



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ**

**Σχεδίαση και Ανάπτυξη Λογισμικού Ενοικίασης  
Ιδιωτικών Αυτοκινήτων με Χρήση  
Recommendation Algorithm**

Διπλωματική Εργασία

Κασιδιάρη Αργυρώ-Γενοβέφα

Επιβλέπουσα: Ασπασία Δασκαλοπούλου

Βόλος 2020





**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

**ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ**

**ΥΠΟΛΟΓΙΣΤΩΝ**

**Σχεδίαση και Ανάπτυξη Λογισμικού Ενοικίασης  
Ιδιωτικών Αυτοκινήτων με Χρήση  
Recommendation Algorithm**

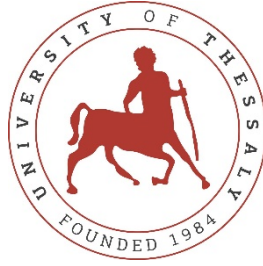
Διπλωματική Εργασία

Κασιδιάρη Αργυρώ-Γενοβέφα

Επιβλέπουσα: Ασπασία Δασκαλοπούλου

Βόλος 2020





**UNIVERSITY OF THESSALY**

**SCHOOL OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING**

**Design and Development of a Private Car Rental  
Application using Recommendation Algorithm**

Diploma Thesis

Kasidiari Argyro-Genovefa

Supervisor: Aspasia Daskalopoulou

Volos 2020



Εγκρίνεται από την Επιτροπή Εξέτασης:

Επιβλέπουσα **Ασπασία Δασκαλοπούλου**  
Επίκουρος Καθηγήτρια, Τμήμα Ηλεκτρολόγων Μηχανικών  
και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Χαρίκλεια Τσαλαπάτα**  
Μέλος Ε.ΔΙ.Π, Τμήμα Ηλεκτρολόγων Μηχανικών και  
Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Μέλος **Ελένη Τουσίδου**  
Μέλος Ε.ΔΙ.Π, Τμήμα Ηλεκτρολόγων Μηχανικών  
και Μηχανικών Υπολογιστών, Πανεπιστήμιο Θεσσαλίας

Ημερομηνία έγκρισης: 20-03-2020





## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τα άτομα που με βοήθησαν, ο καθένας με τον τρόπο του, στην ολοκλήρωση της διπλωματικής εργασίας αυτής και γενικά την ολοκλήρωση των σπουδών μου.

Καταρχάς, οφείλω ένα μεγάλο μέρος της επιτυχίας αυτής στην οικογένεια μου η οποία πίστευε σε εμένα και με υποστήριξε με κάθε πιθανό τρόπο. Επιπλέον, όλα τα άτομα που με περιέβαλαν στην διάρκεια των σπουδών και αποτελούσαν ένα μεγάλο μέρος της καθημερινότητας μου είχαν μεγάλο αντίκτυπο καθώς με βοήθησαν να παραμένω χαρούμενη και να μην σταματάω να προσπαθώ ποτέ για τους στόχους μου.

Τέλος, θα ήθελα να ευχαριστήσω τους καθηγητές του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών οι οποίοι με εφοδίασαν με όλες τις απαραίτητες γνώσεις για την περάτωση των σπουδών μου αλλά ιδιαιτέρως θα ήθελα να αναφέρω τα άτομα που ανέλαβαν την επίβλεψη της διπλωματικής μου· την κυρία Ασπασία Δασκαλοπούλου, την κυρία Χαρίκλεια Τσαλαπατά και την κυρία Ελένη Τουσίδου. Η καθοδήγηση τους ήταν πολύτιμη.



## ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ

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

Η Δηλούσα

(Υπογραφή)  
Κασιδιάρη Αργυρώ-Γενοβέφα  
Οκτώβριος 2020



## Περίληψη

Η ραγδαία εξέλιξη της τεχνολογίας τα τελευταία χρόνια καθώς και η εισχώρησή της στην καθημερινότητα των ανθρώπων έχει ως αποτέλεσμα την δημιουργία εφαρμογών οι οποίες διευκολύνουν όλους τους τομείς του ατόμου. Δεδομένου ότι ειδικά στον ανεπτυγμένο κόσμο ένα αναπόσπαστο κομμάτι της ζωής είναι το αυτοκίνητο είναι αναμενόμενο να δημιουργούνται εφαρμογές οι οποίες διευκολύνουν τον κόσμο που δεν έχει την δυνατότητα ή τυχαίνει να μην έχει στην κατοχή του αμάξι. Έτσι, στην προσπάθεια αυτή, ξεκίνησαν να δημιουργούνται εφαρμογές ενοικίασης αυτοκινήτων. Δυστυχώς όμως το κόστος ενοικίασης συνεχώς αυξάνεται και δεν είναι ανεκτό από πολλούς. Για τον λόγο αυτό ξεκίνησε η ιδέα του διαμοιρασμού αυτοκινήτου και πλέον δίνεται η δυνατότητα να ενοικιάζονται αμάξια ιδιωτών σε πολύ χαμηλότερες τιμές. Τα τελευταία χρόνια αναπτύχθηκαν και οι κατάλληλες εφαρμογές οι οποίες φέρνουν τον διαμοιρασμό αυτοκινήτων ακόμα πιο κοντά στον καθημερινό άνθρωπο. Από τις εφαρμογές αυτές, όμως, λείπει η εξατομίκευση που έχει ανάγκη ο άνθρωπος αυτός έτσι ώστε να διευκολύνεται στις αναζητήσεις του. Σε αυτό ήρθε να βοηθήσει η εφαρμογή αυτή που δημιουργήθηκε στα πλαίσια διπλωματικής εργασίας.



## **Abstract**

The rapid technological development that is seen in the last years and the way it has penetrated the lives of everyone resulted to the development of different applications that would help every aspect of the human life. Considering that, especially, in the developed world, car driving is considered as given it was only a matter of time until applications that could help those who cannot afford to buy a car or choose to not do so, drive one when they need it. As expected, a lot of car renting applications became popular. Unfortunately, though, nowadays the cost of renting a car is really high so not everyone can afford it and not everyone should. That's how the idea of car sharing was born. That means that people should be able to rent a car from a fellow individual when he/she is not using it. Of course, this idea reduced the cost of car renting noticeably. Recently, a lot of applications began developing that could help both sides of car sharing by bringing them in contact. Although, there is one thing missing from these application and that is the customisation that helps people choose the car they need. That's why the application that was developed for this thesis uses a custom recommendation algorithm that helps the user to find their ideal car.





## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

<i>Ευχαριστίες</i> .....	<i>ix</i>
<b>ΥΠΕΥΘΥΝΗ ΔΗΛΩΣΗ ΠΕΡΙ ΑΚΑΔΗΜΑΪΚΗΣ ΔΕΟΝΤΟΛΟΓΙΑΣ ΚΑΙ ΠΝΕΥΜΑΤΙΚΩΝ ΔΙΚΑΙΩΜΑΤΩΝ</b> .....	<b><i>xi</i></b>
<i>Περίληψη</i> .....	<i>xiii</i>
<i>Abstract</i> .....	<i>xv</i>
<b>Κεφάλαιο 1</b> .....	<b>1</b>
<b>Ψηφιακές Κοινότητες και Υπηρεσίες</b> .....	<b>1</b>
<b>1.1 Κοινότητες</b> .....	<b>1</b>
1.1.1 Ορισμός .....	1
1.1.2 Είδη κοινοτήτων.....	1
<b>1.2 Ψηφιακές Κοινότητες</b> .....	<b>1</b>
1.2.1 Ορισμός .....	1
1.2.2. Οφέλη.....	1
1.2.3 Εξέλιξη .....	2
<b>1.3 Υπηρεσίες Ψηφιακών Κοινοτήτων</b> .....	<b>3</b>
1.3.1 Υπηρεσίες Κοινωνικής Δικτύωσης .....	3
1.3.2 Υπηρεσίες Εκμάθησης.....	3
<b>Κεφάλαιο 2</b> .....	<b>4</b>
<b>Διαμοιρασμός Αυτοκινήτων</b> .....	<b>4</b>
<b>2.1 Το Αυτοκίνητο</b> .....	<b>4</b>
2.1.1 Η ανάγκη για ταχεία μετακίνηση και η εξάπλωση του αυτοκινήτου .....	4
2.1.2 Προβλήματα λόγω της τιμής.....	4
<b>2.2 Ενοικίαση Αυτοκινήτων</b> .....	<b>4</b>
2.2.1 Ιστορία.....	4
2.2.2 Αύξηση παροχών και τιμών.....	5
<b>2.3 Διαμοιρασμός Αυτοκινήτων (Carsharing)</b> .....	<b>5</b>
2.3.1 Τι είναι ο διαμοιρασμός αυτοκινήτων .....	5

