



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**Ιστότοπος περιήγησης πόλης με χρήση της MongoDB
για διαχείριση δεδομένων και γεωχωρικών ερωτημάτων**

Διπλωματική Εργασία

Κυριακή Κόφφα

Επιβλέπουσα: Ελένη Τουσίδου

Σεπτέμβριος 2025



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

**Ιστότοπος περιήγησης πόλης με χρήση της MongoDB
για διαχείριση δεδομένων και γεωχωρικών ερωτημάτων**

Διπλωματική Εργασία

Κυριακή Κόφφα

Επιβλέπουσα: Ελένη Τουσίδου

Σεπτέμβριος 2025



UNIVERSITY OF THESSALY
SCHOOL OF ENGINEERING
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

**City navigation website using MongoDB
for data management and geospatial querying**

Diploma Thesis

Kyriaki Koffa

Supervisor: Eleni Tousidou

September 2025

Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπουσα Ελένη Τουσίδου

Ε.ΔΙ.Π., Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος Βασιλακόπουλος Μιχαήλ

Καθηγητής, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος Δασκαλοπούλου Ασπασία

Αναπληρώτρια Καθηγήτρια, Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Θα ήθελα να ευχαριστήσω πρωτίστως την επιβλεπούσα καθηγήτρια Ελένη Τουσίδου, στην οποία είμαι ευγνώμων για την υποστήριξη, κατανόηση και υπομονή που μου έχει δείξει κατά τη διάρκεια της συνεργασίας μας. Θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου που στάθηκαν στο πλευρό μου καθόλη τη διάρκεια των σπουδών μου και ειδικά τον αδερφό μου Γεώργιο Κόφφα, χωρίς τον οποίο δε νομίζω να είχα φτάσει στη θέση στην οποία βρίσκομαι τώρα. Τέλος θα ήθελα να ευχαριστήσω γενικά το προσωπικό και τους φοιτητές του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Θεσσαλίας που με βοήθησαν να βρω ότι το πραγματικό μου πάθος είναι η τέχνη.

ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

«Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ρητά ότι η παρούσα διπλωματική εργασία, καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας, αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή/και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή. Δηλώνω επίσης ότι τα αποτελέσματα της εργασίας δεν έχουν χρησιμοποιηθεί για την απόκτηση άλλου πτυχίου. Αναλαμβάνω πλήρως, ατομικά και προσωπικά, όλες τις νομικές και διοικητικές συνέπειες που δύναται να προκύψουν στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής».

Η Δηλούσα

Κυριακή Κόφφα

Διπλωματική Εργασία

Ιστότοπος περιήγησης πόλης με χρήση της MongoDB για διαχείριση δεδομένων και γεωχωρικών ερωτημάτων

Κυριακή Κόφφα

Περίληψη

Με την ολοένα και αυξανόμενη χρήση των γεωχωρικών δεδομένων και των Συστημάτων Γεωγραφικών Πληροφοριών (Geographical Information Systems - GIS) για διάφορους σκοπούς, από τον πολεοδομικό σχεδιασμό μέχρι και την υγειονομική περίθαλψη, στην παρούσα εργασία επιχειρούμε να εστιάσουμε στην αναζήτηση και χαρτογράφηση διαδρομής βάσει των κυριότερων σημείων ενδιαφέροντος (ΣΕ) σε μία πόλη για την ευκολότερη εξερεύνησή της. Για αυτό τον σκοπό υλοποιήσαμε μια διαδικτυακή εφαρμογή, όπου δίνεται η δυνατότητα στον χρήστη να αναζητήσει τα ενδιαφέροντα σημεία μίας περιοχής και, παράλληλα, να βρει μια διαδρομή την οποία μπορεί να ακολουθήσει προκειμένου να περάσει από τα σημεία που έχει επιλέξει. Επικεντρωθήκαμε στην πόλη του Βόλου, όπου, χρησιμοποιώντας γεωχωρικά δεδομένα του OpenStreetMap, στήσαμε μια MongoDB βάση δεδομένων στην οποία μπορούν να εκτελεστούν γεωχωρικά ερωτήματα για την ικανοποίηση των λειτουργικών απαιτήσεων. Επιπλέον, αναπτύξαμε ευριστικό αλγόριθμο για την εύρεση μονοπατιού στον γράφο του οδικού δικτύου και των πεζοδρόμων μιας πόλης, το οποίο περνάει από συγκεκριμένα σημεία της πόλης, ελαχιστοποιώντας σε σημαντικό βαθμό την απόσταση που πρέπει να διανυθεί.

Λέξεις-κλειδιά:

Σημεία Ενδιαφέροντος (ΣΕ), Συστήματα Γεωγραφικών Πληροφοριών (GIS), Χαρτογράφηση Διαδρομής, Πρόβλημα του Πλανώδιου Πωλητή (TSP), Αλγόριθμος Dijkstra, NoSQL, MongoDB, OpenStreetMap

Diploma Thesis
City navigation website using MongoDB
for data management and geospatial querying
Kyriaki Koffa

Abstract

With the ever increasing use of geospatial data and Geographical Information Systems (GIS) for various purposes, from urban planning to providing healthcare, we are attempting to take an approach that focuses on searching and route mapping based on Points of Interests (POI) in a city for easier exploration. For this purpose we built a web application which helps the user to search an area's points of interest, and, at the same time, to find a route that they can follow in order to pass through the points they have chosen. We focused on the city of Volos where, using geospatial data provided by OpenStreetMap, we set up a MongoDB database with geospatial querying capabilities to make the functions of the website possible. In addition, we developed a heuristic algorithm for finding a path in the graph of a city's road network and sidewalks, which passes through specific points in the city, significantly minimizing the distance that must be traveled.

Keywords:

Points of Interest, Geographical Information Systems (GIS), Route Mapping, Traveling Salesman Problem (TSP), Dijkstra algorithm, NoSQL, MongoDB, OpenStreetMap

Πίνακας περιεχομένων

Ευχαριστίες	ix
Περίληψη	xii
Abstract	xiii
Πίνακας περιεχομένων	xv
Κατάλογος σχημάτων	xvii
1 Εισαγωγή	1
1.1 Αντικείμενο της διπλωματικής	1
1.1.1 Συνεισφορά	2
1.2 Οργάνωση του τόμου	2
2 Σχετικές Εφαρμογές/Εργασίες	5
2.1 Εισαγωγή	5
2.2 Σχετικές εφαρμογές και ιστότοποι	5
3 Εργαλεία	7
3.1 Εισαγωγή	7
3.2 OpenStreetMap και OSMnx	7
3.3 MongoDB	8
3.4 Backend Κώδικας	9
3.5 Front-End	10
4 Βάση Δεδομένων και Αλγόριθμοι	11
4.1 Εισαγωγή	11

4.2	Συλλογή Δεδομένων	11
4.3	Βάση Δεδομένων	13
4.3.1	Μορφή Συλλογών	14
4.3.2	Γεωχωρικά ερωτήματα στη βάση	19
4.4	Εύρεση διαδρομής	22
5	Εμφάνιση και Λειτουργίες Εφαρμογής	29
5.1	Εισαγωγή	29
5.2	Αρχική Σελίδα	29
5.3	Log In/ Register/ Log Out	30
5.4	Search in Radius	33
5.4.1	Σχόλια	36
5.4.2	Διαχειριστής και Ενημέρωση Σημείου Ενδιαφέροντος	38
5.5	Search in Path	41
6	Συμπεράσματα	47
6.1	Σύνοψη και συμπεράσματα	47
6.2	Μελλοντικές επεκτάσεις	48
	Βιβλιογραφία	49

Κατάλογος σχημάτων

4.1	Διαδικασία συλλογής και προεπεξεργασίας δεδομένων	12
4.2	Μετασχηματισμός δεδομένων ΣΕ	13
4.3	Εύρεση επιμέρους μονοπατιών	25
5.1	Αρχική σελίδα	30
5.2	Σελίδα σύνδεσης σε λογαριασμό	31
5.3	Μήνυμα αποτυχίας	31
5.4	Μήνυμα επιτυχίας στην σύνδεση λογαριασμού	31
5.5	Σελίδα δημιουργίας λογαριασμού	31
5.6	Σημάνσεις λάθους στη φόρμα εγγραφής	32
5.7	Προειδοποίηση αν το όνομα χρήστη χρησιμοποιείται ήδη	32
5.8	Επιτυχής εγγραφή	32
5.9	Επιτυχής αποσύνδεση	32
5.10	Σελίδα αναζήτησης σε ακτίνα “Search in Radius”	33
5.11	Φίλτρο ειδών ΣΕ	34
5.12	Η σελίδα αναζήτησης σε ακτίνα σε χρήση	34
5.13	Σελίδα αποτελέσματος αναζήτησης σε ακτίνα	35
5.14	Παράθυρο με τις πληροφορίες του Σημείου Ενδιαφέροντος	35
5.15	Σχόλια Σημείου Ενδιαφέροντος	36
5.16	Πλαίσιο σχολίων σε μη παρουσία σχολίων	36
5.17	Πλαίσιο υποβολής καινούργιου σχολίου	36
5.18	Παράδειγμα υποβολής σχολίου	37
5.19	Συνέχεια παραδείγματος υποβολής σχολίου	37
5.20	Κουμπιά στη σελίδα αποτελέσματος αναζήτησης σε ακτίνα ως διαχειριστής	38
5.21	Σελίδα για προσθήκη ή ενημέρωση Σημείου Ενδιαφέροντος	38

5.22 Σελίδα ενημέρωσης σε χρήση	39
5.23 Πλαίσια ενημέρωσης πληροφοριών ΣΕ	40
5.24 Παράθυρο στοιχείων για ενημέρωση	40
5.25 Προσθήκη νέου ΣΕ	41
5.26 Πλαίσιο ενημέρωσης ΣΕ	41
5.27 Πλαίσιο ενημέρωσης ΣΕ σε χρήση	41
5.28 Μήνυμα επιτυχίας προσθήκης ή ενημέρωσης Σημείου Ενδιαφέροντος	41
5.29 Σελίδα αναζήτησης διαδρομής “Search in Path”	42
5.30 Σελίδα αναζήτησης διαδρομής σε χρήση	43
5.31 Σελίδα αναμονής υπολογισμού διαδρομής	43
5.32 Αποτέλεσμα αναζήτησης διαδρομής	44
5.33 Παράθυρο πληροφοριών απόστασης και χρόνου	45

Κεφάλαιο 1

Εισαγωγή

Με την τεχνολογία να παίζει σημαντικό ρόλο σε ολοένα και μεγαλύτερο κομμάτι της καθημερινότητάς μας και το πλήθος των διαθέσιμων δεδομένων να αυξάνεται συνεχώς, παρατηρείται, μεταξύ άλλων, αυξανόμενο ενδιαφέρον για την αξιοποίηση γεωχωρικών δεδομένων από διαφορετικές βιομηχανίες και υπηρεσίες [1]. Πεδία που έχουν επωφεληθεί από αυτήν την αξιοποίηση είναι ο τουρισμός, ο δημόσιος τομέας, η διαχείρηση πληροφοριών, ο πολεοδομικός σχεδιασμός και η παροχή περίθαλψης [2, 3, 4, 5], ενώ ταυτόχρονα αρκετές υπηρεσίες για ατομική μετακίνηση ή μαζική μεταφορά έχουν κάνει ευκολότερη τη ζωή των πολιτών μέσω της χρήσης ψηφιακών χαρτών και του παγκόσμιου συστήματος εντοπισμού θέσης (Global Positioning System - GPS). Παρά το γεγονός ότι εφαρμογές που έχουν υλοποιηθεί για τις ανάγκες των παραπάνω βιομηχανιών και υπηρεσιών εκτελούν σε ικανοποιητικό βαθμό τον στόχο της εύρεσης διαδρομής ανάμεσα σε δύο σημεία, δεν έχουν συνήθως ως επίκεντρο την εξερεύνηση των τοπικών πολιτισμικών πόλων έλξης και χώρων αναψυχής. Σε αυτό το πλαίσιο και έχοντας αναγνωρίσει την έλλειψη μίας ανάλογης λύσης, προχωρήσαμε στην υλοποίηση του θέματος της συγκεκριμένης διπλωματικής εργασίας.

1.1 Αντικείμενο της διπλωματικής

Ο κύριος στόχος της παρούσας διπλωματικής εργασίας είναι η παραγωγή λογισμικού που ενθαρρύνει και διευκολύνει την εξερεύνηση μιας πόλης σε άτομα με λίγη έως και μηδενική οικειότητα με την πόλη αυτήν. Η ανθρωποκεντρική προσέγγισή μας, σε αντίθεση με μία προσέγγιση που επικεντρώνεται μόνο στο αποτέλεσμα, για παράδειγμα στην απλή εύρεση μίας διαδρομής, ενθαρρύνει τη φυσική άσκηση μέσα από το περπάτημα, επιδεικνύει τα πολιτι-

σημεία καθώς και τα σημεία αναψυχής μιας πόλης, και έμμεσα ενισχύει τον τουρισμό. Η μεθοδολογία που χρησιμοποιήσαμε για να απευθυνθούμε σε αυτές τις ανάγκες ήταν η δημιουργία μίας εφαρμογής ιστοτόπου ανοιχτής πρόσβασης η οποία μέσω των γεωχωρικών δεδομένων μίας πόλης δίνει στους χρήστες τη δυνατότητα να αναζητήσουν σημεία ενδιαφέροντος σε μία περιοχή, να δημιουργήσουν διαδρομές που προσπελαύνουν δεδομένα σημεία ενδιαφέροντος και να προσθέσουν πληροφορίες σε αυτά. Επιλέξαμε ως μελέτη περίπτωσης της εφαρμογής μας την πόλη του Βόλου.

1.1.1 Συνεισφορά

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. MongoDB για αποθήκευση και επεξεργασία γεωχωρικών δεδομένων: Η MongoDB είναι μία ευρέως χρησιμοποιούμενη NoSQL βάση δεδομένων. Παρέχει και τη λιγότερο γνωστή δυνατότητα για γεωχωρικά ερωτήματα σε γεωχωρικού τύπου δεδομένα, που αποτελεί και τον λόγο που την επιλέξαμε.
2. Ευριστικός αλγόριθμος εύρεσης μονοπατιού που περνάει από συγκεκριμένα σημεία: Για τη δημιουργία διαδρομής που περνάει από όλα τα σημεία ενδιαφέροντος που ζητάει ο χρήστης, χρειάστηκε να βρεθεί προσεγγιστική λύση σε πρόβλημα που ανάγεται στο Πρόβλημα Περιοδεύοντος Πωλητή.
3. Χρήση OpenStreetMap API: Η εφαρμογή μας ενσωματώνει το API του OpenStreetMap για την εμφάνιση των γεωχωρικών δεδομένων. Επιλέχθηκε για την υποστήριξη ανοιχτού λογισμικού και εξαιτίας της στατικής φύσης των δεδομένων μας.

1.2 Οργάνωση του τόμου

Εργασίες και εφαρμογές σχετικές με το αντικείμενο της διπλωματικής παρουσιάζονται στο Κεφάλαιο 2, όπου αναφέρονται τα κύρια χαρακτηριστικά τους καθώς και τα σημεία όπου διαφοροποιούνται από την παρούσα εφαρμογή. Το Κεφάλαιο 3 αναγράφει όλα τα εργαλεία και τις βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση της διπλωματικής εργασίας. Στο Κεφάλαιο 4 περιγράφουμε τη διαδικασία συλλογής, επεξεργασίας και χρήσης των δεδομένων της εφαρμογής που αναπτύξαμε, τη δημιουργία της βάσης δεδομένων και των γεωχωρικών ερωτημάτων σε αυτή, καθώς και τον αλγόριθμο για την εύρεση διαδρομής. Στο

Κεφάλαιο 5 παρουσιάζουμε την εμφάνιση της εφαρμογής και τις λειτουργικότητές της από την οπτική πλευρά ενός χρήστη. Τέλος στο Κεφάλαιο 6 συζητάμε μελλοντικές επεκτάσεις της παρούσας εφαρμογής.

Κεφάλαιο 2

Σχετικές Εφαρμογές/Εργασίες

2.1 Εισαγωγή

Η παρούσα διπλωματική εργασία εστιάζει στη διαδικασία της πλοϊγησης σε μια πόλη με στόχο την εξερεύνησή της. Αυτό το θέμα έχει προσελκύσει πολλούς να το επιλύσουν από διαφορετική σκοπιά ο καθένας. Ο τρόπος που χρησιμοποιούνται τα δεδομένα που υπάρχουν διαθέσιμα για μία πόλη και τα σημεία ενδιαφέροντός της άλλωστε μπορούν να επηρεάσουν τοπικές οικονομίες και να καθορίσουν, ενδεχομένως, τους τρόπους χρήσης των διάφορων χώρων. Στο πλαίσιο αυτού του κεφαλαίου θα αναφέρουμε κάποιες παρόμοιες εφαρμογές και εργασίες που δουλεύουν πάνω στο κομμάτι αυτό, που είτε μοιάζουν, είτε έχουν παρόμοιες προσεγγίσεις. Συγκεκριμένα θα δούμε τέσσερα παραδείγματα, τριάντα διαφορετικών διαδικτυακών εφαρμογών και μίας διπλωματικής εργασίας.

