λίστας αξιοθέατων που ο χρήστης έχει επισκεφθεί ή έχει επισημάνει ότι επιθυμεί να επισκεφθεί</i>
	getMostVisitedPlace() <i>Ανάκτηση του αξιοθέατου με τις περισσότερες επισκέψεις</i>
	getMostVisitedType() <i>Ανάκτηση υποκατηγορίας αξιοθέατου με περισσότερες επισκέψεις</i>
	getMostFavouritedPlace() <i>Ανάκτηση αξιοθέατου με τα παραπάνω αγαπημένα</i>
	getMostFavouritedType() <i>Ανάκτηση της υποκατηγορίας με τα παραπάνω αγαπημένα</i>
	getRecommended(uuid,location1,location2,type1,type2) <i>Ανάκτηση λίστας προτεινόμενων αξιοθέατων βάση των αποτελεσμάτων των 4 παραπάνω μεθόδων</i>
	getRecommendedRandom(uuid) <i>Ανάκτηση τυχαίας λίστας προτεινόμενων αξιοθέατων</i>
UserRepository	getEmailAndPassword(email,password) <i>Ανάκτηση χρήστη βάση email και κωδικού</i>
	getToken(token) <i>Ανάκτηση χρήστη βάση token</i>
	getUserMy(uuid) <i>Ανάκτηση χρήστη βάση UUID του χρήστη</i>

Πίνακας 5.2: Περιγραφή των μεθόδων των Controller του backend ανά αρχείο

Όνομα αρχείου	Path	Μέθοδος & Περιγραφή
PlaceController	[GET] /place/all	findAll() Ανάκτηση όλων των αξιοθέατων
	[GET] /place/id	findById() Ανάκτηση αξιοθέατου βάση
	[GET] /place/prefecture	getPrefectureByName(name) Ανάκτηση περιφέρειας βάση ονόματος
	[GET] /place/subcategories	getSubcategories(place_id) Ανάκτηση υποκατηγοριών αξιοθέατου
	[GET] /place/bySubcategories	getPlacesBySubcategories() Ανάκτηση λίστας αξιοθέατων βάση υποκατηγοριών του
	[GET] /place/subcategories/all	getAllSubcategories() Ανάκτηση υποκατηγοριών αξιοθέατου
	[GET] /place/comments	getPlaceComments(place_id, user_id) Ανάκτηση σχολίων αξιοθέατου
	[POST] /place/addComment	addComment(text,user_id,place_id, PrivateNote) Προσθήκη σχολίου στο αξιοθέατο
	[POST] /place/updateComment	updateComment(comment) Ενημέρωση σχολίου
	[DELETE] /place/deleteComment	deleteComment(comment_id) Διαγραφή σχολίου
UserController	[POST] /user/sign_up	signup(user) Εγγραφή χρήστη
	[POST] /user/login	login(user) Συνδέει τον χρήστη
	-	private doLogin(user) Ιδιωτική μέθοδος για σύνδεση χρήστη
	[POST] /user/logout	logout(user) Αποσυνδέει τον χρήστη
	[GET] /user/get	getUserById(user_id) Επιστρέφει χρήστη βάση id
	[GET] /user/stats	getStats(user_id) Επιστρέφει στατιστικά για τον χρήστη
	[GET] /user/recommended	getRecommended(user_id) Επιστρέφει λίστα αξιοθέατων που έχουν επιλεγθεί ως προτεινόμενα
UsernamePlaceController	[GET] /userplace	get(user_id,place_id) Ανάκτηση σχέσης αξιοθέατου-χρήστη
	[POST] /userplace/visited	createVisited(user_id,place_id) Δημιουργία σχέσης αξιοθέατου-χρήστη εάν ο χρήστης δεν είχε προηγουμένως
	[POST] /userplace/favorite	favorite(user_id,place_id,favorite) Εναλλαγή του αξιοθέατου ως αγαπημένου ή όχι
	[POST] /userplace/wishlist	wishlist(user_id,place_id,wishlist) Εναλλαγή του αξιοθέατου ως επιθυμητό για επίσκεψη ή όχι
	[DELETE] /userplace/visited	deleteVisited(user_id,place_id) Διαγραφή σχέσης κατά την αφαίρεση αξιοθέατου από λίστα επισκέψεων
	[DELETE] /userplace/wishlist	deleteWishlist(user_id,place_id) Διαγραφή σχέσης κατά την αφαίρεση αξιοθέατου από λίστα επιθυμιών
	[POST] /userplace/rate	rate(user_id,place_id, rating) Βαθμολόγηση του αξιοθέατου

Πίνακας 5.3: Περιγραφή component,model, service, module του frontend

Όνομα	Περιγραφή
Components	
app	Περιέχει το app-side-menu και τον router
info-panel	Το πάνελ πληροφοριών. Περιέχει το info-images, info-panel-comments & info-user-interact
info-images	Οι εικόνες του αξιοθέατου, αν υπάρχουν
info-panel-comments	Η λίστα σχολίων του αξιοθέατου και η δημιουργία νέου σχολίου, περιέχει το comment
comment	Το σχόλιο
info-user-interact	Η αλληλεπίδραση χρήστη με αξιοθέατο (Δήλωση επίσκεψης, επιθυμίας επίσκεψης, ως αγαπημένου, και βαθμολόγησης)
locations	Η λίστα όλων των αξιοθέατων
map-full	Ο διαδραστικός χάρτης
side-menu	Η γραμμή μενού, περιέχει login-profile, theme-switcher & search
login-profile	Η εγγραφή, σύνδεση και η επιλογή της προσωπικής σελίδας του χρήστη σε περίπτωση που είναι ήδη συνδεδεμένος.
theme-switcher	Το κουμπί εναλλαγής θέματος μεταξύ σκοτεινού και φωτεινού
search	Το πλαίσιο αναζήτησης
user-page	Η προσωπική ιστοσελίδα του χρήστη. Περιέχει το user-places
user-places	Η λίστα αξιοθέατων που εμφανίζεται στην σελίδα χρήστη
Models	
comment.model.ts	Το σχόλιο χρήστη
image.model.ts	Η εικόνα αξιοθέατου
link.model.ts	Ο σύνδεσμος αξιοθέατου
location.model.ts	Η τοποθεσία αξιοθέατου
place.model.ts	Το αξιοθέατο, περιέχει μεταβλητές που δεν υπάρχουν στο backend: <ul style="list-style-type: none"> • latitude, longitude για τις τιμές πλάτους & μήκους, • image αν έχει ή όχι εικόνες το αξιοθέατο, • totalVisits, averageRating: για στατιστικά αξιοθέατου, • rating, favorite, wishlist: αλληλεπίδραση με χρήστη, • Subcategories: από τον πίνακα Place_Subcategory.
prefecture.model.ts	Ο νομός
stats.model.ts	Τα στατιστικά χρήστη
subcategory.model.ts	Η υποκατηγορία αξιοθέατου
user-comment.model.ts	Η σχέση χρήστη-σχολίου, περιέχει το id του σχολίου και το αντικείμενο User
user-place.model.ts	Η σχέση χρήστη-αξιοθέατου, περιέχει την boolean μεταβλητή visited η οποία δεν υπάρχει στο backend
user.model.ts	Ο χρήστης
Services	
api.service.ts	Περιέχει τα get, post & delete request της εφαρμογής
auth-guard.service.ts	Ανακατευθύνει στην κεντρική σελίδα σε περίπτωση που ο χρήστης δεν είναι συνδεδεμένος
auth.service.ts	Οι μέθοδοι εγγραφής, σύνδεσης και ανάκτησης αντικείμενου χρήστη
comment.service.ts	Οι μέθοδοι προσθήκης, διαγραφής και ανάκτησης σχολίων
filter.service.ts	Περιέχει στατικές μεταβλητές που χρησιμοποιούνται κατά το φιλτράρισμα αξιοθέατων
place.service.ts	Οι μέθοδοι ανάκτησης πληροφοριών των αξιοθέατων, των υποκατηγοριών τους και των νομών
user-place.service.ts	Οι μέθοδοι αλληλεπίδρασης χρήστη-αξιοθέατου
Module	
app.module.ts	Δηλώσεις και εισαγωγές της εφαρμογής
app-routing.module.ts	Δηλώσεις των path της εφαρμογής

Αξίζει να αναφερθεί ότι στον κύριο φάκελο της εφαρμογής του frontend υπάρχει το αρχείο styles.css, το οποίο είναι το κύριο αρχείο και οι κανόνες σε

αυτό επιδρούν σε όλη την ιστοσελίδα. Έτσι, εκεί έχουν εισαχθεί οι global μεταβλητές χρωμάτων και των δύο θεμάτων - οι οποίες εναλλάσσονται όπως θα εξηγηθεί στο **Κεφάλαιο 5.9** παρακάτω - καθώς και η γραμματοσειρά, οι εικόνες και το χρώμα φόντου των κύριων κατηγοριών των αξιοθέατων και την κεντράρισμα των πολλαπλών πάνελ της ιστοσελίδας. Επίσης, εδώ βρίσκονται και οι αλλαγές μερικών κανόνων των dependencies leaflet και markercluster καθώς και η προσθήκη νέων κανόνων για τις κλάσεις αυτών.

Τέλος, τα module είναι αρχεία/κλάσεις που περιέχουν τις δηλώσεις των component και των module της Angular που χρησιμοποιούνται στην εφαρμογή.

5.1 Διαδραστικός Χάρτης

Για την υλοποίηση του διαδραστικού χάρτη χρησιμοποιήθηκε η βιβλιοθήκη leaflet στο frontend κομμάτι της εφαρμογής και το πρόγραμμα QGIS για την μετατροπή των συντεταγμένων των αξιοθέατων και των ταμπελών από τιμές στην βάση δεδομένων σε .geojson αρχείο όπως αναφέρθηκε και στο **Κεφάλαιο 3.2**. Η αρχικοποίηση του χάρτη γίνεται στο component map-full με τις μεθόδους initMap() και addGeoJson() όπως φαίνεται στον **Πίνακα 5.4**. Αρχικά στην initMap() δημιουργείτε ο πίνακας με την μέθοδο L.map και τις επιλογές: center [38.5, 24.5] έτσι ώστε το κέντρο να είναι οι συντεταγμένες που υποδεικνύουν το κέντρο της Ελλάδας, zoom 6.5 για το μέγεθος της μεγέθυνσης, zoomSnap για το μέγεθος αυξομείωσης της μεγέθυνση και attributionControl false για να μην εμφανίζεται κάτω αριστερά ο σύνδεσμος του Leaflet. Με την μέθοδο του χάρτη setMaxBounds() αποτρέπουμε τον χρήστη από το να ξεφύγει από τα όρια της Ελλάδας στον χάρτη, τα οποία θέτουμε αν είναι τα αρχικά όρια. Με την μέθοδο L.tileLayer δημιουργούμε την εμφάνιση του χάρτη, επιλέγοντας τον χάρτη VoyagerNoLabels⁸ και θέτοντας τη μέγιστη μεγέθυνση σε 18, την ελάχιστη σε 6, και την αδιαφάνεια σε 0.5. Τέλος θέτουμε τα σημάδια να έχουν μια προκαθορισμένη εικόνα για την αποφυγή μηνυμάτων σφάλματος. Έπειτα καλείται η μέθοδος addGeoJson() στην οποία εισάγουμε το περίγραμμα των νομών το οποίο έχουμε λάβει από το geodata.gov.gr σε μορφή json (Βλέπε **Κεφάλαιο 3.2**). Στις επιλογές του L.geoJson προσθέτουμε το αρχείο nomoi4.json, την μορφοποίηση και με την επιλογή onEachFeature: this.featureUtilitiesPrefectures θέτουμε ότι κατά την αλληλεπίδραση του χρήστη με τους νομούς θα καλείτε η μέθοδος featureUtilitiesPrefectures, η οποία καθορίζει το χρώμα του φόντου του κάθε νομού κατά την εισαγωγή/εξαγωγή του κέρσορα και επικεντρώνει τον νομό στην εικόνα σε περίπτωση επιλογής του. Μετά την αρχικοποίηση του χάρτη και των νομών, με το κάλεσμα της μεθόδου setupMarkers() προσθέτουμε ταμπέλες για τα αξιοθέατα και τα ονόματα των νησιών και των νομών. Όσον αφορά τα αξιοθέατα, οι ταμπέλες εμφανίζονται με ένα εικονίδιο, το οποίο επιλέγεται βάση της κύριας κατηγορίας τους. Επιπλέον, στον κώδικα css της εφαρμογής προσθέτεται χρωματιστό φόντο στα εικονίδια βάση της κατηγορίας τους, όπως φαίνεται στο στιγμιότυπο κώδικα του Πίνακα 6.5. Στην setupMarkers() γίνεται επίσης το φιλτράρισμα των αξιοθέατων βάση της κύριας κατηγορίας τους και των υποκατηγοριών τους, καθώς επίσης και ο έλεγχος της εμφάνισης

⁸ <https://leaflet-extras.github.io/leaflet-providers/preview/#filter=CartoDB.VoyagerNoLabels>

των ταμπελών βάση της μεγέθυνσης του χάρτη. Όταν ο χρήστης επιλέγει οποιαδήποτε από τις ταμπέλες ο χάρτης επικεντρώνεται σε αυτήν και επιπλέον εμφανίζονται πληροφορίες στο παράθυρο των πληροφοριών, όπως θα δούμε στο επόμενο κεφάλαιο.

