



Πανεπιστήμιο Θεσσαλίας  
Πολυτεχνική Σχολή  
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

## Εφαρμογή προχωρημένης αναζήτησης ακινήτων

Διπλωματική Εργασία

**ΒΑΣΙΛΕΙΟΥ Δ. ΜΙΧΑΗΛΙΔΗ**

**Επιβλέπων**

Μιχαήλ Βασιλακόπουλος  
Αναπληρωτής Καθηγητής

Βόλος, Μάιος 2020





Πανεπιστήμιο Θεσσαλίας

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

## Εφαρμογή προχωρημένης αναζήτησης ακινήτων

### Διπλωματική Εργασία

**ΒΑΣΙΛΕΙΟΥ Δ. ΜΙΧΑΗΛΙΔΗ**

Επιτροπή επίβλεψης

Επιβλέπων

Μιχαήλ Βασιλακόπουλος  
Αναπληρωτής Καθηγητής

Συνεπιβλέπουσα

Ελένη Τουσίδου  
Μέλος Ε.ΔΙ.Π

Συνεπιβλέπουσα

Χαρίκλεια Τσαλαπάτα  
Μέλος Ε.ΔΙ.Π

Βόλος, Μάιος 2020



Πανεπιστήμιο Θεσσαλίας  
Πολυτεχνική Σχολή  
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή / της φοιτήτριας που την εκπόνησε. Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ' ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.

Ο/Η συγγραφέας αυτής της εργασίας βεβαιώνει ότι κάθε βοήθεια την οποία είχε για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης βεβαιώνει ότι έχει αναφέρει τις όποιες πηγές από τις οποίες έκανε χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται επακριβώς, είτε παραφρασμένες.



University of Thessaly  
Faculty of Engineering  
Department of Electrical & Computer Engineering

# **Advanced searching application for real estate properties**

## Diploma Thesis

**VASILEIOS D. MICHAILIDIS**

### **Supervisor**

Michail Vasilakopoulos  
Associate Professor

Volos, May 2020



# Περίληψη

Βασικός στόχος της παρούσας διπλωματικής εργασίας είναι η έρευνα, σχεδίαση και ανάπτυξη καινοτόμων φίλτρων αναζήτησης ακινήτων μέσω διαδικτυακής εφαρμογής σε περιβάλλον παγκόσμιου ιστού. Η εφαρμογή αυτή θα περιλαμβάνει ένα σύστημα προβολής και διαχείρισης ακινήτων. Συγκεκριμένα η εφαρμογή έχει ως βασικό σκοπό να βοηθήσει τους χρήστες να βρουν ακίνητα τα οποία ταιριάζουν στις προτιμήσεις τους, καθώς θα δίνει την δυνατότητα εκτός από τα χαρακτηριστικά του ακινήτου να μπορούν να επιλέξουν και τα σημεία ενδιαφέροντος (points of interest) που θα θέλουν να έχουν στην ευρύτερη περιοχή.

Ο χρήστης θα έχει την δυνατότητα να βλέπει τα ακίνητα που είναι προς πώληση και να αγοράζει όποιο επιθυμεί βάσει της εξελιγμένης διαδικασίας αναζήτησης ακινήτων την οποία διαθέτει η εφαρμογή. Τέλος, ο κύριος διαχειριστής (προγραμματιστής) θα έχει πρόσβαση σε όλα τα πεδία και τις σελίδες της εφαρμογής, έχοντάς την δυνατότητα να προσθέτει δεδομένα στη βάση καθώς και να διαχειρίζεται τα φίλτρα αναζήτησης. Ο κύριος διαχειριστής θα έχει την αρμοδιότητά να μεταφέρει τα δεδομένα των ακινήτων στη βάση δεδομένων.

Κύριος στόχος λοιπόν της εργασίας είναι η δυνατότητα της εφαρμογής να παρέχει στους αγοραστές ποικιλία ακινήτων καθώς και ποικιλία σε σημεία ενδιαφέροντος, για να έχουν την δυνατότητα να επιλέξουν τα ακίνητα με τις καλύτερες προδιαγραφές, τόσο του ίδιου του ακινήτου όσο και της περιοχής αυτού. Η εφαρμογή θα επικοινωνεί με μια βάση δεδομένων όπου οι χρήστες θα έχουν την δυνατότητα να ανασύρουν δυναμικά τα στοιχεία από αυτήν. Έτσι θα υπάρχει μια σταθερή επικοινωνία και αλληλεπίδραση μεταξύ της web εφαρμογής και της βάσης δεδομένων. Τέλος σημαντικό χαρακτηριστικό της εφαρμογής θα είναι ο συνδυασμός των χαρακτηριστικών του ακινήτου με τα poi της περιοχής για την δημιουργία φίλτρων αναζήτησης βάση των αποστάσεων μεταξύ τους.

## Λέξεις Κλειδιά

Web App,Laravel, Php, MySQL,House,POI





# Abstract

The main goal of this dissertation is research, design and development of new search filters for real estates on an online application in a global web environment. This application will includes a real estate promotion and management system. Specifically The main purpose of the application is to help users find properties that suit their preferences, as it will enable them in addition to the features of the property to be able to choose the points of interest that they will want to have in surrounding area.

The user will be able to see the properties that are for sale and buy whoever wishes based on the advanced real estate search process that the application has. Finally, the main administrator will have access to all the fields and pages of the application, having the ability to add data to the database as well as manage the search filters. The main administrator will be responsible for transferring estate data to the database.

So the main goal of the work is the possibility of the application to provide buyers with a variety of properties as well variety of points of interest, to be able to choose the properties with the best specifications, both of the property itself and of the area. The application will communicate with a database where users will be able to dynamically retrieve data from it and will be able to modify or create new ones depending on their rights. So there will be a constant communication and interaction between the web application and the database. Finally, an important feature of the application will be the combination of the properties of the property with the poi of the area for the creation of search filters based on the distances between them.

## Keywords

Web App,Laravel, Php, MySQL,House,POI



*Αφιερωμένο στον άνθρωπο που μου έμαθε την αξία της γνώσης. Ευχαριστώ παππού*



# Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ.Βασιλακόπουλο Μιχαήλ για την επίβλεψη αυτής της διπλωματικής εργασίας καθώς και για την πολύτιμη βοήθεια και τις γνώσεις που μου προσέφερε. Θα ήθελα να ευχαριστήσω τους γονείς μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια. Τέλος θα ήθελα να ευχαριστήσω όλους τους καθηγητές του Πανεπιστήμιου Θεσσαλίας για τις γνώσεις και τα εφόδια που μου παρείχαν.



# Πρόλογος

Η απόφαση της δημιουργίας αυτής της εργασίας επήλθε στο τελευταίο έτος της φοίτησής μου, καθώς αναγκάστηκα εγώ και η οικογένειά μου να μετακομίσουμε. Κατά τη διάρκειά αναζήτησης νέου ακινήτου παρουσιάστηκαν προβλήματα και ήρθαμε αντιμέτωποι με αρκετές ελλείψεις και ανακρίβειες στον γενικότερο χώρο της μεσιτικής εργασίας, καθώς και στις ιστοσελίδες που έκαναν αναζήτηση ακινήτων. Εξοπλισμένος από την εμπειρία που είχα εκείνο το καλοκαίρι από την ενασχόληση μου με ένα ηλεκτρονικό μαγαζί, καθώς και με τις απαραίτητες γνώσεις από τη σχολή, αποφάσισα να δημιουργήσω μια εφαρμογή αναζήτησης ακινήτων φιλική για τον χρήστη, εμπλουτισμένη με καινούργια φίλτρα αναζήτησης. Η σωστή και μεθοδική καθοδήγηση από τους επιβλέποντες καθηγητές αλλά και οι συμβουλές και οι παρατηρήσεις τους με βοήθησαν σε μεγάλο βαθμό ώστε να καταφέρω να ολοκληρώσω την εργασία με επιτυχία.





# Περιεχόμενα

<b>Περίληψη</b>	<b>i</b>
<b>Abstract</b>	<b>iii</b>
<b>Ευχαριστίες</b>	<b>vii</b>
<b>Πρόλογος</b>	<b>ix</b>
<b>Περιεχόμενα</b>	<b>xi</b>
<b>Κατάλογος σχημάτων</b>	<b>xv</b>
<b>1 Εισαγωγή</b>	<b>1</b>
1.1 Αντικείμενο της διπλωματικής . . . . .	2
1.1.1 Συνεισφορά . . . . .	2
1.2 Οργάνωση του τόμου . . . . .	2
<b>2 Συγγενικές εργασίες</b>	<b>5</b>
2.1 Εισαγωγή . . . . .	5
2.2 Spitogatos . . . . .	5
2.3 HULI . . . . .	6
2.4 Trulia . . . . .	6
<b>3 Θεωρητικό υπόβαθρο</b>	<b>9</b>
3.1 Μοντέλα και τεχνολογίες που αξιοποιήθηκαν κατά την δημιουργία της web εφαρμογής . . . . .	9
3.1.1 PHP . . . . .	9
3.1.2 HTML - HTML 5 . . . . .	10
3.1.3 CSS . . . . .	10
3.1.4 Laravel . . . . .	10
3.1.5 Php Storm . . . . .	11
3.2 Μοντέλα και τεχνολογίες που αξιοποιήθηκαν κατά την δημιουργία της βάσης δεδομένων . . . . .	11

3.2.1	MySQL . . . . .	12
3.2.2	Apache Web Server . . . . .	12
3.2.3	DataGrip . . . . .	12
<b>4</b>	<b>Ανάλυση απαιτήσεων εφαρμογής</b>	<b>13</b>
4.1	Εισαγωγή . . . . .	13
4.2	Υπολογισμός απόστασης μεταξύ συντεταγμένων . . . . .	13
4.2.1	Μέθοδος μεγάλης απόστασης μεταξύ δύο σημείων (Haversine) . . . . .	13
4.2.2	Μέθοδος Vincenty . . . . .	14
4.2.3	Μέθοδος Σφαιρικός νόμος συνημίτονων . . . . .	15
4.2.4	Ισογώνια προσέγγιση . . . . .	15
4.2.5	Βάση άξονα (αζιμούθιο) . . . . .	16
4.2.6	Περιθώριο λάθους . . . . .	17
4.3	Εύρεση κοντινότερου σημείου ενδιαφέροντος . . . . .	18
4.4	Εύρεση δεδομένων . . . . .	18
<b>5</b>	<b>Επίλυση απαιτήσεων εφαρμογής</b>	<b>19</b>
5.1	Εισαγωγή . . . . .	19
5.2	Τρόπος υπολογισμού απόστασης μεταξύ συντεταγμένων . . . . .	19
5.3	Υπολογισμός κοντινότερου σημείου ενδιαφέροντος . . . . .	19
5.4	Δημιουργία εικονικών δεδομένων (dummy data) . . . . .	20
<b>6</b>	<b>Παρουσίαση Web Εφαρμογής</b>	<b>23</b>
6.1	Αρχική Σελίδα . . . . .	23
6.2	Σελίδα Αναζήτησης . . . . .	23
6.3	Σελίδα Αποτελεσμάτων . . . . .	25
<b>7</b>	<b>Τεχνικές λεπτομέρειες</b>	<b>31</b>
7.1	Λεπτομέρειες υλοποίησης . . . . .	31
7.1.1	MVC μοντέλο . . . . .	31
7.1.2	Migrations και σχεδιασμός της Βάσης . . . . .	32
7.1.3	Routing . . . . .	33
7.1.4	Φόρμα HTML . . . . .	35
7.1.5	Αρχικοποίηση αποστάσεων στη βάση . . . . .	36
7.2	Εγκατάσταση Εφαρμογής . . . . .	38
<b>8</b>	<b>Επίλογος</b>	<b>41</b>
8.1	Σύνοψη και συμπεράσματα . . . . .	41
8.2	Μελλοντικές επεκτάσεις . . . . .	42
<b>I</b>	<b>Παράθεση Κώδικα Front-end και Διαχείριση MVC μοντέλου</b>	<b>43</b>
I.1	Κώδικας HTML και CSS . . . . .	43
I.2	Διαχείριση αρχείων MVC μοντέλου . . . . .	43

---

I.2.1	Δυνατότητες Laravel . . . . .	44
	<b>Βιβλιογραφία</b>	<b>47</b>
	<b>Συνομογραφίες</b>	<b>49</b>
	<b>Ορολογία - Γλωσσάρι</b>	<b>51</b>



# Κατάλογος σχημάτων

2.1	spitogatos.gr . . . . .	5
2.2	huli.gr . . . . .	6
2.3	trulia.com . . . . .	6
4.1	Errors . . . . .	17
4.2	Relative errors . . . . .	18
5.1	Haversine . . . . .	20
5.2	Closest Poi . . . . .	20
5.3	Seeders . . . . .	21
5.4	factories . . . . .	21
5.5	Factory Example . . . . .	22
5.6	Seeder Example . . . . .	22
6.1	Welcome page . . . . .	24
6.2	Search Page . . . . .	26
6.3	Home Button . . . . .	26
6.4	Location filters . . . . .	26
6.5	Location Filter options . . . . .	26
6.6	House specifications . . . . .	27
6.7	Size options min . . . . .	27
6.8	Size options max . . . . .	27
6.9	Find Button . . . . .	27
6.10	Search example . . . . .	28
6.11	Results Page . . . . .	28
6.12	Unsuccessful Search . . . . .	28
6.13	Successful Search . . . . .	29
6.14	House Card . . . . .	30
6.15	New Search button . . . . .	30
7.1	Search controller for distance filter . . . . .	32
7.2	Search controller for Range filter . . . . .	33
7.3	Migrations . . . . .	34

---

7.4	Migration table example	34
7.5	Models	34
7.6	Data Base	35
7.7	Routing	35
7.8	Form part1	36
7.9	Form part2	36
7.10	Form part3	37
7.11	Form part4	37
7.12	Laravel	39
7.13	Project start page	40
I.1	html css code pt1	43
I.2	html css code pt2	44
I.3	html css code pt3	45
I.4	html css code pt4	45
I.5	html css code pt5	46
I.6	html css code pt6	46

# Κεφάλαιο 1

## Εισαγωγή

Στη σημερινή εποχή, όπου η τεχνολογία εισχωρεί εκθετικά στις ζωές όλων, και πολλά προβλήματα από το κοντινό παρελθόν μας πλέον λύνονται με το πάτημα ενός κουμπιού, ένας χώρος όπου δεν συμβαδίζει με τα σημερινά δεδομένα, ιδιαίτερα στην χώρα μας, είναι αυτός της μεσιτικής εργασίας, και γενικά της διαδικασίας αναζήτησης ακινήτου. Σε μια εποχή λοιπόν όπου η ενδεδειγμένη μορφή αναζήτησης αγαθών γίνεται ηλεκτρονικά τα ακίνητα δεν θα πρέπει να αποτελούν εξαίρεση, όταν είναι ένα από τα πιο πολύτιμα αγαθά για τον σύγχρονο άνθρωπο.

Από τα προϊστορικά κιόλας χρόνια, ο άνθρωπος θέλησε να εξασφαλίσει την ζεστασιά και προστασία που το προσέφερε ένα σπίτι, μια οικία από τα άγρια θηρία και τα στοιχεία της φύσης. Με το πέρασμα των χρόνων, η αξία της οικίας για τον άνθρωπο παραμένει αναλλοίωτη, καθώς γύρω από αυτήν οι άνθρωποι συσπειρώνονται, δημιουργούν την οικογένεια τους, επενδύουν τα όνειρα τους και έχουν την δυνατότητα να την διαμορφώσουν με βάση την προσωπική τους αισθητική. Για τον άνθρωπο, το σπίτι του θεωρείται το καταφύγιο του, στο οποίο καταφεύγει κλείνοντας την πόρτα σε οτιδήποτε τον απασχολεί και τον προβληματίζει θέτοντας έτσι τους δικούς του όρους και προϋποθέσεις μακριά από κοινωνικές συμβάσεις και νόρμες. Επιπλέον, η επιχειρηματική δραστηριότητα γύρω από την οικία τα τελευταία χρόνια ακμάζει καθώς πολλοί είναι αυτοί που επιλέγουν να επενδύουν τα χρήματά τους στην αγορά ακινήτων προκειμένου να τα αξιοποιήσουν αργότερα ως επένδυση για άμεση οικονομική απόσβεση μέσω της ενοικίασης ή μεταπώλησης. Επομένως, διαχρονικά η κατοικία εξασφαλίζει την συναισθηματική και κοινωνική άνθιση στη ζωή του ανθρώπου και αποτελεί αυθαίρετο δικαίωμα κάθε ανθρώπου ανεξαρτήτου καταβολών. Έχοντας λοιπόν μεγάλη αξία για τον μέσο άνθρωπο, η αγορά ενός δεν είναι μια απλή διαδικασία. Απαιτεί σκέψη, σχεδιασμό και θα πρέπει η αναζήτησή τους να γίνεται με τον πλέον εξελιγμένο τρόπο, παρέχοντας όλες τις αναγκαίες πληροφορίες τόσο για το ίδιο το ακίνητο όσο και για τις ανάγκες που αυτό παρέχει με βάση την τοποθεσία στην οποία βρίσκεται.