2.2 Σχετικές εφαρμογές και ιστότοποι

Η Google Maps [6] είναι από τις γνωστότερες διαδικτυακές εφαρμογές που αφορούν χάρτες, εξερεύνηση περιοχών και πλοϊγηση. Έχοντας πρόσβαση στο εύρος δεδομένων της Google, ο χάρτης έχει πλήθος πληροφοριών, όπως καταστήματα, ξενοδοχεία, εστιατόρια, προτάσεις για δραστηριότητες και μετακινήσεις. Για τα καταστήματα και τις υπηρεσίες που εμφανίζονται στον χάρτη παρέχει το ωράριο λειτουργίας, άλλες βασικές πληροφορίες καθώς και τον σύνδεσμο για την ιστοσελίδα του καταστήματος, αν υπάρχει. Η Google Maps δίνει έμφαση στην εύρεση συντομότερης διαδρομής ανάμεσα σε δύο σημεία, και χρησιμοποιεί δεδομένα όπως την τωρινή τοποθεσία του χρήστη, τις παλιές αναζητήσεις του αλλά και τη χρονική στιγμή

που κάνει ο χρήστης μία αναζήτηση για να δώσει πληροφορίες κίνησης ή και της τρέχουσας κατάστασης ενός καταστήματος, μουσείου ή άλλου επαγγελματικού χώρου (ανοιχτό - κλειστό).

Η OpenStreetMap [7] είναι ένας ενεργός διαδικτυακός χάρτης με δυνατότητες πλοήγησης που περιέχει ανοιχτού τύπου δεδομένα για σημεία ενδιαφέροντος. Όλα τα δεδομένα παρέχονται εθελοντικά από τους χρήστες, και αφορούν δρόμους, καταστήματα, μουσεία κλπ. Για αυτόν τον λόγο τα δεδομένα δεν είναι πλήρη για όλες τις περιοχές αλλά εξαρτώνται από τις πληροφορίες που παρέχουν οι χρήστες της εφαρμογής. Παρέχεται η δυνατότητα για αναζήτηση της συντομότερης διαδρομής ανάμεσα σε δύο σημεία, καθώς και πληροφόρηση για τα διάφορα σημεία ενδιαφέροντος πάνω στον χάρτη. Παρόλο που τα δεδομένα που χρησιμοποιούνται στην παρούσα εργασία έχουν συλλεχθεί από αυτόν τον ιστότοπο, είναι διαφορετικός ο τρόπος αξιοποίησής τους.

Η Citymapper [8] είναι μία εφαρμογή οργάνωσης εξόδων και πλοήγησης. Κύρια χρήση της είναι για την εύρεση της καλύτερης διαδρομής, ανάλογα με τα κριτήρια του χρήστη, από ένα σημείο στο άλλο. Ένα από αυτά τα κριτήρια για παράδειγμα είναι να διαλέξει ανάμεσα στη γρηγορότερη διαδρομή από ένα αρχικό σημείο σε ένα τελικό και στη διαδρομή που περνάει κυρίως από κεντρικούς δρόμους για να αποφύγει δρόμους που μπορεί να μην έχουν επαρκή φωτισμό το βράδυ. Παρέχει τη δυνατότητα εύρεσης διαδρομής με χρήση διάφορων μέσων, είτε μαζικής μεταφοράς, είτε με τα πόδια ή το ποδήλατο, είτε με αμάξι, και παράλληλα προσφέρει πληροφορίες για κυκλοφορία σε πραγματικό χρόνο.

Τέλος η διπλωματική της Καλλιόπης Γιαννίκογλου, “Χαρτογράφηση διαδρομών μιας πόλης με χρήση του OpenStreetMap και σχεδιασμός βέλτιστων διελεύσεων” [9], αφορά μία εφαρμογή για τον σχεδιασμό μιας διαδρομής ενός διανομέα, βάσει του πλήθους των πακέτων και του αριθμού των διαθέσιμων διανομέων. Αν και ο τελικός στόχος διαφέρει από τη δική μας εφαρμογή ιστοτόπου, υπάρχουν κοινά σημεία στο πρόβλημα που προκύπτει, το οποίο είναι ο σχεδιασμός της διέλευσης από συγκεκριμένα σημεία στην πόλη με τον συντομότερο δυνατό τρόπο.

Κεφάλαιο 3

Εργαλεία

3.1 Εισαγωγή

Για την καλύτερη κατανόηση των επόμενων κεφαλαίων, θα προχωρήσουμε στην επεξήγηση του τρόπου με τον οποίο λειτουργεί ο ιστότοπος και ο διακομιστής αναφέροντας αναλυτικά τα εργαλεία που χρησιμοποιήθηκαν. Συγκεκριμένα, στο κεφάλαιο αυτό θα γίνει μία αναδρομή των εργαλείων που χρησιμοποιήθηκαν για τη συλλογή των δεδομένων (OpenStreetMap, OSMnx), το στήσιμο της βάσης (MongoDB), την επικοινωνία βάσης με ιστότοπο και την επεξεργασία των δεδομένων (Python, Flask, Networkx, λοιπές βιβλιοθήκες), και τέλος τη λειτουργία και εμφάνιση του ιστοτόπου (Jinja, Leaflet, Bootstrap 4, jQuery).

3.2 OpenStreetMap και OSMnx

Το OpenStreetMap [7] είναι ο χάρτης, του οποίου τα δεδομένα και το API χρησιμοποιήθηκαν σε αυτήν την εργασία. Είναι ένας δωρεάν, ανοιχτός παγκόσμιος χάρτης, στον οποίο συνεισφέρουν εθελοντές παγκοσμίως [10] και τον οποίο διαχειρίζεται η OpenStreetMap Foundation [11]. Ο ιστότοπος αυτός προσφέρει ελεύθερα προς χρήση δεδομένα για πολλές ιστοσελίδες και εφαρμογές. Ο λόγος που επιλέχθηκε μεταξύ άλλων εναλλακτικών είναι πως περιέχει ανοιχτά δεδομένα υπό την “Open Database License” [12], τα δεδομένα δηλαδή μπορούν να χρησιμοποιηθούν ελεύθερα, αρκεί να γίνει η κατάλληλη αναφορά στην πηγή τους και να μη χρησιμοποιηθούν για κερδοσκοπικούς σκοπούς.

Για να αποκτήσουμε αυτά τα δεδομένα και να τα αποθηκεύσουμε στη βάση μας, χρησιμοποιήσαμε το OSMnx πακέτο της Python, που προσφέρει εύκολες μεθόδους για την χρήση

και επεξεργασία των δεδομένων του δικτύου των δρόμων, των αστικών υπηρεσιών και των σημείων ενδιαφέροντος από το OpenStreetMap [13]. Επιπρόσθετα, είναι ανοιχτός κώδικας, και κατά συνέπεια επιτρέπει την ελέυθερη χρήση του.

Οι πληροφορίες που αντλήσαμε από τα εργαλεία αυτά αφορούν το οδικό δίκτυο του δήμου Βόλου, καθώς και διάφορα Σημεία Ενδιαφέροντος (ΣΕ) του δήμου, συμπεριλαμβανομένων αρχαιολογικών χώρων, εκθεσιακών χώρων, κάστρων, συντριβανιών, κήπων, ιστορικών χώρων και κτηρίων, μνημείων και αγαλμάτων, μουσείων, πάρκων, παιδικών χαρών, χώρων αναψυχής, τουριστικών αξιοθεάτων και ξαγνάντων.

3.3 MongoDB

Η MongoDB [14], όπου αποθηκεύτηκαν τα δεδομένα που χρειάζονται για τις λειτουργίες του ιστοτόπου, είναι μία “NoSQL” βάση δεδομένων με δυνατότητα γεωχωρικών ερωτημάτων [15]. Οι NoSQL βασεις δεδομένων, όρος που συνήθως χρησιμοποιείται για την αναφορά σε μη-σχεσιακές βάσεις δεδομένων, έχουν διάφορες μορφές ανάλογα με το μοντέλο των δεδομένων τους, όπως έγγραφα (document), τιμές-κλειδιά (key-value), πλατιάς-στήλης (wide-column), και γράφοι. Είναι γνωστές για την ικανότητά τους να προσαρμόζονται εύκολα στις ανάγκες της κάθε περίπτωσης, είτε αυτό αφορά την πολυμορφία των δεδομένων, είτε τον όγκο τους [16, 17]. Ο τύπος που χρησιμοποιήθηκε για τη διπλωματική εργασία είναι βάση δεδομένων εγγράφων.

Μία βάση δεδομένων εγγράφων (document database ή document-oriented database) είναι μία βάση που αποθηκεύει πληροφορίες σε ζεύγη πεδίων-τιμών σε έγγραφα. Αυτές οι τιμές παίρνουν διάφορες μορφές, μεταξύ των οποίων συμβολοσειρές (strings), αριθμοί, ημερομηνίες, πίνακες, ακόμα και αντικείμενα. Έγγραφα μπορούν να αποθηκευτούν σε μορφές JSON, BSON και XML. Μία ομάδα εγγράφων αποτελεί μία συλλογή (collection). Τα πεδία αυτών των εγγράφων δεν είναι απαραίτητο να είναι ίδια, όπως θα συνέβαινε σε έναν πίνακα σχεσιακής βάσης. Το γεγονός αυτό προσφέρει επιπλέον προσαρμοστικότητα στις NoSQL βάσεις. Κάποια από τα πλεονεκτήματα χρήσης βάσεων εγγράφων αντί σχεσιακών είναι πως τα δεδομένα δε χρειάζονται κανονικοποίηση για πιο αποτελεσματική λειτουργία, αντιθέτως αποκανονικοποιημένα δεδομένα είναι θεμιτά, η ευρεία χρήση JSON εγγράφων, και το δυναμικό σχήμα των εγγράφων [18, 19].

3.4 Backend Κόδικας

Ο πηγαίος κώδικας για το αλγορίθμικό κομμάτι και την επικοινωνία μεταξύ διακομιστή και χρήστη στον ιστότοπο είναι γραμμένος σε Python [20]. Επιλέχθηκε για τη μεγάλη γκάμα λειτουργικών μονάδων τρίτων, τις ενσωματωμένες ικανότητές της, όπως και για όλους τους διαθέσιμους πόρους για την εκμάθηση και χρήση της. Ακολουθεί ταχεία αναφορά όλων των πακέτων τρίτων που χρησιμοποιήθηκαν στον κώδικα Python της εργασίας.

- **PyMongo:** Δεδομένου ότι ο πηγαίος κώδικας είναι σε Python, και η MongoDB χρησιμοποιείται για τη βάση δεδομένων, το PyMongo χρησιμοποιήθηκε για τη σύνταξη των ερωτημάτων προς τη βάση. Το πακέτο αυτό είναι ο εγγενής οδηγός (driver) της Python για τη Mongo [21].
- **NetworkX:** Όταν αναζητούμε διαδρομή, το δίκτυο δρόμων της πόλης απεικονίζεται σαν γράφος και, στη συνέχεια, η εύρεση διαδρομής αντιστοιχεί σε εύρεση μονοπατιού από έναν κόμβο του γράφου σε άλλο. Το NetworkX είναι ένα πακέτο της Python που κάνει αυτή τη διαδικασία δυνατή, και εύκολη στην εφαρμογή της, καθώς επιτρέπει τη δημιουργία, μετατροπή και επεξεργασία γράφων [22].
- **Flask:** Το Flask είναι ένα πλαίσιο εφαρμογής Διεπαφής Πύλης Διακομιστή Ιστού (Web Server Gateway Interface - WSGI) [23]. Επιλέχθηκε για το γεγονός ότι, παρόλο που είναι φιλικό προς τη χρήση για αρχαρίους, έχει τη δυνατότητα να αυξήσει την πολυπλοκότητα των εφαρμογών που το χρησιμοποιούν, ανάλογα με τις ανάγκες τους.
- Άλλες βιβλιοθήκες που χρησιμοποιήθηκαν για πιο μικρούς ρόλους είναι η **bcrypt** [24], για την κωδικοποίηση των κωδικών χρηστών, η **random** [25] για την δημιουργία τυχαίων αριθμών για τα κλειδιά των σημείων ενδιαφέροντος, η **datetime** [26] για την καταγραφή ημερομηνιών, η **ast** [27] για τη μετατροπή δεδομένων συντεταγμένων που πάρθηκαν από το frontend της εφαρμογής σε μορφή συμβολοσειράς, σε πίνακα, είτε αριθμών, είτε πινάκων αριθμών, η **copy** [28] για την αντιγραφή δεδομένων στη μνήμη για την επεξεργασία τους και η **math** [29] για να μπορέσουν να γίνουν πιο περίπλοκες μαθηματικές πράξεις που δεν μπορούσαν να γίνουν με τους απλούς συντελεστές.

3.5 Front-End

Δεδομένου ότι το αντικείμενο της παρούσας διπλωματικής είναι η ανάπτυξη μιας διαδικτυακής εφαρμογής, χρησιμοποιήθηκε κώδικας της HTML για τη σχεδίαση των ιστοσελίδων, κώδικας της CSS για την διαμόρφωση της εμφάνισής τους, και η γλώσσα JavaScript για τον χειρισμό της HTML και προσθήκη διαδραστικότητας. Επίσης, για τον χειρισμό της HTML έγινε χρήση της jQuery, μίας βιβλιοθήκης της JavaScript που είναι συμβατή με πολλούς περιηγητές, με δυνατότητες διαχείρισης συμβάντων και ανακλήσεων πληροφοριών από τον διακομιστή [30].

Άλλα εργαλεία που χρησιμοποιήθηκαν για την εμφάνιση και λειτουργία του ιστοτόπου είναι:

- **Flask Jinja:** Το εργαλείο Flask, για να αποδώσει το αρχείο HTML που θα εμφανιστεί τελικά στον ιστότοπο χρησιμοποιεί την Jinja2. Η Jinja είναι μία γρήγορη και επεκτάσιμη μηχανή προτύπου ιστού, που δίνει τη δυνατότητα σε ειδικά σημεία του προτύπου να λειτουργούν σαν μεταβλητές για να γραφεί κώδικας παρόμοιος στη σύνταξη με την Python. Από το backend στέλνονται δεδομένα στο frontend που χρησιμοποιούνται για να συμπληρωθούν οι μεταβλητές των ειδικών σημείων του προτύπου και για να αποδοθεί το τελικό αρχείο HTML [31].
- **Leaflet:** Για την απεικόνιση του χάρτη και για τη διαδραστικότητα του χρήστη με αυτόν, χρησιμοποιήσαμε το Leaflet. Το Leaflet είναι μία ανοιχτού κώδικα βιβλιοθήκη JavaScript με πολλές λειτουργικότητες για χάρτη, συμβατότητα με πολλαπλές πλατφόρμες και ευχέρεια στη μορφοποίηση [32]. Ο χάρτης που εμφανίζεται στον ιστότοπο είναι αυτός που προστίθεται στο “tile” στρώμα-επίπεδο. Στον ιστότοπό μας χρησιμοποιήσαμε τον χάρτη που παρέχεται από το OpenStreetMap, όπως αναφέρθηκε στην παράγραφο 3.2.
- **Bootstrap 4:** Για την εμφάνιση του ιστοτόπου, χρησιμοποιήθηκε το Bootstrap [33], συγκεκριμένα η έκδοση Bootstrap 4. Είναι ένα πλαίσιο με πολλά εργαλεία, εύκολο στη χρήση που χρησιμοποιείται ευρέως από πολλές διαδικτυακές εφαρμογές. Οι δυνατότητες που προσφέρει είναι συμβατές για ιστοσελίδες που εμφανίζονται σε περιηγητές υπολογιστή και κινητού, κάνοντας τη μετατροπή από τη μία μορφή στην άλλη ομαλή, με τα στοιχεία της σελίδας να προσαρμόζονται κατάλληλα στο μέγεθος και στο σχήμα της οθόνης που χρησιμοποιεί ο χρήστης.

Κεφάλαιο 4

Βάση Δεδομένων και Αλγόριθμοι

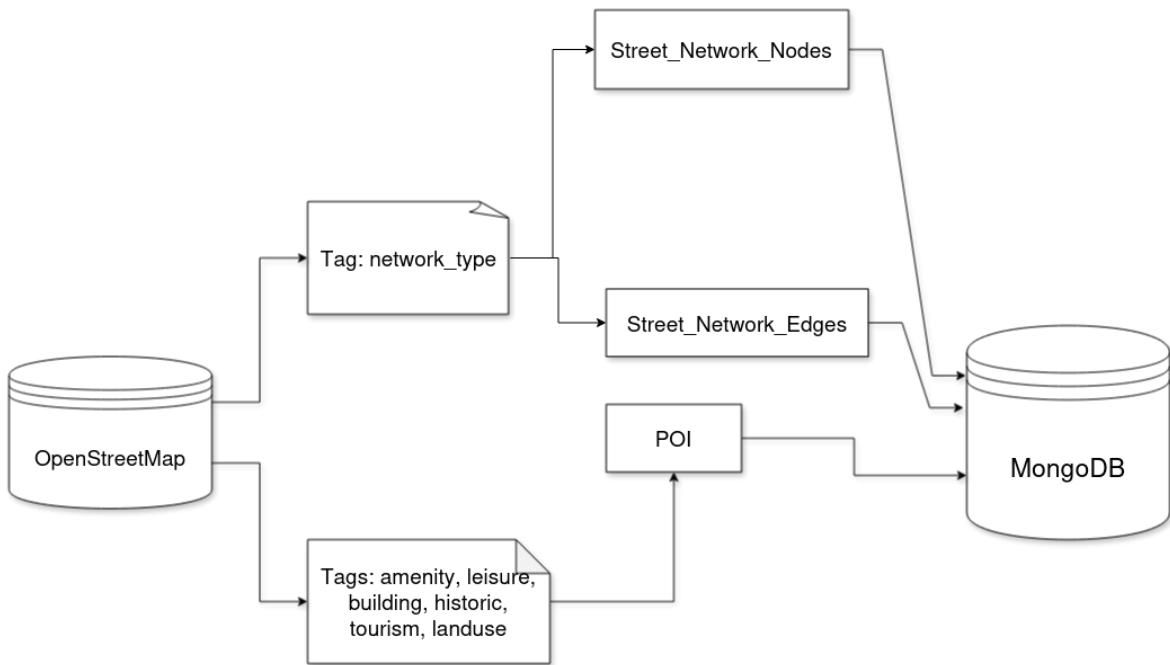
4.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναλύσουμε τον τρόπο που συλλέχθηκαν τα απαραίτητα δεδομένα, τη διαδικασία που ακολουθήσαμε για να στήσουμε τη βάση δεδομένων στην οποία αποθηκεύουμε τα δεδομένα, και τα γεωχωρικά ερωτήματα προς τη βάση τα οποία χρησιμοποιήσαμε για την υλοποίηση των δυνατοτήτων του ιστοτόπου. Τέλος, θα περιγράψουμε σε βάθος τον αλγόριθμο που υλοποιήσαμε για την εύρεση διαδρομής σε χρόνο $\mathcal{O}(M^2 \log(M))$, όπου M ο αριθμός των ΣΕ που επισκεπτόμαστε.