Πίνακας 5.4: Οι μέθοδοι `initMap` και `addGeoJson`, 2 από τις μεθόδους που χρησιμοποιούνται για την αρχικοποίηση του χάρτη.

```
private initMap(): void {
  MapFullComponent.map = L.map('map', {
    center: [38.5, 24.5],
    zoom: 6.5,
    zoomSnap: 0.1,
    attributionControl: false
  });
  MapFullComponent.map.setMaxBounds(MapFullComponent.map.getBounds());
  const tiles =
L.tileLayer('https://{s}.basemaps.cartocdn.com/rastertiles/voyager_nolabels/{z}/{x}/{y}{r}.png', {
  maxZoom: 18,
  minZoom: 6,
  opacity: 0.5
});
tiles.addTo(MapFullComponent.map);
L.Marker.prototype.options.icon = this.DEFAULT_ICON; //to prevent 404 error on default
marker images
  this.addGeoJson();
  this.mapLoaded = true;
}
private addGeoJson() {
  var myStyle = {
    "color": "black",
    "fillColor": "white",
    "weight": 3,
    "opacity": 1
  };
  L.geoJson(nomoi4, {
    style: myStyle,
    onEachFeature: this.featureUtilitiesPrefectures
  }).addTo(MapFullComponent.map);
  this.setupMarkers();
}
```

Πίνακας 5.5: Απόσπασμα κώδικα που καθορίζει το χρώμα φόντου των εικονιδίων που χρησιμοποιούνται ως ταμπέλες για τα αξιοθέατα. Αφορά μόνο τις κατηγορίες Μουσείο, Ορόσημο και Φυσικό Θέαμα.

```
.place-type-icon.museum,
[src*="museum"] {
  background-color: rgba(165, 219, 255, 0.75) !important;
}
.place-type-icon.landmark,
[src*="landmark"] {
  background-color: rgba(164, 164, 164, 0.75) !important;
}
.place-type-icon.natural,
[src*="natural"] {
  background-color: rgba(72, 255, 121, 0.75) !important;
}
```


5.2 Πάνελ με πληροφορίες

Οι πληροφορίες του κάθε αξιοθέατου είναι χωρισμένες σε 4 πίνακες της βάσης δεδομένων: *Place*, *Image*, *Link*, *Subcategory* και οι πληροφορίες χρήστη-αξιοθέατου σε 2 πίνακες: *Username_Place*, *Username_Comment*. Οι πληροφορίες για κάθε νομό βρίσκονται στους πίνακες *Prefecture*, *PrefectureLocation & Place*. Όλες οι παραπάνω πληροφορίες πρέπει να εμφανίζονται στον χρήστη όταν επιλέγει ένα αξιοθέατο ή νομό με απλό και κατανοητό τρόπο. Επιπλέον για τα αξιοθέατα υπάρχουν και χρήσιμες συγκεντρωτικές πληροφορίες/στατιστικά όπως οι συνολικές επισκέψεις και ο μέσος όρος κριτικών, ενώ για τους νομούς είναι χρήσιμο να υπάρχει λίστα για όλα τα αξιοθέατα που βρίσκονται σε καθένα από αυτούς. Για αυτόν τον σκοπό δημιουργήθηκε στο frontend κομμάτι το component *info-panel*, το οποίο έχει με την σειρά του τα παρακάτω 3 υπό component: *info-images* για τις εικόνες, *info-panel-comments* για τις κριτικές και τις σημειώσεις, και *info-user-interact* για την αλληλεπίδραση του χρήστη με το αξιοθέατο. Σε αυτό το κεφάλαιο θα περιγραφούν αυτά τα component και θα δοθούν στιγμιότυπα οθόνης και κώδικα όπου χρειαστεί.

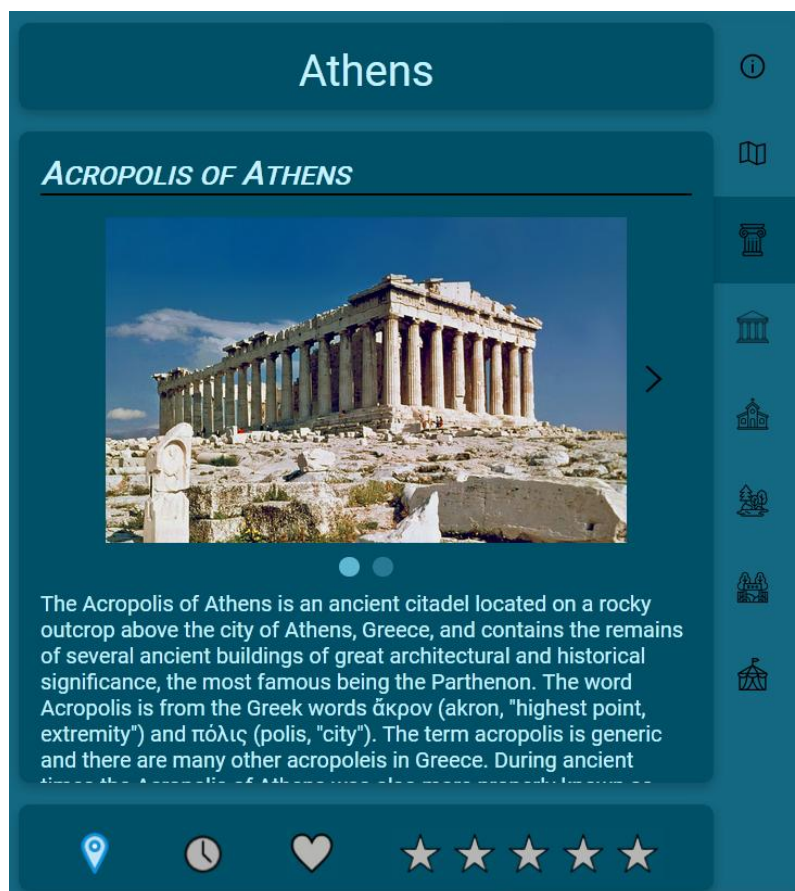
Όταν επισκεφθεί οποιαδήποτε σελίδα της εφαρμογής ένας χρήστης το πάνελ περιέχει μία καρτέλα με βοηθητικές πληροφορίες για την εκάστοτε υποσελίδα. Αυτή η αρχικοποίηση του πάνελ γίνεται απευθείας στον κώδικα του frontend και δεν ανακτάται από το backend. Για κάθε νέο αξιοθέατο ή νομό που θα επιλεγεί εμφανίζεται μια νέα καρτέλα, ενώ εάν ο χρήστης επιλέξει, για παράδειγμα, κάποιο αξιοθέατο στον χάρτη για το οποίο έχει ήδη ανοιχτεί σελίδα, θα επικεντρωθεί αυτή η σελίδα αν δεν είναι ήδη επικεντρωμένη. Στα δεξιά από την κάθε καρτέλα υπάρχει μια λίστα με σελιδοδείκτες, μέσω των οποίων ο χρήστης μπορεί να πλοηγηθεί από καρτέλα σε καρτέλα. Ο κάθε σελιδοδείκτης αποτελείται από το εικονίδιο της κύριας κατηγορίας του αξιοθέατου (σε περίπτωση βοηθητικής πληροφορίας και νομού υπάρχουν άλλα αντίστοιχα εικονίδια όπως φαίνεται στην **Εικόνα 5.1**) σε κανονική χρήση. Συγκεκριμένα τα εικονίδια της **Εικόνας 5.1** από πάνω προς τα κάτω αντιστοιχούν σε: χρήσιμες πληροφορίες, νομός, κατηγορία ορόσημο, κατηγορία μουσείο, κατηγορία μοναστήρι, κατηγορία φυσικό τοπίο, κατηγορία ζωολογικός κήπος και κατηγορία πάρκο. Κατά την επαφή του κέρσορα με αυτόν εμφανίζεται ο τίτλος του αξιοθέατου ενώ το εικονίδιο αντικαθίσταται από ένα κουμπί για το κλείσιμο της καρτέλας. Αν ο χρήστης κλείσει την τελευταία καρτέλα τότε το πάνελ εξαφανίζεται και θα εμφανιστεί ξανά όταν ο χρήστης επιλέξει κάποιο νέο αξιοθέατο ή αλλάξει υποσελίδα.

Για την αλληλεπίδραση του χρήστη με το αξιοθέατο υπάρχει το κουτί στο κάτω μέρος της καρτέλας το οποίο περιέχει τα κουμπιά με εικονίδια κατά σειρά: *Επίσκεψη*, *Επιθυμία επίσκεψης*, *Αγαπημένο*, και *5 κουμπιά για βαθμολογία από 1 μέχρι 5 αστέρια*. (Περισσότερα στο **Υπό Κεφάλαιο 5.4.1**) Το κουτί αυτό δεν εμφανίζεται αν ο χρήστης δεν είναι συνδεδεμένος.

Σε ένα κουτί πάνω από το πάνελ είναι η τοποθεσία του αξιοθέατου. Μέσα στο πάνελ εμφανίζονται πληροφορίες για αυτό, εάν υπάρχουν, όπως φαίνεται στην **Εικόνα 5.2**: (α) Το όνομα του αξιοθέατου, (β) μία φωτογραφία ή ένα καρουζέλ με φωτογραφίες αν το αξιοθέατο διαθέτει πάνω από μία, (γ) περιγραφή του αξιοθέατου, (δ) οι δευτερεύων κατηγορίες του, (ε) στατιστικά επισκεψιμότητας και μέσου όρου βαθμολόγησης του, (στ) χρήσιμοι σύνδεσμοι,

(ζ) μία περιοχή δημιουργίας σχόλιου και από κάτω της (η) η λίστα με τα σχόλιά του. Σε περίπτωση που ένα σχόλιο είναι ιδιωτικό, εμφανίζεται μόνο στον χρήστη που το δημιούργησε με το λεκτικό “noted” αντί “said”.

Στο πεδίο πεδίο ονόματος αξιοθέατου εμφανίζεται πέρα από αυτό, και το όνομα του νομού ή το όνομα της υποσελίδας στις αντίστοιχες περιπτώσεις. Η φωτογραφία εμφανίζεται μόνο εάν το αξιοθέατο διαθέτει κάποια εγγραφή στον πίνακα Image, και αν έχει παραπάνω από μία τότε εμφανίζονται βέλη αριστερά και δεξιά για την βοήθεια πλοήγησης στην προηγούμενη και στην επόμενη εικόνα αντίστοιχα, καθώς και μία μπάρα πλοήγησης εικόνων με τελείες που αντιπροσωπεύουν όλες τις εικόνες, κάτω από την εικόνα. Η περιγραφή του αξιοθέατου ανακτάτε από τον πίνακα Place και δεν έχει κάποιο συγκεκριμένο όριο λέξεων. Οι δευτερεύων κατηγορίες ανακτούνται από τους πίνακες PlaceSubcategory και Subcategory όπου στο backend κατά την κλήση της get /place/subcategories αρχικά ανακτά όλες τα ονόματα των υποκατηγοριών που έχει το αξιοθέατο από τον PlaceSubcategory και έπειτα ανακτά πληροφορίες για αυτές (χρώμα φόντου και χρώμα χαρακτήρων) από τον πίνακα Subcategory. Τα στατιστικά συνολικών επισκέψεων και μέσου όρου βαθμολογιών ανακτώνται με τα πρώτα δύο sql query που φαίνονται στον **Πίνακα 5.6** αντίστοιχα. Οι χρήσιμοι σύνδεσμοι επίσης ανακτώνται από τον πίνακα Link, εκτός του συνδέσμου προς τα google maps ο οποίος δημιουργείται όπως φαίνεται στον **Πίνακα 3.1**. Για το πλαίσιο δημιουργίας σχολίου θα μιλήσουμε περισσότερο στο **Κεφάλαιο 5.4** και τέλος για την ανάκτηση όλων των δημόσιων σχολίων καλείτε το τρίτο query που φαίνεται στον **Πίνακα 5.7** και αναλύεται περαιτέρω και αυτό στο **Κεφάλαιο 5.4**.



