

Ευφυές Γεωγραφικό Σύστημα Δυναμικής Αναζήτησης Δομών Υγείας

Κουτρούδης Ιωάννης

Διπλωματική Εργασία

Επιβλέπων: Κωνσταντίνος Μπλέκας

Ιωάννινα, Ιούνιος, 2022



Τμήμα Μηχ. Η/Υ & Πληροφορικής
Πανεπιστήμιο Ιωαννίνων

**Department of Computer Science & Engineering
University of Ioannina**

Αφιερώσεις

Αφιερωμένη στην Οικογένεια μου

-

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου κ. Κωνσταντίνο Μπλέκα για την καθοδήγηση του κατά την διάρκεια της εκπόνησης της διπλωματικής εργασίας μου. Ένα μεγάλο ευχαριστώ στην οικογένεια μου και τους φίλους μου που με στήριξαν όλο αυτό το διάστημα με κάθε τρόπο.

-

Περίληψη

Η κατασκευή χαρτών είναι μία από τις αρχαιότερες δραστηριότητες του ανθρώπου. Σύμφωνα με τους αρχαιολόγους τα παλαιότερα σχέδια που έχουν βρεθεί και θα μπορούσαν να χαρακτηριστούν χάρτες χρονολογούνται ακόμα και 30.000 χρόνια πριν. Έχουν δε βρεθεί χαραγμένοι χάρτες σε σπηλιές, σε κομμάτια οστράκου ή σε κομμάτια οστών.

Η υγεία είναι ο παράγοντας που μετράει την φυσική, ψυχολογική ή ακόμα και την πνευματική κατάσταση ενός ζώντος οργανισμού. Σύμφωνα με το καταστατικό του Παγκόσμιου Οργανισμού Υγείας (1946) η υγεία είναι «η κατάσταση της πλήρους σωματικής, ψυχικής και κοινωνικής ευεξίας και όχι μόνο η απουσία ασθένειας ή αναπηρίας». Έτσι λοιπόν, η έννοια της υγείας, δεν αποδίδεται μόνο από την ιατρική, αλλά και από άλλους παράγοντες όπως είναι το περιβάλλον, η οικονομία, η εργασία κ.α.

Στην παρούσα διπλωματική εργασία ασχολούμαστε με την χαρτογράφηση των δομών υγείας στον Ελλαδικό χώρο. Η διαδικασία αυτή αποτυπώνεται σε μια ιστοσελίδα χρησιμοποιώντας κάποια κριτήρια όπως περιοχή ,κατηγορία, αριθμός κλινών για τη χαρτογράφηση των νοσοκομείων καθώς και δήμος και περιοχή για τη χαρτογράφηση των ιατρικών διαγνωστικών κέντρων τα οποία ανακτώνται μέσω μιας βάσης δεδομένων που κατασκευάσαμε και περιέχει τις αντίστοιχες πληροφορίες.

Ο αριθμός των νοσοκομείων αντιστοιχείται στα 139 ενώ ο αριθμός των ιατρικών διαγνωστικών κέντρων είναι μεταβαλλόμενος σύμφωνα με τον αριθμό των δεδομένων που υπάρχουν στην ιστοσελίδα που εκτελούμε το web scraping. Στη βάση δεδομένων μας υπάρχουν τρία tables τα οποία περιέχουν τις πληροφορίες για τα νοσοκομεία ,τα διαγνωστικά κέντρα και κάποιες αξιολογήσεις αστεριών που εκτελούμε για τα νοσοκομεία.

Στην πρώτη φάση πριν ανοίξουμε τον σέρβερ εκτελούμε την διαδικασία του web scraping για να ανακτήσουμε τα νέα δεδομένα σε περίπτωση που υπάρχουν. Σε περίπτωση που δεν υπάρχουν το πρόγραμμα εκτελείται μετά την εκτέλεση του web scraping δουλεύοντας με τα ήδη υπάρχοντα δεδομένα που είναι περασμένα στην βάση μας.

Στην δεύτερη φάση αφότου τρέξουμε τον σέρβερ βλέπουμε στο πάνω μέρος της ιστοσελίδας ένα μενού 5 επιλογών. Κάθε επιλογή αντιστοιχεί σε μία λειτουργία.

Στην τρίτη φάση επιλέγοντας κάποια από τις επιλογές 3-4-5(Search For Hospitals-Search For Diagnostic Centers-Rate Hospitals) φέρνουμε τα δεδομένα στην αντίστοιχη ιστοσελίδα ώστε να μπορούμε να επιλέξουμε κριτήρια.

Στην τέταρτη φάση αφότου επιλέξουμε τα κριτήρια στέλνουμε τα δεδομένα που επιλέχθηκαν στο backend ώστε να εκτελέσουμε κάποιες λειτουργίες που αποσκοπούν στην χαρτογράφηση και άσκηση κριτικής.

Στην πέμπτη φάση έχοντας εκτελέσει τις λειτουργίες αυτές στέλνουμε τα επεξεργασμένα δεδομένα ξανά στην ιστοσελίδα μας ώστε να επιτευχθεί η χαρτογράφηση και η κριτικές.

Επιπρόσθετα, στο τελικό στάδιο εκτελούμε σε καθεμία από τις σελίδες την τελική της λειτουργία η οποία θα είναι είτε εύρεση διαδρομών ανάμεσα σε δύο σημεία είτε εμφάνιση μέσου όρου αστεριών των αξιολογήσεων αντίστοιχα.

Με την χαρτογράφηση αυτή δίνεται η ευκολία να αντιληφθούμε την τοποθεσία των αντίστοιχων δομών υγείας που αναζητούμε καθώς και την κοντινότερη διαδρομή αυτών από την τοποθεσία μας. Για την εύρεση την απόστασης αυτής χρησιμοποιήσαμε τον τύπο της απόστασης μεταξύ δύο σημείων του Καρτεσιανού επιπέδου ($AB = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$) όπου x_2, y_2 είναι οι συντεταγμένες του ενός σημείου και x_1, y_1 οι συντεταγμένες του δεύτερου σημείου.

Λέξεις Κλειδιά: Χαρτογράφηση, δομές υγείας, δεδομένα ,ομαδοποίηση, απόσταση σημείων

-

Abstract

Making maps is one of the most ancient human activities. According to archaeologists the oldest designs that have been found and could be characterized as maps date back as far as 30.000 years ago. They have been found engraved in caves, in pieces of shell or in pieces of bone.

Health is the factor that measures the physical, psychological or even mental state of a living organism. According to statute of the World Health Organization (1946) health is “the state of complete physical, mental and social well-being and not just the absence of illness or disability”. Thus, the concept of health is not only attributes by medicine, but also by other factors such as the environment, the economy, work, etc.

In the present dissertation we deal with mapping of health structures in Greece.

This process is captured on a website using some criteria such as area, category, number of beds to map hospitals as well as municipality and area for the mapping of medical diagnostic centers which are retrieved through a database that we have constructed and contains the corresponding information.

The number of hospitals corresponds to 139 while the number of medical diagnostic centers varies according to the number of data available on the website where we perform web scraping.

In our database there are three tables which contain the information about the hospitals, the diagnostic centers, and some star ratings that we perform for the hospitals.

In the first phase before opening the server we perform the web scraping process to retrieve the new data in case there are. In case they do not exist, the program is run after

the web scraping is performed, working with the already existing data that is passed to our database.

In the third phase, by selecting one of the options 3-4-5 (Search for Hospitals-Search For Diagnostic Centers-Rate Hospitals) we bring the data to the respective website so that we can select criteria.

In the fourth phase, after selecting the criteria, we send the selected data to the backend in order to perform some functions aimed at mapping and critique.

In the fifth phase, having performed these functions, we send the processed data back to our website in order to achieve mapping and reviews.

In addition, in the final stage we perform in each of the pages its final function which will be either finding routes between two points or displaying an average of stars of the evaluations, respectively.

This mapping makes it easy to understand the location of the respective health structures we are looking for as well as the closest route to them from our location. To find this distance we used the type of distance between two points of the Cartesian plane $AB = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ where x_2, y_2 are the coordinates of one point and x_1, y_1 are the coordinates of the second point.

Keywords: Mapping, health structures, data, grouping, point distance

• Πίνακας εικόνων

Εικόνα 1 Εκτέλεση εφαρμογής	17
Εικόνα 2 Αρχεία Εργασίας	18
Εικόνα 3 Import 1.....	19
Εικόνα 4 Import 2.....	20
Εικόνα 5 Import 3.....	20
Εικόνα 6 Import 4.....	20
Εικόνα 7 Import 5.....	20
Εικόνα 8 Schema της βάσης.....	25
Εικόνα 9 Δημιουργία και σύνδεση βάσης.....	26
Εικόνα 10 CSV Νοσοκομείων.....	27
Εικόνα 11 CSV Διαγνωστικών Κέντρων.....	27
Εικόνα 12 Η βάση μέσω SQLiteStudio.....	30
Εικόνα 13 Data Νοσοκομείων.....	31
Εικόνα 14 Data για τις κριτικές.....	31
Εικόνα 15 Data Διαγνωστικών Κέντρων.....	31
Εικόνα 16 Σύνδεση βάσης.....	32
Εικόνα 17 Αρχεία για δημιουργία του site.....	36
Εικόνα 18 Jinja Contents.....	37
Εικόνα 19 Δημιουργία app.....	37
Εικόνα 20 Παράδειγμα Δημιουργίας Route.....	38
Εικόνα 21 Παράδειγμα Υλοποίησης Ajax.....	41

<i>Εικόνα 22 Ενδεχόμενο Επιτυχίας Ajax.....</i>	<i>42</i>
<i>Εικόνα 23 Κλήση jsonify.....</i>	<i>42</i>
<i>Εικόνα 24 Ομαδοποίηση Δεδομένων για πλαίσια επιλογών 1.....</i>	<i>44</i>
<i>Εικόνα 25 Παράδειγμα Υλοποίησης Web Scraping.....</i>	<i>45</i>
<i>Εικόνα 26 Ομαδοποίηση Δεδομένων για πλαίσια επιλογών 2.....</i>	<i>46</i>
<i>Εικόνα 27 Σενάρια για πλαίσια επιλογών 1.....</i>	<i>46</i>
<i>Εικόνα 28 Ομαδοποίηση Δεδομένων για πλαίσια επιλογών 3.....</i>	<i>47</i>
<i>Εικόνα 29 Εκ νέου ομαδοποίηση δεδομένων για κριτικές.....</i>	<i>47</i>
<i>Εικόνα 30 Σενάρια για πλαίσια επιλογών 2.....</i>	<i>47</i>
<i>Εικόνα 31 Υλοποίηση web scraping για συντεταγμένες.....</i>	<i>48</i>
<i>Εικόνα 32 Βασικός χάρτης της ιστοσελίδας μας.....</i>	<i>49</i>
<i>Εικόνα 33 Δημιουργία των markers για νοσοκομεία.....</i>	<i>50</i>
<i>Εικόνα 34 Δημιουργία των markers για διαγνωστικά κέντρα.....</i>	<i>50</i>
<i>Εικόνα 35 Υπολογισμός Απόστασης Σημείων.....</i>	<i>51</i>
<i>Εικόνα 36 Υλοποίηση Δρομολόγησης.....</i>	<i>51</i>
<i>Εικόνα 37 Πρώτο στάδιο άσκησης κριτικής.....</i>	<i>53</i>
<i>Εικόνα 38 Δεύτερο στάδιο άσκησης κριτικής.....</i>	<i>54</i>
<i>Εικόνα 39 Χαρτογράφηση Νοσοκομείων 1.....</i>	<i>55</i>
<i>Εικόνα 40 Χαρτογράφηση Νοσοκομείων 2.....</i>	<i>55</i>
<i>Εικόνα 41 Χαρτογράφηση Νοσοκομείων 3.....</i>	<i>56</i>
<i>Εικόνα 42 Χαρτογράφηση Νοσοκομείων 4.....</i>	<i>56</i>
<i>Εικόνα 43 Χαρτογράφηση Νοσοκομείων 5.....</i>	<i>57</i>
<i>Εικόνα 44 Χαρτογράφηση Νοσοκομείων 6.....</i>	<i>57</i>
<i>Εικόνα 45 Χαρτογράφηση Διαγνωστικών Κέντρων 1.....</i>	<i>58</i>
<i>Εικόνα 46 Χαρτογράφηση Διαγνωστικών Κέντρων 2.....</i>	<i>58</i>
<i>Εικόνα 47 Χαρτογράφηση Διαγνωστικών Κέντρων 3.....</i>	<i>59</i>

- ## Εισαγωγή

- ### Στόχοι της Διπλωματικής Εργασίας

Βασικός στόχος της παρούσας διπλωματικής εργασίας είναι η επίτευξη της χαρτογράφησης ενός σημαντικού φορέα που δεν είναι άλλος από αυτόν της υγείας με σκοπό την διευκόλυνση εύρεσης της καταλληλότερης δομής για την αντιμετώπιση των εκάστοτε παθήσεων από πλευράς νοσοκομείων αλλά και εκπόνησης διαφόρων εξετάσεων για την βέλτιστη ακεραιότητα της σωματικής και ψυχικής υγείας του ανθρώπου. Ο βασικός στόχος της εργασίας αναλύεται με τους επιμέρους στόχους:

- Επικοινωνία frontend & backend
- Ομαδοποίηση δεδομένων μέσω κριτηρίων
- Εκμάθηση λειτουργίας ιστοσελίδας
- Χρήση βάσης δεδομένων με δυναμικό αλλά και στατικό τρόπο
- Ανάκτηση πληροφοριών μέσω web scraping
- Απεικόνιση δεδομένων πάνω στον χάρτη
- Λειτουργίες των δεδομένων

- ### Περιεχόμενα της Διπλωματικής Εργασίας

Το περιεχόμενο της παρούσας διπλωματικής εργασίας αποτελείται από 6 κεφάλαια.

Στο δεύτερο κεφάλαιο παρουσιάζεται η έννοια και ο ορισμός της χαρτογράφησης καθώς γίνεται και μια αναφορά στο ιστορικό της κομμάτι. Στη συνέχεια γίνεται μια ανάλυση των κλάδων της χαρτογράφησης και επακολουθεί μια σύντομη αναφορά σε αυτά διότι περιέχουν στοιχεία που αφορούν αναφορικά την υλοποίηση της εργασίας μας.

Στο τρίτο κεφάλαιο αναφέρεται η έννοια αλλά και κάποια παραδείγματα δομών υγείας και στην συνέχεια πραγματοποιείται μια διακλάδωση των δύο κατηγοριών υγείας, Δημόσια και Ιδιωτική και γίνεται μια αναφορά σε αυτές.

Στο τέταρτο κεφάλαιο γίνεται αναφορά στον περιβάλλον που χρησιμοποιήσαμε για την υλοποίηση της εργασίας καθώς και αναλυτικές πληροφορίες για τις προγραμματιστικές γλώσσες και τις βιβλιοθήκες που χρησιμοποιήσαμε.

Στο πέμπτο κεφάλαιο περιγράφονται τα δεδομένα που χρησιμοποιήσαμε καθώς και το περιβάλλον που υλοποιήσαμε την βάση. Στην συνέχεια αναφέρεται η σύνδεση της βάσης με το backend και τέλος η χρήση του web scraping για την ανάκτηση των δεδομένων από ιστοσελίδα και χρήση των δεδομένων αυτών στην βάση μας.

Στο έκτο κεφάλαιο παρουσιάζεται ο τρόπος δημιουργίας και χρήσης της ιστοσελίδας. Επίσης δίνεται έμφαση στην επικοινωνία μεταξύ backend & frontend. Επίσης γίνεται αναφορά και ανάλυση στις μεθόδους, τρόπους που χρησιμοποιήσαμε.

Στο έβδομο και τελευταίο κεφάλαιο αναφέρεται ο τρόπος ομαδοποίησης των δεδομένων και η διαδικασία μέχρι την απεικόνιση τους στον χάρτη. Τέλος αναλύεται ο τρόπος υπολογισμού της απόστασης μεταξύ των σημείων αλλά και η διαδικασία εύρεσης της μικρότερης απόστασης σε συνδυασμό με την απεικόνιση της διαδρομής μέσω δρόμων(δρομολόγηση).

- # Χαρτογράφηση

- ## Ορισμός και έννοια Χαρτογραφίας

Χαρτογραφία ονομάζεται ο επιστημονικός κλάδος της γεωγραφίας που περιλαμβάνει ένα σύνολο προσδιορισμένων μελετών, τεχνικών, ακόμη και εργασιών που αφορούν απεικονίσεις επάνω σε επίπεδη ή σφαιρική επιφάνεια, σε σμίκρυνση, ενός τμήματος ή όλης της γήινης επιφάνειας για τη σύνταξη και έκδοση χαρτών. Η εφαρμογή όλων αυτών των διεργασιών αποτελούν την έννοια της χαρτογράφησης, που γίνεται με διάφορες μεθόδους οι οποίες και ονομάζονται χαρτογραφικές προβολές. Η Χαρτογραφία σχετίζεται στενά με την επιστήμη της Γεωγραφίας, αφού οι χάρτες είναι ένα από τα κυριότερα μέσα παρουσίασης και μελέτης των γεωγραφικών δεδομένων.

- ## Ιστορία της Χαρτογραφίας

Ένα σημαντικότατο εύρημα της προϊστορικής περιόδου είναι ο χάρτης που βρέθηκε στον οικισμό του Τσατάλ Χογιούκ της Τουρκίας και χρονολογείται γύρω στο 7500 π.Χ.. Ο χάρτης έχει τη μορφή τοιχογραφίας, έχει μήκος περίπου 3 μέτρα και απεικονίζει τον οικισμό σε κάτοψη καθώς και ένα γειτονικό ηφαίστειο. Κάτι ιδιαίτερα αξιοσημείωτο σχετικά με τον χάρτη αυτόν είναι η συμφωνία των όσων απεικονίζονται στον χάρτη με τα ευρήματα των αρχαιολογικών ανασκαφών.

Μια σημαντική εξέλιξη στην χαρτογραφία υπήρξε η εμφάνιση της γεωμετρίας. Άλλωστε η λέξη «γεωμετρία» είχε αρχικά την έννοια της «μέτρησης της γης».

Τα πρώτα δείγματα χαρτών που φαίνεται να κατασκευάστηκαν με τη χρήση κάποιων αρχών γεωμετρίας προέρχονται από την Βαβυλώνα. Τα σημαντικότερα ευρήματα εκείνης της περιόδου είναι ένα διάγραμμα που παρουσιάζει τα τέσσερα σημεία του ορίζοντα (2300 π.Χ.) και ένας χαραγμένος χάρτης της ιερής πόλης Νιππούρ (14ος -12ος αιώνας π.Χ.).

Αξιόλογα ευρήματα προέρχονται ακόμα από την περιοχή της Αιγύπτου. Οι «τοπογράφοι» της εποχής έπρεπε κάθε φορά μετά τις πλημμύρες του Νείλου να επαναπροσδιορίζουν τα όρια των ιδιοκτησιών στο έδαφος. Η ανάγκη για ακρίβεια ώθησε τους αρχαίους Αιγυπτίους στην επινόηση τεχνικών που έδιναν αρκετά αξιόλογη ακρίβεια. Ωστόσο δεν έχουν βρεθεί δείγματα χαρτών, παρά μόνο σχέδια μεμονωμένων αγροτεμαχίων, με μία σημαντικότερη εξαίρεση: τον Πάπυρο του Τορίνο. Στον πάπυρο αυτό, που χρονολογείται στο 1300 π.Χ., απεικονίζονται οι θέσεις εξόρυξης χρυσού και Αργύρου στις περιοχές μεταξύ του Νείλου και της Ερυθράς Θάλασσας. Σημειώνονται ακόμα οι θέσεις διαμονής των εργατών, δρόμοι κ.ά. Ο χάρτης αυτός μπορεί κατά πολλούς να θεωρηθεί ο πρώτος γεωλογικός χάρτης της ιστορίας.