4.2 Συλλογή Δεδομένων

Για να συμπληρώσουμε τη βάση μας με τα απαραίτητα γεωχωρικά δεδομένα, σύμφωνα με τα προαναφερθέντα της παραγράφου 3.2, χρησιμοποιήσαμε την ανοιχτή βάση-χάρτη (open map database) του OpenStreetMap ως πηγή των απαραίτητων δεδομένων για τους σκοπούς της εφαρμογής. Για αυτόν τον λόγο καταλήξαμε στη χρήση της βιβλιοθήκης ανοιχτού κώδικα OSMnx η οποία μας δίνει τη δυνατότητα να κατεβάσουμε μαζικά δεδομένα χρησιμοποιώντας συγκεκριμένες ετικέτες “tags” ως το χαρακτηριστικό αναζήτησης. Στο Σχήμα 4.1 φαίνεται αφαιρετικά η διαδικασία προεπεξεργασίας των δεδομένων που συλλέξαμε.

Αρχικά συλλέξαμε όλα τα στοιχεία με παρούσα ετικέτα “network_type” και συντεταγμένες εντός των γεωγραφικών ορίων της ευρύτερης περιοχής της πόλης του Βόλου. Στη συνέχεια, ενεργώντας πάνω στο ανεπεξέργαστο σετ δεδομένων προχωρήσαμε στον διαχωρισμό των δρόμων του αστικού δικτύου και των σημείων τομής αυτών (διασταυρώσεις) σε σετ

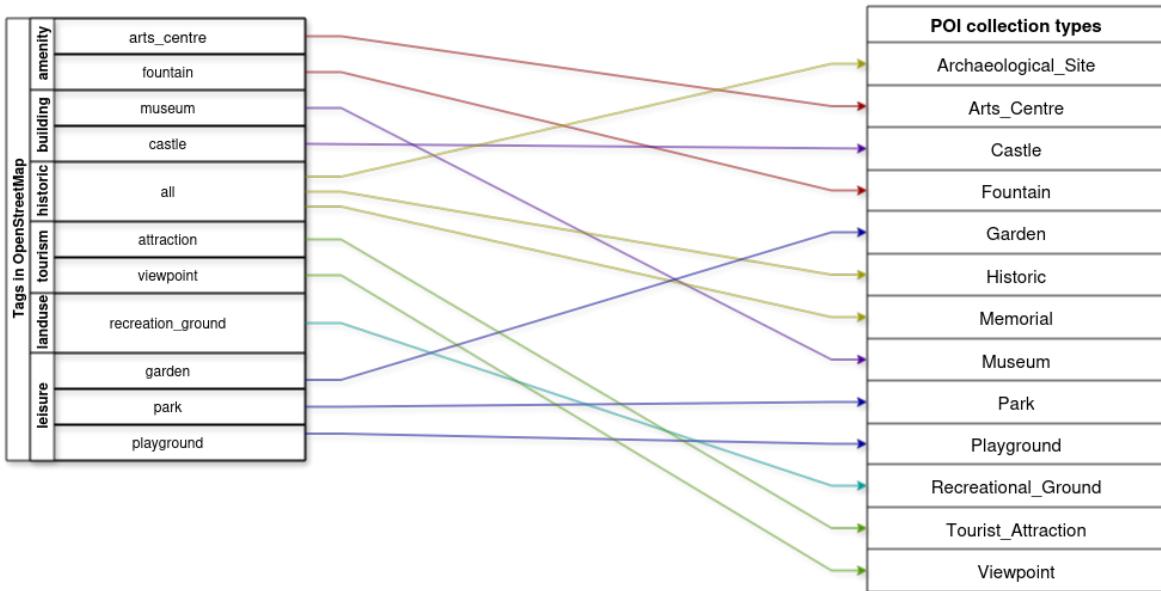


Σχήμα 4.1: Διαδικασία συλλογής και προεπεξεργασίας δεδομένων

ακμών E και σετ κόμβων V αντίστοιχα, μετασχηματίζοντας έτσι τα αρχικά γεωχωρικά δεδομένα μας στη μορφή ενός μη κατευθυνόμενου γραφήματος $G = (V, E)$. Το τελευταίο βήμα πριν σωθούν τα τελικά δεδομένα του γραφήματος στη βάση ήταν η κωδικοποίησή τους σε μορφή JSON, και η μαζική εισαγωγή τους στη συνέχεια στη βάση.

Το επόμενο βήμα στη διαδικασία του Σχήματος 4.1 ήταν η συλλογή των διαφόρων Σημείων Ενδιαφέροντος (ΣΕ) του χάρτη του Βόλου. Όπως φαίνεται στο σχήμα 4.2 συγκεντρώσαμε τα αντικείμενα μέσα στην περιοχή του Βόλου που είχαν τα ακόλουθα ζευγάρια ετικετών-τιμών, και τα κατηγοριοποιήσαμε στα ακόλουθα είδη ΣΕ:

- ετικέτα “amenity”: τιμές “arts_centre” και “fountain” → για τα κέντρα τεχνών και τα συντριβάνια της πόλης
- ετικέτα “building”: τιμές “museum” και “castle” → για τα μουσεία και τα κάστρα της πόλης
- ετικέτα “historic”: τιμές όλα τα αντικείμενα που είχαν αυτήν την ετικέτα → για τα ιστορικά μνημεία και κτήρια
- ετικέτα “tourism”: τιμές “attraction” και “viewpoint” → για τα τουριστικά αξιοθέατα και τα ξάγναντα της πόλης
- ετικέτα “landuse”: τιμές “recreation_ground” → για τους χώρους αναψυχής



Σχήμα 4.2: Μετασχηματισμός δεδομένων ΣΕ

- επικέτα “leisure”: τιμές “garden”, “park” και “playground” → για τους κήπους, τα πάρκα και τις παιδικές χαρές

Αφότου συλλέξαμε και ομαδοποιήσαμε τα ΣΕ με την παραπάνω λογική, μετατρέψαμε το σετ δεδομένων σε μορφή JSON και επακολούθως το σώσαμε στη βάση δεδομένων μας. Έτσι καταλήγουμε στις συλλογές εγγράφων της βάσης, τις οποίες θα αναλύσουμε στην ακόλουθη παράγραφο.

4.3 Βάση Δεδομένων

Το εργαλείο που χρησιμοποιήσαμε για να εισάγουμε, να επεξεργαστούμε και να διαβάσουμε τα δεδομένα μας στη MongoDB βάση δεδομένων ήταν το MongoDB Compass. Το MongoDB Compass προσφέρει μεταξύ άλλων και λειτουργίες επεξεργασίας βάσεων δεδομένων σε γραφικό περιβάλλον. Το γραφικό περιβάλλον δίνει τη δυνατότητα στον χρήστη να εισάγει αρχεία είτε χρησιμοποιώντας απευθείας εντολές της MongoDB Query Language (MQL), είτε μαζικά, μέσω της εισαγωγής πολλαπλών αρχείων σε μορφή JSON. Στα πλαίσια της συγκεκριμένης εργασίας, καταφύγαμε στη χρήση της δεύτερης μεθόδου λόγω της απόκλισης μεταξύ των δεδομένων του σετ αλλά και της ευκολίας χρήσης του γραφικού περιβάλλοντος.

Στη συνέχεια θα αναφερθούμε στις συλλογές POI, Street_Network_Edges και Street_Net-

work_Nodes, οι οποίες περιέχουν τα ΣΕ, τις ακμές/δρόμους και τους κόμβους/διασταυρώσεις του οδικού δικτύου αντίστοιχα και αποτελούν την τελική μορφή των αναγκαίων για την εφαρμογή γεωγραφικών δεδομένων.

4.3.1 Μορφή Συλλογών

Για να παρακάμψουμε τις ανομοιομορφίες στα δεδομένα των συλλογών μας, συγκεκριμένα την έλλειψη συγκεκριμένων πληροφοριών και την πλεονασματικότητα συγκεκριμένων δειγμάτων, επεξεργαστήκαμε την τελική μορφή των αρχείων σε κάθε συλλογή ως εξής:

Συλλογή POI

Στη συλλογή POI (Points of Interest), το κάθε αρχείο D έχει την ακόλουθη μορφή:

$$D_i = (_id_i, properties_i, geometry_i, POID_i) \quad (4.1)$$

όπου $i = 0, 1, \dots, M$ με M τον συνολικό αριθμό των ΣΕ στη βάση.

Συγκεκριμένα:

- $_id$: μοναδικό χαρακτηριστικό τύπου ObjectID, ειδική κλάση της MongoDB, που δίνεται αυτόματα από τη βάση κατά την εισαγωγή του
- $properties$: ένα αντικείμενο της μορφής

$$properties_i = (element_type, name, details, type) \quad (4.2)$$

όπου:

- $element_type$: στοιχείο που έχει πιθανές τιμές “node” στην περίπτωση που είναι σημείο, “way” αν είναι πολύγωνο, ή “relation” αν αποτελεί σχέση πολυγώνων
- $name$: στοιχείο που περιέχει το όνομα του ΣΕ είτε στα αγγλικά είτε στα ελληνικά, εφόσον υπάρχει
- $details$: στοιχείο στο οποίο αναγράφονται οι πληροφορίες που υπήρχαν διαθέσιμες για το ΣΕ
- $type$: στοιχείο το οποίο κατηγοριοποιεί το είδος του ΣΕ προσθέτοντάς του μία από τις τιμές: Archaeological_Site, Arts_Centre, Castle, Fountain, Garden, Historic, Memorial, Museum, Park, Playground, Recreational_Ground, Tourist_Attraction ή Viewpoint

- *geometry*: ένα αντικείμενο της μορφής

$$\text{geometry}_i = (\text{type}, \text{coordinates}) \quad (4.3)$$

όπου:

- *type*: στοιχείο που περιγράφει τη γεωμετρία του ΣΕ και παίρνει τις τιμές “Point” ή “Polygon”
- *coordinates*: πίνακας συντεταγμένων που περιγράφουν την τοποθεσία του ΣΕ
- *POID*: στοιχείο που λειτουργεί σαν μοναδικό κλειδί για το κάθε ΣΕ, πιο ευανάγνωστο, το οποίο χρησιμοποιείται στις αναζητήσεις συγκεκριμένων ΣΕ. Έχει μορφή συμβολοσειράς, με τους τρεις πρώτους χαρακτήρες να αντιπροσωπεύουν τι τύπος ΣΕ είναι, τους επόμενους έξι την ημερομηνία που προστέθηκαν στη βάση σε μορφή YY-MM-DD, και τους τελευταίους 4 χαρακτήρες έναν τυχαίο τετραψήφιο αριθμό από το 0 έως το 9999 (π.χ. PRK2503210743 για ένα πάρκο).

Για πιο αποδοτική αναζήτηση στη συλλογή, έχουν δημιουργηθεί ένα ευρετήριο στο στοιχείο “properties.type” για ευκολότερη αναζήτηση συγκεκριμένων ειδών ΣΕ, και ένα γεωχωρικό “2dsphere” ευρετήριο στο στοιχείο “geometry” για ταχύτερα αποτελέσματα σε γεωχωρικά ερωτήματα.

Συλλογή Street_Network_Nodes

Στη συλλογή Street_Network_Nodes, το κάθε αρχείο *D* έχει την ακόλουθη μορφή:

$$D_i = (_id_i, \text{properties}_i, \text{geometry}_i) \quad (4.4)$$

όπου $i = 0, 1, \dots, V$ με V τον συνολικό αριθμό σταυροδρομίων/κόμβων της βάσης.

Συγκεκριμένα:

- *_id*: ένας μοναδικός ακέραιος αριθμός-κλειδί
- *properties*: ένα αντικείμενο της μορφής

$$\text{properties}_i = (y, x, osmid) \quad (4.5)$$

όπου:

- y : το γεωγραφικό πλάτος του κόμβου, το οποίο ακολουθεί το πρότυπο WGS84 για τον προσδιορισμό της τιμής στο εύρος $[-90^\circ, 90^\circ]$
- x : το γεωγραφικό μήκος του κόμβου, το οποίο ομοίως παίρνει τιμές στο εύρος $[-180^\circ, 180^\circ]$
- $osmid$: ένας χαρακτηριστικός αριθμός “osmid” που αποτελεί την ταυτότητα που ανατίθεται στον κόμβο από το OpenStreetMap, και ο οποίος χρησιμοποιείται σε ζεύγη των δύο από το Street_Network_Edges για να περιγράψει ακμές
- $geometry$: ένα αντικείμενο της μορφής

$$geometry_i = (type, coordinates) \quad (4.6)$$

όπου:

- $type$: στοιχείο με τιμή “Point”
- $coordinates$: πίνακας συντεταγμένων y, x για τον κόμβο.

Για ταχύτερη προσπέλαση στα δεδομένα των κόμβων μέσω γεωχωρικών ερωτημάτων, προστέθηκε και σε αυτήν τη συλλογή ένα γεωχωρικό ευρετήριο τύπου “2dsphere” στο στοιχείο “geometry”.

Συλλογή Street_Network_Edges

Στη συλλογή Street_Network_Edges, το κάθε αρχείο D έχει την ακόλουθη μορφή:

$$D_i = (_id_i, properties_i, geometry_i) \quad (4.7)$$

όπου $i = 0, 1, \dots, E$ με E τον συνολικό αριθμό δρόμων/ακμών στη βάση.

Συγκεκριμένα:

- $_id$: μοναδικό χαρακτηριστικό τύπου ObjectId, ειδική κλάση της MongoDB, που δίνεται αυτόματα από τη βάση κατά την εισαγωγή του
- $properties$: ένα αντικείμενο της μορφής:

$$properties_i = (u, v, length) \quad (4.8)$$

όπου

- u και v : στοιχεία που αντιστοιχούν σε “osmid” κόμβων της συλλογής “Street_Network_Nodes” που ενώνει η κάθε ακμή
- $length$: στοιχείο που περιγράφει το μήκος ενός δρόμου-ακμής σε μέτρα
- $geometry$: ένα αντικείμενο της μορφής

$$geometry_i = (type, coordinates) \quad (4.9)$$

όπου:

- $type$: στοιχείο με τιμή “LineString”, δεδομένου ότι όλοι οι δρόμοι αποτελούν μια γραμμή
- $coordinates$: πίνακας με λίστα ζεύγων συντεταγμένων τα οποία αν προβληθούν στον OpenStreetMap χάρτη σειριακά ξαναχτίζουν τον δρόμο.

Για τη μεγαλύτερη αποδοτικότητα των γεωχωρικών ερωτημάτων, προστέθηκε παρομοίως με τη συλλογή Street_Network_Edges ένα γεωχωρικό ευρετήριο τύπου “2dsphere” στο στοιχείο “geometry”.

Συλλογή Comments

Έκτος από τα δεδομένα που συλλέχθηκαν από το OpenStreetMap, τα οποία μπορούμε να αναφέρουμε ως ‘αναγκαία’ δεδομένα για τη λειτουργία της εφαρμογής, για να προσθέσουμε ένα διεπαφικό στοιχείο μεταξύ χρηστών της εφαρμογής και δεδομένων ήταν απαραίτητη η δημιουργία επιπλέον συλλογών ώστε να μπορούν να συλλεχθούν και να διαβαστούν τα δεδομένα χρηστών. Ένα από αυτά είναι τα σχόλια που μπορούν να αφήσουν οι χρήστες του ιστοτόπου σε ΣΕ. Αυτά αποθηκεύονται στην αποκλειστική συλλογή Comments, όπου κάθε αρχείο D έχει την ακόλουθη μορφή:

$$D_i = (_id_i, guest_usr_i, comment_i, POI_id_i, time_i) \quad (4.10)$$

όπου $i = 0, 1, \dots, C$ με C τον συνολικό αριθμό των σχολίων στη βάση.

Συγκεκριμένα:

- $_id$: ένας αύξοντας ακέραιος αριθμός-μοναδικό κλειδί “_id” που δίνεται κατά την εισαγωγή του αρχείου στη βάση,
- $guest_usr$: μία συμβολοσειρά που αντιπροσωπεύει το όνομα χρήστη

- *comment*: κείμενο που έχει τα περιεχόμενα του σχολίου
- *POI_id*: ένα στοιχείο που αντιστοιχεί στο “POID” του ΣΕ στο οποίο ανήκει το σχόλιο
- *time*: ένα “timestamp” αντικείμενο που περιγράφει την ημερομηνία και ώρα (σε UTC) που αποθηκεύτηκε το σχόλιο στη βάση

Με σκοπό την ταχύτερη ανάκτηση των σχολίων χρηστών κατά την επίσκεψη στον ιστότοπο, στη συλλογή προστέθηκαν δύο ευρετήρια, ένα στο “*_id*” για αναζήτηση εσωτερικά συγκεκριμένου σχολίου και ένα στο “*POI_id*” για την ανάκληση σχολίων συγκεκριμένου ΣΕ.

Συλλογή Users

Η τελευταία συλλογή για να ολοκληρωθεί η εμπειρία των χρηστών του ιστοτόπου δημιουργήθηκε με στόχο την αποθήκευση βασικών στοιχείων για τη δημιουργία προφίλ χρηστών. Το κάθε αρχείο *D* έχει τη μορφή:

$$D_i = (_id_i, username_i, password_i, email_i, adminStatus_i) \quad (4.11)$$

όπου $i = 0, 1, \dots, N$ με N τον συνολικό αριθμό χρηστών στη βάση.