Εικόνα 5.1: Το πάνελ πληροφοριών αξιοθέατου.

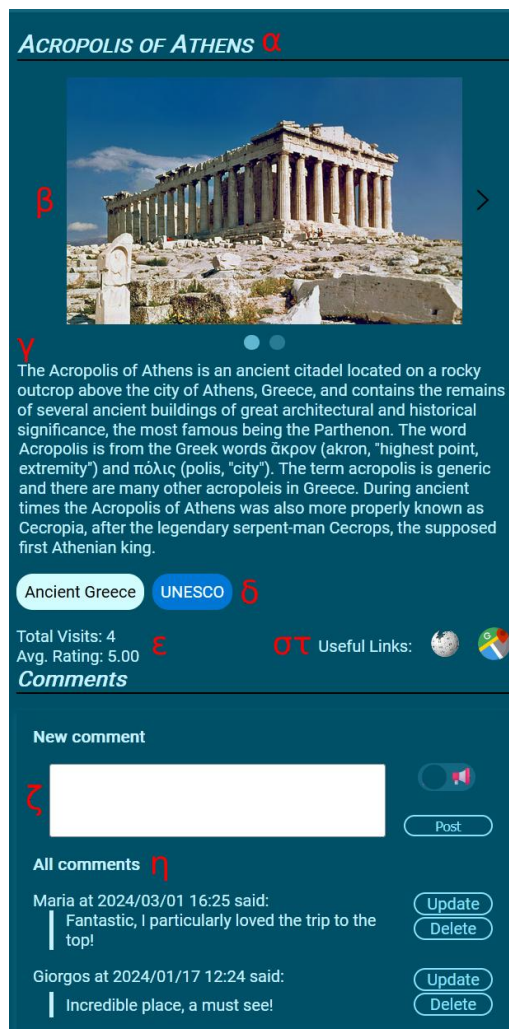
Πίνακας 5.6: Παράδειγμα query που τρέχουν κατά την δημιουργία του πάνελ πληροφορίας

```
--Επιλογή συνολικών επισκέψεων
select COALESCE(CAST(NULLIF( count(up),0) as text), 'None')
from username_place up
where place_id = :place_id and wishlist<>TRUE;

--Επιλογή μέσου όρου βαθμολογίας
select COALESCE(CAST(ROUND(AVG (rating),2) as text), 'N/A')
from username_place up
where place_id = :place_id and rating <> 0 and wishlist<>TRUE;
```

5.3 Δημιουργία λογαριασμού χρήστη (Sign up/Login/Logout)

Στο frontend κομμάτι δημιουργήθηκε, στο κάτω μέρος του πλαισίου μενού, μια διεπαφή με δύο καρτέλες, της εγγραφής και της σύνδεσης, όπως φαίνεται και στο **Κεφάλαιο 6** στην **Εικόνα 6.1 (γ, δ)**.



Εικόνα 5.2: Παράδειγμα πάνελ πληροφοριών.

Όσον αφορά την επικοινωνία frontend-backend, για την υλοποίηση της εγγραφής χρήστη δημιουργήθηκε το post request /user/sign_up, το οποίο όταν καλείτε από το frontend περνάει ένα αντικείμενο Username με το email,

όνομα χρήστη και κωδικό και στο backend απλά αποθηκεύει τον χρήστη στην βάση δεδομένων, δημιουργώντας ένα UUID για εκείνον. Μετά την εγγραφή ο χρήστης συνδέεται αυτόματα. Για να συνδεθεί ένας ήδη εγγεγραμμένος χρήστης χειροκίνητα μπορεί, σε διαφορετική καρτέλα της ίδιας διεπαφής, να πληκτρολογήσει το email και τον κωδικό του. Επίσης, για την αποφυγή αυτής της διαδικασίας κάθε φορά που εισέρχεται στην εφαρμογή, δημιουργήθηκε η μεταβλητή token τύπου UUID στην βάση, η οποία σώζεται στην τοπική αποθήκευση κατά την πρώτη σύνδεση του χρήστη, και έχει διάρκεια 30 μέρες η οποία επαναφέρεται κάθε φορά που ο χρήστης συνδέεται ξανά. Τέλος, για την αποσύνδεση χρήστη δημιουργήθηκε ένα κουμπί στην σελίδα χρήστη το οποίο εκτελεί το post request /user/logout και έτσι αποσυνδέει τον χρήστη, με το backend να ενημερώνει την χρονομέτρηση του token και της τελευταίας σύνδεσης του χρήστη.

5.4 Αλληλεπίδραση χρήστη με αξιοθέατα

Οποιαδήποτε αλληλεπίδραση με τα αξιοθέατα ο χρήστης την κάνει στο πάνελ με τις πληροφορίες των αξιοθέατων. Για την επίσκεψη, επιθυμία επίσκεψης, σήμανση ως αγαπημένο, και την βαθμολόγηση δημιουργήθηκαν κουμπιά τα οποία είτε εκτελούν post request είτε delete request. Για τον σχολιασμό χρησιμοποιήθηκε το αντικείμενο *textarea* για την εισαγωγή του κειμένου, και ένα slider με την βοήθεια css κώδικα, για την εναλλαγή μεταξύ δημόσιας κριτικής και ιδιωτικής σημείωσης (Περισσότερα για αυτό το slider στο **Κεφάλαιο 5.9**).

5.4.1 Επίσκεψη, επιθυμία επίσκεψης, βαθμολόγηση, σήμανση ως αγαπημένο

Ο πίνακας που χρειάστηκε να δημιουργηθεί για την σχέση χρήστη και αξιοθέατου είναι ο UsernamePlace. Στο μοντέλο στο backend κομμάτι χρειάστηκε να δημιουργηθεί η κλάση *UserPlaceKey* η οποία και λειτουργήσε ως το κλειδί του πίνακα. Έπειτα δημιουργήθηκαν οι κλάσεις repository *UserPlaceRepository* με τις μεθόδους που αναφέρονται στον **Πίνακα 5.1** και controller *UsernamePlaceController*, με τις αντίστοιχες REST μεθόδους που αναφέρονται στον **Πίνακα 5.2**. Όλες οι μέθοδοι ελέγχουν αρχικά αν υπάρχει κάποια σχέση με τον χρήστη και το αξιοθέατο, στην περίπτωση δηλαδή που ο χρήστης πατήσει πρώτα την σήμανση ως αγαπημένο (/userplace/favorite) ή την βαθμολογία (/userplace/rate). Έτσι λοιπόν σε αυτήν την περίπτωση δημιουργείται η σχέση με την μεταβλητή wishlist να αρχικοποιείται ως false, το οποίο δηλαδή σημαίνει ότι ο χρήστης έχει επισκεφθεί το αξιοθέατο. Τα post request /userplace/visited & /userplace/wishlist δρουν διαφορετικά. Σε περίπτωση που το visited δεν βρει σχέση, τότε όπως και στις favorite & rate θα δημιουργήσει μία νέα εγγραφή στον πίνακα UsernamePlace. Αν όμως βρει σχέση, αυτό μπορεί να σημαίνει 2 πράγματα: Είτε ότι υπάρχει ήδη σχέση αλλά με τον χρήστη να έχει δηλώσει πως θέλει να επισκεφθεί το αξιοθέατο στο μέλλον. Τότε, το μόνο που χρειάζεται να γίνει είναι να αλλάξει η μεταβλητή wishlist από *true* σε *false*. Παρομοίως, στην wishlist αν βρεθεί σχέση θα αλλάξει μεταβλητή wishlist από *false* σε *true*. Για να δουλέψει αυτή η λογική ήταν σημαντικό ο χρήστης να μη μπορεί να κάνει post request για να δηλώσει ότι έχει επισκεφθεί ή θέλει να επισκεφθεί μια τοποθεσία όταν αντιστοίχως το

έχει κάνει ήδη. Έτσι στο frontend μαζί με την εναλλαγή της κατάστασης visited και wishlist εν αλλάζονται και τα request, από post σε delete. Άρα όταν ο χρήστης πατήσει το “αναμμένο” κουμπί επίσκεψης ή επιθυμίας επίσκεψης εκτελείτε το αντίστοιχο delete request και διαγράφετε η εγγραφή.

5.4.2 Σχόλια χρήστη στα αξιοθέατα

Αρχικά χρειάστηκε να δημιουργηθεί ο πίνακας της βάσης δεδομένων ο οποίος αναφέρεται στο **Υποκεφάλαιο 4.1.4**. Στο backend δημιουργήθηκαν τέσσερις καινούργιες κλάσεις (*Comment*, *UsernameComment*, *UserCommentKey*, *CommentRepository*) και επιπλέον προστέθηκαν τέσσερις μέθοδοι στην κλάση *PlaceController*. Η κλάση του μοντέλου *Comment* περιέχει τις μεταβλητές *comment_id*, *text*, *privateNote*, *timeCreated* και *timeUpdated*, καθώς και την λίστα *usernameComments*. Η κλάση *UsernameComment* περιέχει τις μεταβλητές *comment*, *user*, *place* που αντιστοιχούν στις κλάσεις *Comment*, *User*, *Place* αντίστοιχα. Επιπλέον ως κλειδί της κλάσης *UsernameComment* δημιουργήθηκε η serializable κλάση *UserCommentKey*. Τέλος η κλάση *CommentRepository* περιέχει τα SQL queries τα οποία χρησιμοποιούν οι νέες GET, POST & DELETE μέθοδοι. Οι μέθοδοι που δημιουργήθηκαν και καλούνται από το frontend είναι 1 GET (*getPlaceComments*), 2 POST (*addComment*, *updateComment*), 1 DELETE (*deleteComment*). Στους **Πίνακες 5.7, 5.8, 5.9 & 6.10** εμφανίζεται ο κώδικας αυτών των μεθόδων.

Πίνακας 5.7: Μέθοδος *getPlaceComments* η οποία επιστρέφει όλα τα δημόσια σχόλια και όλα τα σχόλια του χρήστη και η μέθοδος *getCommentsByPlaceId* με το sql query.

```

Στην κλάση PlaceController.java :
@GetMapping("/comments")
public List<Comment> getPlaceComments(String place_id) {
    return commentRepository.getCommentsByPlaceId(place_id);
}

Στην κλάση CommentRepository.java :
@Query(value = "select * from comment where comment_id in " +
    "(select comment_id from username_comment where place_id =:place_id " +
    "and (private_note = 'FALSE' or (private_note = 'TRUE' and user_id =:user_id)))" +
    "order by time_created desc", nativeQuery = true)
List<Comment> getCommentsByPlaceId(@Param("place_id") String place_id,
@Param("user_id") UUID user_id);
    
```

Πίνακας 5.8: Μέθοδος *deleteComment* για την διαγραφή σχολίου.

```

@DeleteMapping("/deleteComment")
public void deleteComment(String comment_id) {
    Optional<Comment> c =
    commentRepository.getByCommentIdMy(UUID.fromString(comment_id));
    c.ifPresent(comment -> {
        Optional<UsernameComment> ouc =
        userCommentRepository.getByCommentId(UUID.fromString(comment_id));
        ouc.ifPresent(uc-> {
            userCommentRepository.delete(uc);
            commentRepository.delete(comment);
        });
    });
}
    
```

Πίνακας 5.9: Μέθοδος `addComment`, για την δημιουργία του σχολίου. Πρέπει πρώτα να δημιουργηθεί το σχόλιο (μεταβλητή `c`) και έπειτα το κλειδί για την συσχέτιση του σχολίου με το αξιοθέατο και τον χρήστη (μεταβλητή `uc`).

```
@PostMapping("/addComment")
public void addComment(String text, String user_id, String place_id, String privateNote) {
    UsernameComment uc = new UsernameComment();
    Optional<Username> ou = userRepository.getByUserIdMy(UUID.fromString(user_id));
    Optional<Place> op = placeRepository.findById(place_id);

    Comment c = new Comment();
    c.setText(text);
    c.setPrivateNote(Boolean.parseBoolean(privateNote));
    c.setTimeCreated(LocalDateTime.now());
    c.setTimeUpdated(LocalDateTime.now());
    uc.setPlace(op.get());
    uc.setUser(ou.get());
    uc.setComment(c);
    commentRepository.save(c);
    UserCommentKey uck = new UserCommentKey(c.getComment_id(),
ou.get().getUser_id(), op.get().getPlace_id());
    uc.setId(uck);
    userCommentRepository.save(uc);
}
```