Τέλος, η τοποθεσία του ακινήτου είναι ένα από τα πιο βασικά χαρακτηριστικά του, καθώς το που βρίσκεται επηρεάζει άμεσα την τιμή του. Επιπλέον, με τα σημερινά δεδομένα οι αγοραστές δεν ψάχνουν μόνο ένα ακίνητο που θα ικανοποιεί τα θέλω τους με τα χαρακτηριστικά του, αλλά να τους παρέχει άμεση πρόσβαση στις αναγκαίες υπηρεσίες και αγαθά.

## 1.1 Αντικείμενο της διπλωματικής

Ο κύριος στόχος αυτής της διπλωματικής εργασίας είναι η δημιουργία καινοτόμων φίλτρων αναζήτησης ακινήτων, βάση του περιβάλλοντος τους, δηλαδή τα σημεία ενδιαφέροντος που τα περιβάλλουν. Αυτό θα επιτευχθεί μέσω μιας εφαρμογής παγκοσμίου ιστού (web application) που θα παρέχει στους χρήστες της την δυνατότητα να αναζητούν ακίνητα, τα οποία θα ταιριάζουν απόλυτα στις ανάγκες και τις προδιαγραφές τους. Η εφαρμογή θα παρέχει στους χρήστες ένα σύστημά προβολής ακινήτων καθώς και το πιο σημαντικό, θα τους δίνει την δυνατότητα να χρησιμοποιήσουν εξειδικευμένα φίλτρα για τους σκοπούς αναζήτησης τους. Τα φίλτρα αυτά θα αντιπροσωπεύουν όχι μόνο τα μεμονωμένα χαρακτηριστικά του σπιτιού, αλλά θα παρέχουν και αποστάσεις από τα πλησιέστερα σημεία ενδιαφέροντος, και η αναζήτηση θα μπορεί να δομηθεί γύρω από τις αποστάσεις αυτές.

Σκοπός είναι να δημιουργηθεί ένας καινούργιος τρόπος αναζήτησης ακινήτων, όχι μόνο βάση των χαρακτηριστικών του σπιτιού, που μέχρι τώρα όλες οι αντίστοιχες εφαρμογές παγκοσμίου ιστού παρέχουν, αλλά και με βάση τα χαρακτηριστικά της περιοχής της οποία βρίσκεται το ακίνητο, και τις σημεία ενδιαφέροντος που το περιβάλλουν. Ο χρήστης ουσιαστικά να μπορεί να βρεί εύκολα και αποδοτικά το ακίνητο που αναζητά και ταυτόχρονα να μπορέσει να επιλέξει και τα σημεία ενδιαφέροντος γύρω από το ακίνητο, έτσι ώστε να μπορεί να έχει τις απαραίτητες υπηρεσίες και αγαθά που κρίνει ο ίδιος αναγκαία.

### 1.1.1 Συνεισφορά

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Δημιουργήθηκαν δυναμικές σελίδες διαχείρισης και αποθήκευσης δεδομένων συνδεδεμένες με MySQL [16] βάση.
2. Υλοποιήθηκαν προηγμένα φίλτρα αναζήτησης.
3. Υλοποιήθηκε πλατφόρμα προβολής ακινήτων.
4. Υλοποιήθηκαν συναρτήσεις υπολογισμού γεωγραφικών αποστάσεων για την υποστήριξη των προηγμένων φίλτρων.
5. Υλοποιήθηκε Φόρμα υποδοχής των χρηστών.
6. Δημιουργήθηκαν πλασματικά δεδομένα (dummy data) για την βάση δεδομένων.

## 1.2 Οργάνωση του τόμου

Παρόμοιες εφαρμογές με αυτή της διπλωματικής παρουσιάζονται στο κεφάλαιο 2. Το κεφάλαιο 3 αναλύει το θεωρητικό υπόβαθρο σχετικά με τα τεχνολογικά μοντέλα που χρησιμοποιήθηκαν σε αυτή την διπλωματική εργασία. Στο κεφάλαιο 4 πραγματοποιούμε την ανάλυση των απαιτήσεων της web εφαρμογής. Το κεφάλαιο 5 αναλύει πώς αντιμετωπίστηκαν και επιλύθηκαν τα προβλήματα και οι απαιτήσεις του κεφαλαίου 4. Το κεφάλαιο 6 παρουσιάζει την εμφάνιση και τα



χαρακτηριστικά της εφαρμογής. Αυτό σημαίνει ότι παρουσιάζει όλες τις σελίδες και λεπτομέρειες σχετικά με αυτές, με βάση ορισμένων προδιαγραφών. Στο κεφάλαιο 7 αναλύονται εκτενέστερα ορισμένες τεχνικές λεπτομέρειες οι οποίες χρήζουν ιδιαίτερης προσοχής και ανάλυσης. Σε αυτό το κεφάλαιο περιγράφεται επίσης και η διαδικασία που πρέπει να ακολουθηθεί για την σωστή εγκατάσταση της web εφαρμογής έτσι ώστε να είναι απολύτως λειτουργική. Τέλος το κεφάλαιο 8 αποτελεί τον επίλογο της εργασίας και τονίζονται τα βασικά συμπεράσματα της αλλά και περιγράφονται ορισμένες μελλοντικές αναβαθμίσεις βελτιώσεις της web εφαρμογής.



## Κεφάλαιο 2

# Συγγενικές εργασίες

### 2.1 Εισαγωγή

Η αναζήτηση ακινήτων στην σύγχρονη εποχή είναι μια διαδικασία που ο μέσος άνθρωπος θα έρθει κάποια στιγμή αντιμέτωπος με αυτή. Είναι μια διαδικασία που θεωρείται κομβικό σημείο για στη ζωή του συγχρόνου ανθρώπου και απαιτεί οργάνωσή και μεθοδικότητα. Τα τελευταία 15 χρόνια πολλές ιστοσελίδες παγκοσμίου ιστού παρέχουν την δυνατότητα να αναζητήσει κάποιος το ιδανικό ακίνητο για τον ίδιο και την οικογένειά του. Ανά τα χρόνια όλο και περισσότερα μεσιτικά γραφεία καταφεύγουν στον παγκόσμιο ιστό για να προβάλλουν τα προϊόντα τους και να έρθουν σε επαφή με όλο και περισσότερο κόσμο. Στη συνέχεια παραθέτονται μερικές από τις πολλές εφαρμογές που υπάρχουν αυτή τη στιγμή διαθέσιμες για τον καταναλωτή.

### 2.2 Spitogatos



Σχήμα 2.1: spitogatos.gr

Ο Spitogatos Σχ. 2.1 είναι ιστοσελίδα αγγελιών ακινήτων στην Ελλάδα, με περισσότερα από 750.000 ακίνητα προς πώληση ή ενοικίαση. Μετά την ίδρυσή του το 2006 και την αρχική ερασιτεχνική ενασχόληση των ιδρυτών του, ο Spitogatos.gr υπήρξε η πρώτη ελληνική startup που άντλησε κεφάλαια από venture capital fund το 2009 και από το 2014 ανήκει στον όμιλο της Real Web, που ελέγχει, μεταξύ άλλων, το Immobiliare.it, ιστοσελίδα αγγελιών ακινήτων στην Ιταλία. Ο Spitogatos.gr είναι μέρος του Spitogatos Network και μαζί με το Spiti24.gr, το Tospitimou.gr και το Spitogatos.com.cy, συναπαρτίζουν τη μεγαλύτερη βάση δεδομένων ακινήτων και επαγγελματιών του Real Estate στην Ελλάδα και την Κύπρο. Από το 2017, το Spitogatos Network δημιούργησε το brand Indomio και απέκτησε παρουσία σε ξένες χώρες, λανσάροντας νέες διεθνείς

ιστοσελίδες αγγελιών ακινήτων. Σημειώνοντας πάνω από 40 εκατομμύρια επισκέψεις και 1 δισεκατομμύριο εμφανίσεις σελίδων το χρόνο. Τέλος διαθέτει και εφαρμογή για κινητά (android,ios). (URL : <https://www.spitogatos.gr/>)

### 2.3 HULI



Σχήμα 2.2: huli.gr

Το HULI είναι ιστοσελίδα η οποία κάνει αναζητήσεις για πληθώρα προϊόντων και υπηρεσιών, καθώς και ακινήτων. Η εταιρία ιδρύθηκε το 2016 και το 2018 πρόσθεσε τη αναζήτηση ακινήτων στις παροχές τις. Μέσω αυτής της ιστοσελίδας μπορεί κανείς να αναζητήσει ποικιλία προϊόντων και υπηρεσιών, όπως ακίνητα, ρούχα, ταξίδια, εργασία, ειδήσεις, ταινίες, συνταγές και πολλά άλλα. Το HULI παρέχει ακόμα την δυνατότητα να αναζητήσει κανείς σπίτια τα οποία βρίσκονται κοντά σε σταθμούς μετρό όπως και την δυνατότητα να ζωγραφίσει σε χάρτη περιοχή στην οποία θέλει να περιοριστεί η αναζήτηση ακινήτου. (URL : <https://www.huli.gr/>)

### 2.4 Trulia



Σχήμα 2.3: trulia.com

Η ιστοσελίδα Trulia υπήρξε έμπνευση για την δημιουργία αυτής της διπλωματικής εργασίας. Η ιστοσελίδα αυτή απευθύνεται σε αυτούς που αναζητούν ακίνητα στις Ηνωμένες Πολιτείες της Αμερικής. Είναι ένα παράδειγμα ολοκληρωμένης, φιλική προς τον χρήστη και καλοσχεδιασμένη εφαρμογή παγκοσμίου ιστού. Δημιουργήθηκε στον San Francisco το 2005 και δίνει σημασία τόσο στα χαρακτηριστικά των ακινήτων, όσο και στις γειτονιές τις οποίες βρίσκονται τα ακίνητα.

Παρέχει στους χρήστες την δυνατότητα αφού επιλέξουν μια περιοχή αναζήτησης να δουν και χαρακτηριστικά της περιοχής, όπως εγκληματικότητα, διαθέσιμα σημεία ενδιαφέροντος, ευκαιρίες εργασίας και πολλά άλλα. Ακόμα παρέχει την δυνατότητα αγοράς ενοικίασης η ακόμα και υποθήκης ενός ακινήτου. Το Trulia διαθέτει τεράστια βάση δεδομένων καθώς καλύπτει όλη την Αμερική, και εξελίσσεται συνεχώς για να παρέχει την καλύτερη δυνατή εμπειρία αναζήτησης ακινήτων σε χρήστες αλλά και πωλητές. (URL : <https://www.trulia.com/>)



## Κεφάλαιο 3

# Θεωρητικό υπόβαθρο

### 3.1 Μοντέλα και τεχνολογίες που αξιοποιήθηκαν κατά την δημιουργία της web εφαρμογής

Για την δημιουργία της εφαρμογής στον παγκόσμιο ιστό, ως βασική γλώσσα υλοποίησης χρησιμοποιήθηκε η διαδικτυακού τύπου γλώσσα PHP [3] η οποία ενδείκνυται για τη δημιουργία σελίδων με δυναμικό περιεχόμενο. Επίσης, έγινε χρήση της γλώσσας προγραμματισμού HTML [5] με σκοπό την ευκολότερη αξιοποίηση της γλώσσας CSS [5] για την κάλυψη των γραφικών χαρακτηριστικών - αναγκών της εφαρμογής. Ο βασικός λόγος επιλογής των πιο πάνω γλωσσών προγραμματισμού είναι το γεγονός ότι επιθυμούμε η εφαρμογή να είναι όσο το δυνατόν πιο φιλική ως προς την χρήση της «User Interface» όσο και για την διαχείριση της. Για την υποστήριξη της PHP αλλά και όλου του περιβάλλοντος στο οποίο θα εκτελείτε η εφαρμογή γίνεται χρήση του APACHE Web Server σε λειτουργικό σύστημα Ubuntu [20].

#### 3.1.1 PHP

Η PHP είναι μια γλώσσα προγραμματισμού με κύρια λειτουργία την δημιουργία web-apps , στις οποίες δίνεται η δυνατότητα επικοινωνίας με μια βάση δεδομένων για την διαχείριση του περιεχομένου τους. Κάθε αρχείο PHP θα πρέπει να είναι κατάλληλης μορφής (π.χ. \*.php, \*.php4). Εσωτερικά στα αρχεία PHP δίνεται η δυνατότητα χρήσης κώδικα HTML γεγονός που διευκολύνει στον προγραμματισμό. Ο κώδικας της PHP μεταφράζεται από έναν διερμηνέα (interpreter), ο οποίος υλοποιείται από ένα component (Apache HTTP) ενός server. Ο κώδικας αυτός εκτελείται και ο server στέλνει το αποτέλεσμα στον client μέσω http response με τη μορφή μιας ιστοσελίδας με περιεχόμενο της μορφής html. Πλεονεκτήματα PHP :

1. Υψηλή απόδοση
2. Χαμηλή κατανάλωση πόρων (resources)
3. Δυνατότητα συνδέσης με πολλά συστήματα βάσης δεδομένων
4. Ενσωματωμένες (by default) βιβλιοθήκες για πολλές Web διαδικασίες

5. Εύκολη στη χρήση και στην υλοποίηση
6. Αντικειμενοστραφή υποστήριξη

### 3.1.2 HTML - HTML 5

Όλες οι ιστοσελίδες που εμφανίζονται σε έναν browser είναι αρχεία γραμμένα στην γλώσσα προγραμματισμού HTML (HyperText Markup Language), τα οποία αποτελούνται από τα κείμενα των σελίδων, τη δομή τους, εικόνες και συνδέσμους που μπορεί να περιέχουν για προβολή. Η HTML βασίζεται στη γλώσσα SGML (Standard Generalized Markup Language), η οποία είναι ένα μεγαλύτερο module επεξεργασίας αρχείων. Η HTML λοιπόν ορίζει ένα σύνολο μεταβλητών - χαρακτηριστικών για τις σελίδες, όπως είναι οι επικεφαλίδες, οι τίτλοι, λίστες αντικειμένων και παράγραφοι. Η HTML 5 είναι η τελευταία έκδοση της HTML και προσφέρει περισσότερες δυνατότητες και περισσότερες επιλογές στην δημιουργία φόρμας. Αυτό σημαίνει ότι έχει περισσότερες δυνατότητες διαχείρισης δεδομένων των post requests πεδία όπως E-mail, password, τηλέφωνο κλπ. Ο βασική λογική του σχεδιασμού της είναι το να προσφέρει όλα τα εργαλεία που χρειάζονται για την εμφάνιση και εξυπηρέτηση web εφαρμογών χωρίς να χρειάζεται η εγκατάσταση επιπλέον δομών (plugins). Βασικό πλεονεκτήματα της είναι η δυνατότητα προσθήκης επιπλέον λειτουργικότητας για τον χρήστη το οποίο το πετυχαίνει με την ομαλή ενσωμάτωση και χρήση της CSS.

### 3.1.3 CSS

Η CSS (Cascading Style Sheets) είναι μια γλώσσα η οποία ανήκει στην κατηγορία των γλωσσών που χρησιμοποιούνται για την δυνατότητα ελέγχου της εμφάνισης ενός αρχείου το οποίο είναι της μορφής μιας γλώσσας σήμανσης. Επομένως χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός αρχείου HTML και XHTML. Η CSS είναι μία γλώσσα εξειδικευμένη στο στήσιμο μιας σελίδας ως προς ορισμένα χαρακτηριστικά. Αναλυτικότερα μπορεί να καθορίζει χαρακτηριστικά όπως, χρώματα, στοίχιση, σχήματα, και να διαχειρίζεται μεγέθη, γωνίες και τοποθεσίες αντικειμένων μέσα σε μια σελίδα γεγονός το οποίο εξελίσσει τις δυνατότητες της html. Το CSS3 είναι η τελευταία έκδοση της CSS η οποία εφαρμόζει νέα χαρακτηριστικά όπως στρογγυλεμένες κλίσεις, γωνίες, σκιές, πάχη γραμμών ακόμη και δυνατότητες animation, καθώς και την δυνατότητα multicolumns και grid layouts.

### 3.1.4 Laravel

Το Laravel [14] είναι ένα framework με κομψή και εύκολη σύνταξη, απόλυτα συμβατό με την php, με αρκετές βάσεις δεδομένων αλλά και με πολλά πρόσθετα plugins τα οποία προσφέρονται για μεγαλύτερη διευκόλυνση του χρήστη (προγραμματιστή). Βασικό πλεονέκτημα του laravel είναι η ενσωματωμένη υλοποίηση πολλών βασικών λειτουργιών απαραίτητων για την δημιουργία μιας web εφαρμογής. Ακόμη η laravel παρέχει τα laracasts τα οποία είναι στο μεγαλύτερο μέρος τους δωρεάν αναλυτικά tutorials εκπαίδευσης του framework αλλά και των δομών που χρησιμοποιεί ο χρήστης όπως php, HTML και δομές για αυτόματο έλεγχο της web εφαρμογής μέσω του laravel.