Συγκεκριμένα:

- *_id*: μοναδικό ObjectID “*_id*” που αποθηκεύεται αυτόματα από τη MongoDB με την εισαγωγή του στη βάση
- *username*: το όνομα χρήστη, το οποίο πρέπει να είναι μοναδικό
- *password*: ο κρυπτογραφημένος κωδικός χρήστη
- *email*: η διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη
- *adminStatus*: ένα boolean στοιχείο που παίρνει την τιμή αληθές όταν ο χρήστης είναι διαχειριστής, και ψευδές όταν είναι απλός χρήστης. Για λόγους ασφαλείας η τιμή αυτή ορίζεται αυτόματα ως ψευδής κατά την εγγραφή νέου χρήστη.

Με σκοπό την ταχύτερη ανάκτηση στοιχείων χρηστών κατά την εγγραφή και τη σύνδεση στον λογαριασμό τους, προστέθηκαν δύο ευρετήρια στα πεδία “*_id*” και “*username*” αντίστοιχα.

4.3.2 Γεωχωρικά ερωτήματα στη βάση

Για την υλοποίηση των κύριων δυνατοτήτων που παρέχονται στον ιστότοπο της εργασίας βασιζόμαστε στα αποτελέσματα γεωχωρικών ερωτημάτων που γίνονται στη βάση. Ακολουθεί αναφορά των τελεστών των ερωτημάτων αυτών και η συνεισφορά τους στην εργασία.

- **\$geoWithin:** Τελεστής γεωχωρικού ερωτήματος που εφαρμόζεται στα στοιχεία αρχείων που φέρουν γεωχωρικές πληροφορίες και έχουν ιδανικά τουλάχιστον ένα γεωχωρικό ευρετήριο. Όταν χρησιμοποιείται σε εντολή έχει τη μορφή:

```
$geoWithin: {
    $centerSphere: { [
        [lng, lat],
        radius
    ] }
}
```

όπου:

- **\$centerSphere:** τελεστής σχήματος που ορίζει κύκλο σε σφαίρα
- **[lng, lat]:** παράμετρος του τελεστή **\$centerSphere** που ορίζει το γεωγραφικό μήκος και πλάτος αντίστοιχα του κέντρου του κύκλου στον οποίο αναφερόμαστε
- **radius:** παράμετρος του τελεστή **\$centerSphere** που ορίζει την ακτίνα αυτού του κύκλου σε ακτίνια.

Αυτός ο τελεστής χρησιμοποιήθηκε για να βρούμε τα ΣΕ στην περιοχή που ορίζει ο χρήστης κατά την αναζήτηση σε ακτίνα και κατά την αναζήτηση σε διαδρομή, λειτουργίες που θα συζητηθούν αναλυτικότερα στο κεφάλαιο 5.

- **\$near:** Τελεστής γεωχωρικού ερωτήματος, που επιστρέφει όλα τα αντικείμενα στη βάση που βρίσκονται σε ορισμένη απόσταση από ένα σημείο αναφοράς, ταξινομημένα κατ' αύξουσα σειρά απόλυτης απόστασης μεταξύ τους. Τα αρχεία αυτά έχουν ιδανικά γεωχωρικό ευρετήριο για ταχύτερη ανάκληση. Όταν χρησιμοποιείται σε εντολή έχει τη μορφή:

```

$near: {
    $geometry: {
        type: 'Point',
        coordinates: [lng, lat]
    },
    $minDistance: m_dist,
    $maxDistance: M_dist
}

```

όπου:

- **\$geometry**: τελεστής που ορίζει GeoJSON γεωμετρία, και περιέχει τις παραμέτρους:
 - * **type**: που ορίζει το είδος της GeoJSON γεωμετρίας, η οποία πάντα παίρνει την τιμή 'Point' όταν χρησιμοποιείται στο ερώτημα **\$near**
 - * **coordinates**: που ορίζει τις συντεταγμένες, γεωγραφικό μήκος και πλάτος [lng, lat] του σημείου αναφοράς
- **\$minDistance**: τελεστής για τη μείωση των αποτελεσμάτων βάσει μιας ελάχιστης απόστασης **m_dist** από το σημείο αναφοράς
- **\$maxDistance**: τελεστής για τη μείωση των αποτελεσμάτων βάσει μιας μέγιστης απόστασης **M_dist** από το σημείο αναφοράς

Αυτός ο τελεστής χρησιμοποιήθηκε για την εύρεση κοντινότερου κόμβου, είτε αυτός είναι για τα αρχικά και τελικά σημεία της διαδρομής, είτε η αντιστοίχιση κόμβου με ΣΕ, λειτουργίες που περιγράφονται με περισσότερη λεπτομέρεια παρακάτω.

- **\$geoNear**: Τελεστής γεωχωρικού ερωτήματος που χρησιμοποιείται σε “aggregation pipelines”. Ένα aggregation pipeline περιέχει στάδια στα οποία εκτελούνται διαδοχικά εντολές. Ο τελεστής αυτός χρησιμοποιείται σε στάδιο όπου αναζητούνται αρχεία σε σειρά από το κοντινότερο στο μακρινότερο από συγκεκριμένο σημείο αναφοράς, παρόμοια με το **\$near**. Όταν χρησιμοποιείται έχει τη μορφή:

```

$geoNear: {

```

```

near: {
    type: 'Point',
    coordinates: [lng, lat]
},
distanceField: fieldName,
maxDistance: M_dist,
minDistance: m_dist,
query: {q},
spherical: true
}

```

όπου:

- `near`: παράμετρος που ορίζει το σημείο αναφοράς, χρησιμοποιώντας το `type`: 'Point' για να δηλώσει ότι είναι GeoJSON point, και `coordinates`: [lng, lat] για το γεωγραφικό μήκος και πλάτος του σημείου αναφοράς αντίστοιχα
- `distanceField`: παράμετρος του ονόματος του πεδίου `fieldName` όπου θα εισαχθεί η απόσταση
- `maxDistance`: παράμετρος για τον ορισμό μέγιστης επιθυμητής απόστασης `M_dist` από το σημείο αναφοράς
- `minDistance`: παράμετρος για τον ορισμό ελάχιστης επιθυμητής απόστασης `m_dist` από το σημείο αναφοράς
- `query`: παράμετρος ειδικής συνθήκης `q` που θέλουμε να πληρεί το τελικό αποτέλεσμα
- `spherical`: παράμετρος που παίρνει τιμή `true` όταν δουλεύουμε με 2dsphere γεωχωρικό ευρετήριο.

Χρησιμοποιήθηκε αυτός ο τελεστής για τη συλλογή όλων των δρόμων στην περιοχή με σκοπό το στήσιμο τοπικού οδικού γράφου για την αναζήτηση διαδρομής, καθώς και στην εύρεση επόμενου κοντινότερου κόμβου από έναν κόμβο, πράγμα που συζητείται σε μεγαλύτερη λεπτομέρεια στην επόμενη παράγραφο.

- **\$geoIntersects:** Τελεστής γεωχωρικού ερωτήματος που επιστρέφει όλα τα αρχεία των οποίων τα γεωχωρικά χαρακτηριστικά επικαλύπτονται από ένα GeoJSON αντικείμενο. Όταν χρησιμοποιείται έχει τη μορφή:

```

$geoIntersects: {
    $geometry: {
        type: objectType,
        coordinates: [ coords ]
    }
}

```

όπου:

- **\$geometry:** τελεστής που ορίζει τη γεωμετρία αντικειμένου GeoJSON
- **type:** παράμετρος του τελεστή **\$geometry** που δηλώνει το είδος **objectType** του αντικειμένου
- **coordinates:** παράμετρος του τελεστή **\$geometry** που περιέχει τις **[coords]** συντεταγμένες του αντικειμένου. Αυτές είναι είτε ένα ζεύγος συντεταγμένων γεωγραφικού μήκους και πλάτους αν το αντικείμενο είναι σημείο, είτε πίνακας ζεύγων αν είναι πολύγωνο

Χρησιμοποιήθηκε στην ενημέρωση ΣΕ όταν ο χρήστης σχηματίζει πολύγωνο για ένα ΣΕ. Με τον τελεστή αυτόν αναζητούμε στη βάση αν υπάρχει ήδη κάποιο ΣΕ στην περιοχή που όρισε ο χρήστης σε περίπτωση που εννοούσε κάποιο από αυτά που βρίσκονται ήδη εκεί σύμφωνα με τα δεδομένα της βάσης. Αναφερόμαστε σε μεγαλύτερη έκταση στη διαδικασία αυτή στο κεφάλαιο 5.

4.4 Εύρεση διαδρομής

Εκτός από την αναζήτηση ΣΕ γύρω από ένα κεντρικό σημείο, παρέχεται και η δυνατότητα αναζήτησης διαδρομής από ένα αρχικό σε ένα τελικό σημείο που ορίζει ο χρήστης. Η διαδρομή που αναζητείται, διασχίζει όλα τα ΣΕ που βρίσκονται στην περιοχή μεταξύ των δύο

σημείων. Στόχος είναι να γίνει αυτή η διαδρομή περισσότερο ενδιαφέρουσα, με βάση τα στοιχεία που ζήτησε ο χρήστης.

Η εύρεση της διαδρομής που περνάει από όλα τα ΣΕ της περιοχής μπορεί να αναχθεί στο Πρόβλημα του Περιοδεύοντος Πωλητή (Traveling Salesman Problem), το πρόβλημα δηλαδή της αναζήτησης διαδρομής σε βεβαρυμένο γράφο η οποία περνάει από όλους τους κόμβους του γράφου, βελτιστοποιώντας ταυτόχρονα το συνολικό κόστος της διαδρομής, όπως αυτό δηλώνεται στα βάρη του γράφου [34, 35]. Το ανάλογο στο δικό μας πρόβλημα είναι, θέτοντας κάθε ΣΕ ως ενδιάμεσο κόμβο από τον οποίο θέλουμε να μεταβούμε, αναζητούμε τη βέλτιστη διαδρομή μεταξύ των σημείων αφετηρίας και προορισμού, ώστε τελικά να επισκεφτούμε όλους τους κόμβους/ΣΕ του γράφου. Στην παρούσα εφαρμογή το πρόβλημα διαφοροποιείται ως προς τους κόμβους που θέλουμε να επισκεφτούμε οι οποίοι είναι υποσύνολο του αρχικού συνόλου των κόμβων, καθώς και πως ο τελικός κόμβος είναι προκαθορισμένος (ο προορισμός). Δεδομένου ότι η εύρεση βέλτιστης λύσης για το TSP ανήκει στην κατηγορία NP-hard προβλημάτων και είναι μη προσεγγίσιμη [36], χρησιμοποιήσαμε μία ευριστική προσέγγιση η οποία ρυθμίστηκε για τα δεδομένα του χάρτη του Βόλου.

Η προσέγγιση που χρησιμοποιήσαμε για το πρόβλημα που προκύπτει είναι η ακόλουθη. Έστω P_0 ο αρχικός κόμβος από όπου ξεκινάει η διαδρομή, και P_N ο τελικός κόμβος όπου τελειώνει η διαδρομή. Βρίσκουμε την απόσταση μεταξύ P_0 και P_N (ίση με $2r$) και το κέντρο O του ευθύγραμμου τμήματος P_0P_N . Έχοντας το σημείο O και την ακτίνα r , συγκεντρώνουμε από τη συλλογή Street_Network_Edges όλους τους δρόμους $A \subset E$ που βρίσκονται μέσα στον κύκλο $C(O, r)$ χρησιμοποιώντας γεωχωρικό ερώτημα με τον τελεστή \$geoNear, και στη συνέχεια χτίζουμε έναν τοπικό γράφο με τους δρόμους ως ακμές και ως κόμβους τα σημεία τομής των δρόμων. Για να βρούμε το σύνολο $S \subset POI$ των ΣΕ τα οποία θα επισκεφτούμε, γίνεται γεωχωρική ερώτηση στη συλλογή POI μέσα στον κύκλο $C(O, r)$ για όλα τα ΣΕ των ειδών που ζήτησε ο χρήστης. Για κάθε σημείο του συνόλου S από όπου θέλουμε να περάσουμε εντοπίζουμε τον κόμβο $\{P_i\} \subset V, i = 1, \dots, N - 1$ από τη συλλογή Street_Network_Nodes με γεωχωρικό ερώτημα με τελεστή \$near, με τον οποίο θα το αντιστοιχίσουμε. Αν το ΣΕ είναι σημείο, βρίσκουμε τον πιο κοντινό κόμβο, αλλιώς, αν είναι πολύγωνο, βρίσκουμε τη γωνία που είναι πιο κοντά στο κέντρο O του κύκλου αναζήτησης, και από εκεί τον πιο κοντινό κόμβο.

Έχοντας όλους τους προς επίσκεψη κόμβους ($P_i, i = 0, \dots, N$), το επόμενο βήμα είναι να βρούμε τη σειρά επίσκεψή τους. Φτιάχνουμε μία λίστα για τη σειρά επίσκεψης κάθε

κόμβου, με αφετηρία τον P_0 , και προορισμό τον P_N . Έπειτα, βρίσκουμε τους κόμβους που είναι πλησιέστεροι στο πρώτο και στο τελευταίο στοιχείο και τους τοποθετούμε στη δεύτερη και στην προτελευταία θέση αντίστοιχα. Στη συνέχεια βρίσκουμε τον πλησιέστερο κόμβο προς το δεύτερο και το προτελευταίο στοιχείο κ.ο.κ. Δηλαδή μετά τον P_i , βάζουμε ως P_{i+1} το σημείο για το οποίο η απόσταση $d_{P_i, P_{i+1}}$ είναι η μικρότερη. Ομοίως, πριν από τον P_{N-i} βάζουμε ως P_{N-i-1} το σημείο για το οποίο η απόσταση $d_{P_{N-i}, P_{N-i-1}}$ είναι η μικρότερη. Αυτή η διαδικασία συνεχίζεται μέχρι είτε να τελειώσουν οι κόμβοι προς ταξινόμηση είτε να μείνει ένας, ο οποίος θα μπει στη μέση της λίστας $[P_0, P_1, \dots, P_{N-1}, P_N]$. Η διαδικασία αυτή φαίνεται σε ψευδοκώδικα στον αλγόριθμο 1.

Algorithm 1 Σειρά επίσκεψης κόμβων

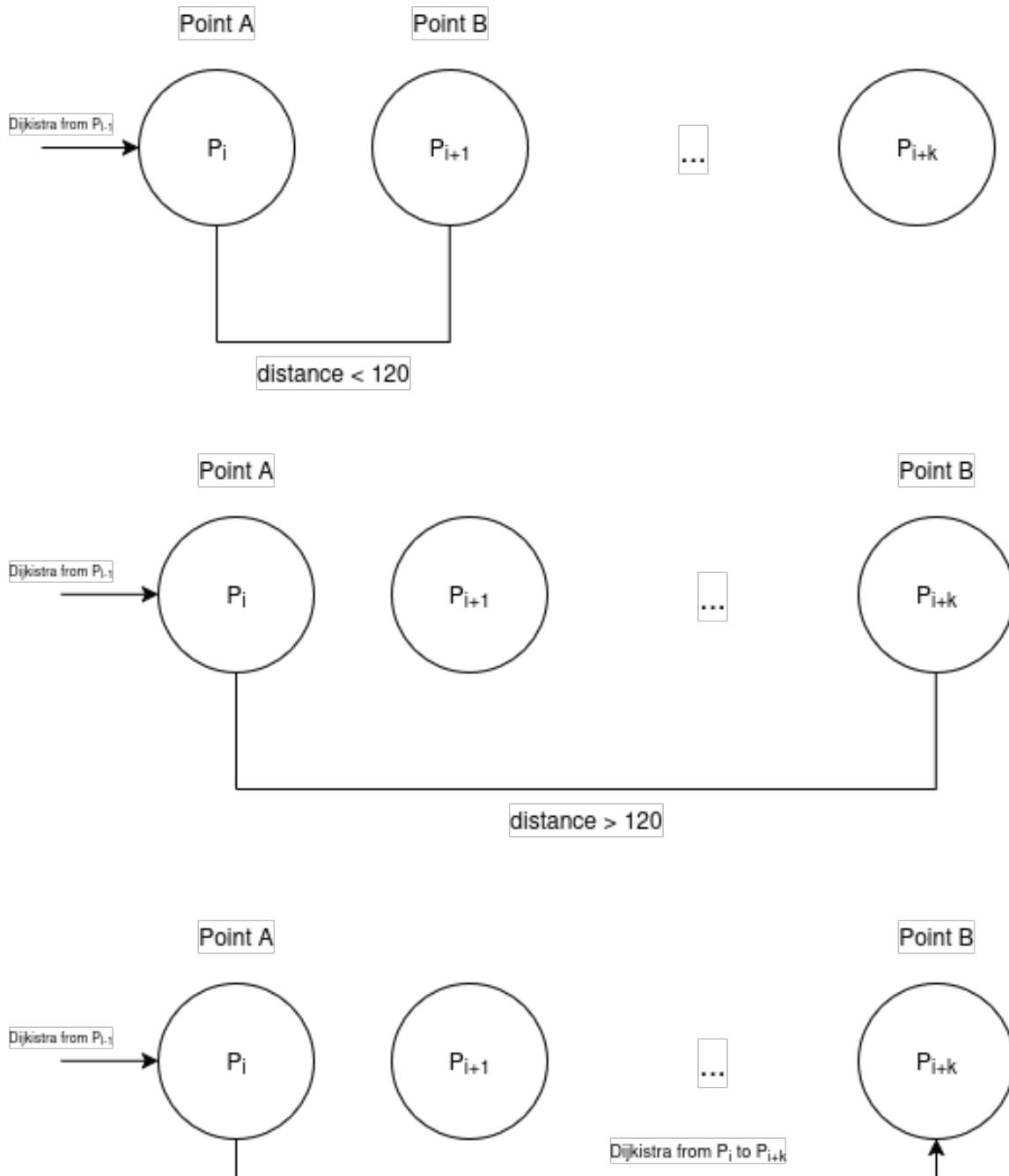
```

1: Let visitList[] where len(visitList) == POInumber + 2 //όπου visitList[] η λίστα με σειρά
   επίσκεψης και POInumber το πλήθος των Σημείων Ενδιαφέροντος
2: visitList[0] := starting node
3: visitList[POInumber+1] := ending node
4: for  $i = 0$  to POInumber/2 do
5:   pointA := nearest node to visitList[ $i$ ]
6:   pointB := nearest node to visitList[POInumber+1 -  $i$ ]
7:   insert(pointA) to visitList[ $i + 1$ ]
8:   insert(pointB) to visitList[POInumber- $i$ ]
9:   mark pointA and pointB as visited
10: end for
11: if unmarked node exists then
12:   visitList[POInumber/2+1] := last remaining node //συμβαίνει αν ο αριθμός των Σημείων
      Ενδιαφέροντος προς επίσκεψη είναι περιπτώση
13: end if