Πίνακας 5.10: Μέθοδος `updateComment`, δέχεται το ανανεωμένο σχόλιο, αλλάζει την ημερομηνία `timeUpdated` στην τωρινή και σώζει το σχόλιο στην βάση.

```
@PostMapping("/updateComment")
public void updateComment(@RequestBody Comment c) {
    c.setTimeUpdated(LocalDateTime.now());
    Comment com = commentRepository.save(c);
}
```

5.5 Επιπλέον κουμπιά επάνω στον χάρτη

Κατά την υλοποίηση των λειτουργικοτήτων της εφαρμογής παρουσιάστηκε το πρόβλημα του χώρου. Έτσι κατά την ανάπτυξη της λειτουργικότητας του φίλτρου αποφασίστηκε να εμφανίζεται πάνω στον χάρτη αντί για κάποιο διπλανό τμήμα ή σε αναδυόμενο παράθυρο. Επιπλέον, το κουμπί 'Εμφάνιση ολόκληρου χάρτη' βρίσκεται και εκείνο πάνω στον χάρτη, στην πάνω αριστερά μεριά του, κάτω από τα κουμπιά ελέγχου του επίπεδο μεγέθυνσης τα οποία είναι ενσωματωμένα στο Leaflet από προεπιλογή. Οι θέσεις αυτών των δυο κουμπιών εμφανίζονται στην **Εικόνα 5.3** με κόκκινη περιγράμμιση. Για την λειτουργικότητα του κουμπιού 'Εμφάνιση ολόκληρου χάρτη' δημιουργήθηκε η μέθοδος `typescript` που αναγράφεται στον **Πίνακα 5.11**, η οποία επιστρέφει τον χάρτη στις αρχικές του συντεταγμένες και επίπεδο μεγέθυνσης.

Πίνακας 5.11: Η μέθοδος που καλείται κατά το πάτημα του κουμπιού 'Εμφάνιση ολόκληρου χάρτη'

```
public centerMap() {
    MapFullComponent.map.setView([38.5, 24.5], 6.5)
}
```

Για να τοποθετηθεί το κουμπί στο μέρος που αναφέρθηκε προηγουμένως χρειάστηκε ο κώδικας css που εμφανίζεται στον **Πίνακα 5.12**.

Πίνακας 5.12: Η CSS κλάση του κουμπιού 'Εμφάνιση ολόκληρου χάρτη'

```
.zoom-out-box {  
  display: grid;  
  cursor: pointer;  
  height: 1.9rem;  
  width: 1.9rem;  
  right: 44.85rem;  
  top: -14rem;  
  position: relative;  
  
  place-self: center;  
  place-content: start;  
  z-index: 999;  
  background-image: url("/assets/center.png");  
  background-repeat: no-repeat;  
  background-size: contain;  
  background-color: white;  
  border-radius: 4px;  
  border: 2px solid rgba(0,0,0,0.2);  
}
```



Εικόνα 5.3: Το κουμπί 'Εμφάνιση ολόκληρου χάρτη' με κόκκινη περιγράμμιση

Όσον αφορά το φιλτράρισμα των αξιοθέατων υπήρξαν δύο υλοποιήσεις, μία για τις κύριες κατηγορίες και μία για τις δευτερεύων. Οι κύριες κατηγορίες βρίσκονται στον πίνακα "Places" της βάσης και στο frontend κομμάτι βρίσκονται απευθείας επάνω στο εισαγόμενο αρχείο με τα αξιοθέατα. Έτσι για την υλοποίηση αυτού του φίλτρου δεν χρειάστηκε επικοινωνία με το backend. Το φιλτράρισμα εκτελείτε κάθε φορά που ο

χρήστης επιλέγει διαφορετική κύρια κατηγορία, όπου καλείτε η μέθοδος `filterByType`. Έτσι κατά το κάλεσμα της μεθόδου `setupMarkers()` στο `MapFull` component αρχικοποιούνται τα σημάδια των αξιοθέατων με το φίλτρο να είναι η μέθοδος `filterCategory`, η οποία επιστρέφει τα μόνο τα αξιοθέατα των οποίων η κατηγορία είναι ίδια με την στατική μεταβλητή `FilterService.filter`, εκτός αν εκείνη έχει την τιμή "All", όπου τα επιστρέφει όλα. Με παρόμοια λογική λειτουργεί το φιλτράρισμα των ταμπελών, το οποίο μπορεί να δεχθεί μία εκ των τριών τιμών: `Island`, `All`, `None`, για την εμφάνιση των ονομασιών των νησιών, όλων (νησιών και νομών) ή κανενός αντίστοιχα.

Για το φιλτράρισμα των υποκατηγοριών χρειάστηκε επικοινωνία με το backend, καθώς αυτές βρίσκονται στους πίνακες `PlaceSubcategory` & `Subcategory`. Επίσης, θα έπρεπε ως λειτουργικότητα να υπάρχει τρόπος ο χρήστης να επιλέξει παραπάνω από μία υποκατηγορία ταυτόχρονα, έτσι ώστε να εμφανίζονται μόνο αξιοθέατα που κατέχουν και τις 2 ή παραπάνω υποκατηγορίες αυτές. Κατά την επιλογή υποκατηγορίας από τον χρήστη, εκτελείτε η μέθοδος `filterBySubcategory`. Αρχικά γίνεται έλεγχος εάν ο χρήστης επέλεξε να καθαρίσει τα φίλτρα, όπου και αδειάζει ο πίνακας με τα φίλτρα υποκατηγοριών `FilterService.filterSubCats` και οι μεταβλητές φιλτραρίσματος των κύριων κατηγοριών και ταμπελών `FilterService.filter` & `FilterService.filterLabel` αντίστοιχα. Σε περίπτωση που έχει επιλέξει υποκατηγορία γίνεται έλεγχος αν η υποκατηγορία αυτή έχει επιλεγθεί ήδη, αν όχι τότε προσθέεται στον πίνακα αν ναι τότε αφαιρείται. Τέλος γίνεται κλήση προς το backend όπου θα εκτελεστεί το query που αναφέρεται στον **Πίνακα 5.13** για την επιστροφή των κλειδιών των αξιοθέατων που περιέχουν την μία ή παραπάνω υποκατηγορία που έχει επιλέξει ο χρήστης. Η μεταβλητή `:cats` είναι η λίστα με τις κατηγορίες και η `:count` είναι το μέγεθος της λίστας.

Πίνακας 5.13: Ανάκτηση όλων των αξιοθέατων τα οποία έχουν όλες τις υποκατηγορίες που έχει παραθέσει ο χρήστης.

```
select place_id from Place_Subcategory
where subcategory_name in :cats
group by place_id
having count(distinct(subcategory_name)) = :count;
```

Όταν ολοκληρωθεί το request και λάβει απάντηση το frontend θα εκτελεστεί ξανά η μέθοδος `setupMarkers` αλλά αυτή την φορά το φίλτρο θα είναι η μέθοδος `filterSubcategory`. Εκεί γίνονται τρεις έλεγχοι. Αρχικά ελέγχεται αν έχει ο πίνακας του φίλτρου υποκατηγοριών και ο πίνακας των κλειδιών των αξιοθέατων που πληρούν το φίλτρο είναι κενοί. Αν είναι τότε σημαίνει πως δεν έχει επιλεγθεί φίλτρο για τις υποκατηγορίες άρα επιστρέφει true δηλαδή εμφανίζει όλα τα αξιοθέατα. Ο δεύτερος έλεγχος είναι εάν το αξιοθέατο έχει την κύρια κατηγορία ή αν το φίλτρο κύριας κατηγορίας είναι `All`. Σε αυτές τις περιπτώσεις η μεταβλητή `hasMain` γίνεται true ειδάλλως false. Ο τρίτος και τελευταίος έλεγχος είναι για το εάν το αξιοθέατο έχει τις κατάλληλες υποκατηγορίες. Έτσι γίνεται έλεγχος αν το κλειδί του αξιοθέατου υπάρχει στον πίνακα που πήραμε από το backend και το αποτέλεσμα μπαίνει στην μεταβλητή `hasSub`. Τέλος αν και οι δύο μεταβλητές `hasMain` & `hasSub` είναι αληθές τότε το αξιοθέατο περνάει το φίλτρο και το σημάδι του εμφανίζεται στον χάρτη, σε κάθε άλλη περίπτωση δεν εμφανίζεται. Στον **Πίνακα 5.15** εμφανίζονται τα αποσπάσματα κώδικα των μεθόδων που χρησιμοποιούνται για την υλοποίηση της λειτουργικότητας του φίλτρου.

5.6 Λίστα όλων των αξιοθέατων

Για την ανάκτηση των κατάλληλων και συνοπτικών πληροφοριών για το κάθε αξιοθέατο καλείτε το GET request /place/all στο οποίο εκτελείται το query του **Πίνακα 5.14**. Συγκεκριμένα επιστρέφει: όνομα, τοποθεσία, κύρια κατηγορία, ύπαρξη περιγραφής και ύπαρξη εικόνας. Η μορφοποίηση χρησιμοποιείται ως πρότυπο και στις λίστες που αναφέρονται στο κεφάλαιο 6.7.1.

Πίνακας 5.14: Ανάκτηση περιληπτικής λίστας όλων των αξιοθέατων

```
select distinct(p.place_id), p.name, p.location, p.type,
case when p.description = '' then '0' else '1' end as description,
case when i.image is null then '0' else '1' end as image
from Place p left join Image i on p.place_id = i.place_id order by location, name;
```

5.7 Πληροφορίες και στατιστικά χρήστη

Στην σελίδα χρήστη εμφανίζονται στοιχεία και στατιστικά για τον χρήστη καθώς και λίστες για τα αξιοθέατα τα οποία έχει επισκεφθεί, έχει χαρακτηρίσει ως αγαπημένα, θα ήθελε να επισκεφθεί στο μέλλον και προτεινόμενα αξιοθέατα βάση του ιστορικού του. Οι πληροφορίες που εμφανίζονται είναι οι βασικές πληροφορίες που βρίσκονται στον πίνακα Username δηλαδή το όνομα χρήστη, το email και η ημερομηνία τελευταίας σύνδεσης. Ως στατιστικά αναφέρονται οι 4 τιμές που εμφανίζονται κάτω από τις πληροφορίες: ο αριθμός αξιοθέατων που έχει επισκεφθεί ο χρήστης, ο αριθμός αξιοθέατων που ο χρήστης έχει δηλώσει ότι θα ήθελε να επισκεφθεί, ο αριθμός αξιοθέατων που ο χρήστης έχει δηλώσει ως αγαπημένα και ο μέσος όρος βαθμολογιών του χρήστη. Στον μέσο όρο εξαιρούνται τα αξιοθέατα που δεν έχει βαθμολογήσει και εμφανίζονται δίπλα σε παρένθεση ως επιπλέον στατιστικό. Αυτά στατιστικά δημιουργούνται στο frontend κομμάτι από τα δεδομένα που λαμβάνει από το GET request /user/stats. Αρχικά δημιουργείτε subscription μεταβλητή που κάνει συνδρομή και περιμένει απάντηση από το backend, όταν την λάβει αρχικοποιεί τον πίνακα places της κλάσης. Μέσο φιλτράρισματος αυτού αρχικοποιεί τους πίνακες favPlaces & wisPlaces ελέγχοντας αν οι τιμές του favorite & wishlist είναι true αντίστοιχα. Έπειτα δημιουργεί τοπικά νέο πίνακα με όνομα placesWithrating, δηλαδή πίνακα με αξιοθέατα τα οποία έχουν βαθμολογία, με φιλτράρισμα πάλι από τον πίνακα places, ελέγχοντας αυτήν την φορά εάν η μεταβλητή rating είναι διαφορετική του 0. Με τον πίνακα placesWithrating υπολογίζει τον μέσο όρο όλων των βαθμολογιών και τον αποθηκεύει στην τοπική μεταβλητή average τύπου, από την οποία παίρνει τους πρώτους 4 χαρακτήρες και τους αποθηκεύει στην μεταβλητή averageRating της κλάσης stats. Κατόπιν δημιουργεί μία επιπλέον τοπική μεταβλητή, την unratedCount στην οποία με ένα τελευταίο φιλτράρισμα του πίνακα places παίρνει τον αριθμό των αξιοθέατων τα οποία ο χρήστης έχει επισκεφθεί αλλά δεν έχει βαθμολογήσει.