### 3.1.5 Php Storm

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το Php Storm [17] της jet brains σαν πρόγραμμα επεξεργασίας κώδικα. Το Php Storm είναι ένα εύχρηστο και σύγχρονο πρόγραμμα επεξεργασίας κώδικα. Υποστηρίζει την εγκατάσταση επιπρόσθετων plugins για διάφορες λειτουργίες. Πλεονεκτήματα PhpStorm:

1. Γρήγορη πλοήγηση σε αρχεία, σύμβολα, εκφράσεις και γραμμές
2. Προσφέρει γρήγορη κλήση πληκτρολογίου για επιπλέον εντολές
3. Υποστηρίζει πληθώρα από frameworks όπως Laravel, Symfony, Drupal, WordPress, Zend Framework, Magento, Joomla!, CakePHP, Yii και πολλά άλλα.
4. Αξιοποιεί στο έπακρο πρωτοποριακές τεχνολογίες αιχμής, όπως HTML 5, CSS, Sass, Less, Stylus, CoffeeScript, TypeScript, Emmet και JavaScript, με refactorings, debugging και unit testing διαθέσιμα.
5. Προσφέρει πλήρη υποστήριξη σε όλες τις μορφές γλώσσας Php καθώς και με όλες τις βάσεις SQL.
6. Ευφυής βοήθεια κωδικοποίησης. Εκατοντάδες επιθεωρήσεις φροντίζουν να επαληθεύουν τον κώδικα καθώς πληκτρολογείτε, αναλύοντας ολόκληρο το έργο.
7. Γρήγορη και ασφαλής αναμόρφωση κώδικα. Αναμορφώνει τον κώδικα αξιόπιστα με την ασφαλή Μετονομασία, Μετακίνηση, Διαγραφή, Εξαγωγή μεθόδου, Inline Variable, Push member Up / Pull member Down, Change Signature και πολλά άλλα refactorings.
8. Εύκολο εντοπισμό σφαλμάτων και δοκιμές. Το PhpStorm είναι γνωστό για το Visual Debugger μηδενικής διαμόρφωσης, παρέχοντας εξαιρετική εικόνα για το τι συμβαίνει στην εφαρμογή σε κάθε βήμα. Λειτουργεί με το Xdebug και το Zend Debugger και μπορεί να χρησιμοποιηθεί τόσο τοπικά όσο και απομακρυσμένα. Οι δοκιμές μονάδων με PHPUnit, BDD με Behat και ενσωμάτωση προφίλ είναι επίσης διαθέσιμες.

## 3.2 Μοντέλα και τεχνολογίες που αξιοποιήθηκαν κατά την δημιουργία της βάσης δεδομένων

Για βάση δεδομένων χρησιμοποιήθηκε η γλώσσα MySQL, η οποία είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Για την υποστήριξη της χρησιμοποιήθηκε ο APACHE web server και αυτά εγκαταστάθηκαν και συνδέθηκαν πλήρως σε περιβάλλον Ubuntu 19.10.

### 3.2.1 MySQL

Η MySQL [16] [19] είναι μια από τις πιο διαδεδομένες βάσεις δεδομένων [12]. Η MySQL είναι μια κοινή επιλογή για χρήση σε web εφαρμογές, καθώς προσφέρει πληθώρα δυνατοτήτων. Αρκετές γνωστές εφαρμογές είναι βασισμένες στη MySQL, όπως το Wordpress και άλλα μεγάλα web-services. Ένα σημαντικό πλεονεκτήματα της είναι ότι είναι συμβατή με τα περισσότερα περιβάλλοντα εργασίας όπως Linux, Mac OS, Windows κ.α, δίνοντας με αυτό τον τρόπο την δυνατότητα στον εκάστοτε προγραμματιστή να χρησιμοποιήσει την πλατφόρμα που προτιμά χωρίς να υπάρχουν προβλήματα συμβατότητας. Στην προκειμένη περίπτωση η MySQL χρησιμοποιήθηκε σε Ubuntu μέσω του γραφικού περιβάλλοντος DataGrip [4].

### 3.2.2 Apache Web Server

Ο Apache Web Server [1] [10] είναι ένα module το οποίο παρέχει ένα επαγγελματικού επιπέδου σέρβερ με πολλές δυνατότητες και ελεύθερο κώδικα HTTP (Web) Server. Ο Apache χρησιμοποιείται σε πάρα πολλά λειτουργικά συστήματα όπως Linux, Windows και Mac OS γεγονός το οποίο τον κάνει πολύ εύκολο στην χρήση και διαχείριση του. Στα πλαίσια της διπλωματικής εγκαταστάθηκε μέσω των Linux(Ubuntu).

### 3.2.3 DataGrip

Το DataGrip είναι ένα εργαλείο για τη διαχείριση βάσεων δεδομένων σε MySQL. Είναι γραμμένο σε PHP και είναι συμβατό με όλα τα συστήματα. Προσφέρει δυνατότητες δημιουργίας, επεξεργασίας, και διαγραφής βάσεων δεδομένων, πινάκων, πεδίων μεταβλητών και σειρών. Επιπλέον δίνει την δυνατότητα εκτέλεσης εντολών σε SQL μορφή. Έχει την ικανότητα εισαγωγής και εξαγωγής δεδομένων από και σε διάφορες μορφές, όπως CSV, SQL, XML, Excel κ.α. Πλεονεκτήματα DataGrip :

1. Εύκολη διαχείριση βάσεων δεδομένων (δημιουργία - διαγραφή - μεταφορά κ.α.)
2. Συντήρηση της βάσης
3. Εύκολη διαχείριση πινάκων και πεδίων σε αυτούς
4. Refactorings που λειτουργούν σε αρχεία και schema SQL
5. Όμορφο και περιεκτικό γραφικό περιβάλλον

## Κεφάλαιο 4

# Ανάλυση απαιτήσεων εφαρμογής

### 4.1 Εισαγωγή

Όπως προαναφέρθηκε η εργασία αυτή έχει ως στόχο την αναζήτηση ακινήτων με νέους τρόπους και χρήσης δεδομένων τοποθεσίας, όχι μόνο των ίδιων των ακινήτων, αλλά και του άμεσου περιβάλλον τους. Η χρήση των σημείων ενδιαφέροντος ως φίλτρα αναζήτησης για ένα ακίνητο είναι κάτι το οποίο δεν χρησιμοποιείτε τοιουτοτρόπως αυτή τη στιγμή, σε καμία από τις μεγαλύτερες ιστοσελίδες αναζήτησης ακινήτων. Οι συνθήκες, οι υπολογισμοί, καθώς και τα προβλήματα που επέρχονται με την χρήση των σημείων ενδιαφέροντος ως φίλτρα αναζήτησης αλλά και όλης της υπόλοιπης εργασίας, θα αναφερθούν και θα αναλυθούν σε αυτό το κεφάλαιο.

### 4.2 Υπολογισμός απόστασης μεταξύ συντεταγμένων

Για τον υπολογισμό απόστασης μεταξύ δύο σημείων στην επιφάνεια της Γης, υπάρχουν πολλοί τρόποι και αλγόριθμοι. Μερικοί από τους οποίους παρατίθενται παρακάτω. Όλοι αυτοί οι τύποι είναι για υπολογισμούς με βάση σφαιρικής Γης (αγνοώντας τα ελλειψοειδή φαινόμενα) - η οποία είναι αρκετά ακριβής για τους περισσότερους σκοπούς (Στην πραγματικότητα, η Γη είναι πολύ ελαφρώς ελλειψοειδής. Η χρήση σφαιρικού μοντέλου δίνει σφάλματα τυπικά έως 0,3%). Όλοι οι κώδικες που θα παρατεθούν σε αυτή την ενότητα θα είναι σε γλώσσα Javascript, για την εργασία ο τρόπος που επιλέχθηκε, παραμετροποιήθηκε σε rhp.

#### 4.2.1 Μέθοδος μεγάλης απόστασης μεταξύ δύο σημείων (Haversine)

Αυτή η μέθοδος χρησιμοποιεί τον τύπο «haversine» [13] για να υπολογίσει την απόσταση του μεγάλου κύκλου μεταξύ δύο σημείων - δηλαδή, τη μικρότερη απόσταση πάνω από την επιφάνεια της γης - δίνοντας μια απόσταση «όπως πετά το κοράκι» μεταξύ των σημείων (αγνοώντας δηλαδή οποιοδήποτε λόγο η υψομετρική διαφορά).

$$\text{Haversine } a = \sin^2(\Delta\phi/2) + \cos \phi_1 * \cos \phi_2 * \sin^2(\Delta\lambda/2)$$

$$\text{Φόρμουλα: } c = 2 * \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad d = R * c$$

Όπου  $\varphi$  είναι το γεωγραφικό μήκος,  $\lambda$  είναι το γεωγραφικό πλάτος,  $R$  είναι η ακτίνα της Γης (μέση ακτίνα = 6,371km); Σημειώστε ότι οι γωνίες πρέπει να είναι σε ακτίνια για να περάσουν στις τριγωνομετρικές λειτουργίες)

JavaScript:

```
const R = 6371e3; // metres
const φ1 = lat1 * Math.PI/180; // φ, λ in radians
const φ2 = lat2 * Math.PI/180;
const Δφ = (lat2-lat1) * Math.PI/180;
const Δλ = (lon2-lon1) * Math.PI/180;

const a = Math.sin(Δφ/2) * Math.sin(Δφ/2) +
  Math.cos(φ1) * Math.cos(φ2) *
  Math.sin(Δλ/2) * Math.sin(Δλ/2);
const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));

const d = R * c; // in metres
```

## 4.2.2 Μέθοδος Vincenty

Οι τύποι αναπτύχθηκαν από τον Thaddeus Vincenty (1975) [21] για τον υπολογισμό γεωδαιτικών αποστάσεων μεταξύ ενός ζεύγους σημείων γεωγραφικού πλάτους / μήκους σε ένα ελλειψοειδές μοντέλο της Γης. Σε αντίθεση με τη μέθοδο Haversine για τον υπολογισμό της απόστασης σε μια σφαίρα, αυτοί οι τύποι είναι μια επαναληπτική μέθοδος και υποθέτουν ότι η Γη είναι ελλειψοειδές.

Αυτός ο τύπος περιλαμβάνει μια άμεση και μια αντίστροφη μέθοδο όπου:

Άμεση μέθοδος: Υπολογίζει τη θέση ενός σημείου που είναι δεδομένη απόσταση και αζιμούθιο από άλλο σημείο

Αντίστροφη μέθοδος: Υπολογίζει τη γεωγραφική απόσταση και το αζιμούθιο μεταξύ δύο δεδομένων σημείων.

Για παράδειγμα, στην έξοδο της Αντίστροφης μεθόδου το  $\lambda$  παράγεται μετά την εκχώρηση πολλών σταθερών, συμπεριλαμβανομένου του μήκους του ημι-μεγάλου άξονα, του μήκους του ημι-δευτερεύοντος άξονα, της ισοπέδωσης, των συντεταγμένων γεωγραφικού πλάτους, των μειωμένων γεωγραφικών πλάτους κ.λπ. Ο στόχος της μεθόδου είναι να ελαχιστοποιηθεί η τιμή του αποτελέσματος  $\lambda$  (δηλαδή όταν τα αποτελέσματα συγκλίνουν σε έναν επιθυμητό βαθμό ακρίβειας).

$$\sin \sigma = \sqrt{(\cos U_2 \sin \lambda)^2 + (\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos \lambda)^2}$$

$$\cos \sigma = \sin U_1 \sin U_2 + \cos U_1 \cos U_2 \cos \lambda$$

$$\sigma = \arctan2(\sin \sigma, \cos \sigma)$$

$$\sin \alpha = \frac{\cos U_1 \cos U_2 \sin \lambda}{\sin \sigma}$$

$$\cos(2\sigma_m) = \cos \sigma - \frac{2 \sin U_1 \sin U_2}{\cos^2 \alpha} = \cos \sigma - \frac{2 \sin U_1 \sin U_2}{1 - \sin^2 \alpha}$$

$$C = \frac{f}{16} \cos^2 \alpha [4 + f(4 - 3 \cos^2 \alpha)]$$

$$\lambda = L + (1 - C)f \sin \alpha \{ \sigma + C \sin \sigma [ \cos (2\sigma_m) + C \cos \sigma (-1 + 2 \cos^2 (2\sigma_m)) ] \}$$

Όταν η διαφορά μεταξύ της τρέχουσας τιμής του  $\lambda$  και της τιμής του  $\lambda$  από την προηγούμενη επανάληψη είναι μικρότερη από την ανοχή σύγκλισης, τότε το τελικό στάδιο της αντίστροφης μεθόδου μπορεί να εκτελεστεί χρησιμοποιώντας τους παραπάνω τύπους.

Γενικά η μέθοδος Vincenty είναι πολύ ακριβής αλλά απαιτεί πολύπλοκους υπολογισμούς, άρα αυξάνει τον χρόνο παραγωγής αποτελεσμάτων και απαιτεί περισσότερους πόρους από το σύστημα.

### 4.2.3 Μέθοδος Σφαιρικός νόμος συνημίτονων

Στην πραγματικότητα, οι περισσότεροι σύγχρονοι υπολογιστές και γλώσσες χρησιμοποιούν αριθμούς κινητής υποδιαστολής 64-bit «IEEE 754», οι οποίοι παρέχουν 15 σημαντικές τιμές ακρίβειας. Με αυτήν την ακρίβεια, ο απλός σφαιρικός νόμος των τύπων συνημίτονων ( $\cos c = \cos a \cos b + \sin a \sin b \cos C$ ) δίνει καλά ρυθμισμένα αποτελέσματα σε αποστάσεις τόσο μικρές όσο λίγα μέτρα στην επιφάνεια της γης. (Σημειώστε ότι η γεωδαιτική μορφή του νόμου των συνημίτονων αναδιατάσσεται από την κανονική, έτσι ώστε το γεωγραφικό πλάτος να μπορεί να χρησιμοποιηθεί άμεσα, αντί για τη συνάθροιση) [18].

Αυτό καθιστά τον απλούστερο νόμο των συνημίτων μια λογική εναλλακτική λύση 1 γραμμής από τον τύπο haversine για πολλούς σκοπούς γεωδαισίας (αν όχι για την αστρονομία). Η επιλογή μπορεί να καθοδηγείται από τη γλώσσα προγραμματισμού, τον επεξεργαστή, το πλαίσιο κωδικοποίησης, τις διαθέσιμες λειτουργίες τριγωνομετρίας (σε διαφορετικές γλώσσες) κ.λπ. - και, για πολύ μικρές αποστάσεις, μπορεί να είναι καταλληλότερη μια ισογώνια προσέγγιση. Ωστόσο αν και απλούστερος, ο νόμος των συνημίτονων είναι ελαφρώς πιο αργός από τον Haversine, στις δοκιμές της εργασίας.

```
Law of cosines: d = acos( sin φ1 * sin φ2 +
                        cos φ1 * cos φ2 * cos Δλ ) * R
```

JavaScript:

```
const φ1 = lat1 * Math.PI/180, φ2 = lat2 * Math.PI/180,
      Δλ = (lon2-lon1) * Math.PI/180, R = 6371e3;
const d = Math.acos( Math.sin(φ1)*Math.sin(φ2) +
                    Math.cos(φ1)*Math.cos(φ2) * Math.cos(Δλ) ) * R;
```

### 4.2.4 Ισογώνια προσέγγιση

Εάν η απόδοση είναι ένα ζήτημα και η ακρίβεια λιγότερο σημαντική, για μικρές αποστάσεις το θεώρημα του Πυθαγόρα μπορεί να χρησιμοποιηθεί σε μια ισόρροπη προβολή [7]:

Formula:

$$\begin{aligned}x &= \Delta\lambda * \cos \varphi_m \\y &= \Delta\varphi \\d &= R * \sqrt{x^2 + y^2}\end{aligned}$$

JavaScript:

```
const x = (λ2-λ1) * Math.cos((φ1+φ2)/2);
const y = (φ2-φ1);
const d = Math.sqrt(x*x + y*y) * R;
```

Αυτό χρησιμοποιεί μόνο μία συνάρτηση τριγωνομετρίας και μία τετραγώνων - σε αντίθεση με τις πολλαπλές συναρτήσεις που χρησιμοποιεί η haversine. Η ακρίβεια είναι κάπως περίπλοκη: στους μεσημβρινούς δεν υπάρχουν σφάλματα, διαφορετικά εξαρτώνται από την απόσταση, το υποστήριγμα άξονος και το πλάτος, αλλά είναι αρκετά μικρά για πολλούς σκοπούς \* (και συχνά ασήμαντα σε σύγκριση με την ίδια τη σφαιρική προσέγγιση).