```

Έχοντας βρει τη σειρά προσπέλασης των κόμβων μέσω του μονοπατιού $P_0 \rightarrow P_1 \rightarrow \dots \rightarrow P_N$, βρίσκουμε τα επιμέρους μονοπάτια $P_i \rightarrow P_{i+1}$ από τον έναν κόμβο στον επόμενο. Για αυτό τον σκοπό επιλέξαμε να χρησιμοποιήσουμε την υλοποίηση του αλγορίθμου Dijkstra που παρέχει η βιβλιοθήκη NetworkX για κάθε ζεύγος κόμβων P_i, P_{i+1} . Σε περίπτωση που για την απόσταση $d_{P_i, P_{i+1}}$ ισχύει $d_{P_i, P_{i+1}} < D$, όπου D μία αρκετά μικρή απόσταση για τα δεδομένα του μεγέθους της πόλης, παραλείπουμε τον δεύτερο κόμβο για να βρούμε τη διαδρομή από τον αρχικό κόμβο P_i στον επόμενο κόμβο της ακολουθίας ($P_{i+k}, k > 1$) που πληροί αυτήν την προϋπόθεση (Σχήμα 4.3). Το σκεπτικό μας για την επιλογή ορίου

απόστασης βασίζεται στη λογική ότι δε θα χρειαστεί να υπολογίσουμε διαδρομή μεταξύ σημείων που βρίσκονται τόσο κοντά επειδή θα έχουν ήδη εξερευνηθεί από τους χρήστες κατά τη διάρκεια του προηγούμενου βήματος. Για την πόλη του Βόλου επιλέξαμε το όριο ως 120 μέτρα εμπειρικά, μετρώντας την απόσταση δύο ΣΕ στην πιο πυκνή από ΣΕ περιοχή. Τέλος ενώνοντας όλα τα επιμέρους μονοπάτια σχηματίζουμε την τελική διαδρομή.



Σχήμα 4.3: Εύρεση επιμέρους μονοπατιών

Συνοπτικά ο συνολικός αλγόριθμος μπορεί να περιγραφεί ως εξής (Αλγόριθμος 2):

Algorithm 2 Εύρεση διαδρομής

```

1: Let multiGraph G
2: Let path           To τελικό συνολικό μονοπάτι
3: start := findNearestNode(starting coordinates)
4: finish := findNearestNode(end coordinates)
5: edges := all the edges in the area
6: insert(edges) in G
7: POInodes := findNearestNode(POI coordinates)
8: use algorithm 1 to sort the visitation order
9: Let movingCondition := TRUE      Συνθήκη για να αλλάξει το αρχικό σημείο του Dijkstra,
   αληθές πάντα εκτός και αν οι κόμβοι για το μονοπάτι είναι πολύ κοντά ο ένας στον άλλο
10: for i = 0 to len(visitationList)−1 do
11:   if movingCondition == TRUE then
12:     pointA := visitationList[i]
13:   end if
14:   pointB := visitationList[i + 1]
15:   if distance(pointA, pointB)≤ D then
16:     movingCondition := FALSE
17:     CONTINUE
18:   end if
19:   movingCondition := TRUE
20:   newPath := dijkstra(pointA, pointB)
21:   insert(newPath) in path
22: end for

```

Ο πηγαίος κώδικας του αλγορίθμου καθώς και ολόκληρης της εφαρμογής που αναπτύχθηκε στο πλαίσιο της παρούσας διπλωματικής εργασίας είναι διαθέσιμος στον σύνδεσμο <https://github.com/Kyriaki-Koffa/VoIEExplore-Diploma-Thesis>.

Για να υπολογίσουμε το κόστος όλου του αλγορίθμου, εάν M είναι ο αριθμός των ΣΕ που επισκεπτόμαστε, η ταξινόμησή τους έχει κόστος $M^2/2$. Με V, E να αποτελούν τους αριθμούς των κόμβων και των ακμών του τοπικού γράφου που χτίζεται, και τον αλγόριθμο Dijkstra να έχει κόστος $\mathcal{O}(E \log(V))$ [37], όλοι οι αλγόριθμοι Dijkstra που εκτελούνται για το κάθε επιμέρους μονοπάτι που υπολογίζεται έχουν κόστος $\mathcal{O}(M E \log(V))$. Συνολικά το κόστος είναι $\mathcal{O}(M^2/2 + M E \log(V))$. Μελετώντας τα δεδομένα μας, βρήκαμε ότι κατά μέσο όρο

(για το 60% των κόμβων) από τον κάθε κόμβο εξέρχονται τρεις ακμές, οπότε $E = (3V)/2$. Για να βρούμε τη σχέση που έχει ο αριθμός των ΣΕ με τον αριθμό κόμβων, πήραμε από ένα τυχαίο σημείο στο κέντρο του Βόλου, που έχει τη μεγαλύτερη πυκνότητα κόμβων, το πλήθος κόμβων που υπάρχουν σε ακτίνα 1000 μέτρων επειδή είναι το μέγιστο όριο που μπαίνει κατά την αναζήτηση σε διαδρομή. Βρέθηκαν περίπου 1000 κόμβοι, οπότε δουλεύουμε με το σκεπτικό ότι το V θα κυμαίνεται κοντά στο 1000. Με όλα τα ΣΕ στη βάση δεδομένων να είναι συνολικά 280, υποθέτουμε πως στη χειρότερη περίπτωση $M = 280$, οπότε σε κάθε τρεις κόμβους θα υπάρχει ένα ΣΕ, δηλαδή $3M \approx V$. Οπότε το κόστος παίρνει τη μορφή:

$$\begin{aligned}\mathcal{O}(M^2/2 + MEl\log(V)) &= \mathcal{O}(M^2/2 + \frac{9M^2}{2}\log(3M)) = \\ &= \mathcal{O}(M^2 + M^2\log(M)) = \mathcal{O}(M^2\log(M))\end{aligned}$$

Ο προσεγγιστικός αλγόριθμος που προκύπτει δεν βγάζει πάντα τα βέλτιστα αποτελέσματα, καθώς υπάρχουν ιδιάζουσες τοπολογίες στις οποίες η διαδρομή θα μπορούσε να είναι και μικρότερη. Αυτές οι τοπολογίες συνήθως έχουν περιοχές με πυκνές συγκεντρώσεις ΣΕ σε ομάδες (clusters) ενώ ταυτόχρονα υπάρχουν πιο μακριά, διάσπαρτα και αραιά ΣΕ στον υπόλοιπο γράφο.

Κεφάλαιο 5

Εμφάνιση και Λειτουργίες Εφαρμογής

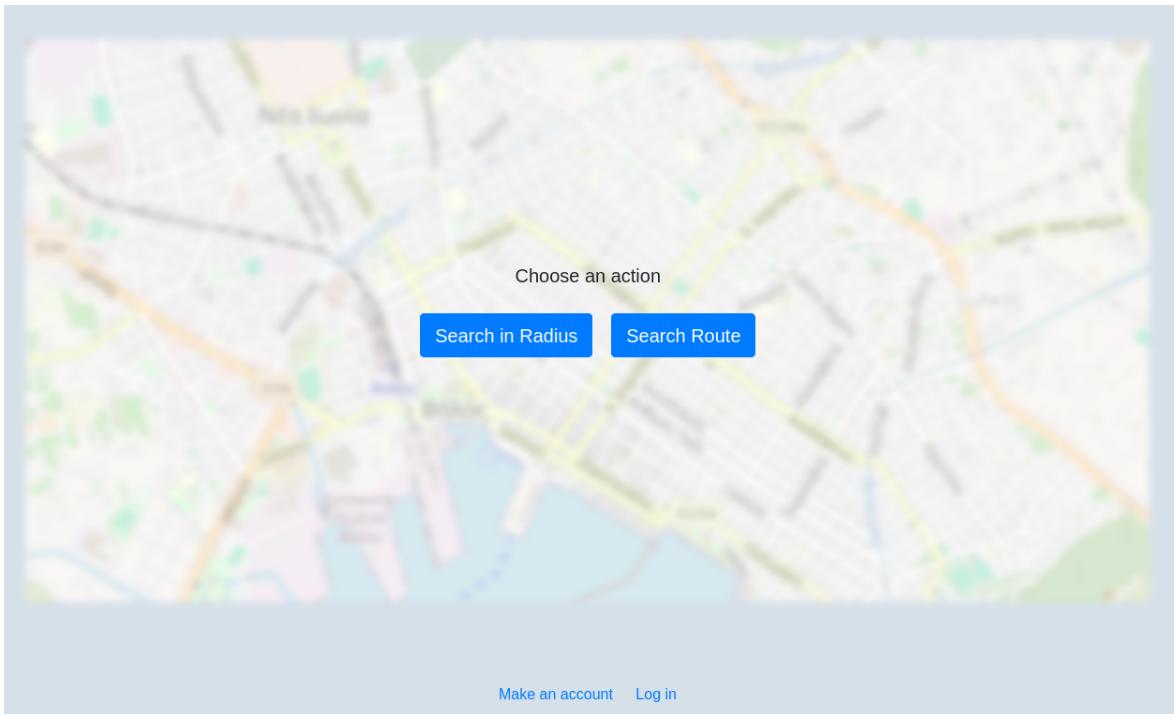
5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα περιγράψουμε τη διαδικασία χρήσης του ιστοτόπου από έναν χρήστη, καθώς και την εμφάνιση κάθε επιμέρους σελίδας. Συγκεκριμένα, θα περιγραφούν αναλυτικά οι δυνατότητες της εφαρμογής: δημιουργία λογαριασμού, εισαγωγή σε λογαριασμό, αναζήτηση σημείων ενδιαφέροντος σε περιοχή, δημιουργία μονοπατιού που περνάει από σημεία ενδιαφέροντος, προσθήκη σχολίων σε σημεία ενδιαφέροντος και ενημέρωση πληροφοριών σημείου ενδιαφέροντος.

5.2 Αρχική Σελίδα

Μπαίνοντας στην αρχική σελίδα, όπως βλέπουμε στο σχήμα 5.1, υπάρχει στο κέντρο η επιγραφή “Choose an action”, και κάτω από αυτήν δύο κουμπιά που ανακατευθύνουν σε σελίδες με τις κύριες λειτουργίες του ιστοτόπου. Το κουμπί “Search in Radius” οδηγεί σε σελίδα όπου μπορούμε να αναζητήσουμε Σημεία Ενδιαφέροντος (ΣΕ) που εντοπίζονται σε κυκλική περιοχή γύρω από ένα κέντρο με δοσμένη ακτίνα, ενώ το “Search in Path” οδηγεί σε σελίδα στην οποία μπορούμε να βρούμε διαδρομή που περνάει από συγκεκριμένα ΣΕ.

Τέλος, στο κάτω μέρος του παραθύρου, κεντραρισμένοι, υπάρχουν οι δύο σύνδεσμοι “Make an account” και “Log In”, για να φτιάξει ο χρήστης λογαριασμό, ή για να συνδεθεί σε προϋπάρχοντα, αν δεν είναι ήδη συνδεδεμένος. Αν είναι συνδεδεμένος, τότε αντί αυτών υπάρχει ένας σύνδεσμος “Log Out” για την αποσύνδεση.



Σχήμα 5.1: Αρχική σελίδα

5.3 Log In/ Register/ Log Out

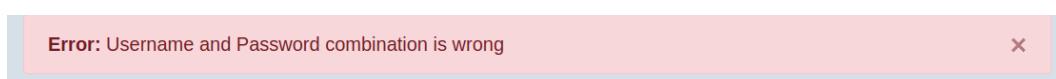
Αν ο χρήστης στην αρχική σελίδα πατήσει τον σύνδεσμο “Log In”, μεταφέρεται στη σελίδα σύνδεσης που φαίνεται στο σχήμα 5.2. Στην κορυφή της σελίδας υπάρχει μία μπάρα με συνδέσμους που ανακατευθύνουν στην αρχική σελίδα και σε σελίδα με πληροφορίες για τον ιστότοπο. Στο κέντρο του παραθύρου υπάρχει μία φόρμα εισόδου, με πεδίο για όνομα χρήστη και κωδικό χρήστη αντίστοιχα. Στο κάτω μέρος του παραθύρου με μικρά γράμματα υπάρχει η επιγραφή “Need an account? Register here”, όπου η λέξη “here” να αποτελεί σύνδεσμο που οδηγεί στη σελίδα εγγραφής.

Όταν είναι κενά τα πεδία των στοιχείων του χρήστη, υπάρχει σήμανση ότι είναι απαραίτητο να συμπληρωθούν. Αν γράψουμε όνομα χρήστη που δεν υπάρχει ή κωδικό που δεν αντιστοιχεί στο όνομα χρήστη που δόθηκε, τότε εμφανίζεται μήνυμα λάθους (Σχήμα 5.3). Διαφορετικά, αν δεν προκύψει κάποιο πρόβλημα, εμφανίζεται το μήνυμα επιτυχίας (Σχήμα 5.4) και επανακατευθυνόμαστε στην αρχική σελίδα.

Επιλέγοντας τον σύνδεσμο “Register”, ανακατευθυνόμαστε στη σελίδα του σχήματος 5.5 που περιέχει τα απαραίτητα στοιχεία για την εγγραφή. Στην κορυφή υπάρχει η μπάρα με τους συνδέσμους για την αρχική σελίδα και αυτή με τις πληροφορίες, και στο κέντρο μία φόρμα για εγγραφή, με πεδίο για όνομα χρήστη, διεύθυνση ηλεκτρονικού ταχυδρομείου και κωδικό.

The screenshot shows the starting page of the volExplore application. At the top, there is a blue header bar with the text "Starting Page" on the left and "About volExplore" on the right. Below the header is a light gray main area containing a login form. The form has two input fields: "Username:" and "Password:", each with a placeholder text ("Type username" and "Type password" respectively) and a red border indicating they are required. Below the fields is a blue "Log In" button. At the bottom of the main area, there is a small link: "Need an account? Register [here](#)".

Σχήμα 5.2: Σελίδα σύνδεσης σε λογαριασμό



Σχήμα 5.3: Μήνυμα αποτυχίας



Σχήμα 5.4: Μήνυμα επιτυχίας στην σύνδεση λογαριασμού

The screenshot shows the starting page of the volExplore application, similar to the one in Figure 5.2 but with a different form. At the top, there is a blue header bar with the text "Starting Page" on the left and "About volExplore" on the right. Below the header is a light gray main area containing a registration form. The form has three input fields: "E-mail:", "Username:", and "Password:", each with a placeholder text ("example@place.domain", "Type username", and "Type password" respectively) and a red border. Below the fields is a blue "Register" button. At the bottom of the main area, there is a small link: "Already have an account? Log in [here](#)".

Σχήμα 5.5: Σελίδα δημιουργίας λογαριασμού

Σε περίπτωση λάθους, αν δηλαδή δοθεί λάθος δομημένη διεύθυνση ταχυδρομείου, ή αν δε μπει όνομα χρήστη ή αν μπει κωδικός μικρότερος των πέντε χαρακτήρων ή μεγαλύτερος των είκοσι, εμφανίζονται οι προειδοποιήσεις του σχήματος 5.6.

E-mail:

Invalid e-mail

Username:

Please fill out this field

Password:

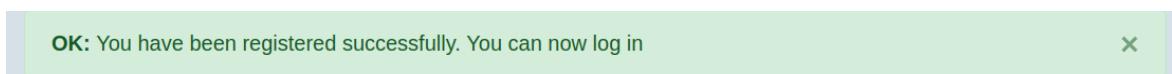
Password must be 5-20 characters long

Σχήμα 5.6: Σημάνσεις λάθους στη φόρμα εγγραφής

Αν ο χρήστης προσπαθήσει να εγγραφεί με όνομα που χρησιμοποιείται ήδη από άλλον λογαριασμό, εμφανίζεται το μήνυμα λάθους (Σχήμα 5.7). Άλλιώς, αν δεν προκύψουν άλλα προβλήματα, εμφανίζεται μήνυμα επιτυχίας (Σχήμα 5.8) και επανακατευθύνεται στην σελίδα σύνδεσης σε λογαριασμό.