- **Κλάδοι της Χαρτογραφίας**

- **Θεματική Χαρτογραφία**

Θεματική Χαρτογραφία είναι ο κλάδος της Χαρτογραφίας που έχει ως αντικείμενο τη σύνθεση θεματικών χαρτών. Το κύριο αντικείμενο της Θεματικής Χαρτογραφίας είναι η γραφική απεικόνιση και ειδικότερα η χαρτογραφική αναπαράσταση με κατάλληλες τεχνικές, φαινομένων που έχουν κατανομή στον γεωγραφικό χώρο, είτε αυτός είναι ο φυσικός είτε ο ανθρωπογενής. Τα φαινόμενα αυτά μπορεί να είναι καταγεγραμμένα μέσω ποιοτικών ή (κυρίως) ποσοτικών χαρακτηριστικών και μεγεθών.

Η Θεματική Χαρτογραφία αποτελεί μια σειρά από καθαρά εφαρμοσμένες διαδικασίες και μεθοδολογίες. Μπορεί να θεωρηθεί και ως επιστήμη, εφόσον βασίζεται σε θεωρητικές επιστήμες όπως τα μαθηματικά, αλλά και ως τέχνη, αφού στόχος της είναι η ισορροπία μεταξύ της αισθητικής και της γρήγορης αντίληψης των φαινομένων που απεικονίζει ο θεματικός χάρτης.

Τα σημαντικότερα θέματα που περιλαμβάνει το γνωστικό αντικείμενο της Θεματικής Χαρτογραφίας είναι:

- Η οπτικοποίηση δεδομένων και φαινομένων που κατανέμονται στον γεωγραφικό χώρο
- Η χωρική δειγματοληψία και η βασική στατιστική επεξεργασία
- Η ομαδοποίηση ποσοτικών δεδομένων
- Η σωστή χρήση του χρώματος μέσα από τη χρωματική θεωρία και τα σχετικά μοντέλα χρωμάτων
- Ο χαρτογραφικός σχεδιασμός και η απόδοση με διάφορους συμβολισμούς (π.χ. σημειακά, γραμμικά και επιφανειακά σύμβολα)
- Η ισαριθμική (ή ισοπληθής) απεικόνιση
- Η χωροπληθής (ή δασυμετρική) απεικόνιση
- Οι χάρτες κουκίδων και τα χαρτογράμματα
- Οι Τοπολογικές και εστιακές/πολυεστιακές απεικονίσεις
- Η Πολυμεταβλητή και Δυναμική χαρτογραφική απόδοση

• **Αναλυτική Χαρτογραφία**

Ονομάζεται και μαθηματική χαρτογραφία και είναι ο κλάδος της Χαρτογραφίας που ασχολείται με την επίλυση χαρτογραφικών προβλημάτων χρησιμοποιώντας μαθηματικούς ή αναλυτικούς τρόπους. Ο όρος *μαθηματική χαρτογραφία* χρησιμοποιούνταν από παλιά, για να περιγράψει ο κλάδος των μαθηματικών που ασχολήθηκε με το πρόβλημα των απεικονίσεων. Ο όρος *αναλυτική χαρτογραφία* χρησιμοποιήθηκε στις αρχές της δεκαετίας του 1960 από τον καθηγητή Waldo Tobler, γεωγράφο και χαρτογράφο στο πανεπιστήμιο του Μίτσιγκαν, ο οποίος και θεωρείται από πολλούς ο πατέρας της Αναλυτικής Χαρτογραφίας.

Η Αναλυτική Χαρτογραφία περιλαμβάνει μαθηματικές έννοιες και μεθόδους που εφαρμόζονται στη σύγχρονη χαρτογραφία και στα Γεωγραφικά Συστήματα Πληροφοριών. Το αντικείμενο της περιλαμβάνει μεταξύ άλλων:

- μεθόδους απεικόνισης της φυσικής γήινης επιφάνειας σε ένα επίπεδο, γνωστές ως χαρτογραφικές προβολές

- μεθόδους μονοδιάστατης και δισδιάστατης παρεμβολής (π.χ. για την προσαρμογή καμπύλων γραμμών)
- μεθόδους αυτόματης γενίκευσης για τη δημιουργία χαρτών μικρής κλίμακας από γεωχωρικά δεδομένα που έχουν συλλεχθεί σε μεγάλες κλίμακες.
- μεθόδους αντιμετώπισης προβλήματος της άρσης πιθανών συμπτώσεων των χαρτογραφικών συμβόλων λόγω της χωρικής μεταβολής (π.χ. μετάθεσης) που υφίστανται κατά τη διαδικασία της γενίκευσης, οι οποίες προκαλούν οπτική σύγχυση.
- μεθόδους δημιουργίας ψηφιακών μοντέλων εδάφους / υψομέτρων και αυτόματης σκίασης του ανάγλυφου
- μεθόδους αυτόματης τοποθέτησης ονοματολογίας στους χάρτες.
- μεθόδους επεξεργασίας χωρικών δεδομένων και δειγματοληψίας.
- μεθόδους αυτόματης επίλυσης προβλημάτων αλληλοτομίας γραφικών αντικειμένων στο δισδιάστατο χώρο (επίπεδο).
- μεθόδους γεωμετρικών μετασχηματισμών όπως π.χ. ο μετασχηματισμός ομοιότητας, ο ομοπαράλληλος μετασχηματισμός που χρησιμοποιούνται συχνά σε χαρτογραφικές εφαρμογές κυρίως στις διαδικασίες γεωαναφοράς σαρωμένων χαρτών και διαγραμμάτων.
- μεθόδους χαρτομετρίας.

Η χαρτομετρία έχει ορισθεί από τη Διεθνή Χαρτογραφική Ένωση (ICA) ως: «*Η μέτρηση και ο υπολογισμός αριθμητικών τιμών από τους χάρτες*».

Τα είδη των μετρήσεων που μπορούν να θεωρηθούν ως βασικές τεχνικές της χαρτομετρίας είναι:

- η μέτρηση αποστάσεων (μηκών)

- η μέτρηση επιφανειών (εμβαδών),
- η μέτρηση γωνιών και διευθύνσεων (π.χ. αζιμούθιο)
- η καταμέτρηση του αριθμού των οντοτήτων που απεικονίζονται σε ένα χάρτη

Σήμερα η Αναλυτική Χαρτογραφία αποτελεί μέρος των θεωρητικών αρχών στην οποία βασίζεται η τεχνολογία των Συστημάτων Γεωγραφικών Πληροφοριών (ΣΓΠ/GIS) και γενικά η Γεωπληροφορική.

• Ψηφιακή Χαρτογραφία

Η ψηφιακή χαρτογραφία είναι η συγκέντρωση δεδομένων και ο συνδυασμός τους ώστε να δημιουργηθεί μια ψηφιακή εικόνα. Ο βασικός σκοπός αυτής της διαδικασίας είναι η δημιουργία χαρτών με υψηλή ακρίβεια. Ωστόσο, βασιζόμενοι κυρίως στην εξέλιξη της τεχνολογίας GPS (Global Positioning System), οι ψηφιακοί χάρτες έχουν πλέον τη δυνατότητα παροχής πολύ περισσότερων πληροφοριών από έναν απλό χάρτη. Συγκεκριμένα μπορούν να επικαιροποιούνται με τις αλλαγές στο οδικό δίκτυο, να παρέχουν πληροφόρηση σχετικά με τη διεξαγωγή έργων και την κυκλοφοριακή κίνηση, να υπολογίζουν αποστάσεις, να εντοπίζουν γεωγραφική θέση κ.α.

- ## Δομές Υγείας

- ### Εισαγωγή στις Δομές Υγείας

Το Σύστημα αποτελείται από επτά (7) Υγειονομικές Περιφέρειες (Υ.Π.): Αττικής, Πειραιώς και Αιγαίου, Μακεδονίας, Θεσσαλίας και Στερεάς Ελλάδος, Πελοποννήσου, Ιονίων Νήσων, Ηπείρου και Δυτικής Ελλάδος, και Κρήτης) οι οποίες διοικούν τρεις (3) βαθμούς *Φροντίδων Υγείας* (Α΄, Β΄ και Γ΄), ως εξής:

- Πρωτοβάθμια Φροντίδα Υγείας (Π.Φ.Υ.): αποτελείται από τα Κέντρα Υγείας (Κ.Υ.) και τις Τοπικές Μονάδες Υγείας (Τ.Μ.Υ.) με σκοπό την πρόληψη, θεραπεία και αποκατάσταση των ασθενών
- Νοσοκομεία: σκοπός είναι η παροχή Δευτεροβάθμιας και Τριτοβάθμιας Φροντίδας Υγείας (Δ.Φ.Υ. και Τ.Φ.Υ.), δηλαδή την ενδονοσοκομειακή περίθαλψη των ασθενών.

Εκτός των παραπάνω τριών κύριων κατηγοριών, υπάρχουν διάσπαρτα Αγροτικά και Περιφερειακά Ιατρεία (Α.Ι. και Π.Ι.) στην Ελληνική επικράτεια τα οποία διοικούνται από τις επτά Υγειονομικές Περιφέρειες.

Πέρα από τα προαναφερθέντα υπάρχουν και οι ιδιωτικές δομές υγείας τα οποία περιλαμβάνουν είτε τα ιατρικά διαγνωστικά κέντρα εταιριών αλυσίδων είτε ιδιωτών όπου σκοπός τους η εκπόνηση διαφόρων εξετάσεων είτε μικροβιολογικών αλλά και πολλών άλλων εξελιγμένης τεχνολογίας. Παράλληλα στις ιδιωτικές δομές ανήκουν και οι ιδιωτικές κλινικές και ιδιωτικά νοσοκομεία τα οποία συσχετίζονται με τα δημόσια νοσοκομεία αλλά είναι επί πληρωμή από πλευράς του ασθενή.

- **Δημόσια Υγεία**

- **Έννοια Δημόσιας Υγείας**

Ο όρος δημόσια υγεία αναφέρεται στην επιστήμη και την τέχνη της πρόληψης των ασθενειών και της επιμήκυνσης της ζωής και της προαγωγής της υγείας. Ο ορισμός αποδίδεται στον Edward Winslow (1920) και στον πληρέστερο ορισμό του δημόσια υγεία είναι η επιστήμη και η τέχνη για την πρόληψη των ασθενειών, την επιμήκυνση τη ζωής και την προαγωγή της σωματικής υγείας και της αποδοτικότητας του ανθρώπου μέσω οργανωμένων προσπαθειών της κοινότητας, την εξυγίανση του περιβάλλοντος, τον έλεγχο των μολύνσεων της κοινότητας, την εκπαίδευση του ατόμου ως προς τις αρχές και την προσωπική υγιεινή, την οργάνωση των ιατρικών και νοσηλευτικών υπηρεσιών για την έγκαιρη διάγνωση και προληπτική θεραπεία της νόσου, καθώς και την ανάπτυξη κοινωνικού μηχανισμού που θα εξασφαλίζει σε κάθε άτομο ένα βιοτικό επίπεδο κατάλληλο για τη διατήρηση της υγείας του.

Η δημόσια υγεία ασχολείται με τις απειλές για την υγεία με βάση την ανάλυση της υγείας του πληθυσμού. Ο πληθυσμός μπορεί να είναι πολύ μικρός ή τόσο μεγάλος ώστε να περιλαμβάνει κατοίκους πολλών ηπείρων (για παράδειγμα, στην περίπτωση της πανδημίας). Οι διαστάσεις της υγείας περιλαμβάνουν «μια κατάσταση πλήρους σωματικής, ψυχικής και κοινωνικής ευεξίας και όχι απλώς η απουσία ασθένειας ή αναπηρίας», όπως ορίζεται από τον Παγκόσμιο Οργανισμό Υγείας.

Η σύγχρονη πρακτική της δημόσιας υγείας απαιτεί διεπιστημονικές ομάδες εργαζομένων στη δημόσια υγεία και επαγγελματίες, με κύριες εκείνων των Ιατρών που ειδικεύονται στη δημόσια υγεία/προληπτική ιατρική/μολυσματικές ασθένειες, των επισκεπτών/τριων υγείας και των νέων αποφοίτων κοινοτικής υγείας του ΠΑΔΑ. Αλλά και άλλων επαγγελματιών όπως: ψυχολόγους, επιδημιολόγους, στατιστικολόγους, Βοηθούς Ιατρικών Λειτουργών, μικροβιολόγους, επιθεωρητές περιβαλλοντικής υγείας/επιθεωρητές δημόσιας υγείας, φαρμακοποιούς, οδοντίατρους, επόπτες δημόσιας υγείας, διαιτολόγους και διατροφολόγους, κτηνίατρους, κοινωνιολόγοι, εμπειρογνώμονες των επικοινωνιών, της βιοηθικής και άλλους.

Επιπλέον, η Δημόσια Υγεία προσδιορίζεται από το σύνολο των οργανωμένων δραστηριοτήτων της πολιτείας και της κοινωνίας, που είναι επιστημονικά τεκμηριωμένες και αποβλέπουν στην

πρόληψη νοσημάτων, στην προστασία και την προαγωγή της υγείας, στην αύξηση του προσδόκιμου επιβίωσης και στη βελτίωση της ποιότητας ζωής και της ευεξίας του πληθυσμού.

- Στην καταγραφή της υγείας, στον εντοπισμό και στην επίλυση προβλημάτων υγείας στην κοινότητα.
- Στη διάγνωση και διερεύνηση προβλημάτων και κινδύνων υγείας στην κοινότητα.
- Στην πληροφόρηση, ενημέρωση και ενδυνάμωση ανθρώπων σε θέματα υγείας.
- Στην ενεργοποίηση/ δραστηριοποίηση της κοινότητας στη συμμετοχή δράσεων και στην επίλυση προβλημάτων.
- Στην ανάπτυξη Πολιτικών και σχεδίων ενδυνάμωσης της ατομικής και κοινοτικής υγείας.
- Στην τήρηση των νόμων και των κανονισμών προστασίας της υγείας και της ασφάλειας.
- Στη δημιουργία ανταγωνιστικών στελεχών Δημόσιας Υγείας.
- Στην αξιολόγηση της αποτελεσματικότητας, προσβασιμότητας και ποιότητας των υπηρεσιών υγείας σε ατομικό επίπεδο και σε επίπεδο κοινότητας.
- Στην έρευνα για στην εφαρμογή καινοτόμων λύσεων στα προβλήματα Δημόσιας Υγείας.

Στο σχεδιασμό και την αξιολόγηση υπηρεσιών υγείας. Στη βελτίωση της πληροφόρησης και της γνώσης για την υγεία, η οποία θα παρέχει στους αρμόδιους χάραξης πολιτικής, στους επαγγελματίες στο χώρο της υγείας και στο ευρύ κοινό τα βασικά στοιχεία και τις πληροφορίες που απαιτούνται για την ενίσχυση της συμμετοχής τους στην προάσπιση και προαγωγή της υγείας.

- **Παραδείγματα Δημόσιας Υγείας**

1. Νοσοκομεία
2. Κέντρα Υγείας
3. Αγροτικά Κέντρα
4. Δημόσια Φαρμακεία
5. Κοινωνικά ιατρεία

- **Ιδιωτική Υγεία**

- **Έννοια Ιδιωτικής Υγείας**

Η Έννοια της ιδιωτικής υγείας δεν απέχει σημασιολογικά από αυτήν της δημόσιας υγείας, το διαχωριστικό της σημασίας τους βασίζεται κυρίως στις δομές και στο αντίτιμο.

Στην σήμερον ημέρα, όπου οι ασθένειες , οι ιοί αλλά και πολλοί άλλοι παράγοντες που επηρεάζουν την ανθρώπινη ακεραιότητα δεν είναι απόλυτα εφικτή η αντιμετώπιση όλων αυτών των προβλημάτων μονάχα από τους δημόσιους φορείς της υγείας, πράγμα που οδήγησε στην ίδρυση αρκετών ιδιωτικών δομών.

- **Παραδείγματα Ιδιωτικής Υγείας**

1. Κλινικές
2. Ιατρικά Διαγνωστικά Κέντρα
- 3.Ιδιωτικά Ιατρεία
- 4.Ιδιωτικά Φαρμακεία
5. Εξειδικευμένης Τεχνολογίας Δομές

Επιπροσθέτως, σύμφωνα με το υπουργείο υγείας η ιδιωτική κλινική είναι η υγειονομική μονάδα στην οποία παρέχονται υπηρεσίες υγείας και νοσηλείας ασθενών, σύμφωνα με τις παραδεδεγμένες σύγχρονες αντιλήψεις και μεθόδους της Ιατρικής Επιστήμης. Είναι δυνατή η παροχή υπηρεσιών υγείας σε εξωτερικούς ασθενείς με τα νομίμως λειτουργικά τμήματα. Επιπρόσθετα τα εργαστήρια και οι μονάδες της κλινικής λογίζονται ως τμήματα αυτής.

- ## Περιβάλλον Υλοποίησης

- ### Γλώσσες Προγραμματισμού

Οι γλώσσες προγραμματισμού που χρησιμοποιήσαμε για να υλοποιήσουμε την διπλωματική εργασία είναι οι παρακάτω

- #### Python

Η Python είναι μια γλώσσα προγραμματισμού που μας επιτρέπει να εργαζόμαστε και να ενσωματώνουμε τα συστήματά μας πιο αποτελεσματικά. Μπορεί να είναι εύκολο να εντοπισθεί είτε είστε προγραμματιστής για πρώτη φορά είτε έχετε εμπειρία σε άλλες γλώσσες. Έχει αναπτυχθεί με άδεια ανοιχτού κώδικα(open source) εγκεκριμένη από την OSI, καθιστώντας την ελεύθερα χρησιμοποιήσιμη και διανεμόμενη, ακόμη και για εμπορική χρήση. Η άδεια χρήσης της διαχειρίζεται από το Python Software Foundation. Χρησιμοποιείται από πολλούς προγραμματιστές εξαιτίας της ποικιλίας των βιβλιοθηκών που μπορούμε να χρησιμοποιήσουμε μέσω αυτής.

Επίσης η python είναι μια γλώσσα υψηλού επιπέδου ερμηνευμένη, γενικής χρήσης. Η σχεδιαστική της φιλοσοφία δίνει έμφαση στην αναγνωσιμότητα κώδικα με την χρήση σημαντικής εσοχής. Είναι δυναμική και συλλέγεται από σκουπίδια(garbage-collected). Υποστηρίζει πολλαπλά παραδείγματα προγραμματισμού, συμπεριλαμβανομένων δομημένου αντικειμενοστραφούς προγραμματισμού. Συχνά περιγράφεται ως γλώσσα 'batteries included' λόγω της ολοκληρωμένης τυπικής βιβλιοθήκης της.

Ο Guido van Rossum άρχισε να εργάζεται στην python στα τέλη της δεκαετίας του 1980 ως διάδοχος της γλώσσας προγραμματισμού ABC και την κυκλοφόρησε πρώτη φορά το 1991 ως Python 0.9.0. Η Python 2.0 κυκλοφόρησε το 2000 και εισήγαγε τις δυνατότητες, όπως

κατανόηση λιστών, συλλογή σκουπιδιών ανίχνευσης κύκλου, καταμέτρηση αναφορών και υποστήριξη Unicode. Η python 3.0 όπου και χρησιμοποιούμε στην εργασία μας, που κυκλοφόρησε το 2008, ήταν μια σημαντική αναθεώρηση που δεν είναι πλήρως συμβατή με παλαιότερες εκδόσεις. Κατατάσσεται σταθερά ως μια από τις πιο δημοφιλείς γλώσσες προγραμματισμού.