Εναλλακτικά, μπορεί να χρησιμοποιηθεί ο τύπος πολικής συντεταγμένης επίπεδης γης: χρησιμοποιώντας τα συν γεωγραφικά πλάτη  $\theta_1 = \pi/2 - \varphi_1$  και  $\theta_2 = \pi/2 - \varphi_2$ , μετά  $d = R * \sqrt{\theta_1^2 + \theta_2^2 - 2 * \theta_1 * \theta_2 * \cos\Delta\lambda}$

#### 4.2.5 Βάση άξονα (αζιμούθιο)

Σε γενικές γραμμές, το τρέχον αρχικό σημείο θα διαφέρει καθώς ακολουθείτε μια μεγάλη διαδρομή κύκλου (ορθόδρομο). Το τελικό σημείο θα διαφέρει από το αρχικό με διαφορετικούς βαθμούς ανάλογα με την απόσταση και το γεωγραφικό πλάτος (αν επρόκειτο να μεταβείτε από 35 ° B, 45 ° A (≈ Βαγδάτη) έως 35 ° B, 135 ° A (≈ Οζάκα), θα ξεκινούσε με κατεύθυνση 60 ° και θα κατέληγε με κατεύθυνση 120 °).

Αυτός ο τύπος είναι για τον αρχικό άξονα (μερικές φορές αναφέρεται ως εμπρός αζιμούθιο [2]) το οποίο αν ακολουθηθεί σε ευθεία γραμμή κατά μήκος ενός τόξου μεγάλου κύκλου θα σας μεταφέρει από το σημείο εκκίνησης έως το τελικό σημείο.

Formula:  $\theta = \text{atan2}(\sin \Delta\lambda * \cos \varphi_2, \cos \varphi_1 * \sin \varphi_2 - \sin \varphi_1 * \cos \varphi_2 * \cos \Delta\lambda)$

where  $\varphi_1, \lambda_1$  is the start point,  $\varphi_2, \lambda_2$  the end point  
( $\Delta\lambda$  is the difference in longitude)

JavaScript:

```
(all angles in radians)
const y = Math.sin(λ2-λ1) * Math.cos(φ2);
const x = Math.cos(φ1)*Math.sin(φ2) -
          Math.sin(φ1)*Math.cos(φ2)*Math.cos(λ2-λ1);
const θ = Math.atan2(y, x);
const brng = (θ*180/Math.PI + 360) % 360; // in degrees
```

Δεδομένου ότι το atan2 επιστρέφει τιμές στο εύρος  $-\pi \dots + \pi$  (δηλαδή  $-180^\circ \dots + 180^\circ$ ), για να ομαλοποιήσει το αποτέλεσμα σε άξονα πυξίδας (στην περιοχή  $0^\circ \dots 360^\circ$ , με  $-ve$  τιμές μεταμορφώθηκαν στην περιοχή  $180^\circ \dots 360^\circ$ ), μετατρέπονται σε μοίρες και μετά χρησιμοποιήστε το  $(\theta + 360)\% 360$ , όπου το% είναι (κυμαινόμενο σημείο) modulo.

Για τον τελικό άξονα, απλώς πάρτε τον αρχικό άξονα, από το τελικό σημείο στο σημείο εκκίνησης και αντιστρέψτε το (χρησιμοποιώντας  $\theta = (\theta + 180)\% 360$ ).

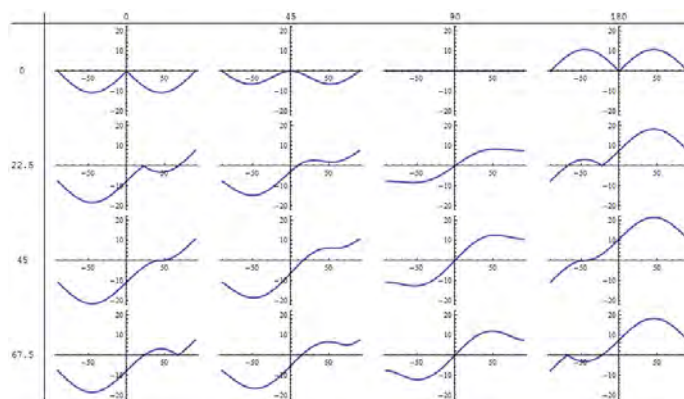
#### 4.2.6 Περιθώριο λάθους

Η απόσταση μπορεί να είναι λάθος έως περίπου 22km ή 0,3%, ανάλογα με τα εν λόγω σημεία.

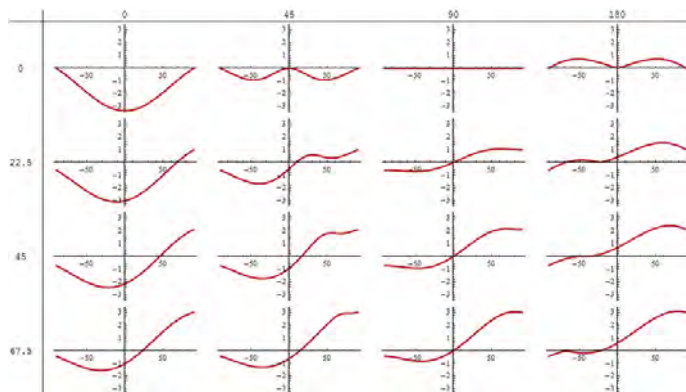
Το σφάλμα μπορεί να εκφραστεί με πολλούς φυσικούς, χρήσιμους τρόπους, όπως πρώτον ως (υπολειπόμενο) σφάλμα, ίσο με τη διαφορά μεταξύ των δύο υπολογισμένων αποστάσεων (σε χιλιόμετρα) και δεύτερον ως σχετικού σφάλματος, ίσο με τη διαφορά που διαιρείται με το "σωστή" (ελλειψοειδής) τιμή.

Τα σφάλματα εξαρτώνται από τα τελικά σημεία. Λόγω της περιστροφικής συμμετρίας του ελλειψοειδούς, της σφαίρας και των διμερών τους (βορρά-νότο και ανατολικά-δυτικά) συμμετριών, μπορεί να τοποθετηθεί ένα από τα τελικά σημεία κάπου κατά μήκος του πρωταρχικού μεσημβρινού (γεωγραφικό μήκος 0) στο βόρειο ημισφαίριο (γεωγραφικό πλάτος μεταξύ 0 και 90) και το άλλο τελικό σημείο στο ανατολικό ημισφαίριο (γεωγραφικό μήκος μεταξύ 0 και 180).

Τα μέγιστα μεγέθη τους είναι περίπου η ισοπέδωση (περίπου 1/300) επί τον κύριο άξονα (περίπου 6700 χλμ), ή περίπου 22 χλμ.



Σχήμα 4.1: Errors



Σχήμα 4.2: Relative errors

### 4.3 Εύρεση κοντινότερου σημείου ενδιαφέροντος

Ένα από τα σημαντικότερα κομμάτια της εργασίας αυτής, είναι η εύρεση του κοντινότερου σημείου ενδιαφέροντος για κάθε ακίνητο. Δηλαδή να μπορεί η εφαρμογή να υπολογίζει για κάθε ακίνητο το κοντινότερο σημείο ενδιαφέροντος ξεχωριστού τύπου. Στην βάση δεδομένων έχουν περαστεί πολλοί τύποι σημείων ενδιαφέροντος όπως σταθμοί μετρό, supermarkets, σχολεία, φαρμακεία, πάρκα και πολλά άλλα. Οπότε για κάθε ακίνητο θα πρέπει η εφαρμογή να βρίσκει το κοντινότερο κάθε τύπου σημείου ενδιαφέροντος και να το αποθηκεύει στη βάση ως στοιχείο του ακινήτου. Έτσι ώστε να λειτουργήσουν σαν φίλτρα αναζήτησης οι αποστάσεις αυτές.

Το κομμάτι αυτό αποτελεί ίσως το πιο σημαντικό κομμάτι της εφαρμογής καθώς είναι αυτό που μας επιτρέπει την χρήση των σημείων ενδιαφέροντος, από τον περιβάλλον χώρο τους, να γίνουν μέρος των χαρακτηριστικών τους έτσι ώστε ο χρήστης να μπορεί να αναζητά ακίνητα με καινούργια φίλτρα.

### 4.4 Εύρεση δεδομένων

Για τη δημιουργία την εκτέλεση καθώς και την δοκιμή της εφαρμογής, πρέπει να υπάρχουν δεδομένα στην βάση. Η εύρεση λοιπόν δεδομένων, όπως τοποθεσίες σε μορφή συντεταγμένων, χαρακτηριστικά των ακινήτων, όπως όροφος, τετραγωνικά κ.α, καθώς και τοποθεσίες και είδη σημείων ενδιαφέροντος, είναι μία πρόκληση. Δυστυχώς για την ακαδημαϊκή φύση της εργασίας δεν μπόρεσαν να βρεθούν υπαρκτά και αρκετά δεδομένα, οπότε η μόνη λύση ήταν η δημιουργία εικονικών δεδομένων (dummy data) [6]. Αυτό επιτεύχθηκε με την δυνατότητα που παρέχει το Laravel και την χρήση βοηθητικών εργαλείων αυτής (helpers) και βιβλιοθηκών, δηλαδή με τη δημιουργία factories και της βιβλιοθήκης faker [9] τα οποία θα αναλυθούν στο επόμενο κεφάλαιο.



## Κεφάλαιο 5

# Επίλυση απαιτήσεων εφαρμογής

### 5.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα αναλύσουμε πως αντιμετωπίστηκαν τα προβλήματα και οι απαιτήσεις που αναφέρθηκαν στο προηγούμενο κεφάλαιο. Χρησιμοποιώντας λοιπόν τα εργαλεία που παρέχει το Laravel, μαθηματικούς τύπους, αλγόριθμους αλλά και πολλά ακόμα εργαλεία.

### 5.2 Τρόπος υπολογισμού απόστασης μεταξύ συντεταγμένων

Η μέθοδος Harvesine κρίθηκε καταλληλότερη για αυτήν την εργασία λόγω την ταχύτητας παραγωγής αποτελεσμάτων και ακρίβειας στα περιθώρια των αποστάσεων που μελετά η συγκεκριμένη εργασία. Χρειάστηκε η μεταφορά της σε php για την χρήση της στο περιβάλλον του Laravel. Όπως παρατηρείται και στο σχήμα 5.1 δημιουργήθηκε μια συνάρτηση στο μοντέλο House όπου λαμβάνει σαν παραμέτρους ( $\$lat2, \$long2$ ) τις συντεταγμένες του σημείου ενδιαφέροντος, με το οποίο θέλουμε να υπολογίσουμε την μεταξύ τους απόσταση. Επειδή η συνάρτηση βρίσκεται στο μοντέλο του ίδιου του ακινήτου, δεν χρειάζεται να περαστούν και οι δικές του συντεταγμένες, καθώς υπάρχουν ήδη στο μοντέλο. Η έννοια του μοντέλου και τα υπόλοιπα χαρακτηριστικά και λειτουργίες της εφαρμογής, θα αναλυθούν εκτενέστερα σε παρακάτω κεφάλαιο.

Αξίζει να σημειωθεί ότι με την χρήση της μεθόδου Vincenty θα παράγονταν πιο ακριβή αποτελέσματα, όσο αναφορά την ακρίβεια των αποστάσεων, αλλά ο χρόνος παραγωγής τους θα ήταν αισθητά μεγαλύτερος. Ακόμα, η διαφορά στην ακρίβεια είναι πολύ μικρή σε τόσο κοντινές αποστάσεις άρα το βασικό πλεονέκτημα της Vincenty εξαφανίζεται.

### 5.3 Υπολογισμός κοντινότερου σημείου ενδιαφέροντος

Για τον υπολογισμό του κοντινότερου σημείου ενδιαφέροντος χρειάζεται να είναι γνωστή η απόσταση μεταξύ ακινήτου και σημείου ενδιαφέροντος. Οπότε όπως παρατηρείτε και στο σχήμα 5.2 δημιουργήθηκε μία συνάρτηση εντός του μοντέλου House (έτσι ώστε να έχουμε άμεση πρόσβαση στα δεδομένα του), λαμβάνοντας μία παράμετρο ( $\$aux$ ) η οποία εμπεριέχει όλα τα σημεία ενδιαφέροντος ενός τύπου π.χ όλα τα σχολεία. Εν συνεχεία δημιουργείται μια νέα συλλογή ( $\$stable$ )

```

House.php
51 public function getDistance($lat2,$long2)
52 {
53
54     $earthRadius = 6371000;
55     $lat1 = $this->latitude;
56     $long1 = $this->longitude;
57
58     $dlat = deg2rad( number: $lat2-$lat1);
59
60     $dlong = deg2rad( number: $long2-$long1);
61
62
63     $a = sin( arg: $dlat/2)*sin( arg: $dlat/2) + cos(deg2rad($lat1))*cos(deg2rad($lat2))*sin( arg: $dlong/2)*sin( arg: $dlong/2);
64
65     $c = 2*atan2(sqrt($a),sqrt( arg: 1-$a));
66
67     $d = $earthRadius*$c;
68
69     return $d;
70
71 }

```

Σχήμα 5.1: Haversine

και μεταφέρονται τα δεδομένα από την παράμετρο σε τοπική μεταβλητή ίδιου τύπου. Ακολουθεί μια επανάληψη τύπου foreach όπου με τη χρήση της συνάρτησης getDistance όπου δημιουργήθηκε πριν υπολογίζεται η απόσταση μεταξύ του ακινήτου και του σημείου ενδιαφέροντος και αποθηκεύεται σε μία μεταβλητή (\$d). Έπειτα αποθηκεύεται στη συλλογή \$table το id (μοναδικός αριθμός κάθε σημείου ενδιαφέροντος) και η απόσταση του από το ακίνητο. Τέλος η συλλογή \$table ταξινομείται ως προς την απόσταση και μεταφέρεται σε μία νέα μεταβλητή, έτσι ώστε να απομονωθούν οι αποστάσεις από τα id, και η συνάρτηση επιστρέφει το πρώτο στοιχείο, δηλαδή το κοντινότερο σημείο ενδιαφέροντος αυτού του τύπου (στην περίπτωση του παραδείγματος που τέθηκε νωρίτερα, το κοντινότερο σχολείο).

```

public function closest($aux)
{
    $table = new Collection;
    $collection = $aux;
    Foreach($collection as $object)
    {
        $d = $this->getDistance($object->latitude,$object->longitude);
        $table->push([$this->id,'object'=>$object->id,'dist'=>$d]);
    }

    $sort = $table->sortBy( callback: 'dist');

    return $sort->values()->first();
}

```

Σχήμα 5.2: Closest Poi

## 5.4 Δημιουργία εικονικών δεδομένων (dummy data)

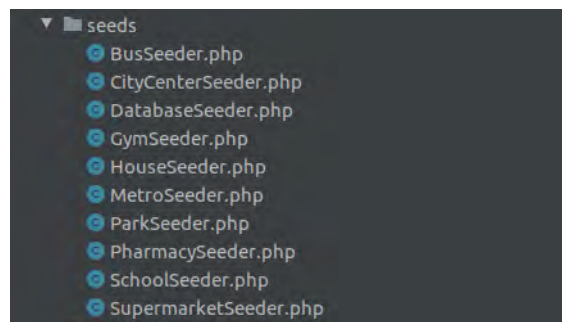
Η πλήρη δημιουργία και σχεδίαση της βάσης δεδομένων θα γίνει αναλυτικά σε παρακάτω κεφάλαιο, οπότε η συγκεκριμένη ενότητα αναλύει την δημιουργία των εικονικών δεδομένων. Με

τη βάση δεδομένων σχεδιασμένη αλλά άδεια, θα πρέπει πρώτα να δημιουργηθεί ένας seeder (τοφοδοτητής), έτσι ώστε να μπορεί να γεμίσει η βάση με δεδομένα. Οι seeders είναι ένας τρόπος εισαγωγής δεδομένων στη βάση μέσω προγραμματισμού και ένα πλεονέκτημα στη χρήση τους είναι ότι μπορεί να εισαχθούν γρήγορα εικονικά δεδομένα στην εφαρμογή. Αυτό επιτυγχάνεται εύκολα με την εντολή:

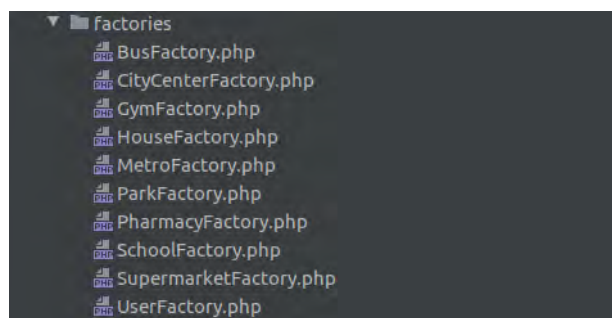
```
php artisan make:seeder (όνομα)Seeder
```

Υπόψιν, θα πρέπει να δημιουργηθεί ένας seeder για κάθε μοντέλο της βάσης βλπ σχήμα 5.3. Όπως προαναφέρθηκε στο κεφάλαιο 4 θα χρειαστεί ακόμα και η χρήση της λειτουργίας factories και της βιβλιοθήκης faker. Στη συνέχεια θα πρέπει να δημιουργηθούν factories για κάθε τύπο σημείου ενδιαφέροντος σχήμα 5.4, το οποίο επιτυγχάνεται ως εξής:

```
php artisan make:factory (όνομα)Factory
```