Σχήμα 5.7: Προειδοποίηση αν το όνομα χρήστη χρησιμοποιείται ήδη



Σχήμα 5.8: Επιτυχής εγγραφή

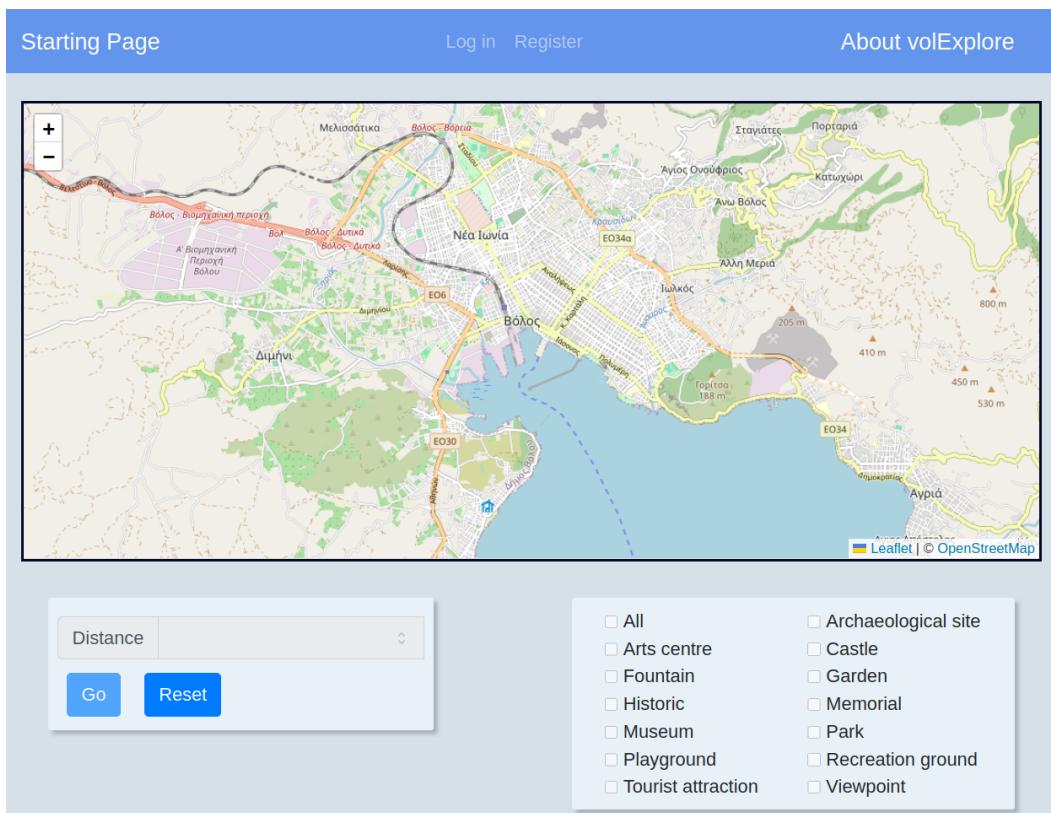
Σε περίπτωση που ο χρήστης είναι συνδεδεμένος και πατήσει τον σύνδεσμο “Log Out”, αποσυνδέεται από τον λογαριασμό του και εμφανίζεται το μήνυμα επιτυχίας (Σχήμα 5.9).



Σχήμα 5.9: Επιτυχής αποσύνδεση

5.4 Search in Radius

Όταν ο χρήστης στην αρχική σελίδα πατήσει το κουμπί “Search in Radius” ανακατευθύνεται στη σελίδα του σχήματος 5.10 για την αναζήτηση ΣΕ σε ακτίνα. Στην κορυφή του παραθύρου υπάρχει μία μπάρα με συνδέσμους για την αρχική σελίδα και τη σελίδα για πληροφορίες, ενώ στη μέση υπάρχουν είτε δύο σύνδεσμοι, ένας για εγγραφή και ένας για είσοδο σε λογαριασμό αν δεν είναι συνδεδεμένος ο χρήστης, όπως φαίνεται στο σχήμα 5.10, είτε ένας σύνδεσμος για αποσύνδεση από τον λογαριασμό.



Σχήμα 5.10: Σελίδα αναζήτησης σε ακτίνα “Search in Radius”

Κάτω από την μπάρα υπάρχει ένα πλαίσιο με τον διαδραστικό χάρτη της OpenStreetMap, εστιασμένο στο κέντρο της πόλης του Βόλου. Δεξιά κάτω από τον χάρτη υπάρχει μία λίστα με τα διαφορετικά είδη ΣΕ (σχήμα 5.11). Στο αριστερό μέρος υπάρχει ένα πλαίσιο εισόδου με την ετικέτα “Distance” (απόσταση), και δύο κουμπιά, το κουμπί “Go” για την υποβολή της απόστασης, και το κουμπί “Reset” για την επαναφόρτωση της σελίδας σε περίπτωση που ο χρήστης θέλει να εκκαθαρίσει τα στοιχεία της και να υποβάλει καινούργια.

Σε αυτή τη σελίδα, όπως φαίνεται στο παράδειγμα χρήσης στο σχήμα 5.12, ο χρήστης πατάει ένα σημείο στον χάρτη, όπου και εμφανίζεται μία πινέζα, η οποία λειτουργεί σαν το

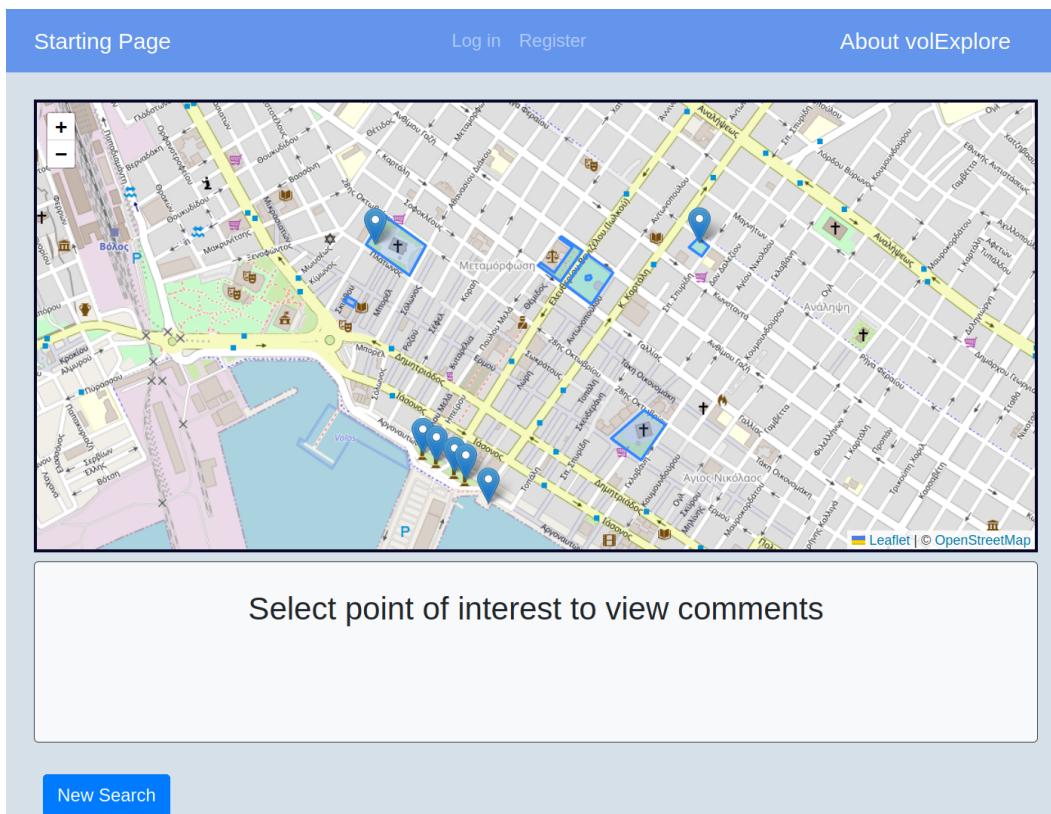
κέντρο της αναζήτησης. Όταν εμφανιστεί η πινέζα, ο πίνακας με τη λίστα των ειδών των ΣΕ ενεργοποιείται και μπορεί ο χρήστης να δράσει πάνω του. Αυτό αποτελεί το φίλτρο αναζήτησης: ο χρήστης μπορεί να επιλέξει τι είδη ΣΕ θέλει να αναζητηθούν στη γύρω περιοχή. Μπορεί να τα διαλέξει είτε όλα, πατώντας το κουτάκι “All”, είτε κάποιον συνδυασμό ανάμεσα σε αυτά που εμφανίζονται. Όταν επιλεχθεί έστω και ένα φίλτρο, τότε ενεργοποιείται το πλαίσιο απόστασης. Εκεί ο χρήστης γράφει την απόσταση σε μέτρα όπου θέλει να αναζητήσει ΣΕ από το σημείο που υπέβαλε στον χάρτη. Τέλος όταν το συμπληρώσει και αυτό, του δίνεται η δυνατότητα να πατήσει το κουμπί “Go” για να παραχθούν τα αποτελέσματα.



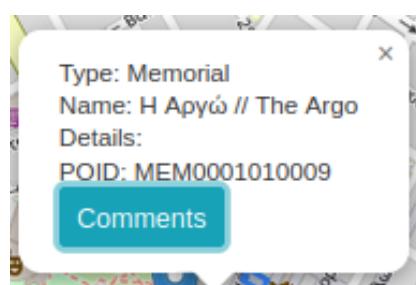
Σχήμα 5.11: Φίλτρο ειδών ΣΕ

Σχήμα 5.12: Η σελίδα αναζήτησης σε ακτίνα σε χρήση

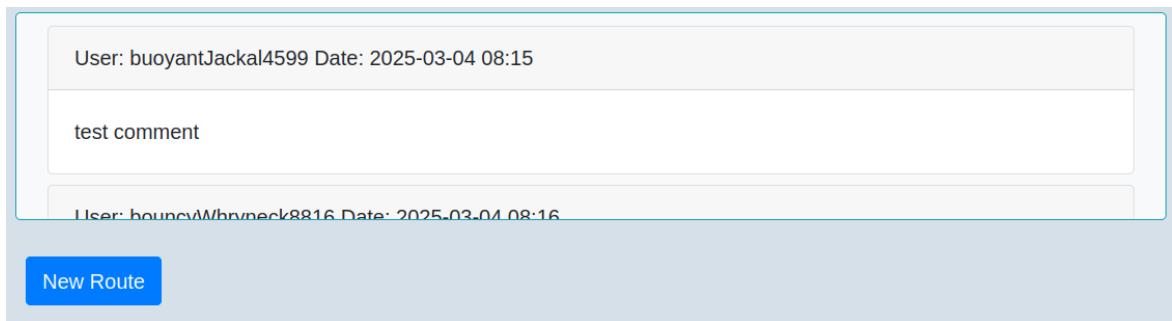
Όταν πατηθεί το κουμπί για την αναζήτηση, φορτώνει καινούργια σελίδα όπως φαίνεται στο σχήμα 5.13. Το σημείο όπου διαφέρει από την προηγούμενη σελίδα είναι ότι στο πλαίσιο με τον χάρτη υπάρχουν μπλε επισημάνσεις, είτε σαν πινέζες είτε σαν πολύγωνα, που δείχνουν πού βρίσκονται τα ΣΕ που βρέθηκαν στην περιοχή. Όταν ο χρήστης πατήσει σε μία από τις επισημάνσεις εμφανίζεται ένα παραθυράκι με τις πληροφορίες του ΣΕ σαν αντό του σχήματος 5.14. Μέσα σε αυτό το παράθυρο που εμφανίζεται υπάρχει και ένα κουμπί “Comments” το οποίο όταν πατηθεί εμφανίζει τα σχόλια που έχουν αναρτηθεί για το συγκεκριμένο ΣΕ στο πλαίσιο που βρίσκεται κάτω από τον χάρτη. Πιο κάτω υπάρχει κουμπί για καινούργια αναζήτηση “New Search”.



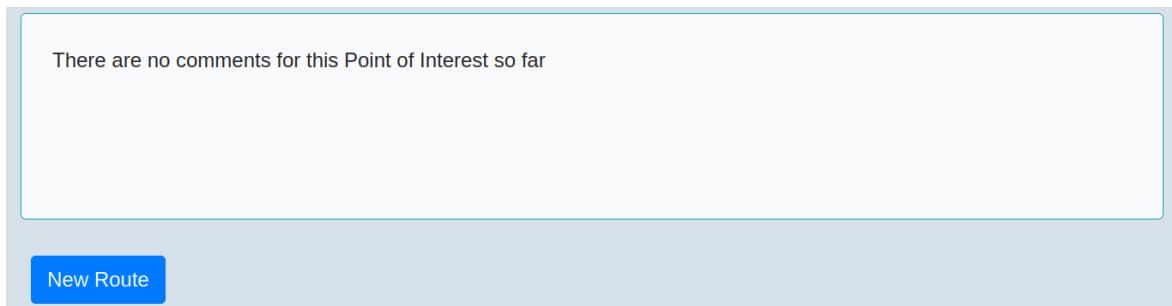
Σχήμα 5.13: Σελίδα αποτελέσματος αναζήτησης σε ακτίνα



Σχήμα 5.14: Παράθυρο με τις πληροφορίες του Σημείου Ενδιαφέροντος



Σχήμα 5.15: Σχόλια Σημείου Ενδιαφέροντος

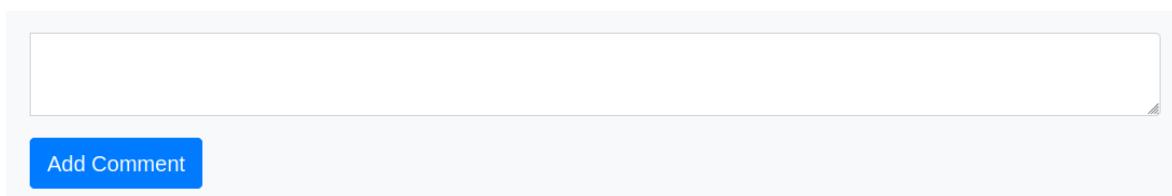


Σχήμα 5.16: Πλαίσιο σχολίων σε μη παρουσία σχολίων

5.4.1 Σχόλια

Όταν ο χρήστης πατήσει το κουμπί εμφάνισης των σχολίων, στο πλαίσιο που αρχικά είχε την επιγραφή “Select point of interest to view comments”, αν υπάρχουν σχόλια, θα εμφανιστούν όπως φαίνονται στο σχήμα 5.15. Πρώτα αναγράφεται το όνομα χρήστη και η ημερομηνία που προστέθηκε το σχόλιο, από κάτω τα περιεχόμενα του σχολίου, και ομοίως ακολουθούν τα επόμενα σχόλια. Αν δεν έχουν υποβληθεί σχόλια ακόμα τότε υπάρχει απλά το μήνυμα “There are no comments for this Point of Interest so far” (Σχήμα 5.16).

Αν ο χρήστης είναι συνδεδεμένος σε λογαριασμό, τότε με την εμφάνιση των σχολίων εμφανίζεται ένα πλαίσιο για υποβολή καινούργιου σχολίου όπως φαίνεται στο σχήμα 5.17. Εκεί ο χρήστης μπορεί να γράψει σχόλιο για το συγκεκριμένο ΣΕ. Όταν τελικά το υποβάλει, γίνεται επαναφόρτωση των σχολίων και εμφανίζεται και το καινούργιο σχόλιο, όπως στο



Σχήμα 5.17: Πλαίσιο υποβολής καινούργιου σχολίου

παράδειγμα στα σχήματα 5.18 και 5.19.

Nice lighting at night!

Add Comment

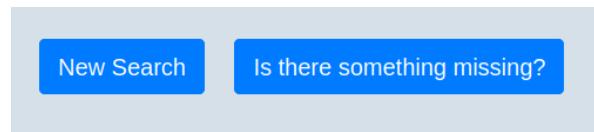
There are no comments for this Point of Interest so far

Σχήμα 5.18: Παράδειγμα υποβολής σχολίου

User: Admin Date: 2025-04-30 10:33

Nice lighting at night

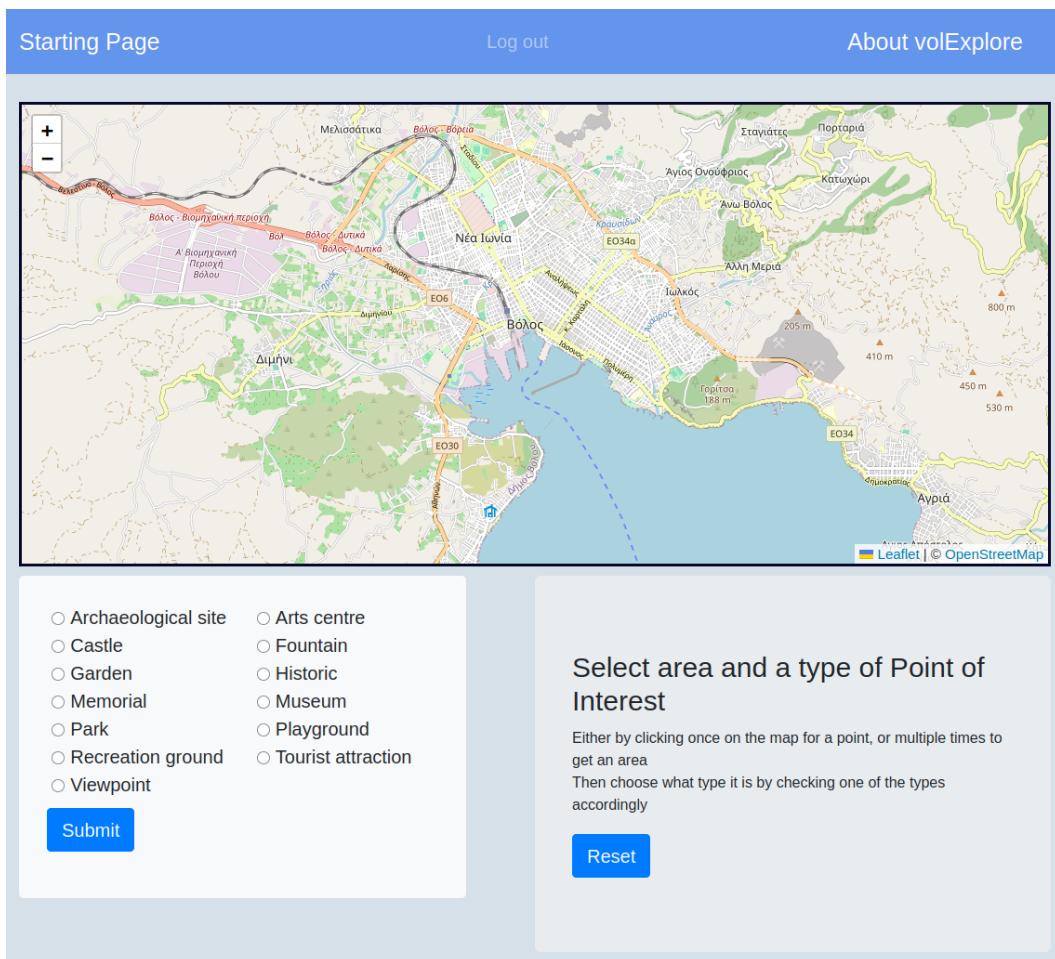
Σχήμα 5.19: Συνέχεια παραδείγματος υποβολής σχολίου