Πίνακας 5.15: Απόσπασμα κώδικα που εμφανίζει τις κύριες μεθόδους που χρησιμοποιούνται για την υλοποίηση των φίλτρων κύριων και δευτερεύων κατηγοριών. Εμφανίζονται οι μέθοδοι `filterByType`, `filterCategory`, `filterBySubcategory` & `filterSubcategory`

```

public filterByType(type: String) {
    FilterService.filter = type;
    this.setupMarkers();
}

filterCategory(feature: any, layer: any) {
    if (FilterService.filter === "All")
        return true;

    return feature.properties.type === FilterService.filter;
}

public filterBySubcategory(cat: String) {
    let bool: boolean = false;
    //Καθαρισμός όλων των φίλτρων
    if (cat === "clear") {
        FilterService.filterSubCats = [];
        FilterService.filter = 'All';
        FilterService.filterLabel = 'All';
    } else {
        let i = FilterService.filterSubCats.findIndex(i => {
            return i === cat
        });
        if (i === -1) { //Δεν υπάρχει, άρα προσθήκη
            FilterService.filterSubCats.push(cat);
            bool = true;
        } else { //Υπάρχει, άρα αφαίρεση
            FilterService.filterSubCats.splice(i, 1);
        }
    }

    PlaceService.getPlacesBySubcategories(FilterService.filterSubCats);
    this.setupMarkers();

    return bool;
}

filterSubcategory(feature: any, layer: any) {
    if (FilterService.filterSubCatIds.length === 0
        && FilterService.filterSubCats.length === 0)
        return true;

    let hasMain = FilterService.filter === "All"
        || feature.properties.type === FilterService.filter;

    let hasSub = FilterService.filterSubCatIds.findIndex(i => {
        return i === feature.properties.place_id
    }) !== -1;

    return hasMain && hasSub;
}

```

Τέλος ελέγχει αν `unratedCount` είναι μεγαλύτερη του μηδενός, και αν είναι θα εμφανίζεται ως στατιστικό ένα μήνυμα με το πόσα αξιοθέατα δεν

έχουν βαθμολογηθεί ακόμα. Στον **Πίνακα 5.16** εμφανίζεται ο κώδικας όλων των παραπάνω εντολών και υπολογισμών.

Πίνακας 5.16: Απόσπασμα κώδικα που υπολογίζει τα στατιστικά βάση της λίστας που λαμβάνει από το backend

```

this.subStats = UserPlaceService.userStatsSubject.subscribe(s => {
  this.places = s.places;
  this.favPlaces = this.places.filter(p => { return p.favorite == "true" });
  this.wisPlaces = this.places.filter(p => { return p.wishlist == "true" });
  let placesWithrating = this.places.filter(p => { return p.rating != "0" });
  let average = (placesWithrating.reduce((total, next) => total + parseFloat(next.rating +
  ""), 0) / (placesWithrating.length)) + "";
  this.stats.averageRating = average.substring(0, 4);
  let unratedCount = this.places.filter(p => { return p.rating === "0" && p.wishlist !=
  "true" }).length;
  if(unratedCount > 0)
    this.stats.totalUnrated = `${unratedCount} not rated`
})
    
```

Στο κάτω μέρος του πάνελ με τις πληροφορίες και τα στατιστικά βρίσκονται τέσσερις λίστες με αξιοθέατα: μία λίστα με όλα τα αξιοθέατα με τα οποία έχει σχέση ο χρήστης, δηλαδή είτε αυτά που έχει επισκεφθεί ήδη είτε αυτά που δήλωσε ότι επιθυμεί να επισκεφθεί στο μέλλον, μία λίστα με όλα τα αξιοθέατα που έχει δηλώσει ως αγαπημένα, μία λίστα με όσα έχει δηλώσει ως ότι επιθυμεί να επισκεφθεί και μία λίστα με προτεινόμενα βάση του ιστορικού επισκέψεών του. Όλες οι λίστες έχουν εμφανισιακά ως πρότυπο την λίστα στην σελίδα “λίστα αξιοθέατων”. Στον **Πίνακα 5.17** εμφανίζεται ο κώδικας CSS τους. Στο κεφάλαιο 6.8 θα περιγραφεί ο τρόπος με τον οποίο δημιουργείτε η λίστα με τα προτεινόμενα αξιοθέατα.

5.8 Προτάσεις αξιοθέατων

Το σύστημα των προτεινόμενων αξιοθέατων παίρνει ως βάση τις προτιμήσεις ενός χρήστη και τις συγκρίνει με τις προτιμήσεις των υπόλοιπων χρηστών. Όσο ο χρήστης χρησιμοποιεί την εφαρμογή και καταγράφει και βαθμολογεί τα αξιοθέατα τα οποίες έχει επισκεφθεί, η βάση δεδομένων εμπλουτίζεται με αυτές τις πληροφορίες, οι οποίες μετέπειτα χρησιμοποιούνται για να γίνουν οι προτάσεις συγκριτικά με τις καταγραφές και τις βαθμολογίες των υπόλοιπων χρηστών.

Υπάρχουν τέσσερις παράμετροι που λαμβάνονται υπ’ όψιν κατά την δημιουργία της λίστας με τα προτεινόμενα αξιοθέατα:

- Η τοποθεσία που έχει επισκεφθεί περισσότερο ο χρήστης (πχ Αθήνα, Σαντορίνη) (**Πίνακας 5.18**)
- Ο τύπος αξιοθέατου που έχει επισκεφθεί περισσότερο ο χρήστης (πχ Μουσείο) (**Πίνακας 5.19**)
- Η τοποθεσία στην οποία έχει τα περισσότερα αγαπημένα αξιοθέατα. (**Πίνακας 5.20**)
- Ο τύπος αξιοθέατου στον οποίο έχει τα περισσότερα αγαπημένα αξιοθέατα. (**Πίνακας 5.21**)

Πίνακας 5.17: Το αρχείο .css του component user-places.

```
.place-list {
  grid-template-rows: 2rem auto;
  height: 14rem;
}

.place {
  grid-template-columns: 8rem 7rem 20rem 4rem 4rem;
}

.places {
  background-color: var(--background2-color);
  height: 12rem;
}

.place-header-div,
.place-list {
  width: 44rem;
}

.place-favorite,
.place-rating {
  text-align: center;
}

.info-ratings {
  display: flex;
}

.info-button {
  width: 1rem;
  height: 1rem;
  background-size: 1rem;
  place-self: center;
}

.info-rate-n {
  background-image: url("/assets/user-interact-images/rate_n.png");
}

.info-rate-y {
  background-image: url("/assets/user-interact-images/rate_y.png");
}

.info-favorite-n {
  background-image: url("/assets/user-interact-images/favorite_n.png");
}

.info-favorite-y {
  background-image: url("/assets/user-interact-images/favorite_y.png");
}
```

Πίνακας 5.18: Τοποθεσία με τις περισσότερες επισκέψεις

```
select p.location
from Place p inner join username_place up on p.place_id = up.place_id
group by p.location
order by count(*) desc
fetch first 1 rows only;
```

Πίνακας 5.19: Τύπος αξιοθέατου με τις περισσότερες επισκέψεις

```
select p.type
from Place p inner join username_place up on p.place_id = up.place_id
group by p.type
order by count(*) desc
fetch first 1 rows only;
```

Πίνακας 5.20: Τοποθεσία με τα περισσότερα αγαπημένα αξιοθέατα

```
select p.location
from Place p inner join username_place up on p.place_id = up.place_id
group by p.location
order by sum(case when up.favorite = TRUE then 1 else 0 end) desc
fetch first 1 rows only;
```

Πίνακας 5.21: Τύπος αξιοθέατου με τα περισσότερα αγαπημένα αξιοθέατα

```
select p.type
from Place p inner join username_place up on p.place_id = up.place_id
group by p.type
order by sum(case when up.favorite = TRUE then 1 else 0 end) desc
fetch first 1 rows only;
```

Έπειτα από την συλλογή αυτών των τεσσάρων μεταβλητών θα τρέξει ακόμα ένα SQL Query (**Πίνακας 5.22**) το οποίο θα γυρίσει τα 10 πρώτα αξιοθέατα - βάση μέσης βαθμολογίας και σύνολο αγαπημένων - τα οποία βρίσκονται σε οποιαδήποτε από τις δύο παραπάνω τοποθεσίες(στο query ως location1 & location2) είτε σε οποιοδήποτε από τους δύο παραπάνω τύπους (στο query ως type & type2).

Πίνακας 5.22: Πρόταση 10 κορυφαίων αξιοθέατων βάση του ιστορικού του χρήστη σε σχέση με τις βαθμολογίες των άλλων χρηστών

```
select distinct(p.place_id), p.name, p.location, p.type, avg(up.rating)
sum(case when up.favorite = TRUE then 1 else 0 end)
from Place p inner join username_place up on p.place_id = up.place_id
where up.user_id <> :uid
and (p.location in (:location1, :location2) or p.type in (:type1, :type2))
group by p.place_id
order by avg(up.rating) desc, sum(case when up.favorite = TRUE then 1 else 0 end) desc,
p.name asc
fetch first 10 rows only;
```

Σε περίπτωση που το παραπάνω query επιστρέψει κενό, το οποίο μπορεί να συμβεί αν ο χρήστης έχει ήδη επισκεφθεί όλα τα αξιοθέατα τα οποία έχουν επισκεφθεί οι υπόλοιποι χρήστες, τότε θα επιστραφούν 10 τυχαία αξιοθέατα τα οποία εκείνος δεν έχει επισκεφθεί, χρησιμοποιώντας το query του **Πίνακα 5.23**.

Πίνακας 5.23: Πρόταση 10 τυχαίων τοποθεσιών τις οποίες δεν έχει επισκεφθεί ο χρήστης

```
SELECT p.place_id, p.name, p.location, p.type
FROM Place p
where p.place_id not in
(select p.place_id from Place p
inner join username_place up on p.place_id = up.place_id
where up.user_id = :uid)
ORDER BY RANDOM()
fetch first 10 rows only;
```

5.9 Εναλλαγή φωτεινού θέματος και σκοτεινού θέματος

Η εναλλαγή θέματος μεταξύ φωτεινού και σκοτεινού και τα δεδομένα που σχετίζονται με αυτή γίνεται μόνο στο front end. Η λογική html, css και typescript βρίσκονται στο όλα component *theme-switcher*. Όπως φαίνεται

στον **Πίνακα 5.24**, ο html κώδικας αποτελείται από 3 στοιχεία: ένα label, ένα input τύπου checkbox και ένα div. Ο σκοπός του κώδικα css (**Πίνακας 5.26**) είναι να μετατρέψει αυτόν τον κώδικα από ένα απλό κουτί ελέγχου που αλλάζει με ένα τικ, σε ένα διακόπτη εναλλαγής ο οποίος θα εμφανίζει ένα φεγγάρι όταν είναι ενεργό το σκοτεινό θέμα και έναν ήλιο όταν είναι ενεργό το φωτεινό θέμα. Τέλος, ο κώδικας typescript (**Πίνακας 5.25**) είναι αυτός ο οποίος εν αλλάζει το θέμα, με την μέθοδο switchTheme() η οποία καλείται κάθε φορά που πατιέται ο διακόπτης/κουμπί καθώς και κατά την αρχικοποίηση του component στην μέθοδο ngOnInit(). Σε αυτήν την μέθοδο δημιουργείτε ένας listener που ακούει κάθε αλλαγή του και εκεί αλλάζει το attribute data-theme και αποθηκεύει στην τοπική αποθήκευση του φυλλομετρητή την κατάλληλη τιμή του θέματος (light για φωτεινό, dark για σκοτεινό). Όταν αλλάζει το data-theme αλλάζει και ο κώδικας css που λαμβάνεται υπόψιν στο αρχικό css αρχείο style.css, όπως αναφέρθηκε στην εισαγωγή του παρόντος κεφαλαίου. Τέλος, γίνεται ο έλεγχος και η ανάκτηση του παρόντος θέματος, και σε περίπτωση που δεν έχει επισκεφθεί στο παρελθόν ο χρήστης την ιστοσελίδα, λαμβάνεται υπόψιν η προτίμηση του χρήστη βάσει του λειτουργικού συστήματός του. Το τελικό αποτέλεσμα εμφανίζεται στην **Εικόνα 5.4**.