Σχήμα 5.3: Seeders



Σχήμα 5.4: factories

Έπειτα σε κάθε factory [8] ορίζονται τα εικονικά δεδομένα και τα χαρακτηριστικά που θα έχουν, όπως τη μορφή και τα όρια από τα οποία θα λαμβάνουν τιμές σχήμα 5.5. Όπως παρατηρείται η βιβλιοθήκη faker δίνει την δυνατότητα να παραχθούν τυχαία δεδομένα συγκεκριμένου τύπου (στο παράδειγμα συνταγμένες για στάσεις λεωφορείων) μέσα από όρια τιμών που μπορεί να εισάγει ο προγραμματιστής. Εν προκειμένου τα όρια είναι ένα τετράγωνο που αντιστοιχεί στην ευρύτερη περιοχή της Αττικής.

```
HouseFactory.php
1  <?php
2
3  /** @var \Illuminate\Database\Eloquent\Factory $factory */
4
5  use ...
6
7
8  $factory->define( class: House::class, function (Faker $faker) {
9      return [
10         'longitude' => $faker->longitude(23.62479, 23.923138),
11         'latitude' => $faker->latitude(38.098071, 37.861552),
12         'description' => $faker->realText($maxNbchars=50, $indexSize=2),
13         'bedrooms' => $faker->numberBetween(1, 8),
14         'size' => $faker->numberBetween(25, 400),
15         'floor' => $faker->numberBetween(0, 12),
16         'heat' => $faker->numberBetween(0, 3),
17         'year' => $faker->numberBetween(1960, 2020),
18         'newBuild' => $faker->boolean,
19         'price' => $faker->numberBetween(25000, 1000000),
20         'singleFloor' => $faker->boolean,
21         'hasElevator' => $faker->boolean,
22
23     ];
24
25 });
```

Σχήμα 5.5: Factory Example

Τέλος τα δεδομένα αυτά πρέπει να περάσουν στην βάση δεδομένων. Τη διαδικασία αυτή, όπως και τον αριθμό των στοιχείων που θα περαστούν, θα την αναλάβει ο seeder όπως παρατηρείται στο σχήμα 5.6.

```
HouseSeeder.php
1  <?php
2
3  use Illuminate\Database\Seeder;
4
5  class HouseSeeder extends Seeder
6  {
7      /**
8       * Run the database seeds.
9       *
10      * @return void
11      */
12      public function run()
13      {
14          factory(App\House::class, 1500)->create();
15      }
16  }
```

Σχήμα 5.6: Seeder Example

## Κεφάλαιο 6

# Παρουσίαση Web Εφαρμογής

Στο παρών κεφάλαιο θα παρουσιαστούν τα σενάρια χρήσης και τις δυνατότητες που προσφέρονται στον χρήστη. Αρχικά θα περιγραφούν οι σελίδες της εφαρμογής αλλά και οι αποδεκτές ενέργειες από τους χρήστες της. Επιπλέον, θα επισημανθούν ορισμένα σημαντικά σημεία κώδικα με στόχο να γίνουν κατανοητές συγκεκριμένες διεργασίες που εκτελούνται χωρίς να γίνονται αντιληπτές από τον χρήστη. Η εφαρμογή υλοποιήθηκε χρησιμοποιώντας την PHP στο back-end και τις HTML και CSS στο front-end κομμάτι της. Όλες οι σελίδες λειτουργούν σωστά σε οποιοδήποτε σύγχρονο browser. Σημαντικό είναι να σημειωθεί ότι ο σκοπός της εργασίας αυτής δεν είναι το front-end κομμάτι, αλλά η σχεδίαση και η χρήση νέων τρόπων αναζήτησης ακινήτων. Όποτε δημιουργήθηκε ένα απλό περιβάλλον front-end για την παρουσίαση και την χρήση των φίλτρων αναζήτησης.

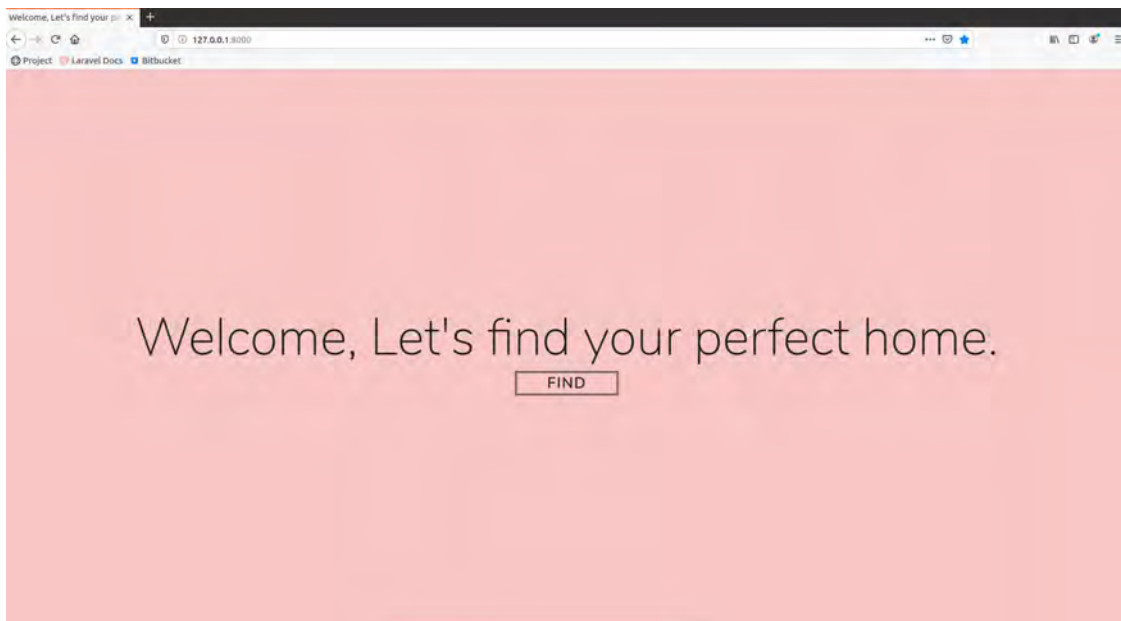
### 6.1 Αρχική Σελίδα

Η αρχική σελίδα της εφαρμογής (welcome page) λειτουργεί ως μια σελίδα υποδοχής για τον χρήστη, όπου ο σκοπός της είναι να τον μεταφέρει στη σελίδα αναζήτησης ακινήτων. Η αρχική σελίδα έχει λιτό σχεδιασμό και αποτελείται μόνο από ένα φόντο απαλού χρώματος, την επιγραφή Welcome, lets find your Perfect Home και ένα κουμπί ανακατεύθυνσης (FIND). Το κουμπί μεταφέρει τον χρήστη στη σελίδα αναζήτησης ακινήτων σχήμα 6.1.

### 6.2 Σελίδα Αναζήτησης

Στη σελίδα αναζήτησης ο χρήστης βρίσκει στη διάθεση του, όλα τα φίλτρα φίλτρα αναζήτησης, τα οποία είναι αστικοποιημένα σε τιμές ουδέτερες σχήμα 6.2. Επάνω παρατηρείται πως έχει τοποθετηθεί μπάρα πλοήγησης όπου εμπεριέχει στο αριστερό τμήμα της, κουμπί με την επιγραφή Perfect Home, το οποίο λειτουργεί ως ανακατεύθυνση στην αρχική σελίδα σχήμα 6.3. Ακριβώς κάτω από την μπάρα πλοήγησης υπάρχει μια επεξήγηση για τις επιλογές των φίλτρων τοποθεσίας όπου δείχνει στον χρήστη τα διαθέσιμα εύρη αναζήτησης.

Κάτω από την επιγραφή Location parameters βρίσκονται τα φίλτρα αναζήτησης ακινήτων βάση απόστασης από τα κοντινότερα σημεία ενδιαφέροντος, σχήμα 6.4. Κάθε φίλτρο τοποθεσίας



Σχήμα 6.1: Welcome page

διαθέτει τις εξής επιλογές, any αρχικοποιημένο για όταν τον χρήστη δεν τον ενδιαφέρει το συγκεκριμένο φίλτρο, Very Close όταν θέλει το συγκεκριμένο σημείο ενδιαφέροντος να βρίσκεται εντός 500 μέτρων, Close όταν βρίσκεται εντός 1000 μέτρων και In General Vicinity όταν βρίσκεται εντός 3000 μέτρων σχήμα 6.5.

Στη συνέχεια βρίσκονται κάποια από τα κλασσικά φίλτρα αναζήτησης ακινήτων, δηλαδή χαρακτηριστικά του ακινήτου, σχήμα 6.6. Ενδιαφέρον από θέμα υλοποίησης είναι η δυνατότητα επιλογής εύρους τιμών, για φίλτρα όπως το μέγεθος, σε τετραγωνικά μέτρα, του σπιτιού, το έτος κατασκευής, ο όροφος και τέλος η τιμή. Στο σχήμα 6.7 δίνονται οι επιλογές για το ελάχιστο του εύρους και στο σχήμα 6.8 οι αντίστοιχες για το μέγιστο. Αξίζει να σημειωθεί ότι αν ο χρήστης εισάγει τιμή ελάχιστου ορίου μεγαλύτερη από την τιμή του μεγίστου, η εφαρμογή θα ανταλλάξει τις τιμές των ορίων. Δηλαδή θα θέσει τη μεγαλύτερη τιμή για άνω όριο και την μικρότερη για κάτω αντίστοιχα. Τα φίλτρα αυτά εμπεριέχονται σε κάθε σελίδα αναζήτησης ακινήτων και γιαυτό τον λόγο επιλέχθηκε να εισαχθούν κάποια βασικά για να εμπλουτιστεί η αναζήτηση.

Τέλος στο κάτω δεξιά μέρος, υπάρχει το κουμπί Find, σχήμα 6.9. Σκοπός του είναι όταν ο χρήστης συμπληρώσει τα φίλτρα που χρειάζεται για την αναζήτηση του, παράδειγμα αναζήτησης σχήμα 6.10, με την χρήση του κουμπιού Find να μεταφερθεί στη σελίδα αποτελεσμάτων.

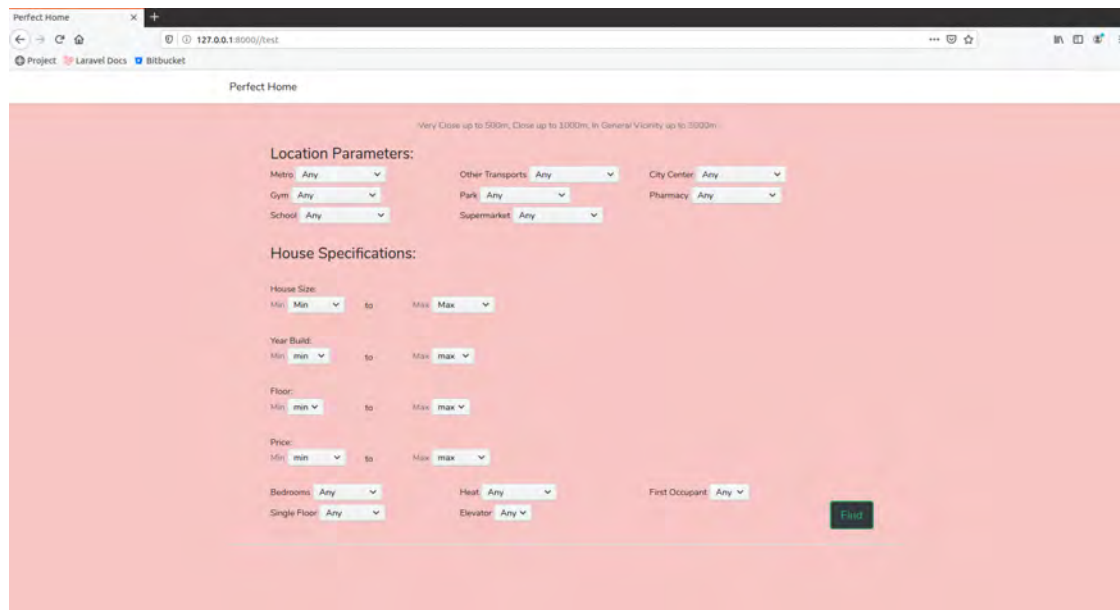
### 6.3 Σελίδα Αποτελεσμάτων

Στη σελίδα αποτελεσμάτων μεταφέρεται ο χρήστης από την σελίδα αναζήτησης. Η σελίδα ακολουθεί τον ίδιο σχεδιασμό με τη σελίδα αναζήτησης, όσο αναφορά τη μπάρα πλοήγησης στο πάνω μέρος και το χρώμα του φόντου, σχήμα 6.11. Σε περίπτωση που η αναζήτηση του χρήστη δεν επιφέρει κανένα αποτέλεσμα, δηλαδή δεν υπάρχει κανένα ακίνητο όπου να πληρεί όλα τα φίλτρα αναζήτησης, του επιστρέφεται το μήνυμα "Sadly, we currently have no House matching your criteria. Try to alter some or just broaden your search range. Thank you" (Δυστυχώς αυτή τη στιγμή δεν διαθέτουμε Σπίτια που να πληρούν τα κριτήρια αναζήτησης σας, δοκιμάστε να αλλάξετε κάποια ή να μεγαλώσετε τα εύρη αναζήτησης. Ευχαριστούμε.), σχήμα 6.12.

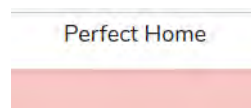
Όταν η αναζήτηση επιστρέφει με ακίνητα που ταιριάζουν στα κριτήρια του χρήστη, αυτά παρουσιάζονται στη σελίδα, σχήμα 6.13. Το κάθε ακίνητο εμπεριέχεται σε ξεχωριστή καρτέλα, σχήμα 6.14, και στο πάνω μέρος εμφανίζεται το μήνυμα "Houses that Perfectly fits your needs" (Σπίτια που ταιριάζουν τέλεια στις ανάγκες σας).

Τέλος ο χρήστης μπορεί να πατήσει το κουμπί Perfect Home, σχήμα 6.3, η το κουμπί New Search στο κάτω μέρος της σελίδας, για να οδηγηθεί πίσω στη σελίδα αναζήτησης, σχήμα 6.15.

Στο παράρτημα I βρίσκεται ο κώδικας HTML και CSS για τις σελίδες αυτού του κεφαλαίου.