Σχήμα 5.20: Κουμπιά στη σελίδα αποτελέσματος αναζήτησης σε ακτίνα ως διαχειριστής

5.4.2 Διαχειριστής και Ενημέρωση Σημείου Ενδιαφέροντος

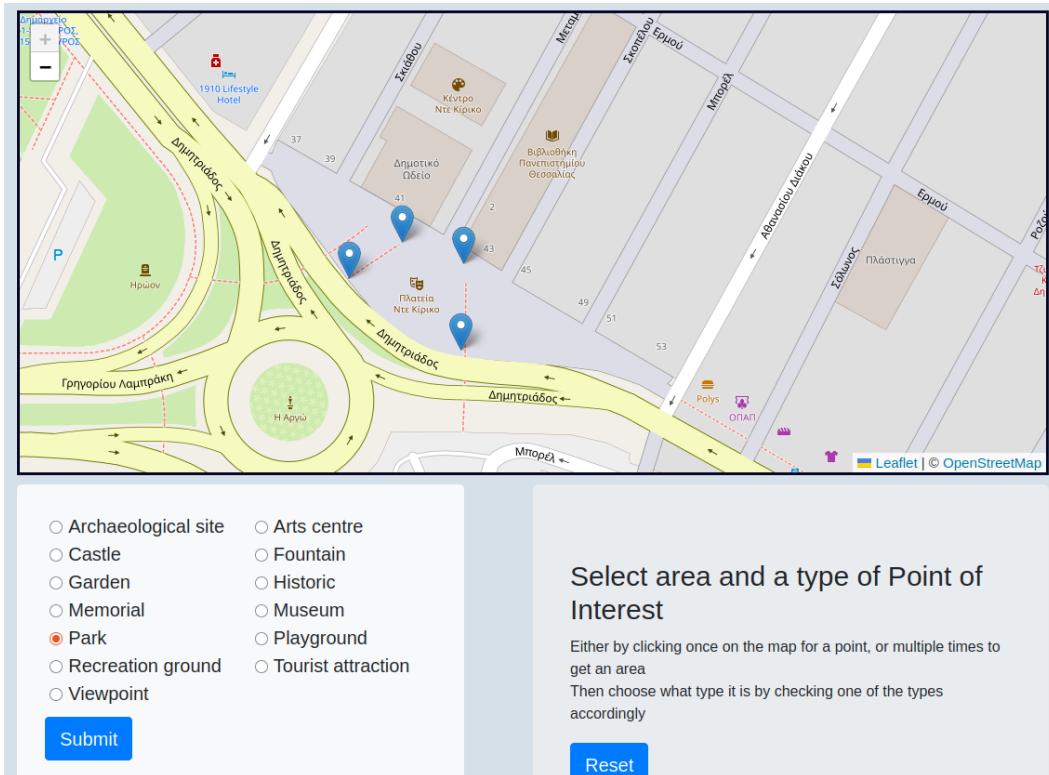
Αν ο χρήστης είναι συνδεδεμένος με λογαριασμό διαχειριστή, εκτός από τις δυνατότητες που παρέχονται σε απλούς χρήστες, στη σελίδα με τα αποτελέσματα της αναζήτησης σε ακτίνα εμφανίζεται και ένα επιπλέον κουμπί “Is there something missing?” (Σχήμα 5.20).



Σχήμα 5.21: Σελίδα για προσθήκη ή ενημέρωση Σημείου Ενδιαφέροντος

Όταν πατηθεί, ανακατευθύνεται σε μία καινούργια σελίδα, η οποία έχει παρόμοια μορφή με την αρχική της αναζήτησης (Σχήμα 5.21). Εκεί ο χρήστης μπορεί είτε πατώντας στον χάρτη μία φορά για ένα σημείο, είτε πολλές φορές για να σχηματίσει πολύγωνο, να δείξει που βρίσκεται το ΣΕ που θέλει να προσθέσει, ή να ενημερώσει ελλιπείς πληροφορίες. Από κάτω

αριστερά υπάρχει μία λίστα με τα είδη ΣΕ. Ο χρήστης θα πρέπει να επιλέξει ένα από αυτά για να υποδείξει τι είναι αυτό που θέλει να ενημερώσει. Από τα δεξιά υπάρχει ένα πλαίσιο που περιγράφει το πώς δουλεύει η σελίδα αυτή, και ένα κουμπί “Reset” για την ανανέωση της σελίδας και προσθήκη διαφορετικών στοιχείων. Στο σχήμα 5.22 φαίνεται η σελίδα σε χρήση.



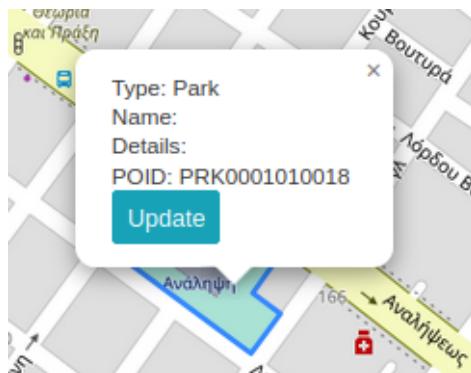
Σχήμα 5.22: Σελίδα ενημέρωσης σε χρήση

Όταν ο χρήστης πατήσει το κουμπί υποβολής ανακατευθύνεται σε καινούργια σελίδα. Εκτός από το πλαίσιο του χάρτη, υπάρχουν δύο επιπλέον πλαίσια, ένα για προσθήκη πληροφοριών για καινούργιο ΣΕ, και ένα για ενημέρωση πληροφοριών (Σχήμα 5.23).

The screenshot shows a user interface for managing points of interest. On the left, there's a form for creating a new POI, featuring input fields for the name and details, and a blue 'Submit new Point of Interest' button. On the right, there's a placeholder box with the text 'Select point of interest to update'.

Σχήμα 5.23: Πλαίσια ενημέρωσης πληροφοριών ΣΕ

Στον χάρτη, αν υπάρχουν προϋπάρχοντα ΣΕ κοντά στην περιοχή που σήμανε ο χρήστης, επισημαίνονται. Αν ο χρήστης πατήσει πάνω σε ένα από αυτά εμφανίζεται ένα παράθυρο σαν αυτό του σχήματος 5.24.



Σχήμα 5.24: Παράθυρο στοιχείων για ενημέρωση

Αν κανένα από τα ΣΕ που εμφανίζονται δεν αντιστοιχούν με αυτό που αναζητούσε ο χρήστης, μπορεί να προσθέσει ένα καινούργιο. Θα πρέπει στη φόρμα υποβολής καινούργιου ΣΕ που βρίσκεται κάτω αριστερά από τον χάρτη να υποβάλει όνομα και πληροφορίες (Σχήμα 5.25). Ο τύπος ΣΕ έχει ήδη επιλεχθεί στην προηγούμενη σελίδα στο φίλτρο.

Αν θέλει ο χρήστης να ενημερώσει ένα από αυτά τα ΣΕ, το πλαίσιο κάτω δεξιά από τον χάρτη με την επιγραφή “Select point of interest to update” ανανεώνεται όταν ο χρήστης πατάει το κουμπί “Update” και εμφανίζει η φόρμα ενημέρωσης στοιχείων (Σχήματα 5.26 και 5.27).

Name of new Point of Interest

Details of new Point of Interest:

Submit new Point of Interest

Σχήμα 5.25: Προσθήκη νέου ΣΕ

Update Name

Update Details:

Submit Update

Update Name

Update Details:

Submit Update

Σχήμα 5.26: Πλαίσιο ενημέρωσης ΣΕ

Σχήμα 5.27: Πλαίσιο ενημέρωσης ΣΕ σε χρήση

Με την υποβολή των στοιχείων, προστίθενται τα καινούργια δεδομένα και ο χρήστης ανακατευθύνεται στην αρχική σελίδα με μήνυμα επιτυχίας (Σχήμα 5.28).

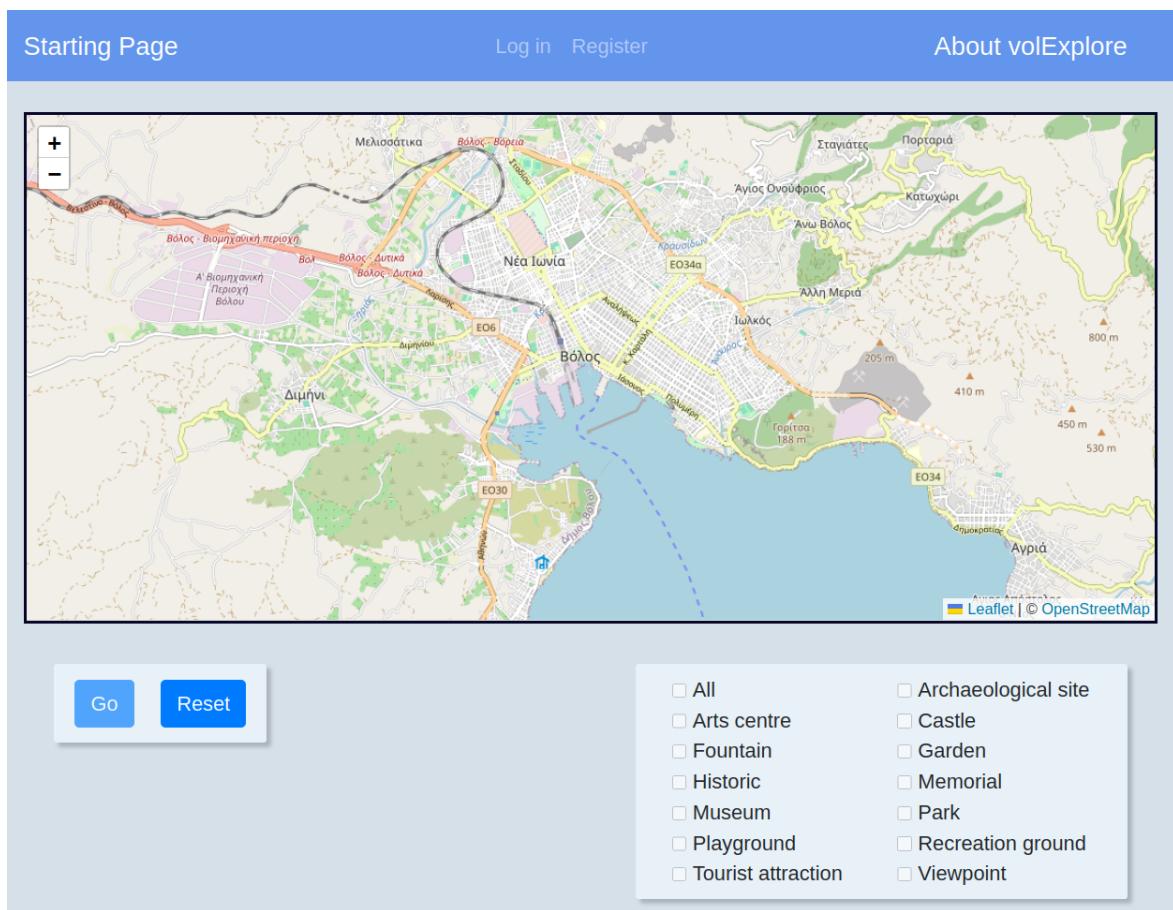
OK: Changes in Point of Interest successful



Σχήμα 5.28: Μήνυμα επιτυχίας προσθήκης ή ενημέρωσης Σημείου Ενδιαφέροντος

5.5 Search in Path

Στην αρχική σελίδα, αν ο χρήστης πατήσει το κουμπί “Search in Path” ανακατευθύνεται στη σελίδα αναζήτησης διαδρομής. Στην κορυφή της σελίδας υπάρχει η μπάρα με τους συνδέ-



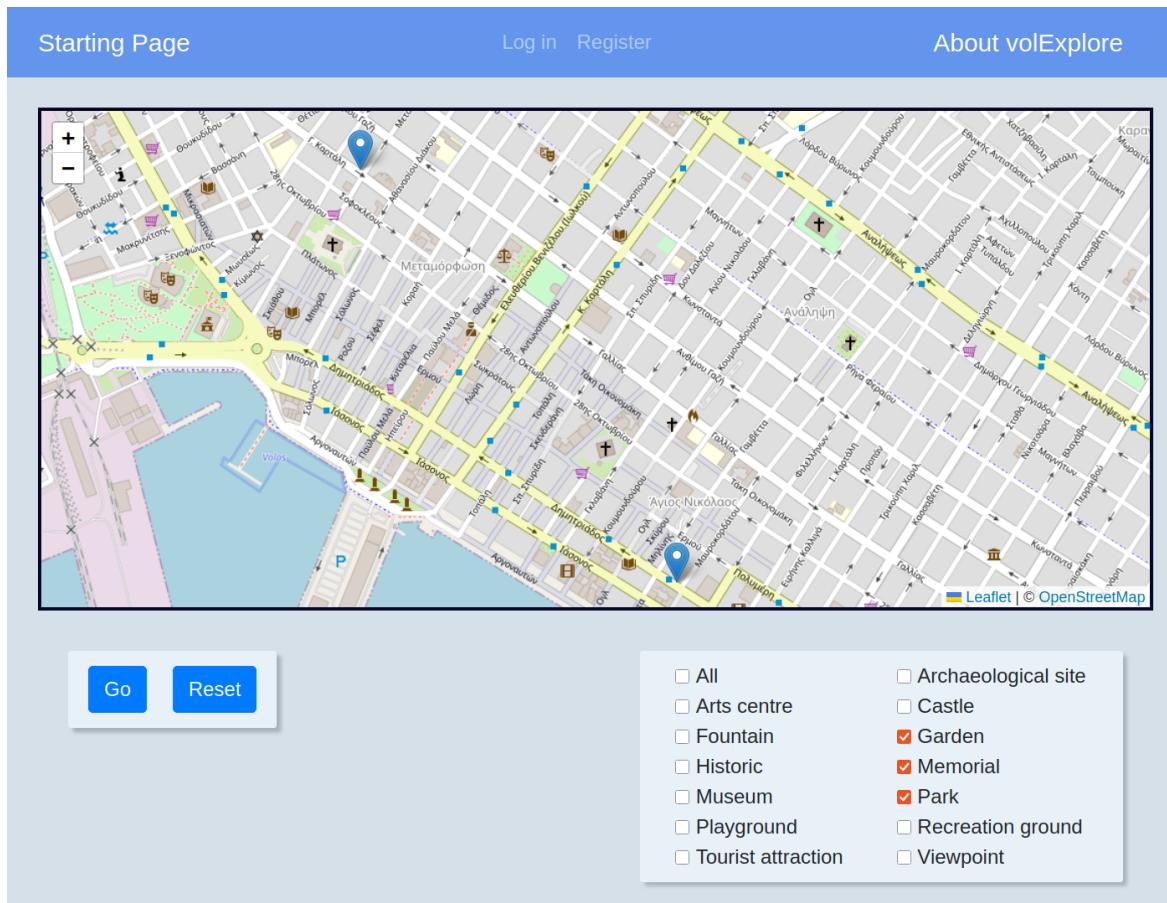
Σχήμα 5.29: Σελίδα αναζήτησης διαδρομής “Search in Path”

σμους για την αρχική σελίδα, τη σελίδα πληροφοριών, και συνδέσμους είτε εγγραφής και εισόδου, είτε αποσύνδεσης ανάλογα με το αν ο χρήστης είναι συνδεδεμένος σε λογαριασμό ή όχι. Κάτω από το πλαίσιο του χάρτη και δεξιά βρίσκεται η λίστα με τα είδη ΣΕ, και αριστερά δύο κουμπιά, ένα υποβολής “Go” και ένα ανανέωσης της σελίδας για υποβολή καινούργιων στοιχείων “Reset” όπως φαίνεται στο σχήμα 5.29.

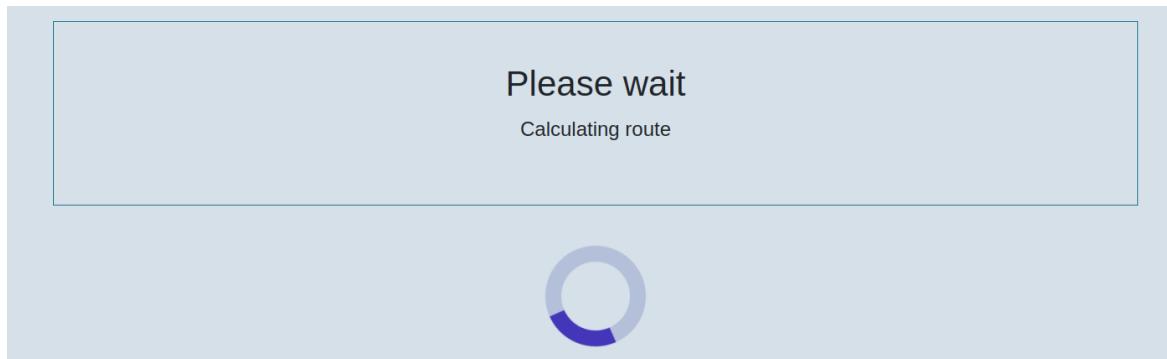
Για να χρησιμοποιήσει τη σελίδα, ο χρήστης επιλέγει στον χάρτη δύο σημεία, ένα για να μαρκάρει την αρχή της διαδρομής που θέλει, και ένα για το τέλος. Στη λίστα επιλέγει, παρόμοια με το πώς περιγράψαμε στην παράγραφο 5.4, τα είδη των ΣΕ από τα οποία θέλει να περάσει. Όταν επιλεχθούν, μπορεί να πατήσει το κουμπί αναζήτησης. Στο σχήμα 5.30 φαίνεται παράδειγμα χρήσης της σελίδας.