Πίνακας 5.24: Κώδικας του αρχείου theme-switcher.component.html

```
<label class="theme-switch" for="checkbox">
  <input type="checkbox" id="checkbox" />
  <div class="slider round" (click)="switchTheme()"> ☾ ☀ </div>
</label>
```

Πίνακας 5.25: Η μέθοδος switchTheme() του αρχείου theme-switcher.component.ts

```
switchTheme() {
  const toggleSwitch = document.querySelector('.theme-switch input[type="checkbox"]') as
  HTMLInputElement;
  if (toggleSwitch === null) return;
  toggleSwitch.addEventListener('change', (e) => {
    if (e.target === null) return
    const target = e.target as HTMLInputElement;
    if (target.checked) {
      document.documentElement.setAttribute('data-theme', 'dark');
      localStorage.setItem('theme', 'dark');
    } else {
      document.documentElement.setAttribute('data-theme', 'light');
      localStorage.setItem('theme', 'light');
    }
  }, false);
  const currentTheme = localStorage.getItem('theme') ? localStorage.getItem('theme'):null;
  if (currentTheme) {
    document.documentElement.setAttribute('data-theme', currentTheme);
    if (currentTheme === 'dark') {
      toggleSwitch.checked = true;
    }
  } else { //no saved theme implying first use of page
    const matchMedia = window.matchMedia('(prefers-color-scheme: dark)');
    if (matchMedia.matches) { //is dark theme
      toggleSwitch.checked = true;
    }
  }
}
```

Πίνακας 5.26: Κώδικας του αρχείου theme-switcher.component.css

```
.theme-switch-wrapper {
  display: flex;
  align-items: center;
}

.theme-switch {
  display: inline-block;
  position: relative;
  height: 2rem;
  width: 4.5rem;
  margin-top: .25rem;
  padding-right: .2rem;
}

.theme-switch input {
  display: none;
}

.slider {
  background-color: var(--background-color);
  cursor: pointer;
  position: relative;
  transition: transform 400ms, outline-color 300ms;
  font-size: 2ch;
}

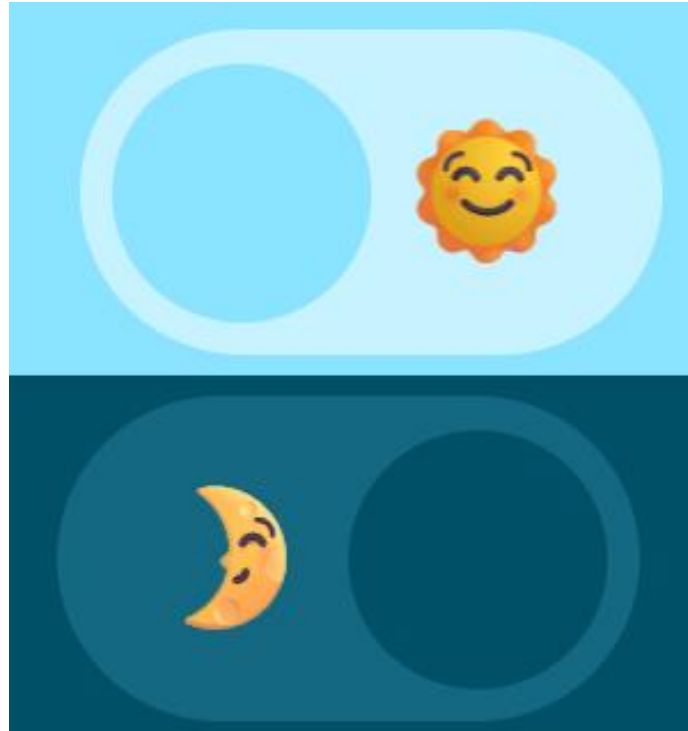
.slider:hover,
.slider:active {
  outline: .2rem solid var(--main-color);
}

.slider:before {
  justify-self: center;
  align-self: center;
  background-color: var(--background2-color);
  bottom: .25rem;
  content: "";
  left: .25rem;
  position: absolute;
  transition: .4s;
  height: 2rem;
  width: 2rem;
}

input:checked+.slider:before {
  transform: translateX(2rem);
}

.slider.round {
  display: flex;
  border-radius: 2.125rem;
  justify-items: center;
  justify-content: center;
  user-select: none;
  line-height: 2.5rem;
}

.slider.round:before {
  border-radius: 50%;
}
```



Εικόνα 5.4: Ο διακόπτης εναλλαγής θέματος, όταν είναι ανοιχτό το φωτεινό θέμα (πάνω) και όταν είναι ανοιχτό το σκοτεινό (κάτω)

5.10 Αναζήτηση αξιοθέατου

Ο χρήστης επιπλέον μπορεί να αναζητήσει οποιοδήποτε αξιοθέατο ή νομό, χρησιμοποιώντας το πεδίο αναζήτησης στην μπάρα μενού (**Εικόνα 6.15**). Εφόσον πληκτρολογήσει τουλάχιστον 3 χαρακτήρες, θα εμφανιστεί δίπλα στην μπάρα μενού μια λίστα με τα αξιοθέατα και τους νομούς που περιέχουν αυτό το λεκτικό. Αυτό επιτυγχάνεται με τον κώδικα που εμφανίζεται στον **Πίνακα 5.25**. Συγκεκριμένα, κάθε φορά που πληκτρολογεί ο χρήστης έναν χαρακτήρα, γίνεται μια αναζήτηση στον στατικό πίνακα της κλάσης, `SearchComponent.fullArr`, ο οποίος στον constructor έχει γεμίσει με όλα τα αξιοθέατα και νομούς από τα αρχεία `places.json` και `nomoi4.json` αντίστοιχα.

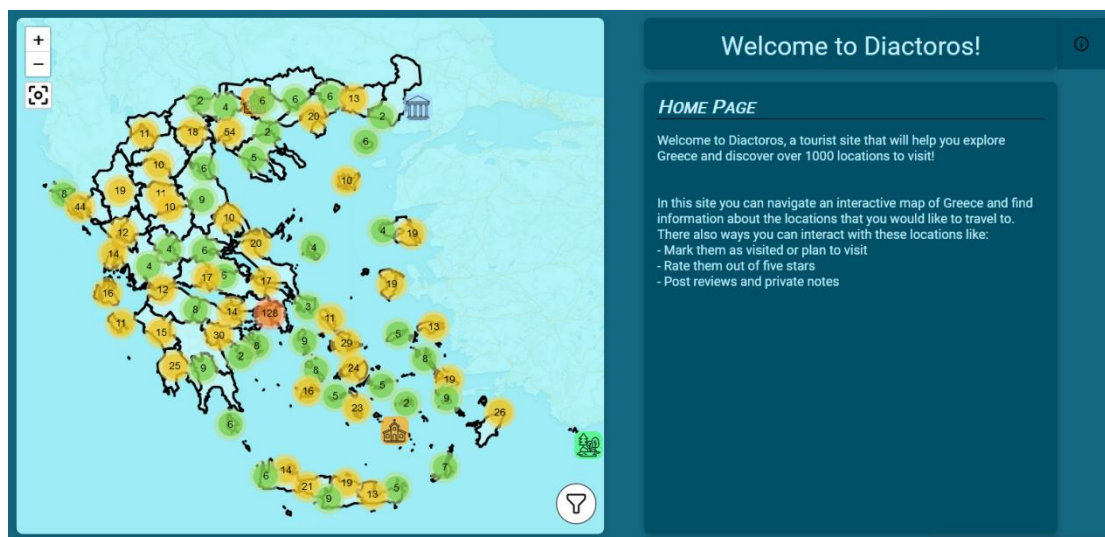
Πίνακας 5.27: Μέθοδος `makeList` που παράγει την λίστα αναζήτησης

```
mmakeList() {
  let curSearchArr: Place[] = []; // Close all previous lists
  SearchComponent.searchArrSubject.next([]);
  let inp = document.getElementById("search") as HTMLInputElement;
  if (inp) {
    let val: HTMLInputElement = inp;
    if (val.value.length < 3) return;
    //make search list
    for (let place of SearchComponent.fullArr) {
      if (place.name.toLowerCase().includes(val.value.toLowerCase())) {
        curSearchArr.push(place);
      }
    }
  }
  SearchComponent.searchArrSubject.next(curSearchArr);
}
```


Κεφάλαιο 6: Αποτέλεσμα και εγχειρίδιο χρήσης

Στο έβδομο κεφάλαιο της παρούσας διπλωματικής εργασίας θα γίνει συζητηθεί το αποτέλεσμα βάση των αρχικών στόχων όπως συζητήθηκαν στα **Κεφάλαια 1 & 2**. Επίσης θα δοθεί έναν εγχειρίδιο χρήσης της εφαρμογής, παραθέτοντας στιγμιότυπα της κάθε λειτουργίας σε δράση.

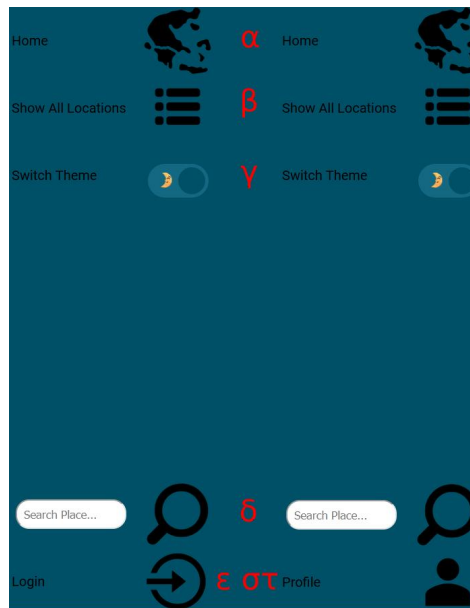
Όταν ένας χρήστης φτάσει στην σελίδα αρχικά καλωσορίζεται από την κύρια σελίδα (**Εικόνα 6.1**) η οποία έχει δύο τμήματα, τον διαδραστικό χάρτη και το πάνελ πληροφορίας, καθώς και την αναδυόμενη μπάρα μενού στο αριστερό μέρος της οθόνης η οποία βρίσκεται σε όλες τις σελίδες του ιστότοπου (**Εικόνα 6.2**). Στις παρακάτω υποενότητες εξηγείται ο τρόπος πλοήγησης όλων των υποσελίδων και τα περιεχόμενά τους.



Εικόνα 6.1: Κύρια Σελίδα.

6.1 Μπάρα μενού

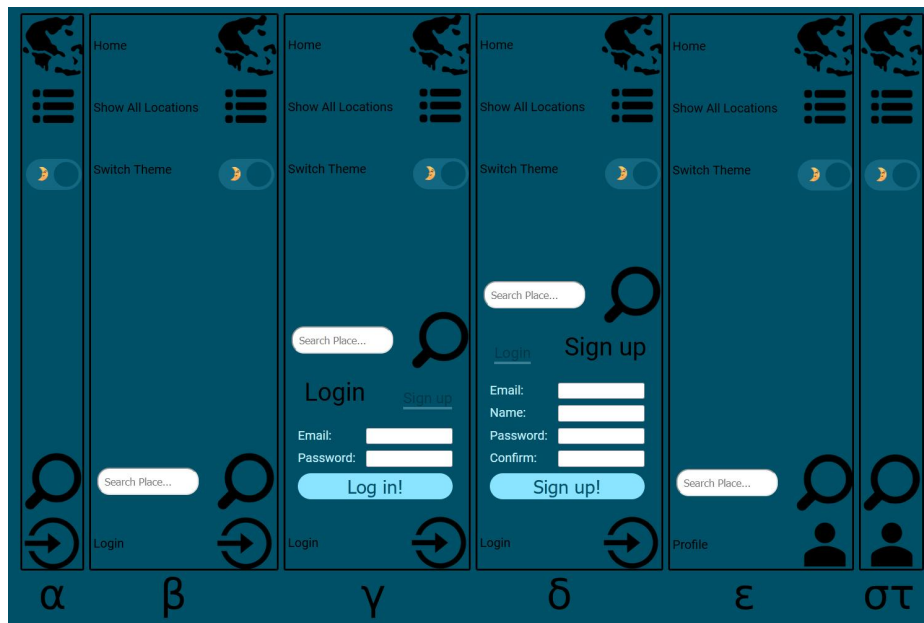
Η μπάρα μενού κατά κανόνα είναι ανασυρμένη και παρέχει μόνο οπτικές πληροφορίες για το κάθε κουμπί/λειτουργία της. Έπειτα από την τοποθέτηση του ποντικιού επάνω της, αναδύεται, εμφανίζοντας περισσότερες πληροφορίες για κάθε μία από τις λειτουργίες της. Συγκεκριμένα μέσω της μπάρας ο χρήστης μπορεί, όπως φαίνεται στην **Εικόνα 6.2**, να επισκεφθεί την (α) κύρια σελίδα και (β) την λίστα αξιοθέατων, (γ) να εναλλάξει το θέμα της ιστοσελίδας μεταξύ φωτεινού και σκοτεινού, (δ) να κάνει αναζήτηση των αξιοθέατων (ε) να δημιουργήσει λογαριασμό ή να συνδεθεί σε αυτόν, (στ) να επισκεφθεί την σελίδα χρήστη του. Στην **Εικόνα 6.3** φαίνονται οι έξι μορφές που μπορεί να πάρει η μπάρα: (α) ανασυρμένη & ο χρήστης αποσυνδεδεμένος (β, γ & δ) επεκταμένη, ο χρήστης αποσυνδεδεμένος και (β) χωρίς επιλογή, (γ) επιλογή του κουμπιού & καρτέλα εγγραφής και (δ) επιλογή του κουμπιού & καρτέλα σύνδεσης, (β) επεκταμένη & ο χρήστης συνδεδεμένος (στ) ανασυρμένη & ο χρήστης συνδεδεμένος.