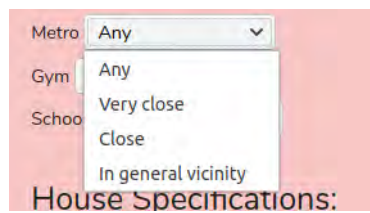
Σχήμα 6.2: Search Page



Σχήμα 6.3: Home Button



Σχήμα 6.4: Location filters



Σχήμα 6.5: Location Filter options



House Specifications:

House Size:  
Min  to Max

Year Build:  
Min  to Max

Floor:  
Min  to Max

Price:  
Min  to Max

Bedrooms  Heat  First Occupant

Single Floor  Elevator

Σχήμα 6.6: House specifications

House Size:  
Min  to Max

Year Build:  
Min  to Max

Floor:  
Min  to Max

Price:  
Min  to Max

Bedrooms  Heat  First Occupant

Single Floor  Elevator

Min   
100sqrm  
150sqrm  
200sqrm

Σχήμα 6.7: Size options min

House Size:  
Min  to Max

Year Build:  
Min  to Max

Floor:  
Min  to Max

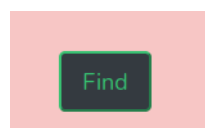
Price:  
Min  to Max

Bedrooms  Heat  First Occupant

Single Floor  Elevator

Max   
150sqrm  
200sqrm  
250sqrm  
250+sqrm

Σχήμα 6.8: Size options max



Σχήμα 6.9: Find Button

Perfect Home

Very Close up to 500m, Close up to 1000m, In General Vicinity up to 3000m

**Location Parameters:**

Metro:  Other Transports:  City Center:

Gym:  Park:  Pharmacy:

School:  Supermarket:

**House Specifications:**

House Size: Min:  to Max:

Year Build: Min:  to Max:

Floor: Min:  to Max:

Price: Min:  to Max:

Bedrooms:  Heat:  First Occupant:

Single Floor:  Elevator:

Σχήμα 6.10: Search example

Perfect Home

Houses that Perfectly fit your needs

Floor: 0	Floor: 1
Size: 66	Size: 153
Bedrooms: 7	Bedrooms: 4
Price: 950683	Price: 125016
Year Built: 1970	Year Built: 1997
Heat: Other First Occupant: Not First Single Floor: Maisonette Elevator: Has elevator Position: 22.7047533, 93.6766	Heat: Other First Occupant: Not First Single Floor: Maisonette Elevator: Has elevator Position: 23.70110137, 93.6991
Points of interest info: Distance to metro: 4109 Distance to other transports: 203 Distance to City Center: 1341 Distance to Park: 331 Distance to Pharmacy: 696 Distance to School: 674 Distance to Supermarket: 3814	Points of interest info: Distance to metro: 2698 Distance to other transports: 198 Distance to City Center: 643 Distance to Park: 152 Distance to Pharmacy: 606 Distance to School: 214 Distance to Supermarket: 833

Σχήμα 6.11: Results Page

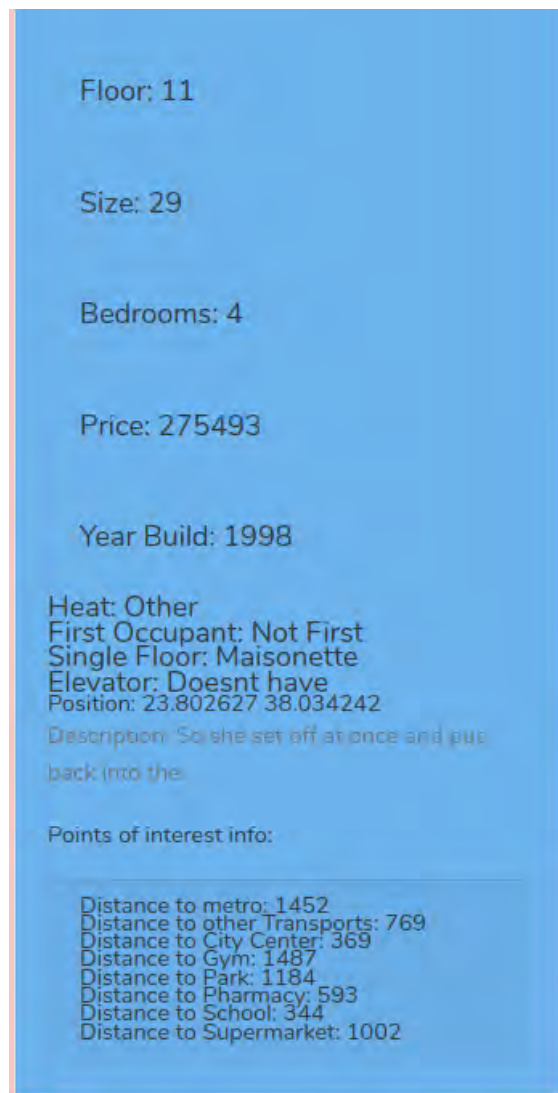
Sadly, we currently have no House matching your criteria. Try to alter some or just broaden your search range. Thank you

Σχήμα 6.12: Unsuccessful Search

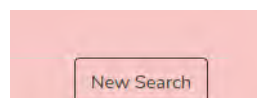
Houses that Perfectly Fits your needs

<b>Floor:</b> 8	<b>Floor:</b> 1
<b>Size:</b> 66	<b>Size:</b> 153
<b>Bedrooms:</b> 7	<b>Bedrooms:</b> 4
<b>Price:</b> 950683	<b>Price:</b> 125016
<b>Year Build:</b> 1970	<b>Year Build:</b> 1997
<b>Heat:</b> Other <b>First Occupant:</b> Not First <b>Single Floor:</b> Maisonette <b>Elevator:</b> Has elevator <b>Position:</b> 23.70479 37.936766 <b>Description:</b> Απολύτως ασφαλές και σύγχρονο διαμέρισμα <b>Conditions:</b> εστιασμένο	<b>Heat:</b> Other <b>First Occupant:</b> Not First <b>Single Floor:</b> Maisonette <b>Elevator:</b> Has elevator <b>Position:</b> 23.701101 37.903691 <b>Description:</b> Βολωμένο 1 WAFS διαμέρισμα <b>Conditions:</b> γενικό
<b>Points of interest info:</b> Distance to metro: 4109 Distance to other Transport: 263 Distance to City Center: 1342 Distance to Gym: 351 Distance to Park: 577 Distance to Pharmacy: 866 Distance to School: 813 Distance to Supermarket: 1614	<b>Points of interest info:</b> Distance to metro: 2698 Distance to other Transport: 106 Distance to City Center: 542 Distance to Gym: 134 Distance to Park: 356 Distance to Pharmacy: 606 Distance to School: 393 Distance to Supermarket: 830

Σχήμα 6.13: Successful Search



Σχήμα 6.14: House Card



Σχήμα 6.15: New Search button

## Κεφάλαιο 7

# Τεχνικές λεπτομέρειες

Στο κεφάλαιο αυτό θα αναλυθούν τεχνικές λεπτομέρειες που έχουν αναφερθεί στην εργασία, καθώς και διάφορες ακόμη που αξίζει να αναφερθούν. Θα αναλυθεί τόσο η σημασία τους όσο και ο τρόπος υλοποίησής τους.

### 7.1 Λεπτομέρειες υλοποίησης

Για την εφαρμογή χρησιμοποιήθηκε το framework της laravel το οποίο προσφέρει αρκετές δυνατότητες όσο στην διαχείριση της εφαρμογής αλλά και της βάσης δεδομένων της. Αναλυτικότερα προσφέρει μια ικανοποιητική υλοποίηση του MVC (Model View Controller) μοντέλου το οποίο είναι μια βασική ιεραρχία υλοποίησης ενός project. Για την βάση δεδομένων προσφέρει τις διαδικασίες των migrations και των seeders οι οποίες αφορούν την δημιουργία του σκελετού της βάσης αλλά και την αρχικοποίηση δεδομένων σε αυτή. Τέλος δίνει την δυνατότητα να συνδεθεί το back-end με το front-end απολύτως λειτουργικά, παρέχοντας παράλληλα πολλές δυνατότητες και εργαλεία για την βέλτιστη συνεργασία.

#### 7.1.1 MVC μοντέλο

Το MVC [11] είναι μια μέθοδος αρχιτεκτονικής λογισμικού η οποία προτείνεται για την υλοποίηση περιβαλλόντων αλληλεπίδρασης με τον χρήστη [15]. Η βασική λογική του μοντέλου είναι να διαχωριστούν οι διαδικασίες προβολής, διαχείρισης και αποθήκευσης δεδομένων για την μετάδοση της πληροφορίας στο χρήστη. Το Model αφορά την διαχείριση της βάσης δεδομένων είτε αυτό αφορά στην αποθήκευση τους σε αυτή είτε την ανάκτηση τους από αυτή. Το View αφορά την παρουσίαση της πληροφορίας στον χρήστη δηλαδή αναφερόμαστε το γραφικό περιβάλλον εμφάνισης της εφαρμογής. Τέλος ο Controller διαχειρίζεται όλη την λογική της εφαρμογής και των δεδομένων της, δηλαδή λαμβάνει τα δεδομένα από το Model τα επεξεργάζεται και τα προβάλλει μέσα από το View στον χρήστη ο οποίος αλληλεπιδρά με την εφαρμογή και επαναφέρει στον Controller δεδομένα και εντολές ώστε να τις διαχειριστεί. Αυτό μας βοηθά καθώς οι διαδικασίες χωρίζονται σε διαφορετικά αρχεία συγκεκριμένου τομέα και λογικής ώστε να διαχειρίζονται με πιο εύκολο τρόπο.

Στο σχήμα 7.1 παρατηρείται ο controller όπου είναι υπεύθυνος για την σελίδα αναζήτησης. Λαμβάνει παράμετρο τύπου αιτήματος, όπου περιέχει μέσα το query που έχει δημιουργεί από την αναζήτηση του χρήστη, δηλαδή ποία φίλτρα έχει χρησιμοποιήσει. Στο σχήμα 7.1 ακόμα παρατηρείται αρχικά οτι στη μεταβλητή \$search τοποθετούνται όλα τα ακίνητα, έτσι ώστε να παραμείνουν όσα ταιριάζουν στην αναζήτηση του χρήστη, αφού ελεγχθούν οι παράμετροι των φίλτρων. Αυτό επιτυγχάνεται με συνθήκες if η οποίες ελέγχουν το query και σύμφωνα με τα στοιχεία που έχει κρατάνε μόνο αυτά που ταιριάζουν στα φίλτρα. Εδώ φαίνεται για το φίλτρο απόστασης Gym, ομοίως γίνεται για κάθε φίλτρο.

Στο σχήμα 7.2 φαίνεται ένα διαφορετικού τύπου φίλτρο, επειδή διαθέτει τις επιλογές ορίων min,max. Εδώ ομοίως υπάρχει μια συνθήκη if η οποία ελέγχει το query, δηλαδή αν ο χρήστης έχει χρησιμοποιήσει το συγκεκριμένο φίλτρο και τι τιμές έχει βάλει. Μετά τον έλεγχο διαλέγονται τα στοιχεία που πληρούν τις προϋποθέσεις και η περίπτωση που ο χρήστης βάλει χαμηλό όριο μεγαλύτερο του υψηλού ή το αντίστροφο, η εφαρμογή το αναγνωρίζει και ανταλλάσσει τις τιμές των ορίων.

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\House;
6 use Illuminate\Http\Request;
7
8 class SearchController extends Controller
9 {
10     public function search(Request $request)
11     {
12         $search= House::All();
13
14         if($request->filled( key: 'gym')) {
15             if($request->gym == 'Vc'){
16                 $search= $search->whereBetween( key: 'gym0', [1,500]);
17             }
18             elseif ($request->gym == 'C') {
19                 $search= $search->whereBetween( key: 'gym0', [1,1000]);
20             }
21             elseif ($request->gym == 'G') {
22                 $search= $search->whereBetween( key: 'gym0', [1,3000]);
23             }
24         }
25         else
26         {
27         }
28     }
29 }

```

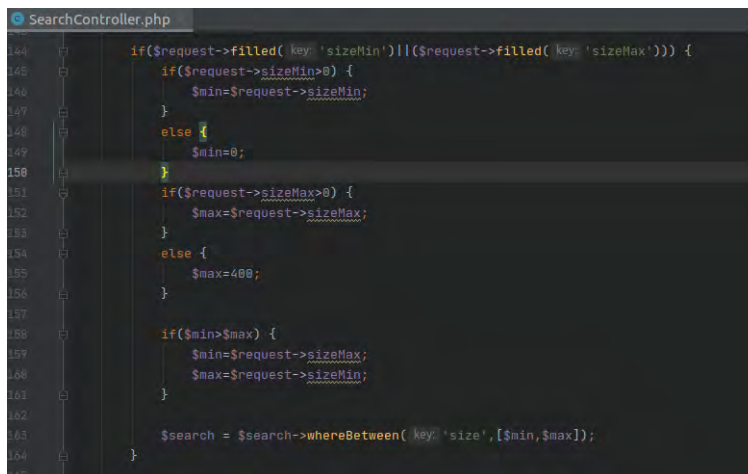
Σχήμα 7.1: Search controller for distance filter

### 7.1.2 Migrations και σχεδιασμός της Βάσης

Τα migrations δημιουργούν τον σκελετό της βάσης δεδομένων, αναλυτικά στο αρχείο ενός migration καθορίζουμε τις μεταβλητές ενός πίνακα, τον τύπο τους, την μοναδικότητα τους, default (προκαθορισμένη) τιμή εάν το επιθυμούμε ακόμα και τις διασυνδέσεις με άλλους πίνακες (foreign keys).

Αρχικά πρέπει να δημιουργηθούν τα migrations για κάθε πίνακα (table) στοιχείων που πρέπει να υπάρχουν στη βάση, σχήμα 7.3. Αυτό επιτυγχάνεται με την εντολή:

```
php artisan make:migration (όνομα)_table
```



```
144     if($request->filled( key: 'sizeMin') || ($request->filled( key: 'sizeMax'))) {
145         if($request->sizeMin > 0) {
146             $min = $request->sizeMin;
147         }
148         else {
149             $min = 0;
150         }
151         if($request->sizeMax > 0) {
152             $max = $request->sizeMax;
153         }
154         else {
155             $max = 400;
156         }
157
158         if($min > $max) {
159             $min = $request->sizeMax;
160             $max = $request->sizeMin;
161         }
162
163         $search = $search->whereBetween( key: 'size', [$min, $max]);
164     }
165 }
```

Σχήμα 7.2: Search controller for Range filter

Για να οριστικοποιηθούν οι πίνακες στη βάση χρειάζεται να τρέξει μια εντολή και όλη η βάση θα δημιουργηθεί εξολοκλήρου:

```
php artisan migrate
```

Ένα παράδειγμα στο σχήμα 7.4.

Κατά την δημιουργία ενός migration δημιουργείται αυτόματα και μια εγγραφή στον πίνακα migrations της βάσης δεδομένων η οποία χρειάζεται ώστε να διατηρείται αρχείο και να γνωρίζει το πρόγραμμα ποια είναι η σειρά με την οποία έχουν δημιουργηθεί τα migrations . Επιπλέον υπάρχει η δυνατότητα αναιρέσης της δημιουργίας σε περίπτωση οποιουδήποτε προβλήματος η αλλαγών σε αρχικό στάδιο. Για την αναίρεση χρησιμοποιείται η εντολή:

```
php artisan migrate:rollback
```

Τέλος για την ολοκλήρωση της βάσης θα πρέπει να δημιουργηθούν τα μοντέλα για πίνακα, σχήμα 7.5 και οι σχέσεις μεταξύ τους. Τα μοντέλα ομοίως δημιουργούνται με την εντολή:

```
php artisan make:model (όνομα)
```

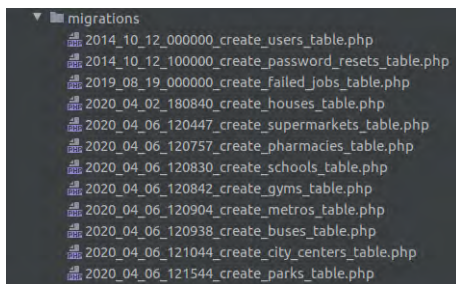
Οι σχέσεις κάθε μοντέλου πχ one-to-many, many-to-many ή many-to-one, δηλώνονται μέσα στο κάθε μοντέλο ξεχωριστά. Στην συγκεκριμένη εργασία υπάρχουν οι σχέσεις του ακινήτου με τα σημεία ενδιαφέροντος, όπου είναι της μορφής one-to-many απο πλευράς σημείων ενδιαφέροντος, καθώς ένα σημείο ενδιαφέροντος μπορεί να βρίσκεται κοντά σε πολλά ακίνητα, ενώ από την πλευρά των ακινήτων είναι το αντίστροφο, κάθε ακίνητο έχει μόνο ένα σημείο ενδιαφέροντος ξεχωριστού τύπου πιο κοντά του.

Τέλος για να γεμίσει η βάση με τα εικονικά δεδομένα πρέπει να τρέξει η εντολή Η τελική μορφή της βάσης βρίσκεται στο σχήμα 7.6

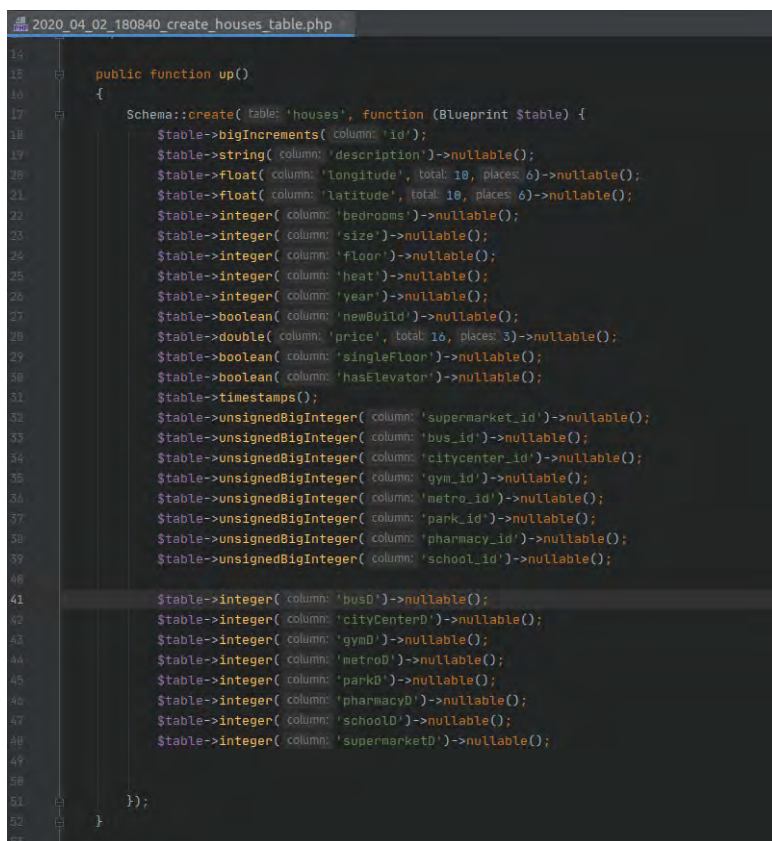
### 7.1.3 Routing

Η δρομολόγηση (routing) για τα url που θα λάβουν οι σελίδες αλλά και την σύνδεση μεταξύ τους, γίνεται στο web.php σχήμα. Εδώ συνδέονται τα url που θα χρησιμοποιεί η εφαρμογή με

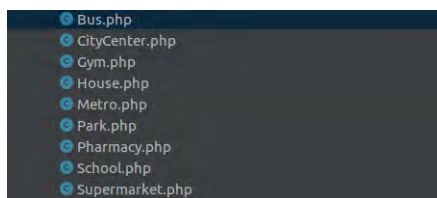




Σχήμα 7.3: Migrations



Σχήμα 7.4: Migration table example



Σχήμα 7.5: Models

την αντίστοιχη λειτουργία που θα ανατεθεί. Συγκεκριμένα στο σχήμα 7.7 παρατηρείται πως το `url/test` είναι συνδεδεμένο με μια συνάρτηση `index` του `house controller`. Στη συνάρτηση αυτή απλά επιστρέφει το `view` (εικόνα) των φίλτρων αναζήτησης.



Table Name	Fields and Data Types
houses	id (bigint unsigned), description (varchar(255)), longitude (double(10,6)), latitude (double(10,6)), bedrooms (int), size (int), floor (int), heat (int), year (int), newBuild (tinyint(1)), price (double(16,3)), singleFloor (tinyint(1)), hasElevator (tinyint(1)), created_at (timestamp), updated_at (timestamp), busD (int), cityCenterD (int), gymD (int), metroD (int), parkD (int), pharmacyD (int), schoolD (int), supermarketD (int)
users	id (bigint unsigned), name (varchar(255)), email (varchar(255)), email_verified_at (timestamp), password (varchar(255)), remember_token (varchar(100)), created_at (timestamp), updated_at (timestamp)
failed_jobs	id (bigint unsigned), connection (text), queue (text), payload (longtext), exception (longtext), failed_at (timestamp)
buses	id (bigint unsigned), longitude (double(10,6)), latitude (double(10,6)), created_at (timestamp), updated_at (timestamp)
schools	id (bigint unsigned), longitude (double(10,6)), latitude (double(10,6)), created_at (timestamp), updated_at (timestamp)
pharmacies	id (bigint unsigned), longitude (double(10,6)), latitude (double(10,6)), created_at (timestamp), updated_at (timestamp)
supermarkets	id (bigint unsigned), longitude (double(10,6)), latitude (double(10,6)), created_at (timestamp), updated_at (timestamp)
parks	id (bigint unsigned), longitude (double(10,6)), latitude (double(10,6)), created_at (timestamp), updated_at (timestamp)
metros	id (bigint unsigned), longitude (double(10,6)), latitude (double(10,6)), created_at (timestamp), updated_at (timestamp)
city_centers	id (bigint unsigned), longitude (double(10,6)), latitude (double(10,6)), created_at (timestamp), updated_at (timestamp)
gyms	id (bigint unsigned), longitude (double(10,6)), latitude (double(10,6)), created_at (timestamp), updated_at (timestamp)
migrations	id (int unsigned), migration (varchar(255)), batch (int)
password_resets	email (varchar(255)), token (varchar(255)), created_at (timestamp)

Σχήμα 7.6: Data Base