Όταν υποβληθούν τα στοιχεία, όσο υπολογίζεται η διαδρομή εμφανίζεται το μήνυμα “Please wait Calculating path” και ένα εικονίδιο φόρτωσης (Σχήμα 5.31).

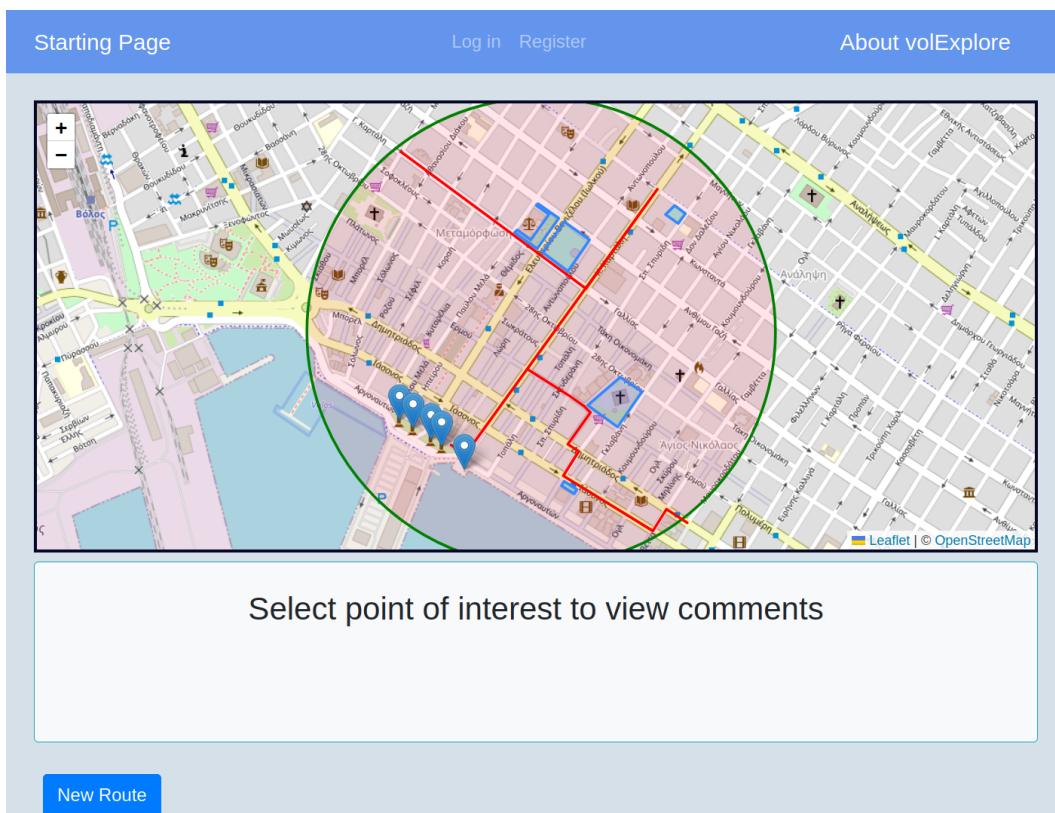
Όταν υπολογισθεί η διαδρομή, εμφανίζεται στον χάρτη ως κόκκινη γραμμή, με κόκκινο κύκλο η περιοχή όπου αναζητήθηκαν ΣΕ, καθώς και όλα τα ΣΕ που βρίσκονται σε αυτήν



Σχήμα 5.30: Σελίδα αναζήτησης διαδρομής σε χρήση

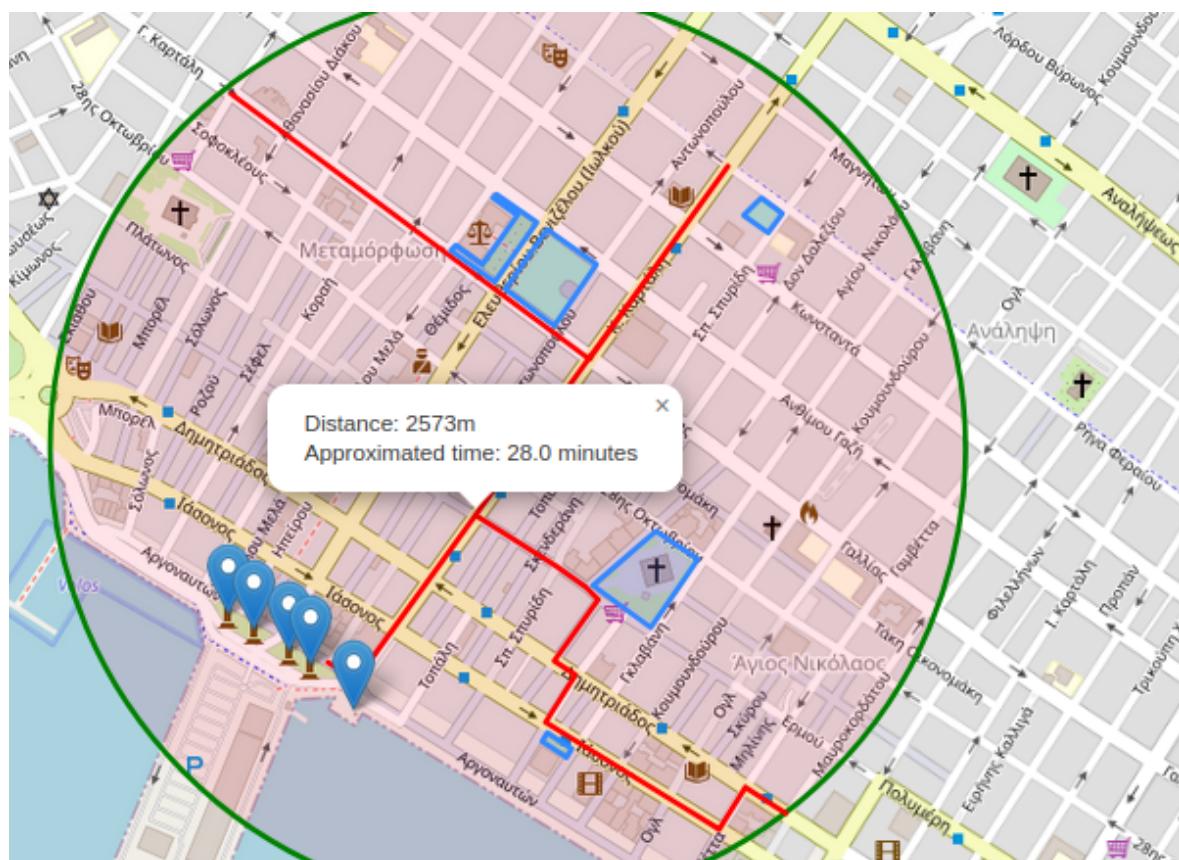


Σχήμα 5.31: Σελίδα αναμονής υπολογισμού διαδρομής



Σχήμα 5.32: Αποτέλεσμα αναζήτησης διαδρομής

και από τα οποία μπορεί να μεταβεί, όπως φαίνεται στο σχήμα 5.32. Αν ο χρήστης πατήσει πάνω στη γραμμή διαδρομής, εμφανίζεται ένα παράθυρο που λέει το μήκος της διαδρομής σε μέτρα και πόσο περίπου χρόνο χρειάζεται για να περπατηθεί σε λεπτά (Σχήμα 5.33). Ομοίως με την παράγραφο 5.4, όταν πατηθεί κάποια σήμανση ΣΕ, εμφανίζονται τα χαρακτηριστικά του.



Σχήμα 5.33: Παράθυρο πληροφοριών απόστασης και χρόνου

Κεφάλαιο 6

Συμπεράσματα

Στην παρούσα διπλωματική εργασία έχουμε εξερευνήσει θέματα γεωχωρικών ερωτημάτων για επίσκεψη/εξερεύνηση σε μία πόλη. Επιπρόσθετα μία εφαρμογή ιστοτόπου υλοποιήθηκε για την ευκολότερη επικοινωνία του χρήστη με τη βάση, και την παρουσίαση των αποτελεσμάτων των αλγορίθμων μας. Σε αυτό το κεφάλαιο, θα συνοψίσουμε τη δουλειά μας, θα συζητήσουμε τα αποτελέσματά μας και θα μιλήσουμε για μελλοντικές επεκτάσεις.

6.1 Σύνοψη και συμπεράσματα

Ο κύριος στόχος της διπλωματικής εργασίας μας ήταν η δημιουργία εφαρμογής για την διεύκρινη της εξερεύνησης της πόλης του Βόλου. Για τον σκοπό αυτό, συλλέξαμε ανοιχτού τύπου, γεωχωρικά δεδομένα από το OpenStreetMap, τα οποία επεξεργαστήκαμε και αποθηκεύσαμε σε μία βάση δεδομένων, στη MongoDB. Για την υλοποίηση ενός από τα γεωχωρικά ερωτήματα που απευθύνθηκαν στη βάση, αναπτύχθηκε ευριστικός αλγόριθμος ο οποίος δίνει μία προσεγγιστική λύση για το Πρόβλημα του Περιοδεύοντος Πωλητή (TSP), συγκεκριμένα για την εύρεση διαδρομής που περνάει από όλα τα σημεία ενδιαφέροντος σε μία περιοχή. Τέλος δημιουργήσαμε έναν ιστότοπο μέσω του οποίου παρουσιάζουμε τα γεωχωρικά δεδομένα και τις διαδρομές με απλό και κατανοητό τρόπο στον χρήστη. Μέσω αυτού, ο χρήστης έχει τη δυνατότητα να δημιουργήσει λογαριασμό και να αφήσει σχόλια σε κάποιο σημείο ενδιαφέροντος που επισκέπτεται, ενώ, επιπλέον, σε περίπτωση που είναι διαχειριστής, μπορεί να πραγματοποιήσει αλλαγές στη βάση.

Η εφαρμογή μας μπορεί να προσφέρει σημαντικές πληροφορίες για ένα σημείο ενδιαφέροντος αλλά δυστυχώς κάποια δεδομένα είναι ελλειπή. Επίσης ο αλγόριθμος εύρεσης μονο-

πατιού δεν είναι ο βέλτιστος εξαιτίας της προσεγγιστικής του φύσης, και έχει το μειονέκτημα ότι μπορεί να εφαρμοστεί μόνο σε μικρές και όχι σε πολύ μεγάλες διαδρομές, των οποίων η αρχή και το τέλος πρέπει να έχουν απόσταση μικρότερη των 2000 μέτρων.

6.2 Μελλοντικές επεκτάσεις

Παρόλο που η εργασία μας προσεγγίζει μια σειρά γεωχωρικών ερωτημάτων που σχετίζονται με την εξερεύνηση μιας πόλης, δεν μπορεί να θεωρηθεί ότι μελετήθηκε και αναλύθηκε το πρόβλημα πλήρως. Υπάρχει δυνατότητα περαιτέρω βελτιστοποίησης στην προσέγγιση ενώ παράλληλα και η ίδια η εφαρμογή θα μπορούσε να επεκταθεί.

Κάποιες προτάσεις για μελλοντικές επεκτάσεις είναι οι ακόλουθες:

1. Συμπερίληψη περισσότερων πόλεων πέραν αυτής του Βόλου.
2. Βελτιστοποίηση του αλγορίθμου εύρεσης μονοπατιού ως προς τη χρήση μνήμης προκειμένου να βελτιωθεί η δυνατότητα εύρεσης διαδρομών μεγαλύτερης απόστασης.
3. Διεύρυνση της λίστας των ειδών των σημείων ενδιαφέροντος και προσθήκη περισσότερων.
4. Συμπλήρωση του σετ δεδομένων από το OpenStreetMap καθώς στην τωρινή του μορφή, τουλάχιστον για την πόλη του Βόλου, είναι ελλειπές.
5. Προσθήκη λειτουρίας “upvote” στα σχόλια των χρηστών για να μπορούν άλλοι χρήστες να δηλώσουν πόσο βοηθητικό ήταν κάποιο σχόλιο.

Βιβλιογραφία

- [1] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.
- [2] Verka Jovanović and Angelina Njeguš. The application of GIS and its components in tourism. *Yugoslav journal of operations research*, 18(2):261–272, 2008.
- [3] Z Nedović-Budić. The impact of GIS technology. *Environment and planning B: Planning and Design*, 25(5):681–692, 1998.
- [4] Anthony GO Yeh. Urban planning and GIS. *Geographical information systems*, 2(877-888):1, 1999.
- [5] Sara L McLafferty. GIS and health care. *Annual review of public health*, 24(1):25–42, 2003.
- [6] Google Maps. <https://www.google.com/maps/>. Ημερομηνία πρόσβασης: 09-06-2025.
- [7] OpenStreetMap. <https://www.openstreetmap.org/>. Ημερομηνία πρόσβασης: 04-04-2025.
- [8] Citymapper. <https://citymapper.com/>. Ημερομηνία πρόσβασης: 09-06-2025.
- [9] Γιαννίκογλου Καλλιόπη. Χαρτογράφηση διαδρομών μιας πόλης με χρήση του OpenStreetMap και σχεδιασμός βέλτιστων διελεύσεων. Διπλωματική εργασία, Πανεπιστήμιο Θεσσαλίας, Ιούνιος 2024.
- [10] OpenStreetMap Wiki. https://wiki.openstreetmap.org/wiki/Main_Page. Ημερομηνία πρόσβασης: 04-04-2025.

- [11] OpenStreetMap Foundation. https://osmfoundation.org/wiki/Main_Page. Ημερομηνία πρόσβασης: 06-04-2025.
- [12] OpenStreetMap Copyright. <https://www.openstreetmap.org/copyright>. Ημερομηνία πρόσβασης: 06-04-2025.
- [13] Geoff Boeing. Modeling and Analyzing Urban Networks and Amenities with OSMnx. *Working paper. URL: https://geoffboeing.com/publications/osmnx-paper/*, 2024.
- [14] MongoDB. <https://www.mongodb.com/>. Ημερομηνία πρόσβασης: 07-04-2025.
- [15] MongoDB - Geospatial Queries. <https://www.mongodb.com/docs/manual/geospatial-queries/>. Ημερομηνία πρόσβασης: 07-04-2025.
- [16] What Is NoSQL? <https://www.mongodb.com/resources/basics/databases/nosql-explained>. Ημερομηνία πρόσβασης: 07-04-2025.
- [17] Jing Han, Haihong E, Guan Le, and Jian Du. Survey on NoSQL database. In *2011 6th International Conference on Pervasive Computing and Applications*, pages 363–366, 2011.
- [18] Ameya Nayak, Anil Poriya, and Dikshay Poojary. Type of NOSQL databases and its comparison with relational databases. *International Journal of Applied Information Systems*, 5(4):16–19, 2013.
- [19] What is a Document Database? <https://www.mongodb.com/resources/basics/databases/document-databases>. Ημερομηνία πρόσβασης: 07-04-2025.
- [20] Python. <https://www.python.org/about/>. Ημερομηνία πρόσβασης: 07-04-2025.
- [21] PyMongo. <https://pypi.org/project/pymongo/>. Ημερομηνία πρόσβασης: 07-04-2025.
- [22] NetworkX. <https://networkx.org/>. Ημερομηνία πρόσβασης: 07-04-2025.

- [23] Flask. <https://flask.palletsprojects.com/en/stable/>. Ημερομηνία πρόσβασης: 08-04-2025.
- [24] bcrypt. <https://www.npmjs.com/package/bcrypt>. Ημερομηνία πρόσβασης: 16-04-2025.
- [25] random — Generate pseudo-random numbers. <https://docs.python.org/3/library/random.html>. Ημερομηνία πρόσβασης: 16-04-2025.
- [26] datetime — Basic date and time types. <https://docs.python.org/3/library/datetime.html>. Ημερομηνία πρόσβασης: 16-04-2025.
- [27] ast — Abstract Syntax Trees. <https://docs.python.org/3/library/ast.html>. Ημερομηνία πρόσβασης: 16-04-2025.
- [28] copy — Shallow and deep copy operations. <https://docs.python.org/3/library/copy.html>. Ημερομηνία πρόσβασης: 16-04-2025.
- [29] math — Mathematical functions. <https://docs.python.org/3/library/math.html>. Ημερομηνία πρόσβασης: 16-04-2025.
- [30] jQuery. <https://jquery.com/>. Ημερομηνία πρόσβασης: 09-04-2025.
- [31] Jinja. <https://jinja.palletsprojects.com/en/stable/>. Ημερομηνία πρόσβασης: 16-04-2025.
- [32] Leaflet. <https://leafletjs.com/>. Ημερομηνία πρόσβασης: 09-04-2025.
- [33] Bootstrap. <https://getbootstrap.com/>. Ημερομηνία πρόσβασης: 08-04-2025.
- [34] Gerhard Reinelt. The Traveling Salesman: Computational solutions for TSP Applications. In *Basic Concepts*, pages 4–30. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.
- [35] Chetna Dahiya and Shabnam Sangwan. Literature review on travelling salesman problem. *International Journal of Research*, 5(16):1152–1155, 2018.

- [36] Richard M. Karp. Reducibility among Combinatorial Problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department*, pages 85–103. Springer US, Boston, MA, 1972.
- [37] M. Barbehenn. A note on the complexity of Dijkstra’s algorithm for graphs with weighted vertices. *IEEE Transactions on Computers*, 47(2):263–, 1998.