Αποσυνδεδεμένος

Συνδεδεμένος

Εικόνα 6.2: Επεξήγηση λειτουργιών μπάρας μενού.





























Εικόνα 6.3: Η μπάρα μενού σε όλες της τις μορφές

6.2 Υποσελίδες

Η “κύρια σελίδα” περιέχει τον διαδραστικό χάρτη και το κουτί πληροφορίας. Το πάνελ πληροφορίας περιέχει αρχικά ένα μήνυμα καλωσορίσματος και μετέπειτα γεμίζει με πληροφορίες ανάλογα το αξιοθέατο ή τον νομό που έχει επιλέξει ο χρήστης μέσω της αλληλεπίδρασής του με τον διαδραστικό χάρτη. Ο χρήστης μπορεί να αυξήσει και να μειώσει την μεγέθυνση του χάρτη ανάλογα με τις ανάγκες του είτε με την χρήση των αντίστοιχων κουμπιών πάνω αριστερά στον χάρτη είτε με την ροδέλα του ποντικιού. Με το κουμπί κάτω από τα δύο κουμπιά ελέγχου μεγέθυνσης μπορεί αν επαναφέρει τον χάρτη στην αρχική του θέση και μεγέθυνση.

Επίσης με το κουμπί φίλτρου στην κάτω δεξιά πλευρά του χάρτη μπορεί να επιλέξει τι είδους αξιοθέατα εμφανίζονται σε αυτόν.

Η υποσελίδα “λίστα αξιοθέατων” έχει δύο τμήματα, την λίστα και το πάνελ πληροφορίας. Η λίστα αξιοθέατων περιέχει όλα τα αξιοθέατα της ιστοσελίδας με συνοπτικές πληροφορίες για αυτά όπως φαίνεται στην **Εικόνα 6.4**. Με την επιλογή οποιασδήποτε εγγραφής στην λίστα, περισσότερες πληροφορίες για το αξιοθέατο φορτώνουν στο πάνελ πληροφορίας στα δεξιά.

Type	Location	Name	Has Descr	Has Images
 Landmark	Achaea	Ancient Theater of Aigeira	No	No
 Natural Sight	Achaea	Bouloumbasi Waterfalls	No	Yes
 Monastery	Achaea	Holy Trinity Orthodox Convent of Akrata	No	No
 Zoo-Aquarium	Achaea	Kalavrita Zoo	No	Yes
 Natural Sight	Achaea	Ladon Springs	No	Yes
 Natural Sight	Achaea	Peristeriona Gorge	No	No
 Museum	Aegina	Archaeological Museum of Aegina	Yes	Yes
 Museum	Aegina	Archaeological Museum of Aphaia	No	Yes
 Landmark	Aegina	Archaeological Site of Palaiochora of Aegina	No	Yes
 Museum	Aegina	Christos Kapralos Museum	No	Yes
 Museum	Aegina	Historical and Folklore Museum of Aegina	No	Yes
 Landmark	Aegina	Temple of Aphaia	Yes	Yes
 Landmark	Aegina	Temple of Apollo	No	Yes
 Landmark	Aegina	Temple of the Greek Zeus	No	Yes
 Natural Sight	Aegina	The Ancient Olive Grove (Elaionas)	No	Yes
 Monastery	Aegina	Virgin Mary Chrysoleontissa Holy Convent	No	Yes
 Museum	Aetolia-Acarmania	Archaeological Museum of Thyreio	No	Yes
 Landmark	Aetolia-Acarmania	Stratos Archaeological Site - Ancient Theater	No	Yes
 Museum	Agrinio	Aetolia-Acarmania Folklore Museum	Yes	Yes
 Museum	Agrinio	Papastratos Archaeological Museum	No	Yes
 Museum	Agrinio	Railway Museum of Agrinio	No	Yes
 Museum	Alonnisos	Alonnisos Museum "Kostas & Angela Mavrikis"	No	Yes
 Museum	Alonnisos	Knowledge Awareness Center	No	Yes
 Landmark	Amorgos	Ancient Arkesini	No	Yes
 Museum	Amorgos	Archaeological Collection of Amorgos	No	Yes
 Landmark	Amorgos	Castle of Amorgos	No	Yes

Εικόνα 6.4: Η λίστα των αξιοθέατων

Η “σελίδα χρήστη” περιέχει επίσης δύο τμήματα, τις πληροφορίες χρήστη και το πάνελ πληροφορίας. Στις πληροφορίες χρήστη περιέχονται τα δεδομένα που εισήγαγε κατά την εγγραφή του ο χρήστης (όνομα χρήστη, email), στατιστικά του χρήστη και λίστες με όλα τα αξιοθέατα τα οποία έχει σχέση επίσκεψης ή επιθυμίας επίσκεψης, που έχει σημειώσει ως αγαπημένα, που θα ήθελε μελλοντικά να επισκεφθεί και που του προτείνονται από το σύστημα βάση αυτών που έχει ήδη επισκεφθεί εκείνος και άλλοι χρήστες. Πληροφορίες ενός χρήστη είναι της μορφής που φαίνονται στην **Εικόνα 6.5**. Τέλος όπως φαίνεται στην **Εικόνα 6.5**, ο χρήστης μπορεί να αποσυνδεθεί πατώντας το κουμπί “Logout”.

Giorgos

[Logout](#)

Email: test@gmail.com
Last Login: 2024-05-08 09:25:03

📍 Visited : 20
 🕒 Wishlist : 2
 ❤️ Favorites : 8
 ★ Average rating : 3.52 (3 not rated)

[All Places](#) | [Favorite Places](#) | [Wishlist Places](#) | [Recommended Places](#)

Type	Location	Name	Rating	Favorite
🏛️ Landmark	Arachova	Clock Tower of Arachova	★★★★★	❤️
🏛️ Museum	Chania	Maritime Museum of Crete	★★★★★	❤️
🏛️ Landmark	Ioannina	Ich Kale Acropolis of Ioannina	★★★★★	❤️
🏛️ Museum	East Attica	Museum of Military School "Evelpidon"	★★★★★	❤️
🏛️ Landmark	Athens	Acropolis of Athens	★★★★★	❤️
🐠 Zoo-Aquarium	East Attica	Attica Zoological Park	★★★★★	❤️
🏛️ Museum	Athens	Jewish Museum of Greece	★★★★★	❤️
🌿 Natural Sight	Aegina	The Ancient Olive Grove (Elaionas)	★★★★★	❤️
🏛️ Museum	Aegina	Christos Kapralos Museum	★★★★★	❤️
🏛️ Landmark	Arachova	Maroni Bridge	★★★★★	❤️

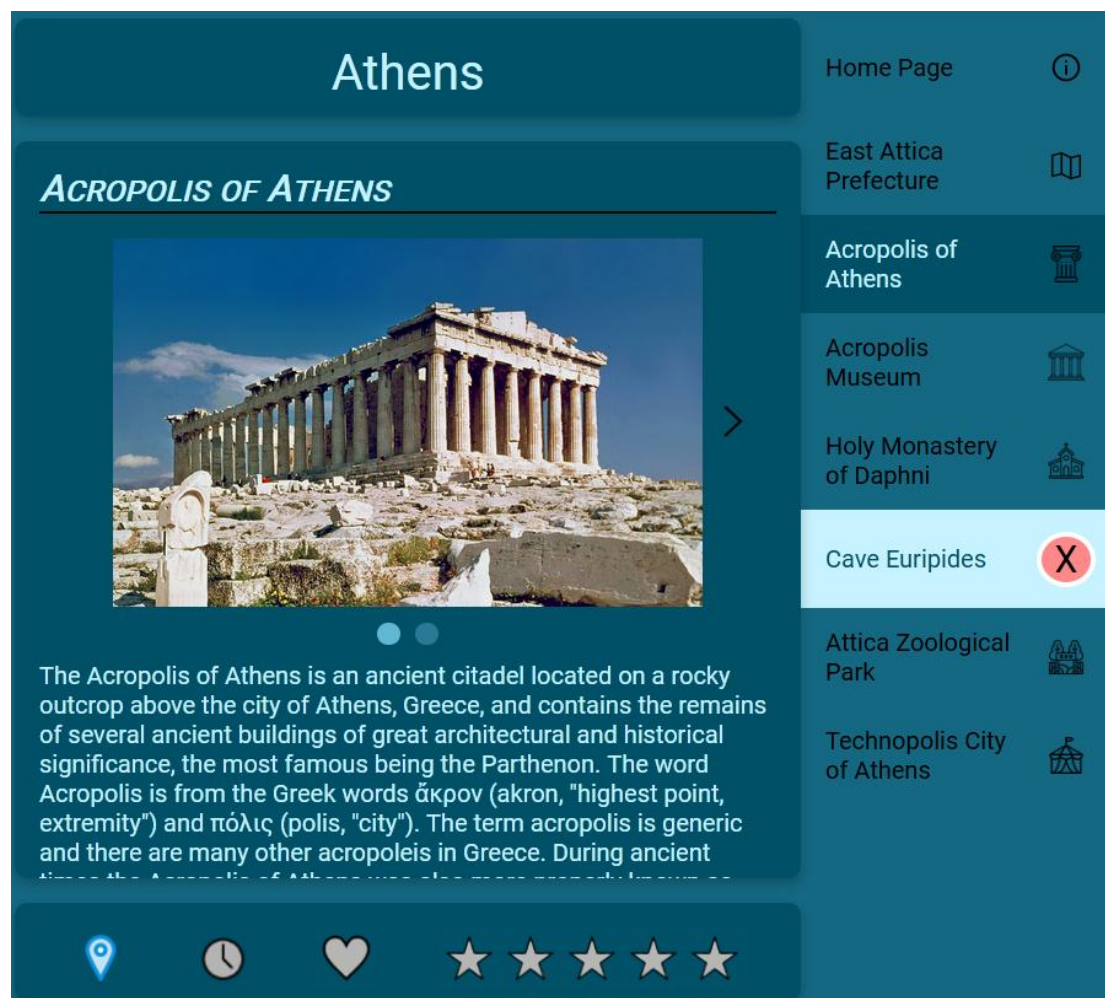
Εικόνα 6.5: Πληροφορίες Χρήστη

6.3 Πάνελ πληροφορίας

Το πάνελ πληροφορίας εμφανίζεται σε όλες τις υποσελίδες. Στην “κύρια σελίδα” εμφανίζεται μέσω της επιλογής αξιοθέατου ή περιφέρειας στον διαδραστικού χάρτη, στην “λίστα αξιοθέατων” μέσω της επιλογής κάποιας εγγραφής του πίνακα και στην “σελίδα χρήστη” μέσω της επιλογής κάποιας εγγραφής σε μία εκ των τριών λιστών της σελίδας. Επίσης, με την χρήση της αναζήτησης αξιοθέατου, το αποτέλεσμα μπορεί να εμφανιστεί στο κουτί πληροφορίας της πλέον ενεργής υποσελίδας, οποιαδήποτε και να είναι εκείνη. Στην **Εικόνα 6.6** φαίνεται ένα στιγμιότυπο μιας κανονικής χρήσης της ιστοσελίδας με το πάνελ πληροφορίες να περιέχει 7 καρτέλες. Οι επτά καρτέλες είναι οι παρακάτω.