- **Javascript**

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

Είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστρεφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού. Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side). Το πρότυπο της γλώσσας κατά τον οργανισμό τυποποίησης ECMA ονομάζεται ECMAScript

Δημιουργήθηκε αρχικά από τον Brendan Eich της εταιρείας Netscape με την επωνυμία Mocha. Αργότερα, η Mocha μετονομάστηκε σε LiveScript, και τελικά σε JavaScript, κυρίως επειδή η ανάπτυξη της επηρεάστηκε περισσότερο από τη γλώσσα προγραμματισμού Java. LiveScript ήταν το επίσημο όνομα της γλώσσας όταν για πρώτη φορά κυκλοφόρησε στην αγορά σε βήτα (beta) εκδόσεις με το πρόγραμμα περιήγησης στο Web, Netscape Navigator εκδοχή 2.0 τον

Σεπτέμβριο του 1995. Η LiveScript μετονομάστηκε σε JavaScript σε μια κοινή ανακοίνωση με την εταιρεία Sun Microsystems στις 4 Δεκεμβρίου, 1995, όταν επεκτάθηκε στην έκδοση του προγράμματος περιήγησης στο Web, Netscape εκδοχή 2.0B3. Η JavaScript απέκτησε μεγάλη επιτυχία ως γλώσσα στην πλευρά του πελάτη (client-side) για εκτέλεση κώδικα σε ιστοσελίδες, και περιλήφθηκε σε διάφορα προγράμματα περιήγησης στο Web. Κατά συνέπεια, η εταιρεία Microsoft ονόμασε την εφαρμογή της σε JScript για να αποφύγει δύσκολα θέματα εμπορικών σημάτων. Το JScript περιλήφθηκε στο πρόγραμμα Internet Explorer εκδοχή 3.0, το οποίο κυκλοφόρησε τον Αύγουστο του 1996. Τον Νοέμβριο του 1996, η Netscape ανακοίνωσε ότι είχε υποβάλει τη γλώσσα JavaScript στο Ecma International (μία οργάνωση της τυποποίησης των γλωσσών προγραμματισμού) για εξέταση ως βιομηχανικό πρότυπο, και στη συνέχεια το έργο είχε ως αποτέλεσμα την τυποποιημένη μορφή που ονομάζεται ECMAScript. Η JavaScript έχει γίνει μία από τις πιο δημοφιλείς γλώσσες προγραμματισμού ηλεκτρονικών υπολογιστών στον Παγκόσμιο Ιστό (Web). Αρχικά, όμως, πολλοί επαγγελματίες προγραμματιστές υποτίμησαν τη γλώσσα διότι το κοινό της ήταν ερασιτέχνες συγγραφείς ιστοσελίδων και όχι επαγγελματίες προγραμματιστές (μαζί με άλλους λόγους). Με τη χρήση της τεχνολογίας Ajax, η JavaScript γλώσσα επέστρεψε στο προσκήνιο και έφερε πιο επαγγελματική προσοχή προγραμματισμού. Το αποτέλεσμα ήταν ένα καινοτόμο αντίκτυπο στην εξάπλωση των πλαισίων και των βιβλιοθηκών, τη βελτίωση προγραμματισμού με JavaScript, καθώς και αυξημένη χρήση της JavaScript έξω από τα προγράμματα περιήγησης στο Web. Τον Ιανουάριο του 2009, το έργο CommonJS ιδρύθηκε με στόχο τον καθορισμό ενός κοινού προτύπου βιβλιοθήκης κυρίως για την ανάπτυξη της JavaScript έξω από το πρόγραμμα περιήγησης και μέσα σε άλλες τεχνολογίες (π.χ. server-side). Η επεξεργασία του κώδικα Javascript και η παραγωγή του τελικού περιεχομένου HTML δεν πραγματοποιείται στο διακομιστή, αλλά στο πρόγραμμα περιήγησης των επισκεπτών, ενώ μπορεί να ενσωματωθεί σε στατικές σελίδες HTML. Αντίθετα, άλλες γλώσσες όπως η PHP εκτελούνται στο διακομιστή (server-side γλώσσες προγραμματισμού).

- **CSS**

Η CSS (*Cascading Style Sheets* – διαδοχικά φύλλα ύφους ή επάλληλα φύλλα ύφους) είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων ύφους που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου που έχει γραφτεί με μια γλώσσα σήμανσης. Χρησιμοποιείται δηλαδή για τον έλεγχο της εμφάνισης ενός εγγράφου που γράφτηκε στις γλώσσες HTML και XHTML, δηλαδή για τον έλεγχο της εμφάνισης μιας ιστοσελίδας και γενικότερα ενός ιστοτόπου. Η CSS είναι μια γλώσσα υπολογιστή προορισμένη να αναπτύσσει στιλιστικά μια ιστοσελίδα δηλαδή να διαμορφώνει περισσότερα χαρακτηριστικά, χρώματα, στοίχιση και δίνει περισσότερες δυνατότητες σε σχέση με την html. Για μια όμορφη και καλοσχεδιασμένη ιστοσελίδα η χρήση της CSS κρίνεται ως απαραίτητη.

- **HTML**

Η γλώσσα σήμανσης HyperText ή HTML είναι η τυπική γλώσσα σήμανσης για έγγραφα που έχουν σχεδιαστεί για εμφάνιση σε πρόγραμμα περιήγησης Ιστού. Μπορεί να υποστηριχθεί από τεχνολογίες όπως τα Cascading Style Sheets(CSS) και γλώσσες δέσμης ενεργειών όπως η JavaScript. Τα προγράμματα περιήγησης Ιστού λαμβάνουν έγγραφα HTML από έναν διακομιστή ιστού ή από τοπική αποθήκευση και αποδίδουν τα έγγραφα σε ιστοσελίδες πολυμέσων. Επίσης περιγράφει την δομή μιας ιστοσελίδας σημασιολογικά και αρχικά περιλάμβανε ενδείξεις για την εμφάνιση του εγγράφου. Με κατασκευές HTML, εικόνες και άλλα αντικείμενα ,όπως διαδραστικές φόρμες μπορούν να ενσωματωθούν στην σελίδα απόδοσης. Επιπρόσθετα, παρέχει ένα μέσο για την δημιουργία δομημένων εγγράφων δηλώνοντας δομική σημασιολογία για κείμενο όπως επικεφαλίδες, παραγράφους, λίστες, συνδέσμους, εισαγωγικά και άλλα στοιχεία. Τα στοιχεία αυτά οριοθετούνται με ετικέτες γραμμένες με χρήση αγκυλών(</>). Στη συνέχεια, η HTML μπορεί να ενσωματώσει προγράμματα γραμμένα σε μια γλώσσα δέσμης ενεργειών όπως η JavaScript. Η συμπερίληψη της CSS καθορίζει την εμφάνιση και την διάταξη του περιεχομένου. Μια γνωστή μορφή HTML, γνωστή ως HTML5, χρησιμοποιείται για την εμφάνιση βίντεο και ήχου, χρησιμοποιώντας κυρίως το στοιχείο <canvas> σε συνεργασία με την JavaScript.

- **SQL**

Η SQL (Structured Query Language) είναι μια συγκεκριμένη γλώσσα που χρησιμοποιείται στον προγραμματισμό και έχει σχεδιαστεί για την διαχείριση δεδομένων που τηρούνται σε ένα σύστημα διαχείρισης σχεσιακής βάσης δεδομένων ή για επεξεργασία ροής σε σύστημα διαχείρισης σχεσιακών ροών δεδομένων. Είναι ιδιαίτερα χρήσιμη για το χειρισμό δομημένων δεδομένων, δηλαδή δεδομένων που ενσωματώνουν σχέσεις μεταξύ οντοτήτων και μεταβλητών. Προσφέρει δύο βασικά πλεονεκτήματα σε σχέση με παλαιότερα API ανάγνωσης-εγγραφής. Πρώτον εισήγαγε την έννοια της πρόσβασης σε πολλές εγγραφές με μία μόνο εντολή. Δεύτερον εξαλείφει την ανάγκη καθορισμού του τρόπου επίτευξης ενός ρεκόρ πχ με ή χωρίς ευρετήριο.

Αρχικά βασισμένη στην σχεσιακή άλγεβρα και τον πολλαπλό σχεσιακό λογισμό, η SQL αποτελείται από πολλούς τύπους δηλώσεων, οι οποίοι μπορούν να ταξινομηθούν ανεπίσημα ως υπογλώσσες. Το πεδίο εφαρμογής της SQL περιλαμβάνει ερώτημα δεδομένων, χειρισμό δεδομένων (εισαγωγή, ενημέρωση και διαγραφή), ορισμό δεδομένων (δημιουργία σχήματος και τροποποίηση) και έλεγχο πρόσβασης δεδομένων.

Η SQL ήταν μια από τις πρώτες εμπορικές γλώσσες που χρησιμοποίησε το σχεσιακό μοντέλο Edgar F. Codd, το οποίο περιγράφηκε στην επιδραστική εργασία του το 1970. Παρά το γεγονός ότι δεν συμμορφώθηκε πλήρως με το σχεσιακό μοντέλο όπως περιγράφεται από τον Codd, έγινε η πιο ευρέως χρησιμοποιούμενη γλώσσα βάσης δεδομένων.

- **Βασικές Λεπτομέρειες Υλοποίησης**

Αρχικά χρησιμοποιήσαμε μια main στην οποία καλούμε τις συναρτήσεις που θέλουμε για την εκκίνηση της εργασίας μας. Πιο συγκεκριμένα έχοντας κάνει import τα αρχεία που θα χρησιμοποιήσουμε αλλά και το application καλούμε κατά σειρά σύμφωνα με το επιθυμητό αποτέλεσμα τις συναρτήσεις μας και στο τέλος το application με σκοπό την έναρξη του σέρβερ.

```
if __name__ == "__main__":  
    scrapping.scrap()  
    createdb.create()  
    app.run()
```

Εικόνα 1 Εκτέλεση εφαρμογής

Στην συνάρτηση scrap εκτελούμε την διαδικασία του web scraping για αυτόν τον λόγο το καλούμε και πρώτο, στην συνέχεια καλούμε την create στην οποία δημιουργούμε την βάση μας αλλά και προσθέτουμε τα δεδομένα. Συνεπώς στο τέλος καλούμε το app.run ώστε να εκκινήσουμε τον σέρβερ έχοντας ήδη εκτελέσει τις προηγούμενες λειτουργίες με σκοπό την σωστή λειτουργία του σέρβερ μας.

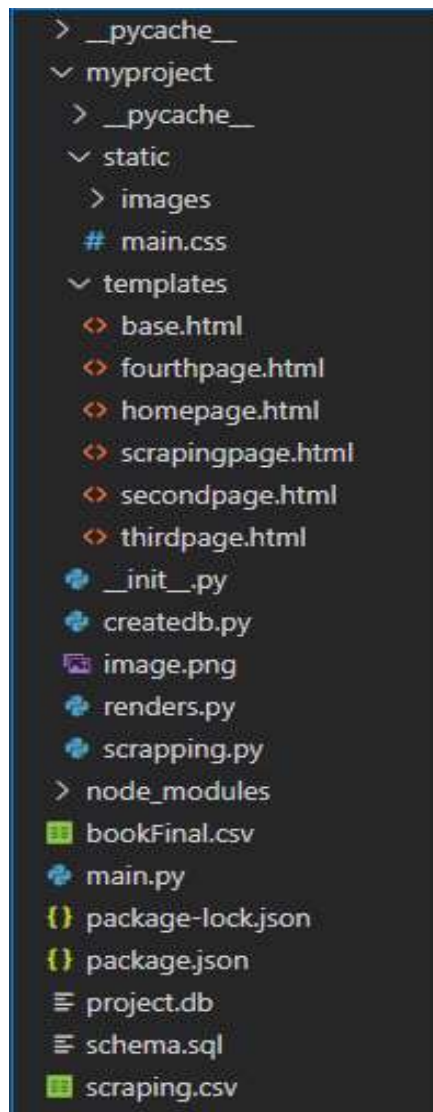
Στην συνέχεια χρησιμοποιούμε το αρχείο `__init__.py` στο οποίο μέσω της Flask δημιουργούμε το application που καλέσαμε στην main. Μέσα στην init κάνουμε import την flask αρχικά και στην συνέχεια το αρχείο renders. Ο λόγος που κάνουμε εδώ αυτό το import και όχι μέσα στην main είναι ότι η λειτουργία του αρχείου renders δεν μας απασχολεί προτού εκκινήσουμε τον σέρβερ αλλά αφότου ξεκινήσει.

Επιπρόσθετα, χρησιμοποιούμε άλλα τρία αρχεία σε python (renders.py, created.py, scrapping.py) στα οποία εκτελούμε σε κάθε ένα μια ξεχωριστή λειτουργία. Όσον αφορά την ιστοσελίδα έχουμε δημιουργήσει 6 αρχεία html και τελευταίο ένα στατικό αρχείο css.

Τέλος χρησιμοποιούμε κάποιες εικόνες για να εμπλουτίσουμε τον σέρβερ και 2 αρχεία csv στα οποία αποθηκεύουμε τα δεδομένα μας προτού τα περάσουμε στην βάση.

- **Αρχεία**

Μια πλήρη εικόνα των αρχείων μας είναι :



Εικόνα Αρχεία Εργασίας

- **Frameworks**

- **Flask**

Το Flask είναι ένα micro web framework γραμμένο σε Python. Ταξινομείται ως μικροπλαίσιο επειδή δεν απαιτεί συγκεκριμένα εργαλεία ή βιβλιοθήκες. Δεν έχει επίπεδο αφαίρεσης βάσης δεδομένων, επικύρωση φόρμας ή άλλα στοιχεία όπου προϋπάρχοντες βιβλιοθήκες τρίτων παρέχουν κοινές λειτουργίες. Ωστόσο υποστηρίζει επεκτάσεις που μπορούν να προσθέτουν δυνατότητες εφαρμογής σαν να είχαν υλοποιηθεί στο ίδιο το Flask. Υπάρχουν επεκτάσεις για

αντικείμενο σχεσιακούς αντιστοιχιστές, επικύρωση φόρμας, χειρισμό μεραμόρφωσης, διάφορες ανοιχτές τεχνολογίες ελέγχου ταυτότητας και πολλά εργαλεία που σχετίζονται με το πλαίσιο.

Για παράδειγμα κάποιες εφαρμογές που χρησιμοποιούν την Flask είναι το Pinterest και το LinkedIn.

- **Selenium**

Το Selenium είναι ένα open source umbrella project για μια σειρά εργαλείων και βιβλιοθηκών που στοχεύουν στην υποστήριξη της αυτοματοποίησης του προγράμματος περιήγησης. Παρέχει ένα εργαλείο αναπαραγωγής για τη σύνταξη λειτουργικών δοκιμών στα περισσότερα σύγχρονα προγράμματα περιήγησης ιστού, χωρίς την ανάγκη εκμάθησης μιας δοκιμαστικής γλώσσας δέσμης ενεργειών. Παρέχει επίσης μια δοκιμαστική γλώσσα για συγκεκριμένο τομέα για τη σύνταξη δοκιμών σε διάφορες δημοφιλείς γλώσσες προγραμματισμού όπως για παράδειγμα JavaScript, C#, Groovy, Java, Perl, PHP, Python, Ruby, Scala. Το selenium τρέχει σε Windows, Linux, macOS και είναι λογισμικό ανοιχτού κώδικα που κυκλοφορεί υπό την άδεια της Apache 2.0.

- **Leaflet**

Το Leaflet είναι μια από τις καλύτερες JavaScript open-source βιβλιοθήκες για διαδραστικούς χάρτες. Με μέγεθος μόλις περίπου 42KB έχει όλα τα χαρακτηριστικά χαρτογράφησης που χρειάζονται οι περισσότεροι προγραμματιστές. Έχει σχεδιαστεί με γνώμονα την απλότητα, την απόδοση και την χρηστικότητα. Λειτουργεί αποτελεσματικά σε όλες τις μεγάλες πλατφόρμες επιτραπέζιων και φορητών υπολογιστών, μπορεί να επεκταθεί με πολλές προσθήκες, έχει ένα όμορφο, εύχρηστο και καλά τεκμηριωμένο API και έναν απλό, ευανάγνωστο πηγαίο κώδικα.

- **Imports**

Τα imports που κάναμε είναι διαμοιρασμένα σε 5 διαφορετικά αρχεία.

-createdb.py

```
import sqlite3
import csv
```

Εικόνα 3 Import 1

-renders.py

```
from flask import render_template,request,jsonify,json
import sqlite3
from myproject import app
```

Εικόνα 4 Import 2

-scrapping.py

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import chromedriver_autoinstaller
import time
```

Εικόνα 5 Import 3

-__init__.py