```

1 web.php
2 ~?php
3
4 Route::get('/', function () {
5     return view('welcome');
6 });
7
8 // Auth: routes(['register' => false]);
9
10 Route::get('/home', 'HomeController@index')->name('home');
11
12 Route::get('/test', 'HomeController@index');
13
14 Route::get('/init', 'HomeController@initialize');
15
16 Route::post('/src', 'SearchController@search')->name('search');
17
18 Route::get('/results', 'HomeController@view')->name('results');

```

Σχήμα 7.7: Routing

### 7.1.4 Φόρμα HTML

Μια φόρμα HTML χρησιμοποιείται για τη συλλογή εισόδου χρήστη. Η είσοδος χρήστη μπορεί στη συνέχεια να σταλεί σε διακομιστή για επεξεργασία. Στην εργασία αυτή για τη δημιουργία της σελίδας αναζήτησης χρησιμοποιήθηκε φόρμα μέσω του Laravel, έτσι ώστε να μπορεί ο χρήστης να επιλέξει τα φίλτρα αναζήτησης που επιθυμεί. Στα σχήματα 7.8 7.9 7.10 7.11 υπάρχει η φόρμα που χρησιμοποιήθηκε.

```

{!! Form::open(['route'=>'search'])!!}
<div>
  <strong id="locationSpecs" class="text">Location Parameters:</strong>
</div>
<div class="row">
  <div class="col-4">
    {{Form::label('metro','Metro')}}
    {{Form::select('metro',['null'=>'Any','V'=>'Very close','C'=>'Close','G'=>'In general vicinity'])}}
  </div>
  <div class="col-4">
    {{Form::label('bus','Other transports')}}
    {{Form::select('bus',['null'=>'Any','V'=>'Very close','C'=>'Close','G'=>'In general vicinity'])}}
  </div>
  <div class="col-4">
    {{Form::label('cityCenter','City center')}}
    {{Form::select('cityCenter',['null'=>'Any','V'=>'Very close','C'=>'Close','G'=>'In general vicinity'])}}
  </div>
  <div class="col-4">
    {{Form::label('gym','Gym')}}
    {{Form::select('gym',['null'=>'Any','V'=>'Very close','C'=>'Close','G'=>'In general vicinity'])}}
  </div>
  <div class="col-4">
    {{Form::label('park','Park')}}
    {{Form::select('park',['null'=>'Any','V'=>'Very close','C'=>'Close','G'=>'In general vicinity'])}}
  </div>
  <div class="col-4">
    {{Form::label('pharmacy','Pharmacy')}}
    {{Form::select('pharmacy',['null'=>'Any','V'=>'Very close','C'=>'Close','G'=>'In general vicinity'])}}
  </div>
  <div class="col-4">
    {{Form::label('school','School')}}
    {{Form::select('school',['null'=>'Any','V'=>'Very close','C'=>'Close','G'=>'In general vicinity'])}}
  </div>
  <div class="col-4">
    {{Form::label('supermarket','Supermarket')}}
    {{Form::select('supermarket',['null'=>'Any','V'=>'Very close','C'=>'Close','G'=>'In general vicinity'])}}
  </div>
</div>

```

Σχήμα 7.8: Form part1

```

<div>
  <strong id="houseSpecifications" class="text">House Specifications:</strong>
</div>
<div>
  <br>
  <strong id="houseSize" class="form-text">House Size:</strong>
</div>
<div class="row">
  <div class="col-2">
    {{Form::label('sizeMin','Min')}}
    {{Form::select('sizeMin',['null'=>'Min','50'=>'50sqm','100'=>'100sqm','150'=>'150sqm','200'=>'200sqm'])}}
  </div>
  <div class="col-1">
    <strong id="to" class="form-text">to</strong>
  </div>
  <div class="col-3">
    {{Form::label('sizeMax','Max')}}
    {{Form::select('sizeMax',['null'=>'Max','100'=>'100sqm','150'=>'150sqm','200'=>'200sqm','250'=>'250sqm','300'=>'300sqm'])}}
  </div>
</div>
<div>
  <br>
  <strong id="yearBuild" class="form-text">Year Build:</strong>
</div>
<div class="row">
  <div class="col-2">
    {{Form::label('yearMin','Min')}}
    {{Form::select('yearMin',['null'=>'Any','1960'=>'1960','1970'=>'1970','1980'=>'1980','1990'=>'1990','2000'=>'2000','2010'=>'2010'])}}
  </div>
  <div class="col-1">
    <strong id="to" class="form-text">to</strong>
  </div>
  <div class="col-2">
    {{Form::label('yearMax','Max')}}
    {{Form::select('yearMax',['null'=>'Any','1970'=>'1970','1980'=>'1980','1990'=>'1990','2000'=>'2000','2010'=>'2010','2015'=>'2015'])}}
  </div>
</div>
<div>
  <br>
  <strong id="floor" class="form-text">Floor:</strong>
</div>

```

Σχήμα 7.9: Form part2

### 7.1.5 Αρχικοποίηση αποστάσεων στη βάση

Όταν η βάση δεδομένων γεμίσει, για να μπορέσει να λειτουργεί ομαλά η εφαρμογή και όλα τα φίλτρα αναζήτησης, θα πρέπει να υπολογισθούν οι αποστάσεις μεταξύ κάθε ακινήτου και σημείου ενδιαφέροντος. Ο τρόπος αναλύθηκε σε προηγούμενο κεφάλαιο. Οι αποστάσεις αυτές αποθηκεύ-

```

<div class="row">
  <div class="col-2">
    {{Form::label('floormin','min')}}
    {{Form::select('floormin',[null=>'min',0=>'0',1=>'1',2=>'2',3=>'3',4=>'4',5=>'5'])}}
  </div>
  <div class="col-1">
    <strong id="to" class="form-text">to</strong>
  </div>
  <div class="col-2">
    {{Form::label('floormax','max')}}
    {{Form::select('floormax',[null=>'max',1=>'1',2=>'2',3=>'3',4=>'4',5=>'5'])}}
  </div>
</div>
<div>
  <br>
  <strong id="price" class="form-text">Price:</strong>
</div>
<div class="row">
  <div class="col-2">
    {{Form::label('pricemin','min')}}
    {{Form::select('pricemin',[null=>'min',25000=>'25000',50000=>'50000',75000=>'75000',100000=>'100000',150000=>'150000',200000=>'200000'])}}
  </div>
  <div class="col-1">
    <strong id="to" class="form-text">to</strong>
  </div>
  <div class="col-3">
    {{Form::label('pricemax','max')}}
    {{Form::select('pricemax',[null=>'max',25000=>'25000',50000=>'50000',75000=>'75000',100000=>'100000',150000=>'150000',200000=>'200000'])}}
  </div>
</div>
<br>
<div class="row">
  <div class="col-6">
    {{Form::label('bedrooms','bedrooms')}}
    {{Form::select('bedrooms',[null=>'Any',1=>'1',2=>'2',3=>'3',4=>'4',5=>'5',6=>'More than 6'])}}
  </div>
  <div class="col-4">
    {{Form::label('heat','heat')}}
    {{Form::select('heat',[null=>'Any',CP=>'Central Petrol',CB=>'Central Gas',PG=>'Private Gas',O=>'Other'])}}
  </div>
</div>

```

Σχήμα 7.10: Form part3

```

<div class="row">
  <div class="col-4">
    {{Form::label('bedrooms','bedrooms')}}
    {{Form::select('bedrooms',[null=>'Any',1=>'1',2=>'2',3=>'3',4=>'4',5=>'5',6=>'More than 6'])}}
  </div>
  <div class="col-4">
    {{Form::label('heat','heat')}}
    {{Form::select('heat',[null=>'Any',CP=>'Central Petrol',CB=>'Central Gas',PG=>'Private Gas',O=>'Other'])}}
  </div>
  <div class="col-4">
    {{Form::label('firstoccupant','first occupant')}}
    {{Form::select('firstoccupant',[null=>'Any','first=>'First'])}}
  </div>
  <div class="col-4">
    {{Form::label('singlefloor','single floor')}}
    {{Form::select('singlefloor',[null=>'Any',S=>'Single',H=>'Halkonetta'])}}
  </div>
  <div class="col-4">
    {{Form::label('elevator','elevator')}}
    {{Form::select('elevator',[null=>'Any',1=>'has'])}}
  </div>
</div>
</div>
<div class="at-auto">
  <button class="btn-lg btn-dark btn-outline-success" type="submit">Find</button>
</div>
</div>
{{Form::close()}}

```

Σχήμα 7.11: Form part4

ονται στον πίνακα των ακινήτων. Δηλαδή κάθε ακίνητο έχει ξεχωριστό κελί για κάθε τύπο σημείου ενδιαφέροντος, και εκεί αποθηκεύεται η απόσταση του από το κοντινότερο. Αυτό πρέπει να γίνει κάθε φορά που εισάγονται νέα ακίνητα ή νέα σημεία ενδιαφέροντος στη βάση, και επιτυγχάνεται με τη συνάρτηση `initialization` που βρίσκεται στον `houseController`. Για μελλοντική χρήση της εφαρμογής, έχουν δημιουργηθεί `events` και `listeners`, έτσι ώστε όποτε εισάγεται νέα δεδομένα στη βάση, μέσω αυτών να μπορεί να τρέχει το `initialization`. Αυτή τη στιγμή γίνεται χειροκίνητα μέσω του `url:/init` της εφαρμογής.

## 7.2 Εγκατάσταση Εφαρμογής

Για την εγκατάσταση της εφαρμογής σε περιβάλλον Ubuntu χρειάζεται ο κώδικας του project, μια βάση δεδομένων, ένας apache server και τις κατάλληλες ρυθμίσεις για τις διασυνδέσεις τους, καθώς και το laravel και η γλώσσα Php.

Αρχικά χρειάζεται να ανοιχθεί ένα τερματικό του περιβάλλοντος linux και εκεί να τρέξουν οι εξής εντολές:

```
sudo apt update
sudo apt install apache2
```

Για την εγκατάσταση του σέρβερ.

```
sudo apt install php7.2 libapache2-mod-php7.2
php7.2-mbstring php7.2-xmlrpc
php7.2-soap php7.2-gd php7.2-xml php7.2-cli php7.2-zip
```

Για την εγκατάσταση της php.

```
curl -sS https://getcomposer.org/installer | sudo php --
--install-dir=/usr/local/bin --filename=composer
```

Για την εγκατάσταση του composer.

```
sudo nano /etc/apache2/sites-available/laravel.conf
```

Έπειτα αντιγραφή και επικόλληση το παρακάτω μέσα στο αρχείο

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    DocumentRoot /var/www/html/MyProject/public
    ServerName example.com

    <Directory /var/www/html/MyProject/public>
        Options +FollowSymlinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Αποθήκευση και έξοδος.

Για την παραμετροποίηση του αρχείου διαχείρισης του apache site για το laravel. Το αρχείο αυτό θα ελέγχει πώς οι χρήστες θα βλέπουν και θα τροποποιούν το περιεχόμενο.

```
sudo a2ensite laravel.conf
sudo a2enmod rewrite
```

Για να επιτραπεί το laravel και το module rewrite. Τέλος γίνεται επανακίνηση του Σέρβερ με την εντολή:

```
sudo systemctl restart apache2.service
```

Η πρόσβαση γίνεται μέσω browser στη σελίδα localhost του συστήματος, όπου θα ανοίξει η αρχική σελίδα του laravel.



Σχήμα 7.12: Laravel

Στη συνέχεια πρέπει να εγκατασταθεί ο mySQL server και να δημιουργηθεί μια βάση. Αυτό επιτυγχάνεται με τα εξής βήματα:

```
sudo apt-get update
sudo apt-get install mysql-server
```

Αφού ολοκληρωθεί η εγκατάσταση, μπορείτε να ξεκινήσετε την υπηρεσία βάσης δεδομένων εκτελώντας την ακόλουθη εντολή:

```
sudo systemctl start mysql
```

Για να βεβαιωθείτε ότι ο διακομιστής βάσης δεδομένων ξεκινά μετά την επανεκκίνηση, εκτελέστε την ακόλουθη εντολή:

```
sudo systemctl enable mysql
```

Επανεκκίνηση mysql

```
sudo systemctl restart mysql
```

Στη γραμμή εντολών, εκτελέστε την ακόλουθη εντολή για να ξεκινήσετε το κέλυφος mysql και εισάχθιτε ως root χρήστης:

```
/usr/bin/mysql -u root -p
```

Όταν σας ζητηθεί κωδικός πρόσβασης, εισαγάγετε τον κωδικό πρόσβασης που έχετε ορίσει κατά την εγκατάσταση, ή εάν δεν έχετε ορίσει, πατήστε Enter για να μην υποβάλετε κωδικό πρόσβασης.

Τέλος απλά τρέξτε την εντολή

```
CREATE DATABASE (ONOMA);
```

Για να δημιουργηθεί μια βάση.

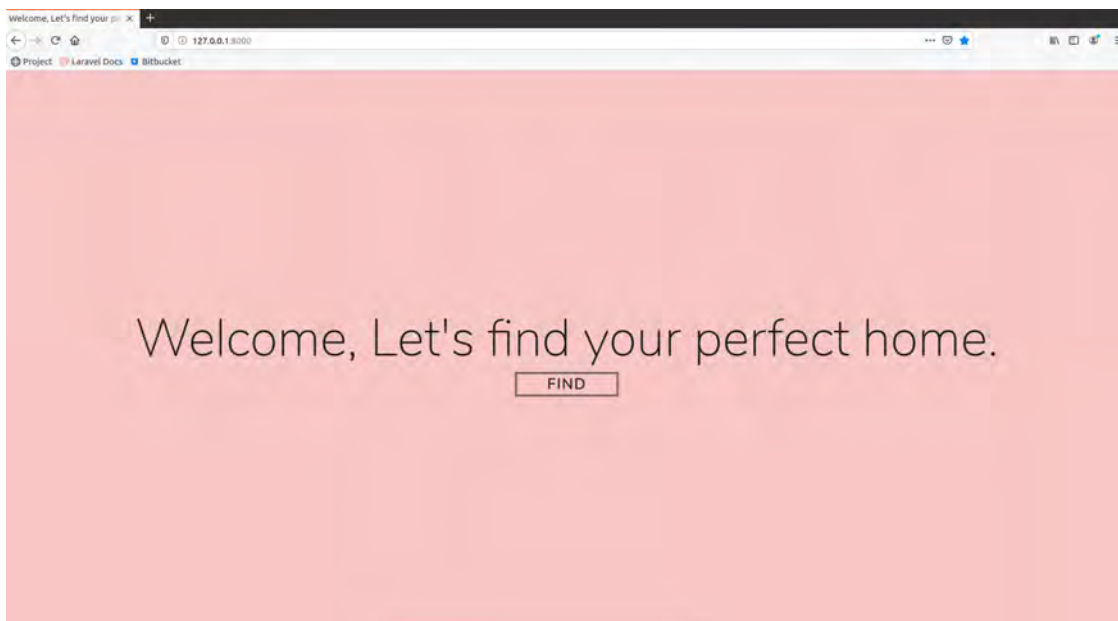
Έπειτα μεταβαίνουμε στον φάκελο που είναι αποθηκευμένο το project και ανοίγουμε με οποιονδήποτε text editor το αρχείο .env έτσι ώστε να οριστικοποιήσουμε τις βασικές ρυθμίσεις. Αυτές είναι ο ορισμός των στοιχείων της βάσης με την οποία θα συνδεθεί η εφαρμογή.

Μετά από την προηγούμενη διαδικασία πρέπει στον φάκελο που βρίσκεται το project να εκτελεστούν με την σειρά οι παρακάτω εντολές έτσι ώστε να δημιουργηθεί το σχήμα της βάσης και να τεθεί σε λειτουργία τοπικό host της εφαρμογής.