Αρχικά η πρώτη καρτέλα “Home Page”, είναι η καρτέλα με τις πληροφορίες της υποσελίδας, η οποία στην προκειμένη περίπτωση είναι η κύρια σελίδα. Η δεύτερη καρτέλα “East Attica Prefecture” περιέχει πληροφορίες για τον νομό ανατολικής αττικής. Η επόμενες έξι καρτέλες περιέχουν πληροφορίες για αξιοθέατα, με κάθε κύρια κατηγορία αξιοθέατου να εμφανίζεται με το αντίστοιχα κατάλληλο εικονίδιο όπως αναφέρθηκε στο **Κεφάλαιο 5.2**, με την τρίτη και την έκτη να έχουν διαφορετικού χρώματος φόντο. Όσον αφορά την τρίτη καρτέλα “Acropolis of Athens” το σκοτεινό φόντο της υποδεικνύει ότι είναι η ενεργή καρτέλα και όπως βλέπουμε στο πάνελ της εμφανίζονται πληροφορίες για την Ακρόπολη. Όσον αφορά την έκτη καρτέλα “Cave of Euripides”, η καρτέλα έχει ανοιχτό φόντο το οποίο υποδεικνύει ότι ο χρήστης έχει τοποθετήσει τον κέρσορα πάνω σε αυτήν. Έτσι,

στην θέση του εικονιδίου εμφανίζεται ένα κόκκινο “X”, το οποίο αν πατηθεί από τον χρήστη θα κλείσει την καρτέλα.



Εικόνα 6.6: Το πάνελ πληροφορίας.

Αν ο χρήστης θέλει να αλληλεπιδράσει με το αξιοθέατο μπορεί να πατήσει οποιοδήποτε από τα κουμπιά που φαίνονται στην **Εικόνα 6.6** για να δηλώσει ότι έχει επισκεφθεί το αξιοθέατο, ότι επιθυμεί να το επισκεφθεί στο μέλλον, να το χαρακτηρίσει ως αγαπημένο, ή να το βαθμολογήσει από 1 μέχρι 5 αστέρια.

6.4 Σύνδεση και εγγραφή χρήστη

Όπως φαίνεται στην **Εικόνα 6.3 (γ) & (δ)**, ο χρήστης μέσω της μπάρας μενού μπορεί να συνδεθεί παρέχοντας το email του και τον κωδικό του και να κάνει εγγραφή στην ιστοσελίδα παρέχοντας το όνομα χρήστη του, email κωδικό και επιβεβαίωση κωδικού.

Κεφάλαιο 7: Συμπέρασμα και μελλοντικές επεκτάσεις

Η εφαρμογή επέτυχε το αρχικό στόχο να συγκεντρώσει υλικό για τα πολυπληθή αξιοθέατα της Ελλάδος και να το εμφανίσει σε έναν ευδιάκριτο, διαδραστικό χάρτη της, διαχωρίζοντας τα ανά κατηγορίες και τύπους. Επίσης κατάφερε να είναι ένας εύκολος τρόπος καταγραφής και βαθμολογίας των αξιοθέατων τα οποία έχει επισκεφθεί ο κάθε χρήστης καθώς και να τον βοηθήσει να ανακαλύψει νέα ενδιαφέρον αξιοθέατα με την χρήση συστήματος προτάσεων.

Συγκεκριμένα, ο χρήστης θα μπορεί σε ελάχιστο χρόνο να δημιουργήσει έναν λογαριασμό και να εξερευνήσει και να μάθει περισσότερες πληροφορίες για τα αξιοθέατα της Ελλάδος γρήγορα και αποτελεσματικά, καθώς και να κρατήσει ιδιωτικές σημειώσεις για το αξιοθέατο αυτό ή να αφήσει δημόσια κριτική. Επίσης αν δεν τον ενδιαφέρει ο διαδραστικός χάρτης, μπορεί να χρησιμοποιήσει την απλή λίστα με τα αξιοθέατα. Τέλος στην προσωπική του σελίδα χρήστη μπορεί να δει διάφορα στατιστικά για τα αξιοθέατά του, καθώς και λίστες με όλα τα αξιοθέατα που έχει επισκεφθεί, που έχει χαρακτηρίσει ως αγαπημένα, και που του έχουν προταθεί να επισκεφθεί από την εφαρμογή.

Άλλες λειτουργικότητες οι οποίες ήταν εκτός του αρχικού πεδίου εφαρμογής και μελλοντικά θα μπορούσαν να εφαρμοστούν για να επεκταθεί η χρησιμότητά της είναι οι παρακάτω:

7.1 Περισσότερες γλώσσες διεπαφής

Έχοντας ως αρχικό στόχο την εξυπηρέτηση των τουριστών, η εφαρμογή δημιουργήθηκε στην πιο διαδεδομένη γλώσσα του κόσμου: τα αγγλικά. Όμως, η λειτουργικότητα της εφαρμογής δεν σταματάει σου τουρίστες από εξωτερικό. Πέρα από ημεδαπούς τουρίστες, οι οποίοι το 2023 ανέρχονταν στο 24% των συνολικών τουριστών⁹, η εφαρμογή μπορεί να χρησιμεύσει και στους απλούς κατοίκους των εκάστοτε περιοχών της Ελλάδος. Για παράδειγμα, στην βάση δεδομένων της εφαρμογής υπάρχουν 48 αξιοθέατα στην Αθήνα και άλλα 41 αξιοθέατα στην Θεσσαλονίκη, οι οποίες είναι οι δύο πιο πολυπληθής πόλεις της Ελλάδος. Συνεπώς, η προσθήκη της ελληνικής γλώσσας θα έπρεπε να είναι από τις πρώτες προτεραιότητες.

Πέρα από την προσθήκη της ελληνικής, η γερμανική ως δεύτερη πιο πολυπληθής ομιλούμενη γλώσσα αλλοδαπών τουριστών¹⁰ θα ήταν επίσης μία εύλογη προσθήκη, και κατά συνέχεια τα γαλλικά, ολλανδικά, ιταλικά και άλλα.

⁹ https://www.statistics.gr/documents/20181/18165299/TOURISMOS_2023.pdf/f0903d6f-e756-fa80-f847-1b591249fb20 σελίδα 1, ανακτήθηκε Φεβρουάριο 2024

¹⁰ https://www.statistics.gr/documents/20181/18165299/TOURISMOS_2023.pdf/f0903d6f-e756-fa80-f847-1b591249fb20 σελίδα 3, ανακτήθηκε Φεβρουάριο 2024

7.2 Περισσότερες πληροφορίες

Παρά την εξονυχιστική προσπάθεια για εύρεση αξιοθέατων, είναι αρκετά πιθανών να έχουν παραλειφθεί σημαντικά ιστορικά αλλά και διάσημα αξιοθέατα, καθώς η εύρεση αυτή έγινε κατά κύριο λόγο χειροκίνητα. Για τον ίδιο λόγο και η καταγραφή των υποκατηγοριών και της περιγραφής, κάθε ενός από τα πάνω από χίλια αξιοθέατα μπορεί να χαρακτηριστεί ελλιπής. Για παράδειγμα, θα μπορούσαν να προστεθούν επιπλέον κατάλληλες υποκατηγορίες στα ήδη υπάρχοντα αξιοθέατα καθώς και να επεκταθούν οι υπάρχουσες υποκατηγορίες όπως τα πάρκα (Park), τα τεμένη (Mosque), αγάλματα (Statue). Πέρα αυτού, θα μπορούσαν να προστεθούν και επιπλέον κατηγορίες και υποκατηγορίες αξιοθέατων όπως παραδείγματος χάρη: χώροι κατασκήνωσης, γραφικές θέες, σημεία για κολύμπι και καταδύσεις, ορειβατικά μονοπάτια, καζίνο, περιοδικές εκδηλώσεις, χιονοδρομικά κέντρα. Καθώς η βασική πηγή πληροφοριών για τα αξιοθέατα ήταν η βικιπαίδεια και εκείνη δεν είχε αρκετές πληροφορίες για την πλειοψηφία τους, θα μπορούσαν να συλλεχθούν πληροφορίες από άλλες πηγές όπως βιβλία, τοπικές ιστοσελίδες ή τουριστικών φυλλαδίων. Όπως αναφέρθηκε και στο **Υποκεφάλαιο 2.1.4**, οι τοπικές ιστοσελίδες παρέχουν αρκετή και χρήσιμη πληροφορία για τα αξιοθέατα του νομού και περιφέρειάς τους. Ακόμη, θα μπορούσαν να προστεθούν επιπλέον χρήσιμοι σύνδεσμοι, όπως πχ οι επίσημες ιστοσελίδες της ελληνικής κυβέρνησης για το εκάστοτε αξιοθέατο, κυρίως των μουσείων.

7.3 Δυνατότητα συζήτησης και φιλτράρισμα σχολίων και χρηστών

Μία επέκταση στην λειτουργικότητα των κριτικών σε κάθε αξιοθέατο θα ήταν η βαθμολόγηση τους καθώς και η δυνατότητα απάντησης και δημιουργίας νήματος συζήτησης. Ακόμη, στην παρούσα υλοποίηση, καθώς δεν υπάρχει εκλεπτυσμένο φιλτράρισμα των κριτικών, οι χρήστες μπορούν να γράψουν οτιδήποτε θέλουν. Σε οποιαδήποτε εφαρμογή στην οποία επιτρέπεται η συγγραφή δημόσιων σχολίων θα πρέπει να υπάρχει και τρόπος φιλτραρίσματος ανεπιθύμητων εκφράσεων και φραγμός χρηστών για αποκλεισμό από την ροή ενός χρήστη εφόσον αυτός δεν επιθυμεί να βλέπει το περιεχόμενο του συγκεκριμένου χρήστη για οποιοδήποτε λόγο, καθώς και αναφορά για χυδαίο περιεχόμενο έτσι ώστε οι διαχειριστές να μπορούν να λάβουν την απαραίτητη δράση.

7.4 Εφαρμογή κινητού και καλύτερη ανταποκρισιμότητα σε κάθετες οθόνες

Πλέον οι περισσότεροι προγραμματιστές κατά την ανάπτυξη των διαδικτυακών εφαρμογών τους λαμβάνουν υπ' όψιν το γεγονός ότι πολλοί αν όχι οι περισσότεροι χρήστες θα χρησιμοποιήσουν την ιστοσελίδα τους από το κινητό τους. Έτσι λαμβάνουν και τα αντίστοιχα μέτρα για να έχει ανταποκρισιμότητα (responsiveness) σε κάθετες οθόνες. Επιπλέον, μια stand alone εφαρμογή κινητού στα android ή/και iOS οικοσυστήματα είναι ένας επιπλέον τρόπος να επεκταθεί η βάση χρηστών. Επίσης, με εφαρμογή σε κινητή συσκευή μας επιτρέπει να εκμεταλλευτούμε την χρήση του GPS για την

Ανάπτυξη τουριστικής εφαρμογής/ιστοσελίδας

υλοποίηση αυτόματης επίσκεψης αξιοθέατων αν ο χρήστης βρίσκεται κοντά τους.

Πηγές

Για την εκπόνηση του παρών εγγράφου

- en.wikipedia.org
- statistics.gr
- visitgreece.com
- allovergreece.com

Για την εκπόνηση της εφαρμογής

- github.com/fastily/jwiki, (Ανακτήθηκε Ιανουάριο 2024)
- en.wikipedia.org
- angular.io/docs
- stackoverflow.com
- leafletjs.com/reference.html
- baeldung.com
- openmaptiles.org/styles,
- <https://geodata.gov.gr/el/dataset/oria-nomon-okkhe>, για την ανάκτηση του χάρτη νομών της Ελλάδος, ανακτήθηκε Σεπτέμβριο 2023
- Για την εκμάθηση του λογισμικού QGIS παρακολουθήθηκε μέρος της σειράς μαθημάτων MapAcademy στο Udemy, <https://www.udemy.com/course/mapacademy/>, (ανακτήθηκε Φεβρουάριο 2024)