```
from flask import Flask
```

Εικόνα 6 Import 4

-main.py

```
from myproject import app
from myproject import scrapping
from myproject import createdb
```

Εικόνα 7 Import 5

Επίσης χρησιμοποιήσαμε τα Jinja που αφορούν το κομμάτι της ιστοσελίδας μας για την διευκόλυνση μας όσον αφορά τα templates αλλά και την διατύπωση των δεδομένων μας στις φόρμες των templates.Επίσης για το κομμάτι της χαρτογράφησης χρησιμοποιήσαμε κάποια plugins της Leaflet,δηλαδή για την απεικόνιση στην χάρτη, την δημιουργία των markers για τα δεδομένα ,τα popups με τις πληροφορίες για κάθε δεδομένο πάνω στα markers αλλά και την διαδικασία της δρομολόγησης μεταξύ δύο σημείων. Τέλος χρησιμοποιήσαμε την Bootstrap 5.1.3 σε κάποια σημεία στην ιστοσελίδα μας με σκοπό την μεγαλύτερη αποτελεσματικότητα αλλά και την ομορφιά καθώς και την Ajax που είναι βιβλιοθήκη της JavaScript με σκοπό την επικοινωνία frontend & backend.

- ## Δεδομένα

- ### Ορισμός και έννοια Βάσης Δεδομένων

Με τον όρο βάση δεδομένων εννοείται μία συλλογή από *συστηματικά μορφοποιημένα* σχετιζόμενα δεδομένα στα οποία είναι δυνατή η ανάκτηση δεδομένων μέσω αναζήτησης κατ' απαίτηση. Ο Αμερικανός επιστήμονας υπολογιστών Τζιμ Γκρέϊ έχει γράψει για τις βάσεις δεδομένων: «Όταν οι άνθρωποι χρησιμοποιούν τις λέξεις *βάση δεδομένων*, διατυπώνουν στην ουσία ότι τα δεδομένα πρέπει να αυτοπροσδιορίζονται και να έχουν μια σχηματική δομή. Αυτό ακριβώς περιγράφουν οι λέξεις *βάση δεδομένων*». Επίσης βάση δεδομένων είναι ένα σύνολο δεδομένων που έχει μια κανονική δομή και που είναι οργανωμένη με τέτοιο τρόπο ώστε ένας υπολογιστής να μπορεί εύκολα να βρει τις επιθυμητές πληροφορίες. Επιπρόσθετα μπορεί γενικά να θεωρηθεί ως μια συλλογή εγγραφών, καθεμία από τις οποίες περιέχει ένα ή περισσότερα πεδία (δηλαδή κομμάτια δεδομένων) σχετικά με κάποια οντότητα (δηλαδή αντικείμενο), όπως ένα άτομο, οργανισμός, πόλη, προϊόν, εργασία της τέχνης, της συνταγής, της χημικής ουσίας ή της αλληλουχίας του DNA.

Για παράδειγμα, τα πεδία για μια βάση δεδομένων που αφορά άτομα που εργάζονται για μια συγκεκριμένη εταιρεία μπορεί να περιλαμβάνουν το όνομα, τον αριθμό αναγνώρισης υπαλλήλου, τη διεύθυνση, τον αριθμό τηλεφώνου, την ημερομηνία έναρξης της απασχόλησης, τη θέση και τον μισθό για κάθε εργαζόμενο.

Ένα ακόμη παράδειγμα μπορεί να θεωρηθεί ένας τηλεφωνικός κατάλογος όπου καθώς αποθηκεύει και οργανώνει σχετιζόμενα τμήματα πληροφορίας, όπως είναι το όνομα και ο αριθμός τηλεφώνου.

Ένα τρίτο και τελευταίο παράδειγμα είναι οι πίνακες δημογραφικών/στατιστικών στοιχείων. Ωστόσο, οι σύγχρονες βάσεις δεδομένων υλοποιούνται κυρίως ψηφιακά, με τυποποιημένες μεθόδους σε ηλεκτρονικούς υπολογιστές, ελέγχονται από λογισμικό DBMS και συνιστούν αντικείμενο επιστημονικής και τεχνικής μελέτης από το αντίστοιχο ακαδημαϊκό πεδίο.

Ειδικότερα, στην επιστήμη της πληροφορικής και στην καθημερινή χρήση των ηλεκτρονικών υπολογιστών, με τον όρο *βάσεις δεδομένων* αναφερόμαστε σε οργανωμένες, διακριτές συλλογές σχετιζόμενων δεδομένων ηλεκτρονικά και ψηφιακά αποθηκευμένων, στο λογισμικό που χειρίζεται τέτοιες συλλογές (Σύστημα Διαχείρισης Βάσεων Δεδομένων, ή *DBMS*) και στο γνωστικό πεδίο που το μελετά. Πέρα από την εγγενή της ικανότητα να αποθηκεύει δεδομένα, η βάση δεδομένων παρέχει μέσω του σχεδιασμού και του τρόπου ιεράρχησης των δεδομένων, τα αποκαλούμενα συστήματα διαχείρισης περιεχομένου, δηλαδή τη δυνατότητα γρήγορης άντλησης και ανανέωσης των δεδομένων.

Οι βάσεις δεδομένων μπορεί να είναι αυτόνομα προγράμματα ή μπορούν να ενσωματωθούν σε άλλα προγράμματα, συμπεριλαμβανομένων εκείνων που θεωρούνται μέρος του λειτουργικού συστήματος. Ο τυπικός υπολογιστής περιέχει πολυάριθμες βάσεις δεδομένων του τελευταίου τύπου. Περιλαμβάνουν τα πολλά αρχεία διαμόρφωσης απλού κειμένου (δηλαδή, αναγνώσιμους από τον άνθρωπο χαρακτήρες) που επιτρέπουν στους χρήστες να τροποποιούν εύκολα τη συμπεριφορά του συστήματος στο Linux και άλλα λειτουργικά συστήματα παρόμοια με Unix, όπως τα `/etc/fstab`, `/etc/hosts` και `/etc/passwd`. Παρομοίως, τα προγράμματα περιήγησης ιστού περιέχουν απλές βάσεις δεδομένων που απαριθμούν τις πιο πρόσφατες ιστοσελίδες που επισκέφθηκαν και τις προτιμήσεις των χρηστών, και τα προγράμματα ηλεκτρονικού ταχυδρομείου περιέχουν απλές βάσεις δεδομένων που παραθέτουν διευθύνσεις e-mail, πρόσφατες επικοινωνίες και προτιμήσεις χρηστών.

Έχουν αναπτυχθεί αρκετοί βασικοί τύποι μοντέλων βάσεων δεδομένων, συμπεριλαμβανομένων των επίπεδων, ιεραρχικών, δικτυακών και σχεσιακών. Τέτοια μοντέλα περιγράφουν όχι μόνο τη δομή των συμβατών βάσεων δεδομένων αλλά και τις λειτουργίες που μπορούν να εκτελεστούν σε αυτές. Συνήθως, μια βάση δεδομένων έχει ένα σχήμα, το οποίο είναι μια περιγραφή του μοντέλου, συμπεριλαμβανομένων των τύπων οντοτήτων που βρίσκονται σε αυτό και των σχέσεων μεταξύ τους.

- **Κατηγορίες Βάσεων Δεδομένων**

- **Επίπεδες Βάσεις Δεδομένων**

Οι επίπεδες βάσεις δεδομένων είναι ο απλούστερος τύπος. Ήταν ο κυρίαρχος τύπος για μεγάλο χρονικό διάστημα και μπορούν ακόμα να είναι χρήσιμοι, ιδιαίτερα για πολύ μικρής κλίμακας και απλές εφαρμογές. Ένα παράδειγμα είναι ένας ενιαίος πίνακας σε χαρτί ή σε αρχείο υπολογιστή που περιέχει μια λίστα εταιρειών με πληροφορίες για κάθε μία, όπως όνομα, διεύθυνση, κατηγορία προϊόντος, όνομα επαφής κ.λπ. Μια επίπεδη βάση δεδομένων μπορεί επίσης να υπάρχει με τη μορφή ενός συνόλου κάρτες ευρετηρίου, καθεμία από τις οποίες περιέχει τις πληροφορίες για μία από τις οντότητες.

- **Επεκτάσιμες(Ιεραρχικές) Βάσεις Δεδομένων**

Στη συνέχεια, τα τελευταία χρόνια έχει δοθεί μεγάλη προσοχή στις βάσεις δεδομένων XML (επεκτάσιμη γλώσσα σήμανσης) λόγω της ικανότητάς τους να εξαλείφουν την παραδοσιακή διαίρεση μεταξύ εγγράφων και δεδομένων, αναλύοντας την πρώτη σε πιο ατομικές, μηχανικές αναζητήσιμες μονάδες. Ορισμένες έννοιες XML ενσωματώνονται επίσης στα κύρια προϊόντα σχεσιακής βάσης δεδομένων.

- **Δικτυακές Βάσεις Δεδομένων**

Τα υπερμέσα μπορεί να θεωρηθούν ως ένας τύπος βάσης δεδομένων δικτύου. Τα υπερμέσα είναι ένα σύστημα ανάκτησης πληροφοριών που βασίζεται σε υπολογιστή που επιτρέπει σε έναν χρήστη να αποκτήσει ή να παρέχει πρόσβαση σε κείμενο, εικόνες (ακίνητες και κινούμενες) και ήχο μέσω υπερσυνδέσμων. Τα περισσότερα υπερμέσα αποτελούνται από υπερκείμενο και το μεγαλύτερο παράδειγμα είναι, φυσικά, ο ιστός.

- **Σχεσιακές Βάσεις Δεδομένων**

Μια σχεσιακή βάση δεδομένων είναι ένας τρόπος οργάνωσης δεδομένων έτσι ώστε να φαίνεται στον χρήστη ότι είναι αποθηκευμένα σε μια σειρά αλληλένδετων πινάκων. Το ενδιαφέρον για αυτό το μοντέλο αρχικά περιορίστηκε στον ακαδημαϊκό κόσμο, ίσως επειδή η θεωρητική βάση δεν είναι εύκολο να κατανοηθεί, και έτσι τα πρώτα εμπορικά προϊόντα, Oracle και DB2, εμφανίστηκαν μόλις το 1980 περίπου. Στη συνέχεια, οι σχεσιακές βάσεις δεδομένων έγιναν ο κυρίαρχος τύπος υψηλής απόδοσης εφαρμογές λόγω της αποτελεσματικότητάς τους, της ευκολίας χρήσης και της ικανότητάς τους να εκτελούν μια ποικιλία χρήσιμων εργασιών που δεν είχαν αρχικά οραματιστεί.

- **Σύστημα διαχείρισης Βάσεων Δεδομένων(DBMS)**

Ένα σύστημα διαχείρισης βάσεων δεδομένων είναι λογισμικό που έχει δημιουργηθεί για να επιτρέπει την αποτελεσματική χρήση και διαχείριση των βάσεων δεδομένων, συμπεριλαμβανομένης της διασφάλισης ότι τα δεδομένα είναι συνεπή και σωστά και διευκολύνουν την ενημέρωσή τους. Για μικρές βάσεις δεδομένων ενός χρήστη, η διαχείριση όλων των συναρτήσεων γίνεται συχνά από ένα μόνο πρόγραμμα. Για μεγαλύτερες βάσεις δεδομένων πολλών χρηστών, συνήθως εμπλέκονται πολλαπλά προγράμματα και γενικά χρησιμοποιείται μια αρχιτεκτονική πελάτη-διακομιστή. Τα πρώτα DBMS αναπτύχθηκαν τη δεκαετία του 1960 σε μια προσπάθεια να κάνουν πιο αποτελεσματική χρήση των νέων συσκευών αποθήκευσης άμεσης πρόσβασης (δηλαδή, μονάδες σκληρού δίσκου) που γίνονταν διαθέσιμες ως συμπληρώματα και ενδεχόμενες αντικαταστάσεις για διάτρητες κάρτες και μαγνητική ταινία. Η λέξη βάση δεδομένων χρησιμοποιείται συνήθως με ευρεία έννοια για να αναφέρεται όχι μόνο στα δομημένα δεδομένα αλλά και στο ΣΔΒΔ που χρησιμοποιείται μαζί τους.

- **Δημιουργία Βάσης Δεδομένων**

Για την δημιουργία της βάσης δεδομένων μας χρησιμοποιήσαμε την SQLite. Δημιουργήσαμε ένα σχήμα των τριών tables που θέλαμε να χρησιμοποιήσουμε στην βάση μας ως εξής:

```

CREATE TABLE IF NOT EXISTS Data(
    H_id INTEGER PRIMARY KEY AUTOINCREMENT,
    H_name STRING NOT NULL,
    H_region STRING NOT NULL,
    H_category STRING NOT NULL,
    H_beds STRING NOT NULL,
    H_yaxis FLOAT NOT NULL,
    H_xaxis FLOAT NOT NULL,
    H_address STRING NOT NULL,
    H_phone STRING NOT NULL
);

CREATE TABLE IF NOT EXISTS RatingData(
    R_id INTEGER PRIMARY KEY AUTOINCREMENT,
    R_H_yaxis FLOAT NOT NULL,
    R_H_xaxis FLOAT NOT NULL,
    R_rating INTEGER NOT NULL,
    R_comments VARCHAR NOT NULL
);

CREATE TABLE IF NOT EXISTS ScrapData(
    S_name STRING NOT NULL,
    S_address STRING NOT NULL,
    S_locality STRING NOT NULL,
    S_region STRING NOT NULL,
    S_activities STRING NOT NULL,
    S_yaxis FLOAT NOT NULL,
    S_xaxis FLOAT NOT NULL,
    S_phone STRING NOT NULL PRIMARY KEY,
    S_website STRING NOT NULL
);

```

Εικόνα 8 Schema της βάσης

Στην συνέχεια με την χρήση της python σε VisualStudioCode ,κάνοντας στο αρχείο created.py import την sqlite3 αλλά και χρησιμοποιώντας τις παρακάτω εντολές της:

```
#create my Database
connection = sqlite3.connect('project.db')

with open('schema.sql') as f:
    connection.executescript(f.read())
```

Εικόνα 9 Δημιουργία και σύνδεση βάσης

Καταφέραμε να δημιουργήσουμε την βάση μας με τα tables που πήραμε από το σχήμα μας.

Όσον αφορά τα tables χρησιμοποιήσαμε την εντολή (CREATE TABLE IF NOT EXISTS) ώστε να αποφύγουμε τυχόν προβλήματα. Θα παρατηρήσουμε ότι η διαφορά των πρώτων δύο tables σε σχέση με το τρίτο είναι ότι δεν χρησιμοποιούμε κάποιο id αυτόματα αυξανόμενο. Ο λόγος είναι ότι θέλαμε να χρησιμοποιήσουμε σαν primary key το S_phone και δεν χρειάστηκε να χρησιμοποιήσουμε id αυτόματα αυξανόμενο.

Για τα δεδομένα των tables χρησιμοποιήσαμε δύο csv αρχεία, πιο συγκεκριμένα για το πρώτο και τρίτο table(Data & ScrapData).

Στην πρώτη περίπτωση δημιουργήσαμε ένα στατικό csv αρχείο το οποίο ονομάζεται bookFinal.csv και περιέχει 139 νοσοκομεία με τις εξής πληροφορίες (Όνομα νοσοκομείου, Περιοχή, Κατηγορία, Αριθμός Κλινών, Χ συντεταγμένη, Υ συντεταγμένη, Διεύθυνση ,Τηλέφωνο). Ο λόγος που κάναμε στατικά αυτό το csv είναι διότι η ίδρυση κάποιου νοσοκομείου δεν είναι και συστηματικό φαινόμενο και δεύτερον διότι θέλαμε να εμπλουτίσουμε τα δεδομένα για το κάθε νοσοκομείο.

Στην δεύτερη περίπτωση το csv μας παίρνει δεδομένα για τα ιατρικά διαγνωστικά κέντρα με τις εξής πληροφορίες(Όνομα διαγνωστικού κέντρου,διεύθυνση,δήμος,περιοχή,Εξετάσεις,Χσυντεταγμένη,Υσυντεταγμένη,τηλέφωνο,Ιστοσελίδα) με την μέθοδο του web scraping και ότι δεδομένα χρειαζόμαστε από την αντίστοιχη ιστοσελίδα Χρυσός Οδηγός(xo.gr) τα αποθηκεύει προσωρινά σε αυτό και στην συνέχεια μέσα στην βάση μας.

Επιπρόσθετα, για το τρίτο table που αφορά τις κριτικές των νοσοκομείων δεν χρησιμοποιήσαμε κάποιο csv αρχείο αλλά εκτελούσαμε δυο λειτουργίες άντληση πληροφοριών μέσω SELECT και προσθήκη στην βάση μέσω του INSERT.Ουσιαστικά το αρχείο αρχικά ήταν άδειο και

διαβάζουμε τα δεδομένα που εμείς οι ίδιοι προσθέτουμε ώστε να επιτύχουμε τις ανάλογες λειτουργίες.

Ο τρόπος που αποθηκεύουμε τα δεδομένα από τα csv αρχεία στην βάση μας είναι ο εξής:

-Για τα νοσοκομεία:

```
#open csv for reading only
file = open('bookFinal.csv','r')

#read the lines from my csv file diaxwironeno me comma
grammes = csv.reader(file,delimiter = ',')
#agnow tin prwti grammi tou csv pou einai oi onomasies twv columns pou
next(grammes)

#diniourgw ana query pou kanw insert ta sugkekrimena dedomena apo to csv arxeio
queryGrammes = "INSERT INTO Data (H_name,H_region,H_category,H_beds,H_yaxis,H_xaxis,H_address,H_phone) VALUES (?,?, ?,?, ?,?, ?,?)"

result = "SELECT * FROM Data"
rows = conn.execute(result).fetchall()

if(len(rows) < 139):
    #pass my datas into my table
    conn.executemany(queryGrammes,grammes)
```

Εικόνα 10 CSV Νοσοκομείων

-Για τα διαγνωστικά Κέντρα:

```
scrappfile = open('scrapping.csv','r')
scrappgrammes = csv.reader(scrappfile,delimiter=',')
next(scrappgrammes)
# for scrap in scrappgrammes:
#     print(scrap)

scrappquery = "INSERT ON DUPLICATE KEY UPDATE INTO Scrapdata (S_name,S_address,S_locality,S_region,S_activities,S_yaxis,S_xaxis,S_phone,S_website) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)"
conn.executemany(scrappquery,scrappgrammes)
```

Εικόνα 11 CSV Διαγνωστικών Κέντρων

Όπου διαβάζοντας το αντίστοιχο κάθε φορά csv αρχείο και διαχωρίζοντας στην πρώτη περίπτωση τα δεδομένα μας με ,(κόμμα) και στην δεύτερη με ;(ερωτηματικό) και αγνοώντας πάντα την πρώτη γραμμή που περιέχει τα ονόματα των στηλών μας εκτελούμε executemany όπου σύμφωνα με το πρότυπο που έχουμε δώσει στο INSERT μας(ποια δεδομένα αλλά και την σειρά τους) γίνεται η εισαγωγή των δεδομένων στην βάση.

• SQLite

Η SQLite είναι μια βιβλιοθήκη εντός διαδικασίας που υλοποιεί μια αυτόνομη μηχανή βάσης δεδομένων SQL χωρίς διακομιστές με μηδενικές ρυθμίσεις παραμέτρων. Ο κώδικας για την SQLite είναι δημόσιος τομέας και επομένως είναι δωρεάν για χρήση για οποιονδήποτε σκοπό, εμπορικό ή ιδιωτικό. Είναι η πιο ευρέως διαδεδομένη βάση δεδομένων στον κόσμο με περισσότερες εφαρμογές από όσες μπορούμε να μετρήσουμε, συμπεριλαμβανομένων πολλών έργων υψηλού προφίλ.

Επίσης είναι μια ενσωματωμένη μηχανή βάσης δεδομένων SQL. Σε αντίθεση με τις περισσότερες βάσεις δεδομένων SQL, την SQLite δεν διαθέτει ξεχωριστή διαδικασία διακομιστή. Επιπρόσθετα, διαβάζει και γράφει απευθείας σε συνηθισμένα αρχεία δίσκου. Μια πλήρης βάση δεδομένων SQL με πολλαπλούς πίνακες, δείκτες, εναύσματα και προβολές, περιέχεται σε ένα ενιαίο αρχείο δίσκου. Η μορφή αρχείου της βάσης δεδομένων είναι cross-platform - μπορείτε ελεύθερα να αντιγράψετε μια βάση δεδομένων μεταξύ συστημάτων 32-bit και 64-bit ή μεταξύ αρχιτεκτονικών big-end και small-endian. Αυτά τα χαρακτηριστικά κάνουν την SQLite μια δημοφιλή επιλογή ως μορφή αρχείου εφαρμογής. Τα αρχεία βάσης δεδομένων SQLite είναι μια προτεινόμενη μορφή αποθήκευσης από τη Βιβλιοθήκη του Κογκρέσου των ΗΠΑ.

Επιπρόσθετα, η SQLite είναι μια συμπαγής βιβλιοθήκη. Με όλες τις δυνατότητες ενεργοποιημένες, το μέγεθος της βιβλιοθήκης μπορεί να είναι μικρότερο από 750 KB, ανάλογα με την πλατφόρμα-στόχο και τις ρυθμίσεις βελτιστοποίησης του μεταγλωττιστή. (Ο κώδικας 64 bit είναι μεγαλύτερος, και ορισμένες βελτιστοποιήσεις μεταγλωττιστή, όπως η επιθετική εισαγωγή συναρτήσεων και το ξετύλιγμα βρόχου μπορεί να προκαλέσουν πολύ μεγαλύτερο μέγεθος του κώδικα αντικειμένου.) Υπάρχει μια αντιστάθμιση μεταξύ χρήσης μνήμης και ταχύτητας. Η SQLite γενικά τρέχει πιο γρήγορα όσο περισσότερη μνήμη της δίνουμε. Ωστόσο, η απόδοση είναι συνήθως αρκετά καλή ακόμη και σε περιβάλλοντα χαμηλής μνήμης. Ανάλογα με τον τρόπο χρήσης του, η SQLite μπορεί να είναι ταχύτερη από το άμεσο I/O του συστήματος αρχείων.

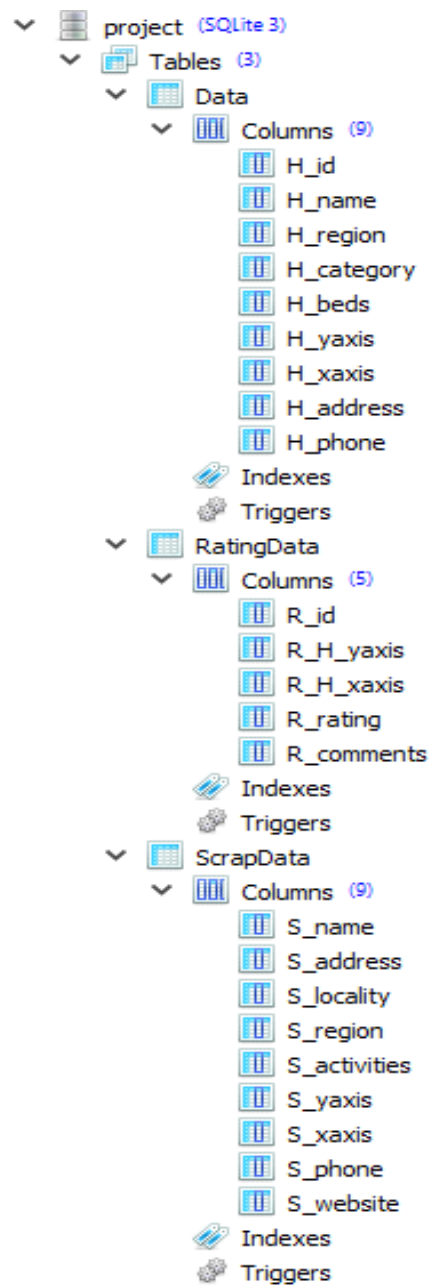
Τέλος, η SQLite ελέγχεται πολύ προσεκτικά πριν από κάθε κυκλοφορία και έχει τη φήμη ότι είναι πολύ αξιόπιστη. Το μεγαλύτερο μέρος του πηγαίου κώδικα SQLite είναι αφιερωμένο αποκλειστικά σε δοκιμές και επαλήθευση. Μια αυτοματοποιημένη σουίτα δοκιμών εκτελεί

εκατομμύρια και εκατομμύρια δοκιμαστικές περιπτώσεις που περιλαμβάνουν εκατοντάδες εκατομμύρια μεμονωμένες δηλώσεις SQL και επιτυγχάνει 100% κάλυψη δοκιμής κλάδου. Ανταποκρίνεται με χάρη σε αποτυχίες εκχώρησης μνήμης και σφάλματα εισόδου/εξόδου δίσκου. Οι συναλλαγές είναι ACID(ατομικότητα, συνέπεια, απομόνωση, μονιμότητα) ακόμα και αν διακόπτονται από σφάλματα συστήματος ή διακοπή ρεύματος. Όλα αυτά επαληθεύονται από τις αυτοματοποιημένες δοκιμές με τη χρήση ειδικών δοκιμαστικών ιμάντων που προσομοιώνουν βλάβες του συστήματος. Φυσικά, ακόμα και με όλες αυτές τις δοκιμές, εξακολουθούν να υπάρχουν σφάλματα. Αλλά σε αντίθεση με ορισμένα παρόμοια έργα (ιδίως εμπορικούς ανταγωνιστές), η SQLite είναι ανοιχτή και ειλικρινής για όλα τα σφάλματα και παρέχει λίστες σφαλμάτων και χρονολογίες αλλαγών κώδικα ανά λεπτό.

Για διευκόλυνση μας πέρα από την τύπωση στο Visual Studio Code χρησιμοποιήσαμε και ένα ακόμη εργαλείο, το SQLiteStudio το οποίο αξιοποιούσαμε για να παρατηρούμε τυχόν αλλαγές στα δεδομένα μας είτε λόγω Web Scraping είτε λόγω των INSERT στην βάση.

Το εργαλείο SQLiteStudio είναι ένα δωρεάν εργαλείο GUI για τη διαχείριση βάσεων δεδομένων SQLite. Είναι δωρεάν, φορητό, διαισθητικό και cross-platform. Το εργαλείο SQLite παρέχει επίσης μερικές από τις πιο σημαντικές δυνατότητες για εργασία με βάσεις δεδομένων SQLite, όπως εισαγωγή, εξαγωγή δεδομένων σε διάφορες μορφές, όπως CSV, XML και JSON.










Μια εικόνα της βάσης μας είναι η παρακάτω:



Εικόνα 12 Η βάση μέσω SQLiteStudio






Και αναλυτικά τα δεδομένα μας:

Για τα νοσοκομεία:

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	H_id	INTEGER							NULL
2	H_name	STRING							NULL
3	H_region	STRING							NULL
4	H_category	STRING							NULL
5	H_beds	STRING							NULL
6	H_yaxis	FLOAT							NULL
7	H_xaxis	FLOAT							NULL
8	H_address	STRING							NULL
9	H_phone	STRING							NULL











Εικόνα 13 Data Νοσοκομείων

Για τις κριτικές:

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	R_id	INTEGER							NULL
2	R_H_yaxis	FLOAT							NULL
3	R_H_xaxis	FLOAT							NULL
4	R_rating	INTEGER							NULL
5	R_comments	VARCHAR							NULL

Εικόνα 14 Data για τις κριτικές

Για τα ιατρικά διαγνωστικά κέντρα:

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	S_name	STRING							NULL
2	S_address	STRING							NULL
3	S_locality	STRING							NULL
4	S_region	STRING							NULL
5	S_activities	STRING							NULL
6	S_yaxis	FLOAT							NULL
7	S_xaxis	FLOAT							NULL
8	S_phone	STRING							NULL
9	S_website	STRING							NULL

Εικόνα 15 Data Διαγνωστικών Κέντρων

- **Σύνδεση βάσης με backend**

Πέρα από την δημιουργία της βάσης μας είναι απαραίτητη και η σύνδεση της με σκοπό να μπορούμε να την αξιοποιήσουμε. Έτσι λοιπόν στο αρχείο `renders.py` αρχικά κάνοντας `import` την `sqlite3` δημιουργούμε μια συνάρτηση την οποία ονομάζουμε `get_db_connection` μέσα στην οποία αρχικά συνδεόμαστε με την βάση μας και στην συνέχεια διαβάζουμε όλα τα δεδομένα που υπάρχουν μέσα σε αυτήν ως εξής:

```
def get_db_connection():  
    conn = sqlite3.connect('project.db')  
    with conn:  
        conn.row_factory = sqlite3.Row  
    return conn
```

Εικόνα 16 Σύνδεση βάσης

Καθώς έχουμε συνδεθεί στην βάση αντιστοιχούμε με γραμμές(rows) τα δεδομένα της ώστε να τα αξιοποιήσουμε επιστρέφοντας στο τέλος την `conn`, που υποδηλώνει την σύνδεση και αυτό έχει ως στόχο την κλήση της συνάρτησης αυτής μέσα σε άλλες που θα είναι απαραίτητη η πρόσβαση στην βάση είτε για άντληση πληροφοριών μέσω `SELECT` είτε για προσθήκη σε αυτήν μέσω `INSERT`.

Όσον αφορά την εντολή (`conn.row_factory = sqlite3.Row`), μετά τη δημιουργία της σύνδεσης, οι σειρές μετατρέπονται σε λίστα αντικειμένων σειράς `SQLite`. Οι στήλες δεδομένων που αντιπροσωπεύονται από αντικείμενα γραμμής μπορούν να ανακτηθούν είτε με τη σειρά (όπως μια λίστα) είτε χρησιμοποιώντας το όνομα του πεδίου ως κλειδί (όπως ένα λεξικό). Όπως ένα λεξικό, μπορείτε επίσης να χρησιμοποιήσετε τη μέθοδο `keys()` του για να λάβετε τη λίστα με τα ονόματα των πεδίων.

- **Web Scraping**

- **Έννοια Web Scraping**

Στον σημερινό ανταγωνιστικό κόσμο, όλοι αναζητούν τρόπους για να καινοτομήσουν και να κάνουν χρήση των νέων τεχνολογιών. Το web scraping (ονομάζεται επίσης εξαγωγή δεδομένων ιστού ή απόξεση δεδομένων) παρέχει μια λύση για όσους θέλουν να αποκτήσουν πρόσβαση σε δομημένα δεδομένα ιστού με αυτοματοποιημένο τρόπο. Η απόξεση ιστού είναι χρήσιμη εάν ο δημόσιος ιστότοπος από τον οποίο θέλετε να λάβετε δεδομένα παρέχει μόνο περιορισμένη πρόσβαση στα δεδομένα.

Μερικές από τις κύριες περιπτώσεις web scraping περιλαμβάνουν παρακολούθηση τιμών, ευφυΐα τιμών, παρακολούθηση ειδήσεων και έρευνα αγοράς μεταξύ πολλών άλλων. Γενικά χρησιμοποιείται από άτομα και επιχειρήσεις που θέλουν να κάνουν χρήση του τεράστιου όγκου των διαθέσιμων στο κοινό δεδομένων για να λάβουν πιο έξυπνες αποφάσεις. Σε αντίθεση με την εγκόσμια διαδικασία της μη αυτόματης εξαγωγής δεδομένων, το web scraping χρησιμοποιεί έξυπνη αυτοματοποίηση για να ανακτήσει εκατοντάδες, εκατομμύρια ή και δισεκατομμύρια σημεία δεδομένων από τα φαινομενικά ατελείωτα σύνορα του Διαδικτύου.

- **Αξιοποίηση Web Scraping**

Το web scraping έχει μια τεράστια γκάμα εφαρμογών. Ένα τέτοιο εργαλείο μπορεί να σας βοηθήσει να αυτοματοποιήσετε τη διαδικασία εξαγωγής πληροφοριών από άλλους ιστότοπους, γρήγορα και με ακρίβεια. Μπορεί επίσης να διασφαλίσει ότι τα δεδομένα που έχετε εξάγει είναι σωστά οργανωμένα, διευκολύνοντας την ανάλυση και την χρήση τους σε άλλα έργα. Στον κόσμο του ηλεκτρονικού εμπορίου χρησιμοποιείται ευρέως για την παρακολούθηση τιμών ανταγωνιστή. Είναι ο μόνος πρακτικός τρόπος για τις επωνυμίες να ελέγχουν την τιμολόγηση των προϊόντων και των υπηρεσιών των ανταγωνιστών τους, επιτρέποντας τους να προσαρμόσουν τις δικές τους στρατηγικές τιμών και να παραμείνουν μπροστά από το παιχνίδι. Οι οργανισμοί έρευνας αγοράς και οι αναλύσεις εξαρτώνται από την εξαγωγή δεδομένων ιστού για να μετρήσουν το συναίσθημα των καταναλωτών παρακολουθώντας τις διαδικτυακές κριτικές προϊόντων, τα άρθρα και τα σχόλια.

Όσον αφορά την χρήση του web scraping για την διπλωματική μας εργασία, το χρησιμοποιήσουμε ώστε να ανακτήσουμε δεδομένα που αφορούν τα ιατρικά διαγνωστικά κέντρα της Ελλάδος. Στα δεδομένα αυτά συμπεριλαμβάνονται πληροφορίες για το όνομα του διαγνωστικού φυσικά, την περιοχή και τον δήμο που ανήκουν, την διεύθυνση της τοποθεσίας τους, ένα σταθερό ή κινητό τηλέφωνο, πληροφορίες για την ιστοσελίδα του καθενός εάν υπάρχει αλλά και τις δραστηριότητες του κάθε διαγνωστικού κέντρου, δηλαδή τις εξετάσεις που μπορεί να κάνει ένας ασθενής. Οι εξετάσεις αυτές έχουν μεγάλης κλίμακας, μπορεί να είναι οι απλές καθημερινές αλλά και διαφόρων ειδών εξειδικευμένες.

- ## Ιστοσελίδα

- ### Δομή Ιστοσελίδας

Η δομή της ιστοσελίδας μας αποτελείται από 5 templates(Home,GreekHospitals,Search For Hospitals,Search For Diagnostic Centers,Rate Hospitals) τα οποία είναι αποτυπωμένα σε μπάρα μενού(nav-bar) όπου επιλέγοντας κάθε μία από τις επιλογές κατευθυνόμαστε στην αντίστοιχη λειτουργία.

Home: Η σελίδα αυτή αποτελεί την αρχική μας στην οποία έχουμε ένα φόντο σε συνδυασμό με κάποια λόγια καλωσορίσματος, αλλά και λίγα με σκοπό της ευκολότερη κατεύθυνση στην ιστοσελίδα μας.

Greek Hospitals: Η δεύτερη κατά σειρά σελίδα συμπεριλαμβάνει κάποιες πληροφορίες για τα νοσοκομεία με σκοπό της διευκόλυνση μας στην επιλογή των κριτηρίων για την χαρτογράφηση των νοσοκομείων. Η μορφή που εμφανίζονται τα δεδομένα μας είναι μέσω ενός πίνακα ,ο οποίος είναι γεμισμένος αυτοματοποιημένα μέσω κατάλληλων ερωτήσεων στην βάση μας.

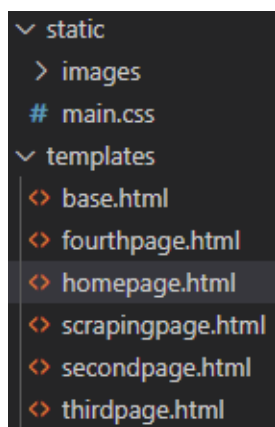
Search For Hospitals: Η Τρίτη σελίδα μας αναφέρεται στην λειτουργία της χαρτογράφησης των νοσοκομείων που έχουμε στην βάση μας. Αποτελείται από έναν χάρτη μέσω της βιβλιοθήκης Leaflet ο οποίος χάρτης είναι κατάλληλα επικεντρωμένος σε τέτοια θέση ώστε να έχουμε την καλύτερη δυνατή όψη του ελλαδικού χάρτη. Στην συνέχεια έχουμε δημιουργήσει τρία πλαίσια επιλογής που αντιστοιχούν στα κριτήρια για την εύρεση των νοσοκομείων(Περιοχή, Κατηγορία, Αριθμός Κλινών) αλλά και δύο κουμπιά όπου το κάθε ένα αποσκοπεί σε μία λειτουργία.

Search For Diagnostic Centers:Στην τέταρτη σελίδα μας απεικονίζεται η χαρτογράφηση των ιατρικών διαγνωστικών κέντρων με δεδομένα από την βάση μας. Τα δεδομένα αυτά σε

αντίθεση με τα δεδομένα των νοσοκομείων που είναι σε στατικό αρχείο, ανακτώνται μέσω της διαδικασίας του web scraping. Έχουμε κατασκευάσει και πάλι δύο πλαίσια επιλογών (Περιοχή, Δήμος) που αντιστοιχούν στα κριτήρια για την χαρτογράφηση των ιατρικών διαγνωστικών κέντρων. Επίσης έχουμε δύο κουμπιά που το κάθε ένα κατά το πάτημα εκτελεί μια λειτουργία.

Rate Hospitals: Η Πέμπτη και τελευταία σελίδα μας έχει να κάνει με την κριτική των νοσοκομείων που επιτυγχάνεται μέσω αστεριών. Σε αυτήν την σελίδα υπάρχει ένα πλαίσιο επιλογών που αφορά την περιοχή των νοσοκομείων αλλά και δυο κουμπιά με δύο λειτουργίες. Η διαδικασία της κριτικής γίνεται με δυναμικό τρόπο δηλαδή εκτελώντας μια κριτική αποθηκεύεται στην βάση μας και όταν γίνει δεύτερη κριτική για το νοσοκομείο αυτό προστίθεται και αυτή στην βάση και επιστρέφουμε τον μέσο όρο αστεριών κάνοντας τις κατάλληλες ερωτήσεις στην βάση μας.

Τα html & css αρχεία που χρησιμοποιήσαμε για την δημιουργία της ιστοσελίδας μας είναι τα εξής:



Εικόνα 17 Αρχεία για δημιουργία του site

Το αρχείο homepage.html αντιστοιχεί στην αρχική μας

Το αρχείο secondpage.html αντιστοιχεί στην σελίδα Greek Hospitals

Το αρχείο thirdpage.html αντιστοιχεί στην σελίδα Search For Hospitals

Το αρχείο fourthpage.html αντιστοιχεί στην σελίδα Rate Hospitals

και το αρχείο scraping.html αντιστοιχεί στην σελίδα Search For Diagnostic Centers

Επιπρόσθετα το αρχείο base.html αποτελεί τον «πατέρα» των υπολοίπων σελίδων καθώς την κληρονομούν διότι αποτελεί το βασικό εμφανισιακό κομμάτι των σελίδων μας. Αυτή την διαδικασία την κάνουμε ώστε να μην χρειάζεται να υπερφορτώνουμε όλα τα html αρχεία μας. Ο

τρόπος της κληρονόμησης επιτυγχάνεται μέσω Jinja. Τέλος για την διαμόρφωση της ιστοσελίδας μας χρησιμοποιούμε ένα css αρχείο(main.css).

Όσον αφορά την κληρονόμηση μέσω Jinja όλα τα html αρχεία μας κληρονομούν το βασικό αρχείο με την εντολή `{%extends 'base.html'%}` και στην συνέχεια για το head ή body χρησιμοποιούμε τα block content τα οποία δηλώνουν σε ποιο σημείο του αρχικού html θα υλοποιηθούν οι λειτουργίες των υπολοίπων. Πιο συγκεκριμένα

```
{% block contentA%}  
{% endblock contentA%}  
  
{% block contentB%}  
{% endblock contentB%}  
  
{% block contentC%}  
{% endblock contentC%}  
  
{% block contentD%}  
{% endblock contentD%}  
  
{% block contentE%}  
{% endblock contentE%}
```

Εικόνα 18 Jinja Contents

με αυτόν τον τρόπο μέσα στο αρχικό html δηλώνουμε ότι θα υλοποιηθεί κομμάτι κώδικα των εκάστοτε αρχείων. Τα block content υπάρχουν και στα αρχεία «παιδιά» ώστε να δηλωθεί το εύρος του κομμάτι κώδικα του καθενός.

• Flask(routes)

Το `@app.route` είναι ένας διακομιστής Python που παρέχει η Flask για να εκχωρεί εύκολα διευθύνσεις URL στην εφαρμογή μας σε λειτουργίες. Είναι εύκολο να καταλάβουμε τι συμβαίνει με την πρώτη ματιά, ο διακομιστής λέει στην `@app` μας κάθε φορά που ένας χρήστης επισκέπτεται τον τομέα της εφαρμογής μας στη δεδομένη route να εκτελέσει την αντίστοιχη λειτουργία.

Η δήλωση του app γίνεται μέσω της Flask `app = Flask(__name__)`

Εικόνα 19 Δημιουργία app

και έτσι κάνοντας import το app μέσα στα αρχεία main.py και renders.py μπορούμε να εκκινήσουμε τον σέρβερ και να δημιουργήσουμε τα routes αντίστοιχα.

Στην προκειμένη περίπτωση εμείς χρησιμοποιήσαμε ένα αρχείο της Python (renders.py) στο οποίο υλοποιήσαμε όλα τα routes που χρειαστήκαμε. Πιο συγκεκριμένα χρησιμοποιήσαμε 7 routes (/home, /secondpage, /thirdpage, /fourthpage, /ratingpage, /scrapingpage). Το κάθε ένα από αυτά αντιστοιχεί σε κάθε μία από τις σελίδες μας. Για παράδειγμα στην περίπτωση που βρισκόμαστε στην σελίδα της χαρτογράφησης των ιατρικών διαγνωστικών κέντρων η μπάρα του φυλλομετρητή μας θα είναι `127.0.0.1:5000/scrapingpage`, όπου «/scrapingpage» είναι το route αυτής της σελίδας. Η διαδικασία των routes είναι απαραίτητη ώστε να επιτύχουμε την επιθυμητή λειτουργία σε κάθε σελίδα αλλά και για να καταφέρουμε την επικοινωνία του frontend & backend.

Έχοντας για παράδειγμα το route της αρχικής μας σελίδας