```
composer update
php artisan migrate:--fresh --seed
composer dump-autoload
php artisan serve
```

Ύστερα η εφαρμογή είναι σε κατάσταση λειτουργίας σε τοπικό server και είναι προσβάσιμη απο οποιοδήποτε browser στο <http://localhost:8000/>.

Τέλος η χρήστης θα πρέπει να πληκτρολογήσει στο url της σελίδας το μονοπάτι /init και μετά να επιστρέψει στην αρχική σελίδα.



Σχήμα 7.13: Project start page

Ο κώδικας της εργασίας βρίσκεται στο gitHub.

<https://github.com/VLidis/Thesis>

## Κεφάλαιο 8

# Επίλογος

Στο κεφάλαιο αυτό θα αναφερθεί το τελικό αποτέλεσμα της διπλωματικής εργασίας αλλά και στο κατά πόσο αυτό το αποτέλεσμα ανταποκρίνεται στον αρχικό στόχο. Ακόμα θα τεθούν ιδέες για την μελλοντική ανάπτυξη της εργασίας και σημαντικές ενημερώσεις για την λειτουργία του για εμπορική χρήση. της web εφαρμογής σε μελλοντικά update

### 8.1 Σύνοψη και συμπεράσματα

Ο αρχικός στόχος της διπλωματικής εργασίας ήταν να εξετάσει την δυνατότητα νέων φίλτρων αναζήτησης ακινήτων και πώς μπορούν να υλοποιηθούν αυτά σε περιβάλλον web. Τη δυνατότητα δηλαδή του χρήστη να μπορεί να αναζητεί ακίνητα, όχι μόνο με του κλασσικού τρόπους βάση χαρακτηριστικών των ιδίων των ακινήτων που υπάρχουν διαθέσιμοι αυτη τη στιγμή στην αγορά, αλλά και με βάση τα χαρακτηριστικά της περιοχής που βρίσκονται τα εν λόγω ακίνητα και τα σημεία ενδιαφέροντος που τα περιβάλλουν.

Αυτό υλοποιήθηκε με επιτυχία, καθώς μέσα από την εφαρμογή που δημιουργήθηκε μέσω αυτής της διπλωματικής, ένας χρήστης μπορεί να άνετα να αναζητήσει ακίνητα με προηγμένο τρόπο και πληθώρα φίλτρων. Τα εικονικά δεδομένα αντιπροσωπεύουν όσο το δυνατόν περισσότερο αληθινά και ο αριθμός τους, παρότι δεν είναι κοντά σε αυτόν ενός εμπορικού site, είναι ικανοποιητικός και αντιπροσωπεύει ένα ικανό δείγμα για την εκτέλεση της εφαρμογής.

Το γραφικό περιβάλλον είναι φιλικό προς τον χρήστη, όπου ήταν ένα βασικό μέλημα της εργασίας, και ο χρόνος ανταπόκρισης της εφαρμογής κινείται σε ικανοποιητικά πλαίσια. Προφανώς το front-end κομμάτι δεν υπήρξε στόχος της διπλωματικής αφού το κύριο μέλημα ήταν η λειτουργία των προηγμένων φίλτρων και ο υπολογισμός αποστάσεων.

## 8.2 Μελλοντικές επεκτάσεις

Ο πρώτος μελλοντικός στόχος της εφαρμογής είναι να μπορέσει να δημιουργηθεί ένα περιβάλλον front-end αντάξιο εμπορικής χρήσης. Αρχικά το γραφικό περιβάλλον να γίνει ελκυστικό και εμπλουτισμένο με βασικές λειτουργίες. Να μπορεί να υποστηρίζει την ύπαρξη διαφορετικών χρηστών, όπως απλών επισκεπτών αλλά και συνεργατών (μεσιτικά γραφεία) τα οποία θα μπορούν να προσθέτουν, προβάλλουν και διαχειρίζονται τα προϊόντα τους.

Η προσθήκη βασικών λειτουργιών διαχείρισης πελατών, και την δυνατότητα αναδρομολόγησης (Link) στα site των μεσιτικών γραφείων για κάθε ακίνητο. Δηλαδή την δυνατότητα ο χρήστης αφού αναζητήσει τα ακίνητα που τον ενδιαφέρουν να μπορεί να μεταφερθεί στη σελίδα του μεσιτικού γραφείου που ανήκει το ακίνητο έτσι ώστε να προχωρήσει στις διαδικασίες αγοράς του, όπως ξενάγηση στον χώρο, διαδικασίες συμβολαίων κτλ. Επιπροσθέτως η προσθήκη λογαριασμών για τους χρήστες (πελάτες) έτσι ώστε να μπορούν να έχουν ιστορικό αναζητήσεων καθώς και λίστα αγαπημένων (wishlist) για να μπορούν να αποθηκεύουν τα ακίνητα τους τους ενδιαφέρουν και να ενημερώνονται μέσω email για τυχόν αλλαγές τους, όπως τιμή η αν πωλήθηκαν κτλ.

Τέλος, η προσθήκη Api έτσι ώστε ο χρήστης να έχει την δυνατότητα προβολής των ακινήτων σε χάρτη και η δυνατότητα να μπορεί να αναζητεί ακίνητα απευθείας από αυτόν.



# Παράρτημα I

## Παράθεση Κώδικα Front-end και Διαχείριση MVC μοντέλου

### I.1 Κώδικας HTML και CSS

Παραθέτονται κομμάτια κώδικα HTML και CSS που χρησιμοποιήθηκαν για το front-end της εφαρμογής.

```
@extends("layouts.app")

@section('content')
    @include('searchView')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-12">
                <div class="card d-flex flex-column"></div>
            </div>
        </div>
    </div>
@endsection
```

Σχήμα I.1: html css code pt1

### I.2 Διαχείριση αρχείων MVC μοντέλου

Τα αρχεία του κώδικα είναι ταξινομημένα σε φακέλους ανάλογα με τις λειτουργικότητα τους. Συγκεκριμένα :

1. App/HTTP/Controllers => Αρχεία διαχείρισης όλων των λειτουργιών της εφαρμογής.
2. App/model => Αρχεία διαχείρισης της βάσης δεδομένων και των διασυνδέσεων μεταξύ των πινάκων.
3. App/StorageLayer => Αρχεία ανάκτησης δεδομένων απο την βάση.
4. Database => Αρχεία δημιουργίας σχήματος βάσης δεδομένων (migrations) και αρχικοποίησης πινάκων (seeds).
5. Resources/views => Αρχεία html με την εμφάνιση των σελίδων της εφαρμογής.

```

@extends("layouts.app")

@section('content')
    <div class="container border-bottom">
        <div class="row justify-content-center">
            @if ($search->isEmpty())
                <div class="title text-xl-center text-dark">
                    Sadly, we currently have no House matching your criteria. Try to alter some or just broaden your search range. Thank you
                </div>
            @else
                <div class="container-fluid">
                    <div class="title text-xl-center mt-1 text-dark">
                        Houses that Perfectly fits your needs
                    </div>
                    </div>
                    @foreach($search as $house)
                        <div class="col-md-4 text-truncate m-1">
                            <div class="card m-1">
                                <div class="card-body bg-info text-dark">
                                    <h5 class="card-body">Floor:
                                        {{ $house->floor }}
                                    </h5>
                                    <h5 class="card-body">Size:
                                        {{ $house->size }}
                                    </h5>
                                    <h5 class="card-body">Bedrooms:
                                        {{ $house->bedrooms }}
                                    </h5>
                                    <h5 class="card-body">Price:
                                        {{ $house->price }}
                                    </h5>
                                    <h5 class="card-body">Year Build:
                                        {{ $house->year }}
                                    </h5>
                                    <h5 class="card-subtitle">Heat:
                                        @if($house->heat=='0')
                                            Central Petrol
                                        @elseif($house->heat=='1')
                                            Central Gas
                                        @elseif($house->heat=='2')

```

Σχήμα Ι.2: html css code pt2

6. .env => Αρχείο αρχικοποίησης απαραίτητων πληροφοριών της εφαρμογής.

## I.2.1 Δυνατότητες Laravel

1. MVC μοντέλο ιεράρχησης.
2. Υλοποιημένες συναρτήσεις αποστολής email μέσω smtp server.
3. Υλοποιημένες συναρτήσεις διαχείρισης αρχείων (upload, download, delete).
4. Υλοποιημένες συναρτήσεις αρχικοποίησης προσβάσεων στις σελίδες (middleware).
5. Εύκολη διασύνδεση αρχείων html και css μεσο της php.

```

        @elseif($house->heat=='2')
            Private Gas
        @else
            Other
        @endif
    </h5>
    <h5 class="card-subtitle">First Occupant:
        @if($house->newBuild==true)
            First
        @else
            Not First
        @endif
    </h5>
    <h5 class="card-subtitle">Single Floor:
        @if($house->singleFloor==true)
            Single Floor
        @else
            Maisonette
        @endif
    </h5>
    <h5 class="card-subtitle">Elevator:
        @if($house->hasElevator==true)
            Has elevator
        @else
            Doesn't have
        @endif
    </h5>
    <h6 class="card-subtitle">Position:
        {{ $house->longitude }} {{ $house->latitude }}
    </h6>
    <p class="card-text text-wrap text-muted">Description:
        {{ $house->description }}
    </p>
    <p class="card-text">Points of interest info:</p>
    <div class="card-footer">
        <h6 class="card-subtitle">Distance to metro:
            {{ $house->metro0 }}
        </h6>
        <h6 class="card-subtitle">Distance to other Transports:
            {{ $house->bus0 }}

```

Σχήμα I.3: html css code pt3

```

            {{ $house->bus0 }}
        </h6>
        <h6 class="card-subtitle">Distance to City Center:
            {{ $house->cityCenter0 }}
        </h6>
        <h6 class="card-subtitle">Distance to Gym:
            {{ $house->gym0 }}
        </h6>
        <h6 class="card-subtitle">Distance to Park:
            {{ $house->park0 }}
        </h6>
        <h6 class="card-subtitle">Distance to Pharmacy:
            {{ $house->pharmacy0 }}
        </h6>
        <h6 class="card-subtitle">Distance to School:
            {{ $house->school0 }}
        </h6>
        <h6 class="card-subtitle">Distance to Supermarket:
            {{ $house->supermarket0 }}
        </h6>
    </div>
</div>
</div>
</div>
</div>
    @endForeach
    @endif
</div>
<div class="button">
    <a href="http://127.0.0.1:8000/test" class="btn float-xl-right btn-outline-dark">New Search</a>
</div>
</div>
endsection

```

Σχήμα I.4: html css code pt4

```

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Welcome, Let's find your perfect home.</title>

    <!-- Fonts -->
    <link href="https://fonts.googleapis.com/css?family=Nunito:200,600" rel="stylesheet">

    <!-- Styles -->
    <style>
      html, body {
        background-color: #f7c6e5;
        color: #1b1e21;
        font-family: 'Nunito', sans-serif;
        font-weight: 200;
        height: 100vh;
        margin: 0;
      }

      .full-height {
        height: 100vh;
      }

      .flex-center {
        align-items: center;
        display: flex;
        justify-content: center;
      }

      .position-ref {
        position: relative;
      }

      .top-right {
        position: absolute;
        right: 10px;
        top: 10px;
      }
    </style>
  </head>
  <body>
    <div class="flex-center position-ref full-height">
      <div class="content">
        <div class="title">
          Welcome, Let's find your perfect home.
        </div>
      </div>
      <div class="button">
        <a href="http://127.0.0.1:8000/test" class="btn m-b-md">Find</a>
      </div>
    </div>
  </body>
</html>

```

Σχήμα Ι.5: html css code pt5

```

.button > a {
  color: #1b1e21;
  padding: 0 50px;
  font-size: 25px;
  font-weight: 600;
  letter-spacing: .1rem;
  text-decoration: none;
  text-transform: uppercase;
  outline-color: #1b1e21;
  outline-style: double;
  margin-top: 100px;
}

.m-b-md {
  margin-bottom: 30px;
}
</style>
</head>
<body>
  <div class="flex-center position-ref full-height">
    <div class="content">
      <div class="title">
        Welcome, Let's find your perfect home.
      </div>
    </div>
    <div class="button">
      <a href="http://127.0.0.1:8000/test" class="btn m-b-md">Find</a>
    </div>
  </div>
</body>
</html>

```

Σχήμα Ι.6: html css code pt6

# Βιβλιογραφία

- [1] Apache Server. <https://httpd.apache.org/>.
- [2] Azimuth. <https://en.wikipedia.org/wiki/Azimuth>.
- [3] Michael Morrison και Lynn Beighley. *Head First PHP & MySQL: A Brain-Friendly Guide*. O'Reilly Media, 1ή έκδοση, 2009.
- [4] DataGrip. <https://www.jetbrains.com/datagrip/>.
- [5] John Duckett. *HTML and CSS: Design and Build Websites*. John Wiley & Sons, 1ή έκδοση, 2011.
- [6] Dummy Data. <https://scotch.io/tutorials/generate-dummy-laravel-data-with-model-factories>.
- [7] Equirectangular approximation. [https://en.wikipedia.org/wiki/Equirectangular\\_projection](https://en.wikipedia.org/wiki/Equirectangular_projection).
- [8] Factories. <https://laravel.com/docs/7.x/database-testing>.
- [9] Faker Tutorial. <https://kode-blog.io/laravel-5-faker-tutorial>.
- [10] Andrew Ford. *Apache 2 Pocket Reference*. O'Reilly Media, 1ή έκδοση, 2008.
- [11] Steven Sanderson και Adam Freeman. *Pro ASP.NET MVC 4*. Apress, 4ή έκδοση, 2012.
- [12] Raghu Ramakrishnan και Johannes Gehrke. *Database Management Systems*. Εκδόσεις ΤΖΙΟΛΑ, 3ή έκδοση, 2014.
- [13] Haversine. <https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere/>.
- [14] Laravel. <https://laravel.com/>.
- [15] MVC. <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>.
- [16] mySQL. <https://www.mysql.com/about/>.
- [17] PhpStorm. <https://www.jetbrains.com/phpstorm/>.

- [18] Spherical law of cosines. [https://en.wikipedia.org/wiki/Spherical\\_law\\_of\\_cosines](https://en.wikipedia.org/wiki/Spherical_law_of_cosines).
- [19] David Stokes. *MySQL and JSON: A Practical Programming Guide*. McGraw-Hill Education, 1ή έκδοση, 2018.
- [20] Ubuntu. <https://ubuntu.com/>.
- [21] Vincenty. [https://en.wikipedia.org/wiki/Thaddeus\\_Vincenty](https://en.wikipedia.org/wiki/Thaddeus_Vincenty).

# Συντομογραφίες

βλπ	βλέπε
κ.λπ.	και λοιπά
κ.ο.κ	και ούτω καθεξής
POI	Points of interest





# Ορολογία - Γλωσσάρι

## Ελληνικός όρος

ακροατές

απάντηση

γεγονότα

δρομολόγηση

εξυπηρετητής

Εμπρόσθιο άκρο

μετανάστευση

οπίσθιο άκρο

πελάτης

φυλλομετρητής

## Αγγλικός όρος

listeners

response

events

routing

server

front-end

migration

back-end

client

browser