```
@app.route('/home')
def homepage():
    return render_template('homepage.html', title='Home Page')
```

Εικόνα 20 Παράδειγμα Δημιουργίας Route

Εκτελούμε την συνάρτηση homepage ώστε όταν κληθεί από τον διακομιστή το συγκεκριμένο route να εκτελεστεί αυτή η συνάρτηση. Στην δική μας περίπτωση κάνοντας κλικ σε κάποια από τις επιλογές πάνω στην μπάρα μενού ειδοποιείται το ανάλογο route και έτσι εκτελείται η αντίστοιχη συνάρτηση αυτού.

• Επικοινωνία Frontend & Backend

Στην μηχανική λογισμικού (software engineering), οι όροι frontend και backend αναφέρονται στον διαχωρισμό ανάμεσα στο επίπεδο παρουσίασης και στο επίπεδο πρόσβασης δεδομένων ενός τμήματος λογισμικού. Στο μοντέλο πελάτη-διακομιστή, ο πελάτης θεωρείται συνήθως το frontend και ο διακομιστής θεωρείται το backend, ακόμη και όταν κάποια εργασία παρουσίασης γίνεται πραγματικά στον ίδιο διακομιστή.

Επιπρόσθετα η επικοινωνία επιτυγχάνεται με τρεις τρόπους στην διπλωματική εργασία μας σύμφωνα με τις απαιτήσεις κάθε φορά. Πιο συγκεκριμένα η απεικόνιση των αρχικών δεδομένων επιτυγχάνεται με την χρήση του render_template που ουσιαστικά εκτελείται GET μέθοδος

δηλαδή στέλνουμε δεδομένα στην ιστοσελίδα μας. Στην συνέχεια εκτελώντας το πρώτο στάδιο και αποκτώντας κάποια δεδομένα που θα παίξουν τον ρόλο κλειδιού για τις επόμενες λειτουργίες χρησιμοποιούμε την AJAX μέσω της γλώσσας JavaScript ώστε να στείλουμε τα δεδομένα κλειδιά στο backend για να γίνει η κατάλληλη επεξεργασία των δεδομένων και οι ερωτήσεις στην βάση για την ανάκτηση των νέων δεδομένων και την εισχώρηση στην τελική ευθεία της χαρτογράφησης ή την εμφάνιση του μέσου όρου αστεριών των νοσοκομείων στην αντίστοιχη περίπτωση. Με την διαδικασία αυτή ουσιαστικά εκτελείται POST μέθοδος δηλαδή στέλνουμε τα δεδομένα μας από το frontend στο backend. Τέλος για να στείλουμε τα επεξεργασμένα και νέα δεδομένα μας ξανά μπροστά και να τα απεικονίσουμε στην ιστοσελίδα μας χρησιμοποιούμε την μέθοδο jsonify στην οποία στέλνουμε ξανά τα δεδομένα μας σε μορφή JSON(ουσιαστικά λεξικού) για την επίτευξη της τελικής λειτουργίας. Η τελική αυτή μέθοδος ανήκει στην POST μέθοδο καθώς εκτελείται ως σενάριο επιτυχίας στην σύνδεση frontend-backend μέσω AJAX.

- **Frontend**

Το frontend εστιάζει στα εξής:

- Markup and web languages such as HTML, CSS, JavaScript, and ancillary libraries commonly used in those languages such as Sass or jQuery
- Asynchronous request handling and AJAX
- Single-page applications (with frameworks like React, Angular or Vue.js)
- Web performance (largest contentful paint, time to interactive, 60 FPS animations and interactions, memory usage, etc.)
- Responsive web design
- Cross-browser compatibility issues and workarounds
- End-to-end testing with a headless browser
- Build automation to transform and bundle JavaScript files, reduce images size... with tools like Webpack or Gulp.js
- Search engine optimization
- Accessibility concerns

- Basic usage of image editing tools such as GIMP or Photoshop
- User Interface

- **Backend**

Το backend εστιάζει στα εξής:

- Scripting languages like PHP, Python, Ruby, Perl, Node.js, or Compiled languages like C#, Java or Go
- Automated testing frameworks for the language being used
- Application Data Access
- Application Business Logic
- Database administration
- Scalability
- High availability
- Security concerns, authentication, authorization
- Software Architecture
- Data transformation
- Backup methods and software

- **Render_template**

Το `render_template` είναι μια μέθοδος της Flask το οποίο κατά την διαδικασία του routing μας επιτρέπει να περάσουμε σε ένα template κάποιο μήνυμα, να δώσουμε σε συνδυασμό με τα Jinja τίτλο στο template αυτό, όπως επίσης μπορούμε να στείλουμε σταθερές και μεταβλητές.

Στην περίπτωση της διπλωματικής μας εργασίας οι λόγοι που χρησιμοποιήσαμε την μέθοδο αυτή είναι για να δώσουμε τίτλο στα templates μας αλλά και να στείλουμε κάποιες μεταβλητές που θέλαμε να απεικονίσουμε στις σελίδες. Πιο συγκεκριμένα, στην αρχική μας το μοναδικό στοιχείο που στείλαμε ήταν ο τίτλος ενώ στις υπόλοιπες σελίδες είχαμε περισσότερα ορίσματα.

times περασμένα στη βάση μας. Για την εμφάνιση των στοιχείων χρησιμοποιήσαμε το PRIMARY KEY που είναι το id και για το οπτικό κομμάτι το όνομα του δεδομένου από την βάση. Η διαδικασία αυτή είναι επίσης μορφής Jinja.

- **Ajax**

Στην συνέχεια αφού στείλαμε τα δεδομένα μας μέσω της μεθόδου render_template στις σελίδες, επιλέγουμε όποια από τα κριτήρια θέλουμε για να προχωρήσουμε στην διαδικασία της χαρτογράφησης και της άσκησης κριτικής. Όταν γίνει η ολοκλήρωση των επιλογών κριτηρίων, τα δεδομένα κριτήρια τα χρειαζόμαστε ώστε να κάνουμε τις κατάλληλες επιλογές μέσα από την βάση. Χρησιμοποιούμε λοιπόν ένα λεξικό το οποίο μετατρέπουμε σε JSON μορφή και στο οποίο προσθέτουμε τα δεδομένα κριτήρια. Στην συνέχεια μέσα σε script tags γράφουμε τον κώδικα σε JavaScript στον οποίο χρησιμοποιήσαμε την Ajax όπου στείλαμε το Json μορφής λεξικό που δημιουργήσαμε. Ένα παράδειγμα της υλοποίησης μας

```
$.ajax({  
  url: "/thirdpage",  
  type: "POST",  
  contentType: "application/json",  
  data: JSON.stringify(lv),  
})
```

Εικόνα 21 Παράδειγμα Υλοποίησης Ajax

στο οποίο επιλέγουμε σε ποιο route θέλουμε να στείλουμε τα δεδομένα μας, στην προκειμένη περίπτωση επιλέξαμε το /thirdpage στο οποίο στείλαμε {περιοχή, κατηγορία, αριθμός κλινών}. Ο τύπος παρακάτω είναι POST διότι θέλουμε να στείλουμε δεδομένα από την σελίδα και όχι τα λάβουμε. Ο τύπος που περιεχομένου που στέλνουμε θέλουμε να έχει json μορφή οπότε και το δηλώνουμε και τέλος αναγράφουμε το λεξικό που θα αποστείλουμε στο backend. Την διαδικασία αυτή την εκτελέσαμε συνολικά τέσσερις φορές. Μια φορά στο thirdpage.html όπου κάναμε την παραπάνω διαδικασία, μία φορά στο scraping.html όπου στείλαμε την δυάδα δεδομένων κριτηρίων που αφορούν την χαρτογράφηση των ιατρικών διαγνωστικών κέντρων (περιοχή και δήμος) και δύο φορές στο fourthpage.html όπου αναφέρεται στην κριτική των νοσοκομείων. Την πρώτη φορά στείλαμε το δεδομένο κριτήριο για την επιλογή των νοσοκομείων σύμφωνα με την επιλεγμένη περιοχή για την άσκηση κριτικής και την δεύτερη φορά στείλαμε τα κάποια επεξεργασμένα δεδομένα που αποτέλεσαν σημαντικό ρόλο στην τελική άσκηση κριτικής.

- **Json-Jsonify**

Τελικό στάδιο για την επικοινωνία μεταξύ του frontend-backend αποτέλεσε η jsonify μέθοδος. Ανήκει στην κατηγορία της POST μεθόδου καθώς μετά τις λειτουργίες που εκτελούμε επιστρέφουμε την jsonify μέθοδο με τα ορίσματα που επιθυμούμε. Πιο συγκεκριμένα η διαδικασία της εκτέλεσης

```
$.ajax({  
  url: "/thirdpage",  
  type: "POST",  
  contentType: "application/json",  
  data: JSON.stringify(lv),  
  success: function(dictionary3){
```

Εικόνα 22 Ενδεχόμενο Επιτυχίας Ajax

όπου συνεχίζεται από την μία πλευρά κατά το τέλος της Ajax ενώ από την άλλη πλευρά η διαδικασία ξεκινάει στις μεθόδους των routes.

```
return jsonify(dictionary3)
```

Εικόνα 23 Κλήση jsonify

Το dictionary3, το οποίο είναι ένα λεξικό περιέχει τα δεδομένα που ανακτήσαμε μετά τις επεξεργασίες που εκτελέσαμε μέσα στις μεθόδους των routes και τελικά τα στέλνουμε και πάλι μπροστά ώστε να ολοκληρωθεί η λειτουργία σε κάθε σελίδα.

Αφότου εκτελεστεί η Ajax του προηγούμενου ερωτήματος, κατά το ενδεχόμενο της επιτυχίας(success) , γίνεται η λήψη των δεδομένων μέσω του διαύλου επικοινωνίας που δημιουργήθηκε και έτσι λαμβάνουμε τα επεξεργασμένα δεδομένα τα οποία είναι έτοιμα για χρήση. Την διαδικασία αυτή την εκτελέσαμε όσες φορές και την Ajax. Στην περίπτωση του /thirdpage στείλαμε δεδομένα για το όνομα, τη διεύθυνση, τον αριθμό επικοινωνίας αλλά και τις x και y συντεταγμένες. Την ίδια ακριβώς διαδικασία εκτελέσαμε και για το /fourthpage με τα ίδια δεδομένα. Με διαφορετικό τρόπο κινηθήκαμε όμως στο /ratingpage.

Η μεταξύ τους διαφορά είναι ότι επειδή στην fourthpage.html χρειαστήκαμε να αποκτήσουμε πολλαπλή επικοινωνία δημιουργήσαμε ένα επιπλέον route το οποίο δεν αντιστοιχεί σε κάποια σελίδα αλλά το χρησιμοποιήσαμε ως δίαυλο επικοινωνίας για να μπορέσουμε να στείλουμε και να λάβουμε δεδομένα από το frontend στο backend και αντίστροφα περισσότερες από μία φορές. Η διαδικασία αυτή αφορά την κριτική των νοσοκομείων όπου αρχικά στείλαμε τα πρώτα δεδομένα για την αναπαράσταση τους στα πλαίσια επιλογών, στην συνέχεια στείλαμε πίσω αυτά τα δεδομένα για να τα επεξεργαστούμε και να δημιουργήσουμε εκ νέου κριτήρια τα οποία μέσω

του πρώτου route τα στείλαμε στην σελίδα για την επιλογή των αστεριών και την δυνατότητα αναφοράς κάποιων σχολίων για το επιλεγμένο νοσοκομείο. Τέλος τα δεδομένα αυτά τα στέλνουμε ξανά πίσω ώστε να κάνουμε την εκτίμηση του μέσου όρου αστεριών του νοσοκομείου στο οποίο ασκήθηκε η κριτική και την ανάκτηση του πιο πρόσφατου σχολίου που παραδόθηκε. Έτσι με το δεύτερο και επιπλέον route στέλνουμε τα τελικά μας δεδομένα μπροστά με σκοπό την απεικόνιση τους ως τελικό αποτέλεσμα της κριτικής.

- ## Χρήση Δεδομένων

- ### Ομαδοποίηση Δεδομένων

Για την ομαδοποίηση των δεδομένων μας κινηθήκαμε με δύο τρόπους .Στην πρώτη περίπτωση που αφορά την χαρτογράφηση των νοσοκομείων και την άσκηση κριτικής δουλέψαμε με τον ίδιο τρόπο. Έχοντας τα δεδομένα μας στην βάση επιλέξαμε τα δεδομένα για τις περιοχές, τις κατηγορίες και τον αριθμό των κλινών, επιλέγοντας τα δεδομένα χωρίς διπλοτιμές και ταξινομημένα ως προς καθένα από αυτά.

```
beds = conn.execute('SELECT DISTINCT H_beds FROM Data ORDER BY H_beds').fetchall()
regions = conn.execute('SELECT DISTINCT H_region FROM Data ORDER BY H_region').fetchall()
categories = conn.execute('SELECT DISTINCT H_category FROM Data ORDER BY H_category').fetchall()
```

Εικόνα 24 Ομαδοποίηση Δεδομένων για πλαίσια επιλογών 1

Οι τρεις αυτές λίστες περιέχουν τα προαναφερθέντα δεδομένα, το beds περιέχει τους αριθμούς των κλινών, το regions τις περιοχές και το categories τις κατηγορίες των νοσοκομείων. Τα δεδομένα αυτά προορίζονται για τα πλαίσια επιλογών μέσα στις φόρμες των σελίδων. Στην δεύτερη περίπτωση που αναφέρεται στην χαρτογράφηση των ιατρικών διαγνωστικών κέντρων, τα δεδομένα μας τα ανακτούμε μέσω του web scraping. Η ιστοσελίδα που χρησιμοποιήσαμε είναι αυτή του χρυσού οδηγού([www](http://www.xo.gr/) [HYPERLINK "http://www.xo.gr/".](http://www.xo.gr/) [HYPERLINK "http://www.xo.gr/"](http://www.xo.gr/) [HYPERLINK "http://www.xo.gr/"](http://www.xo.gr/) [gr](http://www.xo.gr/)) όπου μέσω του πηγαίου κώδικα της ιστοσελίδας βρήκαμε και ανακτήσαμε τα δεδομένα μας για την κάθε περίπτωση. Ένα παράδειγμα εκτέλεσης είναι αυτό της ιστοσελίδας του κάθε ιατρικού διαγνωστικού κέντρου.

```

websites = driver.find_elements_by_class_name('listingMainCallToActions')
for website in websites:
    try:
        #print(website.text)
        web = website.find_element_by_xpath('..//li/a').get_attribute('href')
        weblist.append(web)
        # print(web)
    except Exception:
        weblist.append('')

```

Εικόνα 25 Παράδειγμα Υλοποίησης Web Scraping

Χρησιμοποιώντας την selenium και το webdriver της αποκτήσαμε πρόσβαση στα δεδομένα της ιστοσελίδας. Αρχικά εντοπίζουμε όλες τις ιστοσελίδες σύμφωνα με το όνομα κλάσης που είναι δηλωμένα και στην συνέχεια διατρέχοντας μια for loop εντοπίζουμε την κάθε μία εξ αυτών ξεχωριστά και τα προσθέτουμε σε μία λίστα ώστε να τα έχουμε συγκεντρωμένα. Ο τρόπος που πήραμε το κάθε ένα είναι έχοντας αρχικά βρει την κοινή κλάση, προχωρούμε μέσω του μονοπατιού ώστε να βρούμε το όνομα σαν href του καθενός. Τέλος ο λόγος που χρησιμοποιήσαμε το try-except είναι διότι δεν είχαν όλα τα διαγνωστικά κέντρα ιστοσελίδα αναγραμμένη και έπρεπε να τα προσπερνάμε για να μην δημιουργηθεί πρόβλημα στην αντιστοίχιση των δεδομένων.

Για να μπορέσουμε όμως να φτάσουμε έως το σημείο της ανάκτησης των δεδομένων έπρεπε να δώσουμε τις κατάλληλες εντολές στον webdriver. Αρχικά εκτελώντας την συνάρτηση στο αρχείο scraping.py ανοίγουμε τον webdriver ο οποίος στην δική μας περίπτωση αντιστοιχεί στο Chrome και στην πορεία τοποθετήσαμε το κείμενο που θέλαμε να έχουμε ως κριτήριο αναζήτησης στην μπάρα αναζήτησης και τέλος κάναμε κλίκ στην αναζήτηση έχοντας βρει τα στοιχεία τους από τον πηγαίο κώδικα.

Αργότερα, μόλις φορτώσει η πρώτη σελίδα γίνεται εμφάνιση ενός παραθύρου των cookies και ένα διαφημιστικό το οποίο πρέπει να αφαιρεθεί για να μην δημιουργηθούν προβλήματα στην όλη διαδικασία. Επομένως, βρήκαμε μέσω του πηγαίου κώδικα τα δύο αυτά πλαίσια και εντοπίζοντας στην μία περίπτωση το OK για τα cookies και το X για το διαφημιστικό κάναμε το κλίκ για την εξαφάνισή τους. Η διαδικασία αυτή χρειάστηκε να γίνει μόνο για την πρώτη σελίδα, κατά το πέρασμα στις επόμενες σελίδες δεν είναι απαραίτητο γιατί έχουν ήδη απαλειφθεί.

Στην συνέχεια, για να μπορέσουμε να λάβουμε όλα τα δεδομένα πρέπει να συνεχίσουμε την διαδικασία για όλες τις σελίδες. Αυτό το πέτυχαμε εντοπίζοντας μέσω του πηγαίου κώδικα της ιστοσελίδας το Next κουμπί, ώστε να το πατάμε και να προχωράμε στην επόμενη σελίδα εφόσον

έχουμε τελειώσει την τρέχουσα σελίδα. Έτσι μόλις ολοκληρωθεί η διαδικασία του web scraping κλείνει ο φυλλομετρητής και τα στοιχεία έχουν αποθηκευτεί στο αρχείο που τα τοποθετήσαμε(scraping.csv).

Αμέσως μετά γίνεται κάνουμε κλήση της created.py όπου προσθέτει όλα τα δεδομένα μας στην βάση και πλέον είμαστε ικανοί να εκτελέσουμε τις επιλογές μέσα από την βάση μας.

Τελικά έχοντας ολοκληρωθεί η διαδικασία του web scraping, ο τρόπος που παίρνουμε τα δεδομένα μας ώστε να τα ομαδοποιήσουμε για την περίπτωση της χαρτογράφησης των ιατρικών διαγνωστικών κέντρων είναι ο εξής:

```
localities = conn.execute('SELECT DISTINCT S_locality FROM ScrapData ORDER BY S_locality').fetchall()
regions = conn.execute('SELECT DISTINCT S_region FROM ScrapData ORDER BY S_region').fetchall()
```

Εικόνα 26 Ομαδοποίηση Δεδομένων για πλαίσια επιλογών 2

Οι δύο αυτές επιλογές γίνονται εξίσου και πάλι επιλέγοντας τα δεδομένα μας χωρίς τις διπλοτιμές και ταξινομημένα ως προς τα ίδια τα δεδομένα. Το localities περιέχει τους δήμους και το regions τις περιοχές των διαγνωστικών που θα τοποθετηθούν στα πλαίσια επιλογών. Η διαδικασία αυτή αφορά την ομαδοποίηση των δεδομένων μας για την GET μέθοδο. Όσον αφορά την POST μέθοδο, όταν λαμβάνουμε τα επιλεγμένα δεδομένα από τα πλαίσια επιλογών σε json μορφή, αρχικά τα μετατρέπουμε σε δεδομένα που μπορούμε να τα χρησιμοποιήσουμε ώστε να κάνουμε τα SELECT που θέλουμε στην βάση και στην συνέχεια για την πρώτη περίπτωση των νοσοκομείων εκτελούμε τα παρακάτω Select

```
if(region == "Select Region" and category == "Select Category"):
    filteredAxis = cur.execute('SELECT H_name,H_xaxis,H_yaxis,H_address,H_phone FROM Data WHERE H_beds = ?',(bed,)).fetchall()
elif(region == "Select Region" and bed == "Select Number of Beds"):
    filteredAxis = cur.execute('SELECT H_name,H_xaxis,H_yaxis,H_address,H_phone FROM Data WHERE H_category = ?',(category,)).fetchall()
elif(category == "Select Category" and bed == "Select Number of Beds"):
    filteredAxis = cur.execute('SELECT H_name,H_xaxis,H_yaxis,H_address,H_phone FROM Data WHERE H_region = ?',(region,)).fetchall()
elif(category == "Select Category"):
    filteredAxis = cur.execute('SELECT H_name,H_xaxis,H_yaxis,H_address,H_phone FROM Data WHERE H_region = ? AND H_beds = ?',(region,bed)).fetchall()
elif(region == "Select Region"):
    filteredAxis = cur.execute('SELECT H_name,H_xaxis,H_yaxis,H_address,H_phone FROM Data WHERE H_category = ? AND H_beds = ?',(category,bed)).fetchall()
elif(bed == "Select Number of Beds"):
    filteredAxis = cur.execute('SELECT H_name,H_xaxis,H_yaxis,H_address,H_phone FROM Data WHERE H_category = ? AND H_region = ?',(category,region)).fetchall()
else:
    filteredAxis = cur.execute('SELECT H_name,H_xaxis,H_yaxis,H_address,H_phone FROM Data WHERE H_category = ? AND H_region = ? AND H_beds = ?',(category,region,bed)).fetchall()
```

Εικόνα 27 Σενάρια για πλαίσια επιλογών 1

Παίρνοντας όλες τις περιπτώσεις επιλογών με σκοπό την σωστή ανταπόκριση από την βάση μας. Στα select αυτά χρησιμοποιούμε τα δεδομένα που λάβαμε μέσω ajax και τα τοποθετούμε ως

κριτήρια για την εύρεση των επιθυμητών δεδομένων. Ο λόγος που πήραμε τόσες περιπτώσεις είναι διότι μπορεί να μην επιλεγθούν όλες οι επιλογές, επομένως θα πρέπει να γίνουν τα select με όσα διαθέσιμα κριτήρια διαθέτουμε σε κάθε περίπτωση.

Για την άσκηση της κριτικής αφότου λάβουμε τα δεδομένα από τα ο πλαίσιο επιλογών τα οποίο είναι οι περιοχές εκτελούμε

```
ratingAxis = cur.execute('SELECT H_name,H_xaxis,H_yaxis,H_address,H_phone FROM Data WHERE H_region = ?',(region,)).fetchall()
```

Εικόνα 28 Ομαδοποίηση Δεδομένων για πλαίσια επιλογών 3

για να πάρουμε από την βάση τα δεδομένα σύμφωνα με αυτήν την περιοχή. Έπειτα εκτελούμε εκ νέου ομαδοποίηση των δεδομένων, μόλις λάβουμε τα δεδομένα μετά την άσκηση της κριτικής,

```
avgQuery = cur.execute("SELECT AVG(R_rating) FROM RatingData WHERE R_H_yaxis = ? AND R_H_xaxis = ?",(y,x)).fetchall()
countRating = cur.execute("SELECT COUNT(R_rating) FROM RatingData WHERE R_H_yaxis = ? AND R_H_xaxis = ?",(y,x)).fetchall()
commentsQuery = cur.execute("SELECT R_comments FROM RatingData WHERE R_H_yaxis = ? AND R_H_xaxis = ? ORDER BY R_id DESC LIMIT 1",(y,x)).fetchall()
```

Εικόνα 29 Εκ νέου ομαδοποίηση δεδομένων για κριτικές

όπου σύμφωνα με τις συντεταγμένες των νοσοκομείων αντιλαμβανόμαστε για ποιο νοσοκομείο θα κάνουμε την κριτική και έτσι υπολογίζουμε τον μέσο όρο αστεριών του τρέχον νοσοκομείου αλλά και κρατάμε το πιο πρόσφατο σχόλιο που έγινε.

Τέλος για την ομαδοποίηση των δεδομένων για τα ιατρικά διαγνωστικά κέντρα εφόσον έχει ανανεωθεί η βάση μας μετά την διαδικασία του web scraping εκτελούμε

```
if(region == "Select Region"):
    scrapAxis = cur.execute("SELECT S_name,S_address,S_activities,S_yaxis,S_xaxis,S_phone,S_website FROM ScrapData WHERE S_locality = ?",(locality,)).fetchall()
else(locality == "Select locality"):
    scrapAxis = cur.execute("SELECT S_name,S_address,S_activities,S_yaxis,S_xaxis,S_phone,S_website FROM ScrapData WHERE S_region = ?",(region,)).fetchall()
else:
    scrapAxis = cur.execute("SELECT S_name,S_address,S_activities,S_yaxis,S_xaxis,S_phone,S_website FROM ScrapData WHERE S_region = ? AND S_locality = ?",(region,locality)).fetchall()
```

Εικόνα 30 Σενάρια για πλαίσια επιλογών 2

και ομαδοποιούμε τα δεδομένα μας σύμφωνα με την περιοχή και τον δήμο που επιλέχθηκαν.

• Ανάκτηση Συντεταγμένων

Η ανάκτηση των συντεταγμένων για τα νοσοκομεία ήταν απλή διαδικασία εφόσον τα είχαμε στο csv αρχείο επομένως τα λαμβάνουμε κάνοντας απλά ερώτηση στην βάση μας. Όσον αφορά τις συντεταγμένες των ιατρικών διαγνωστικών κέντρων η διαδικασία ήταν διαφορετική καθώς

ανακτούμε τις συντεταγμένες μέσω web scraping και επειδή δεν είναι φανερά διατυπωμένες στην ιστοσελίδα, έπρεπε να ψάξουμε λίγο διαφορετικά στον πηγαίο κώδικα και να εκτελέσουμε κάποιες διαδικασίες ώστε να καταφέρουμε να έχουμε στην διάθεση μας τις συντεταγμένες. Πιο συγκεκριμένα το μόνο σημείο που υπήρχαν οι συντεταγμένες ήταν μέσα σε ένα href το οποίο βρισκόταν μέσα σε onclick function της JavaScript. Επομένως βρίσκοντας αρχικά που βρίσκεται αυτό το onclick function μέσω του ονόματος κλάσης παίρνουμε ολόκληρο το κείμενο μέσα στο onclick function και εκτελούμε αυτήν την επεξεργασία ώστε να καταλήξουμε στις δύο συντεταγμένες σε double μορφή.

```
coords = driver.find_elements_by_class_name('address')

for coord in coords:
    try:
        coordinates = coord.find_element_by_xpath('..//span/a').get_attribute('onclick')
        split = coordinates.split()

        ##ycoordinate
        y1 = split[1]
        x1 = split[2]
        x2 = x1.split(',')
        x3 = x2[0]
        finalx = x3.replace("'", '')

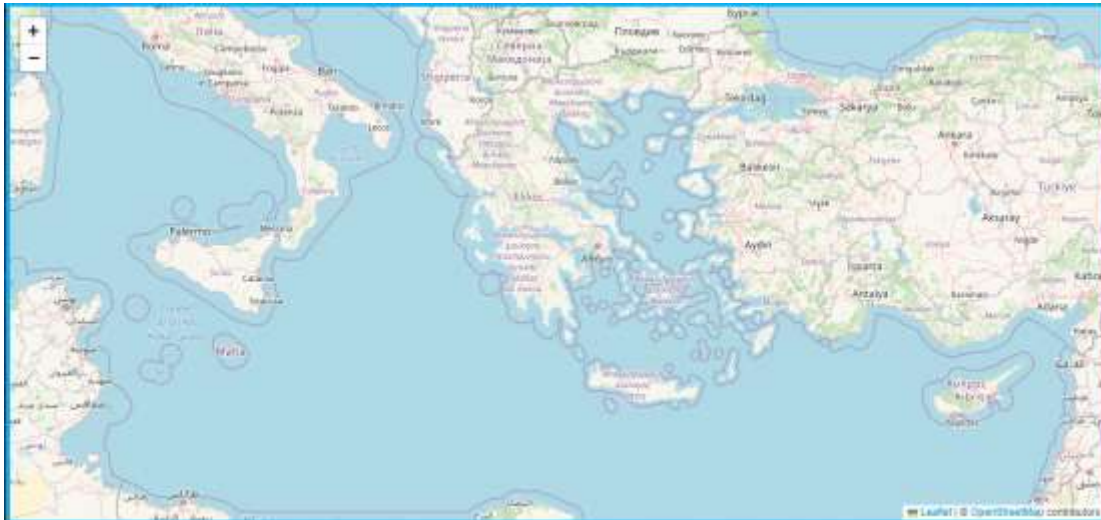
        ##xcoordinate
        y2=y1.split('(')
        y3 = y2[1]
        y4 = y3.split(',')
        y5 = y4[0]
        finaly = y5.replace("'", '')
        #cords.append('')
        cords.append([finaly,finalx])
    except Exception as ex:
        cords.append(['', ''])
```

Εικόνα 31 Υλοποίηση web scraping για συντεταγμένες

Σπάσαμε λοιπόν το κείμενο αυτό με split μερικές φορές και καταλήξαμε στις δύο συντεταγμένες όπου τις προσθέσαμε σε μια τελική λίστα. Χρησιμοποιήσαμε και πάλι try-except διότι δεν είχαν όλα τα διαγνωστικά πληροφορίες για την τοποθεσία τους, συνεπώς αυτά που δεν είχαν τα προσπερνούμε για συμπληρώνουμε στην θέση τους το κενό για να μην δημιουργηθεί πρόβλημα αντιστοίχισης ανάμεσα στα δεδομένα μας.

- **Χαρτογράφηση Δεδομένων**

Την χαρτογράφηση των δεδομένων μας την πέτυχαμε χρησιμοποιώντας κώδικα JavaScript ενσωματωμένο μέσα στα html αρχεία μέσα σε script tabs. Σε όλα τα αρχεία όπου εκτελούμε είτε την λειτουργία της χαρτογράφησης είτε της άσκησης κριτικής έχουμε προσθέσει έναν χάρτη μέσω Leaflet ο οποίος αποτελεί τον βασικό μας χάρτη για όλες τις λειτουργίες.



Εικόνα 32 Βασικός χάρτης της ιστοσελίδας μας

Έχουμε βάλει ως σημείο επίκεντρο στον χάρτη το Άργος και με τέτοιο ζουμ ώστε να υπάρχει η καλύτερη δυνατή όψη του. Στην περίπτωση όπου κατά την επιλογή των κριτηρίων δεν επιλεγεί κανένα από τα πλαίσια τότε θα μας εμφανιστεί ένα μήνυμα μορφής alert(ειδοποίηση) που μας ενημερώνει ότι δεν υπάρχει κάποιο νοσοκομείο/διαγνωστικό κέντρο με τα αντίστοιχα δεδομένα και γίνεται επαναφόρτιση της σελίδας.

- **Χαρτογράφηση Νοσοκομείων**

Όσον αφορά την χαρτογράφηση των νοσοκομείων, από την στιγμή που λάβουμε τα τελικώς επεξεργασμένα και ομαδοποιημένα δεδομένα μας κατά την επιτυχία μέσω Ajax, περάσαμε όλα τα δεδομένα από το λεξικό που μεταφέραμε σε μια λίστα ώστε να είναι πιο εύκολη η χρήση τους. Στην συνέχεια διατρέχοντας αυτήν την λίστα χωρίσαμε τα δεδομένα μας σε x, y συντεταγμένες και στις υπόλοιπες πληροφορίες των νοσοκομείων. Τις συντεταγμένες τις χρησιμοποιήσαμε για να περάσουμε στον χάρτη βάσει αυτών τα markers που θα δηλώνουν την τοποθεσία τους και ταυτόχρονα τις υπόλοιπες πληροφορίες τις προσθέσαμε σε popups(παραθυράκια πληροφοριών) τα οποία είναι συνδεδεμένα με τα markers και κατά το

πάτημα του κλικ πάνω στα markers εμφανίζεται αυτό το παράθυρο το οποίο περιέχει πληροφορίες για το όνομα, τη διεύθυνση και τον αριθμό τηλεφώνου του αντίστοιχου νοσοκομείου. Την παραπάνω διαδικασία εκτελέσαμε ως εξής:

```
for(i=1;i<coordinates.length;i++){  
    var marker = L.marker([coordinates[i][1],coordinates[i][2]]);  
    marker.addTo(map);  
    var popup = marker.bindPopup(String(coordinates[i][8]) + "<br> Address: " + String(coordinates[i][3]) + "<br>Phone: " + String(coordinates[i][4]));  
    popup.addTo(map);  
}
```

Εικόνα 33 Δημιουργία των markers για νοσοκομεία

όπου `coordinates[i][1]` και `coordinates[i][2]` είναι οι συντεταγμένες ενώ τα υπόλοιπα είναι οι επιπλέον πληροφορίες. Με αυτόν τον τρόπο δημιουργήσαμε όσα markers χρειαζόμαστε για την κάθε περίπτωση και στην συνέχεια τα προσθέταμε στον χάρτη. Ακολουθούμενη είναι η διαδικασία της δημιουργίας των popups. Το κίτρινο marker είναι μέθοδος της Leaflet ενώ το μπλε είναι το αντικείμενο μας το οποίο και χρησιμοποιούμε για τα popups ώστε να γίνεται η αντιστοίχιση του marker και των πληροφοριών. Την ίδια διαδικασία έχουμε ακολουθήσει και στα αρχικά στάδια της σελίδας της άσκησης κριτικής προτού φτάσουμε στο βασικό κομμάτι της κριτικής. Για την εύρεση της κοντινότερης απόστασης και της δρομολόγησης θα αναφερθούμε παρακάτω.

- **Χαρτογράφηση Διαγνωστικών Κέντρων**

Στην χαρτογράφηση των διαγνωστικών κέντρων η διαδικασία της χαρτογράφησης είναι παρόμοια με αυτή των νοσοκομείων. Έχοντας ολοκληρωθεί και εδώ η επιτυχία μετά την Ajax προσθέτουμε τα δεδομένα μας από το λεξικό σε μια λίστα. Στην προκειμένη περίπτωση και πάλι το πρώτο μέρος του χωρισμού είναι αυτό για τα markers όπου χρησιμοποιούμε τις συντεταγμένες μόνο για την απεικόνιση στην χάρτη και στο δεύτερο σκέλος τα popups τα οποία περιέχουν πληροφορίες για το όνομα, τη διεύθυνση, τις δραστηριότητες, αριθμός επικοινωνίας και ιστοσελίδα των ιατρικών διαγνωστικών κέντρων. Η διαδικασία είναι η ακολουθούμενη:

```
for(i=1;i<datas.length;i++){  
    var marker = L.marker([datas[i][4],datas[i][3]]);  
    marker.addTo(map);  
    var popup = marker.bindPopup(String(datas[i][0]) + "<br>" + String(datas[i][1]) + "<br>Activities: " + String(datas[i][2])  
    + "<br>" + String(datas[i][5]) + "<br>Website: " + String(datas[i][6]));  
    popup.addTo(map);  
}
```

Εικόνα 34 Δημιουργία των markers για διαγνωστικά κέντρα

• Εύρεση και απεικόνιση κοντινότερης διαδρομής

Για την απεικόνιση της δρομολόγησης ανάμεσα στα δύο σημεία χρησιμοποιήσαμε το routing της Leaflet καθώς και μερικές μαθηματικές πράξεις. Πιο συγκεκριμένα δημιουργήσαμε δύο custom markers στους οποίους έχουμε δώσει δύο διαφορετικά εικονίδια ώστε να είναι πιο κατανοητή η χαρτογράφηση. Το ένα εικονίδιο αντιστοιχεί στην δική μας τοποθεσία και το άλλο στην τοποθεσία προορισμού(το κοντινότερο νοσοκομείο/ιατρικό διαγνωστικό κέντρο).Την εκτίμηση του κοντινότερου σημείου την υπολογίσαμε με τον τύπο υπολογισμού απόστασης μεταξύ δύο σημείων. Το ένα σημείο είναι οι συντεταγμένες του δικού μας σημείου και το άλλο σημείο οι συντεταγμένες τους σημείου προορισμού. Για τον σωστό υπολογισμό χρησιμοποιήσαμε το πρόθεμα Math της JavaScript και τα sqrt , pow για τον υπολογισμό της ρίζας και του τετραγώνου αντίστοιχα.

```
var k = Math.sqrt((Math.pow(datas[i][4]-lat,2) + Math.pow(datas[i][3]-long,2)) );
```

Εικόνα 35 Υπολογισμός Απόστασης Σημείων

Υπολογίζοντας λοιπόν την απόσταση από την τοποθεσία μας και όλων των σημείων θέτοντας την πρώτη απόσταση ως την μικρότερη,καθώς συνεχίζουμε την for loop αναλόγως από τις ανισοτικές σχέσεις μεταξύ των αποστάσεων καταλήγουμε στην μικρότερη απόσταση η οποία είναι και το ζητούμενο.Ταυτόχρονα για το σημείο προορισμό κρατάμε και τις πληροφορίες του popup του ώστε να τα αντιστοιχίσουμε με το ειδικό marker.Τέλος την απεικόνιση της δρομολόγησης την επιτυγχάνουμε με τον παρακάτω τρόπο:

```
var routing = L.Routing.control({
  waypoints: [
    L.latLng(arrayForLine[0][0],arrayForLine[0][1]),
    L.latLng(arrayForLine[1][0],arrayForLine[1][1])
  ],
  createMarker: function(i,wp,nwps){
    if(i===0){
      return L.marker(wp.latLng, {icon:myIcon}).bindPopup("My Location");
    }
    else{
      return L.marker(wp.latLng,{icon:greenIcon}).bindPopup(destinationPopup);
    }
  }
}).addTo(map);
```

Εικόνα 36 Υλοποίηση Δρομολόγησης

όπου arrayForLine είναι μια λίστα που στην πρώτη της υπολίστα έχουμε προσθέσει τις συντεταγμένες του ενός σημείου και στην δεύτερη τις συντεταγμένες του άλλου. Με την μέθοδο

createMarker δηλώνουμε τα δύο ειδικά markers μας όπου στην θέση icon βάζουμε τα custom markers και στο bindpopup τα παραθυράκια με τις πληροφορίες. Στην συνάρτηση αυτή έχουμε πάρει και ένα if-else statement στο οποίο δηλώνουμε ότι εάν είμαστε στην αρχή βρισκόμαστε στο σημείο αφετηρία αλλιώς στο σημείο προορισμού. Προσθέτουμε και το routing στον χάρτη και απεικονίζεται η δρομολόγηση. Ταυτόχρονα μέσω της βιβλιοθήκης που χρησιμοποιούμε στην πάνω δεξιά γωνία του χάρτη υπάρχει και ένα μικρό παράθυρο στο οποίο βρίσκονται κάποιες αναλυτικές πληροφορίες για την χιλιομετρική διαδρομή από το σημείο αφετηρία έως το σημείο προορισμού.

- **Άσκηση Κριτικής Νοσοκομείων**






Στην άσκηση της κριτικής μόλις επιλέξουμε την περιοχή που θέλουμε και κάνουμε κλικ στο κουμπί Search εμφανίζονται τα αντίστοιχα νοσοκομεία και στην συνέχεια για να κάνουμε κριτική πρέπει να κάνουμε κλικ στο marker που θέλουμε. Μόλις επιλέξουμε το νοσοκομείο εμφανίζεται ένα παράθυρο το οποίο είναι σε wrapper-container tag όπου έχουμε την δυνατότητα να επιλέξουμε τον αριθμό αστεριών που θέλουμε να δώσουμε στο νοσοκομείο από ένα έως πέντε και στην συνέχεια να αφήσουμε και κάποιο σχόλιο. Κατά την ολοκλήρωση αυτού του σταδίου πατώντας save εξαφανίζεται αυτό το παράθυρο και εμφανίζεται το επόμενο στο οποίο αναγράφεται ο μέσος όρος αστεριών του συγκεκριμένου νοσοκομείου καθώς και το τελευταίο σχόλιο που καταχωρήθηκε. Τέλος πατώντας το close κουμπί επιστρέφουμε ξανά στην σελίδα αυτή για νέα κριτική.

- **Αποτελέσματα Άσκησης Κριτικής**

Το πρώτο παράθυρο που μας εμφανίζεται για να εκτελέσουμε την κριτική μας:

Rate this Hospital

What's your experience about this hospital?



COMMENTS

Καλή εξυπηρέτηση!

Save your rating

Εικόνα 37 Πρώτο στάδιο άσκησης κριτικής

όπου για παράδειγμα έχουμε επιλέξει αριθμό αστεριών 4 και καταχωρήσαμε το σχόλιο Καλή εξυπηρέτηση!. Κάνοντας κλικ στο Save your rating εξαφανίζεται το συγκεκριμένο παράθυρο και εμφανίζεται το παρακάτω:

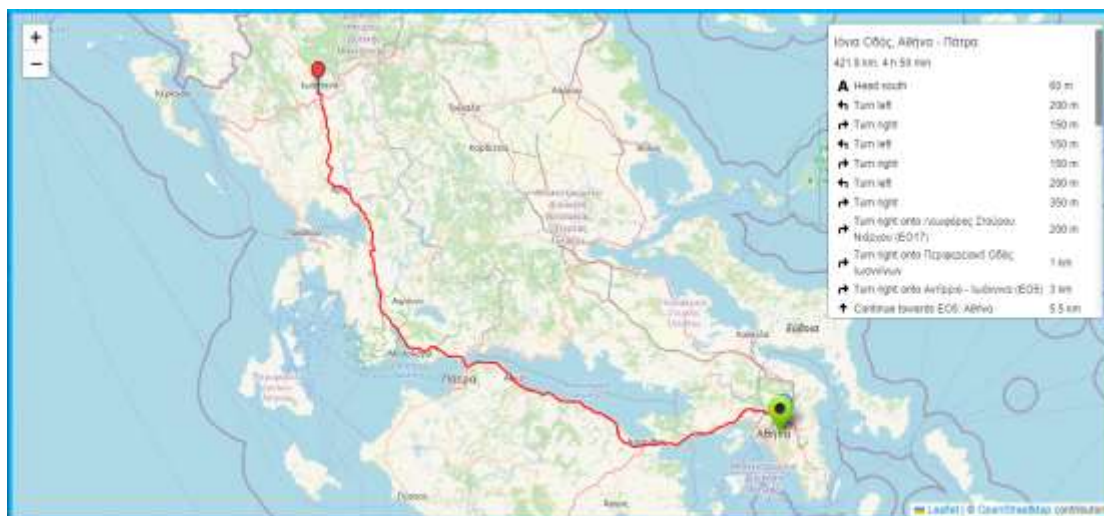


Εικόνα 38 Δεύτερο στάδιο άσκησης κριτικής

το οποίο μας αναγράφει ότι έχουν γίνει συνολικά 2 κριτικές για το συγκεκριμένο νοσοκομείο, ο μέσος όρος αστεριών είναι 3.5 και το τελευταίο σχόλιο που καταχωρήθηκε είναι αυτό που καταχωρήσαμε προηγουμένως.

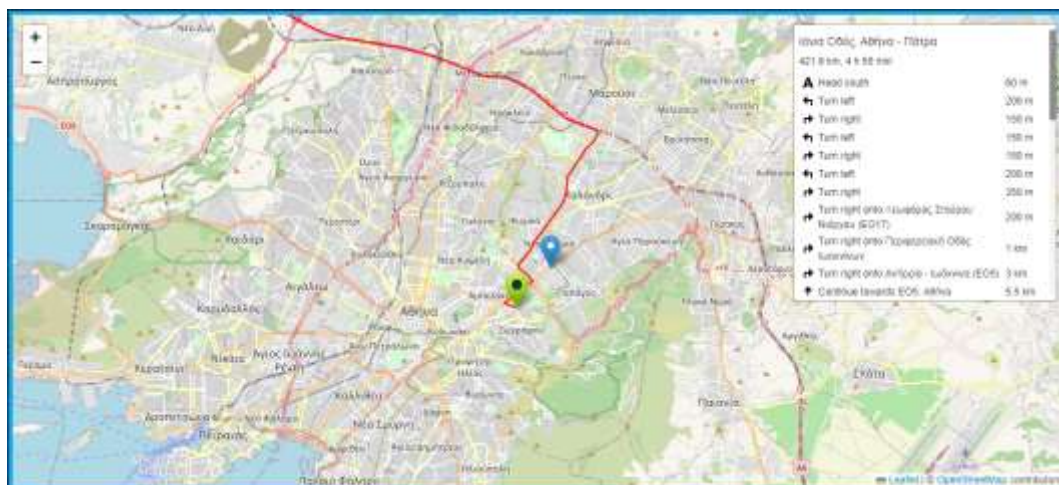
- **Αποτελέσματα Χαρτογράφησης Νοσοκομείων**

Επιλέγοντας για παράδειγμα περιοχή Αθήνα, Κατηγορία Γενικό Νοσοκομείο και Αριθμό Κλινών στο εύρος 600 με 700 το αποτέλεσμα είναι:



Εικόνα 39 Χαρτογράφηση Νοσοκομείων 1

όπου το κόκκινο σημαδάκι είναι η δική μας τοποθεσία, τα μπλε είναι τα υπόλοιπα νοσοκομεία που προκύπτουν από αυτά τα κριτήρια και το πράσινο είναι το νοσοκομείο προορισμός δηλαδή το κοντινότερο. Μια κοντινότερη εικόνα των νοσοκομείων από το προηγούμενο παράδειγμα είναι:



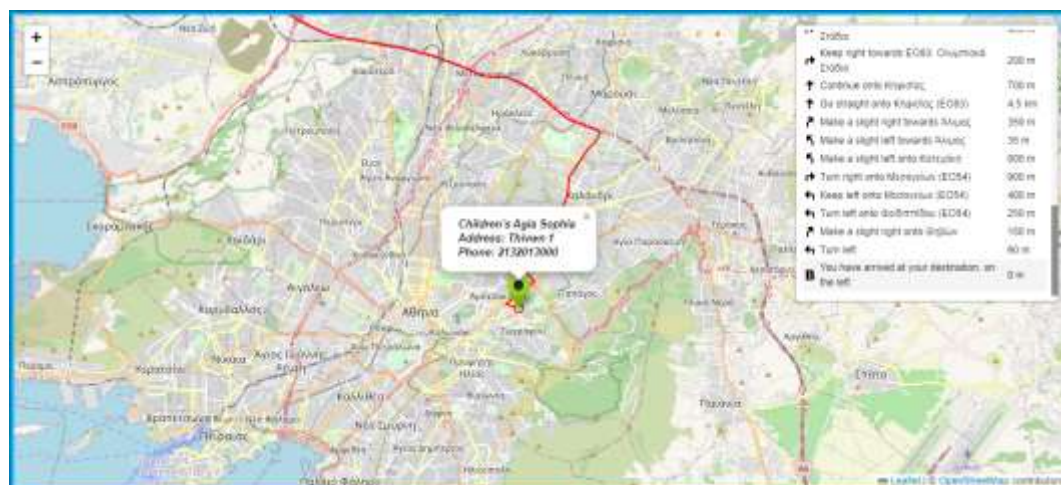
Εικόνα 40 Χαρτογράφηση Νοσοκομείων 2

Επιπλέον μια κοντινότερη εικόνα των οδηγίων δρομολόγησης μέχρι την άφιξη στον προορισμό:

Στάδιο	200 m
Keep right towards ΕΟ83: Ολυμπιακό Στάδιο	200 m
Continue onto Κηφισίας	700 m
Go straight onto Κηφισίας (ΕΟ83)	4.5 km
Make a slight right towards Άλιμος	350 m
Make a slight left towards Άλιμος	35 m
Make a slight left onto Κατεχάκη	800 m
Turn right onto Μεσογείων (ΕΟ54)	900 m
Keep left onto Μεσογείων (ΕΟ54)	400 m
Turn left onto Φειδιππίδου (ΕΟ54)	250 m
Make a slight right onto Θηβών	150 m
Turn left	90 m
You have arrived at your destination, on the left	0 m

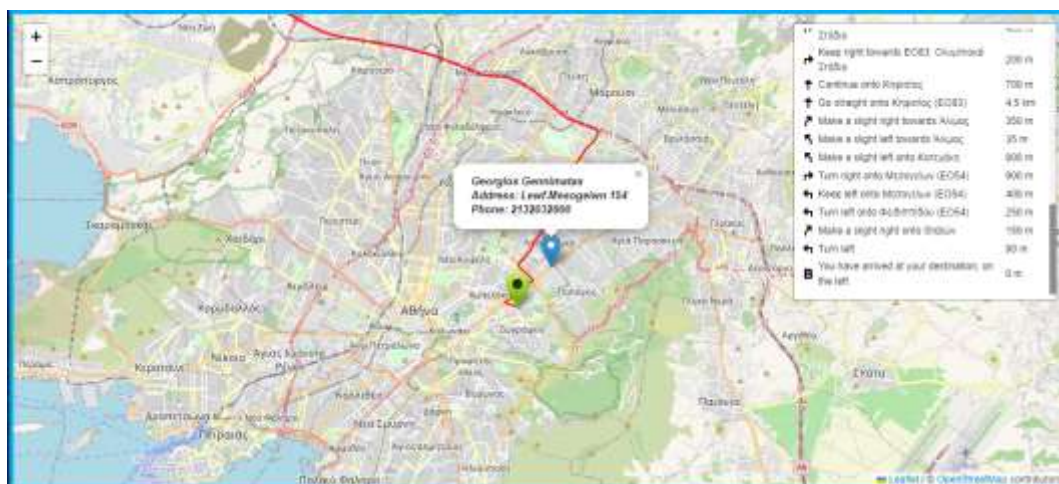
Εικόνα 41 Χαρτογράφηση Νοσοκομείων 3

και τέλος πατώντας κλικ στο marker προορισμό εμφανίζεται το παρακάτω παράθυρο



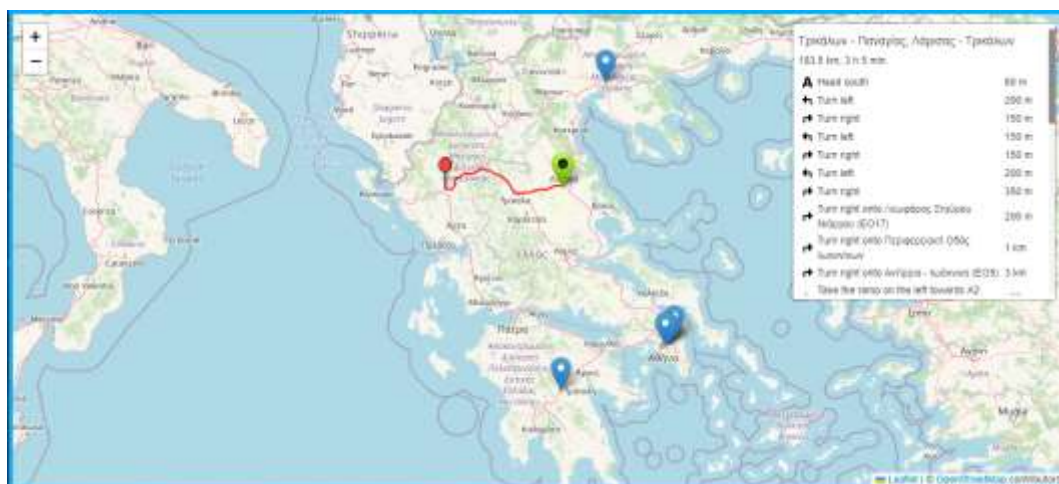
Εικόνα 42 Χαρτογράφηση Νοσοκομείων 4

Ενώ πατώντας κλικ στην άλλη επιλογή νοσοκομείο:



Εικόνα 43 Χαρτογράφηση Νοσοκομείων 5

Φυσικά σε περίπτωση που δεν απαιτήσουμε τόσα κριτήρια και βάλουμε για παράδειγμα μόνο κατηγορία νοσοκομείου τα markers θα είναι περισσότερα. Για παράδειγμα με κατηγορία Στρατιωτικό Νοσοκομείο χωρίς περιοχή και αριθμό κλινών:

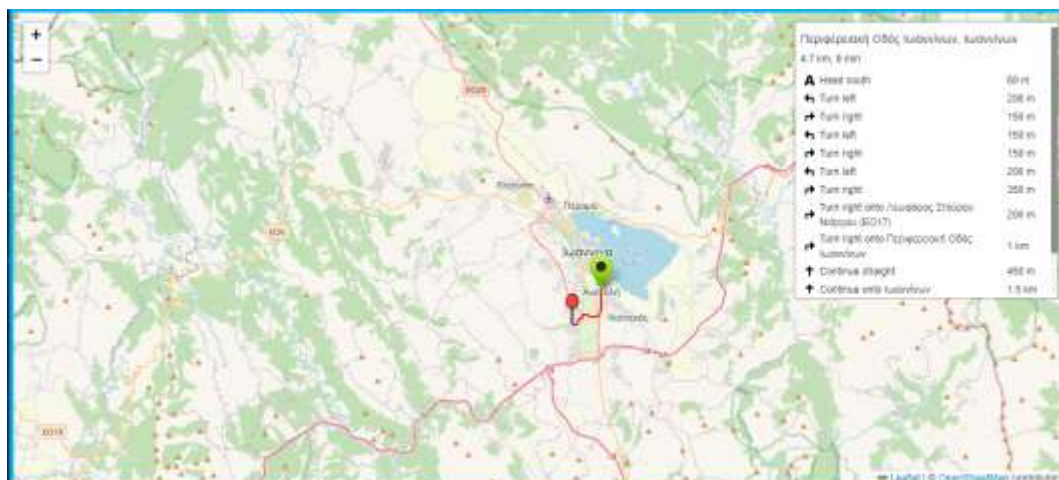


Εικόνα 44 Χαρτογράφηση Νοσοκομείων 6

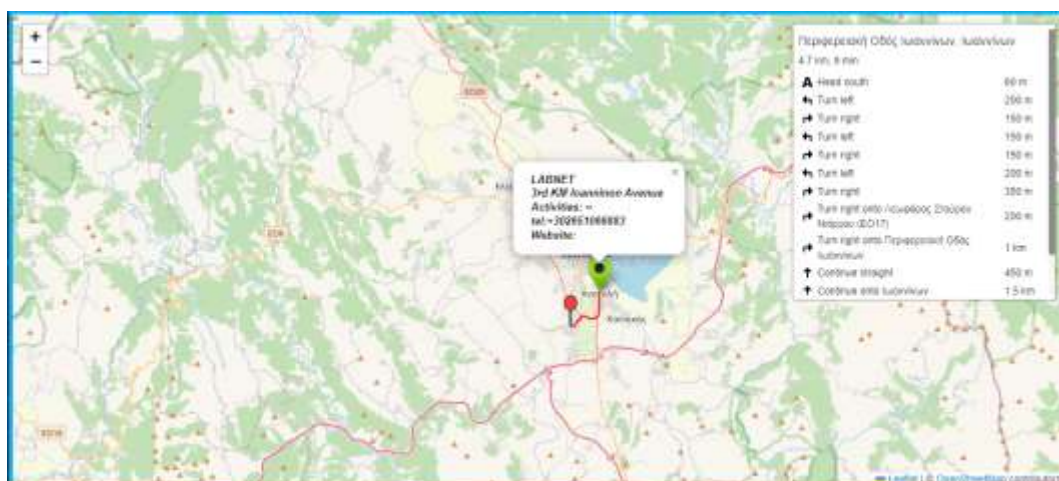
• Αποτελέσματα Χαρτογράφησης Διαγνωστικών Κέντρων

Τέλος για την χαρτογράφηση των ιατρικών διαγνωστικών κέντρων επιλέγοντας για παράδειγμα περιοχή Ιωάννινα και δήμο Ανατολή:

Εικόνα Χαρτογράφηση Διαγνωστικών Κέντρων 1

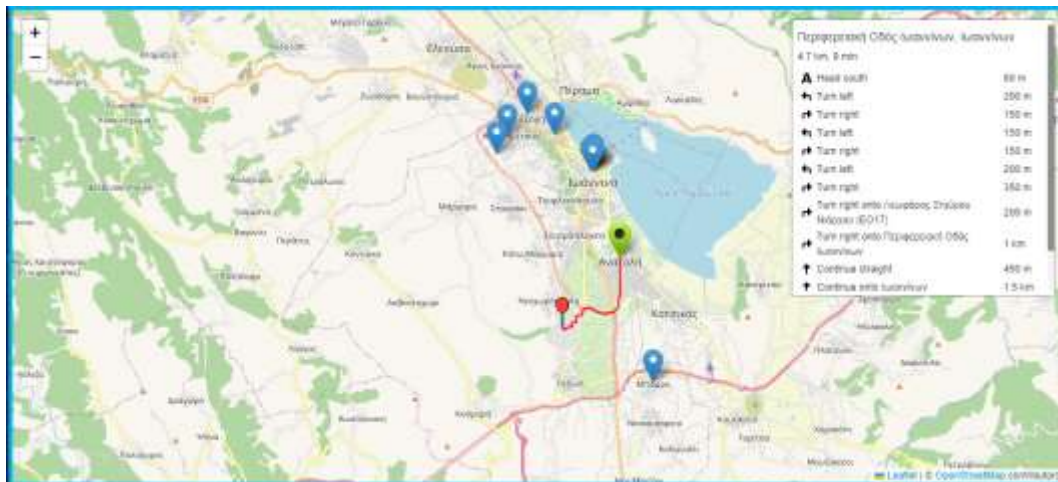


όπου κάνοντας κλικ πάνω στο διαγνωστικό κέντρο προορισμό εμφανίζεται το παράθυρο πληροφοριών:



Εικόνα Χαρτογράφηση Διαγνωστικών Κέντρων 2

Αντίστοιχα όπως και για την χαρτογράφηση νοσοκομείων εάν επιλέξουμε μόνο περιοχή και όχι δήμο για παράδειγμα και πάλι Ιωάννινα θα έχουμε το εξής αποτέλεσμα:



Εικόνα Χαρτογράφηση Διαγνωστικών Κέντρων 3

Βιβλιογραφία-Παραπομπές

1. , International Health Conference, New York
2. , Βάσεις της Χαρτογραφίας, Θεσσαλονίκη
- 3.
- 4., Γ. Μαθηματική Χαρτογραφία, Αθήνα, Εργαστήριο Τοπογραφίας, Εθνικό Μετσόβιο Πολυτεχνείο

5. (Νάκος, 2006), Β. Αναλυτική Χαρτογραφία, Αθήνα, Σχολή Τοπογράφων Μηχανικών, Τομέας Τοπογραφίας, Εθνικό Μετσόβιο Πολυτεχνείο
6. Ενιαίος ψηφιακός χάρτης, Ανακτήθηκε στις 12 Ιουλίου 2021
<https://www.gpsworld.com/a-milestone-in-digital-mapping>
7. (Winslow, Charles-Edward Amory ,1920), «The Untilled Field of Public Health». Modern Medicine 2: 183–191.
8. (International Health Conference, 1946), Frequently asked questions from the "Preamble to the Constitution of the World Health Organization" International Health Conference, 1946
9. Joint Task Group on Public Health Human Resources· Advisory Committee on Health Delivery & Human Resources· Advisory Committee on Population Health & Health Security (2005). Building the public health workforce for the 21st century. Ottawa: Public Health Agency of Canada. OCLC 144167975.
10. «Database Definition». The Linux Information Project. Ανακτήθηκε στις 16 Μαΐου 2013.
11. "General Python FAQ — Python 3.9.2 documentation". *docs.python.org*. Archived from the original on 24 October 2012. Retrieved 28 March 2021.
12. «ECMAScript Language Overview» (PDF). 23 Οκτωβρίου 2007. σελ. 4. Αρχειοθετήθηκε από το πρωτότυπο (PDF) στις 26 Μαρτίου 2009. Ανακτήθηκε στις 3 Μαΐου 2009.
<https://web.archive.org/web/20090326234420/http://www.ecmascript.org/es4/spec/overview.pdf>

13. «Learn to style HTML using CSS - Learn web development | MDN». *developer.mozilla.org* .Ανακτήθηκε στις 3 Νοεμβρίου 2021.
14. "HTML 4.0 Specification — W3C Recommendation — Conformance: requirements and recommendations". w3. World Wide Web Consortium. December 18, 1997. Archived from the original on July 5, 2015. Retrieved July 6, 2015. <https://www.w3.org/TR/REC-html40-971218/conform.html#deprecated>
15. <https://www.techtarget.com/searchdatamanagement/definition/SQL>
16. <https://leafletjs.com>
17. <https://www.zyte.com/learn/what-is-web-scraping/>
18. <https://hackersandslackers.com/flask-routes/>
19. (Codesido,2009). "What is front-end development?". Theguardian.com. Retrieved 17 January 2019.
20. (Thapliyal, 2016), "Difference Between Frontend and Backend MVC – JoomlaTuts". joomlatuts.net. Archived from the original on 2016-12-30. Retrieved 2016-12-30.
21. <https://www.selenium.dev>