2.3.2 Πρώτα βήματα .....	5
2.3.3 Είδη.....	6
2.3.4 Εξάπλωση .....	6
<b>Κεφάλαιο 3 .....</b>	<b>8</b>
<b>Recommendation Algorithm .....</b>	<b>8</b>
3.1 Ορισμός .....	8
3.2 Γνωστές εφαρμογές Recommendation Algorithm .....	8
3.2.1 Κοινωνικά δίκτυα .....	8
3.2.2 Υπηρεσίες streaming.....	9
3.2.3 Διαδικτυακά καταστήματα.....	9
3.3 Έλεγχος αποτελεσματικότητας .....	9
3.3.1 Μελέτες χρηστών.....	9
3.3.2 Διαδικτυακές εκτιμήσεις .....	10
3.4 Γιατί χρησιμοποιούνται Recommendation Algorithms στην εφαρμογή; .....	10
3.5 Πώς χρησιμοποιείται Recommendation Algorithm στην εφαρμογή;.....	10
<b>Κεφάλαιο 4 .....</b>	<b>12</b>
<b>Περιγραφή της Εφαρμογής .....</b>	<b>12</b>
4.1 Παρόμοιες εφαρμογές .....	12
4.1.1 Σύγκριση εφαρμογών.....	12
4.2 Σχεδιασμός εφαρμογής.....	13
4.2.1 Σκοπός εφαρμογής.....	13
4.2.2 Απαιτήσεις εφαρμογής.....	13
4.2.3 Εργαλεία που χρησιμοποιήθηκαν .....	13
4.3 Επεξήγηση κώδικα .....	15
4.3.1.1 Εγγραφή ενοικιαστή.....	15
4.3.1.2 Εγγραφή ενοικιάζοντος .....	17
4.3.1.3 Εγγραφή καινούριου αυτοκινήτου .....	19
4.3.2 Σύνδεση χρήστη .....	22

4.3.3 Αποσύνδεση χρήστη.....	23
4.3.4 Αναζήτηση αυτοκινήτου .....	24
4.4 Γενική παρουσίαση εφαρμογής .....	26
4.5 Επίπεδο της εφαρμογής .....	34
4.6 Χρήση και δοκιμή της εφαρμογής .....	34
<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>35</b>
<b>ΠΑΡΑΡΤΗΜΑΤΑ .....</b>	<b>36</b>
<b>ΠΑΡΑΡΤΗΜΑ Α.....</b>	<b>37</b>
<b>ΛΙΣΤΑ ΓΡΑΦΗΜΑΤΩΝ.....</b>	<b>37</b>
<b>ΠΑΡΑΡΤΗΜΑ Β.....</b>	<b>38</b>
<b>ΛΙΣΤΑ ΕΙΚΟΝΩΝ .....</b>	<b>38</b>



# Κεφάλαιο 1

## Ψηφιακές Κοινότητες και Υπηρεσίες

### 1.1 Κοινότητες

#### 1.1.1 Ορισμός

Ως κοινότητα ορίζεται μία κοινωνική ενότητα από ανθρώπους με κοινές αρχές που κινούνται στο ίδιο περιβάλλον, αλληλεπιδρούν μεταξύ τους, ασκούν επιρροή ο ένας στον άλλον και υπακούν στους ίδιους κανόνες.

#### 1.1.2 Είδη κοινοτήτων

Οι κοινότητες διακρίνονται από 3 είδη ανάλογα με τον συνδετικό κρίκο των μελών τους. Αυτά είναι:

- i) Γεωγραφικές κοινότητες οι οποίες αποτελούνται από ανθρώπους από την ίδια εδαφική περιφέρεια,
- ii) Πολιτισμικές κοινότητες οι οποίες αποτελούνται από άτομα με κοινή πολιτισμική κληρονομιά και κουλτούρα,
- iii) Εθνικές κοινότητες οι οποίες αποτελούνται από άτομα που κατάγονται από το ίδιο έθνος.

[1]

### 1.2 Ψηφιακές Κοινότητες

#### 1.2.1 Ορισμός

Ως ψηφιακή κοινότητα ορίζεται μία κοινότητα τα μέλη της οποίας έχουν κάποιο κοινό ενδιαφέρον και αλληλεπιδρούν κατά κύριο λόγο μέσω του διαδικτύου.

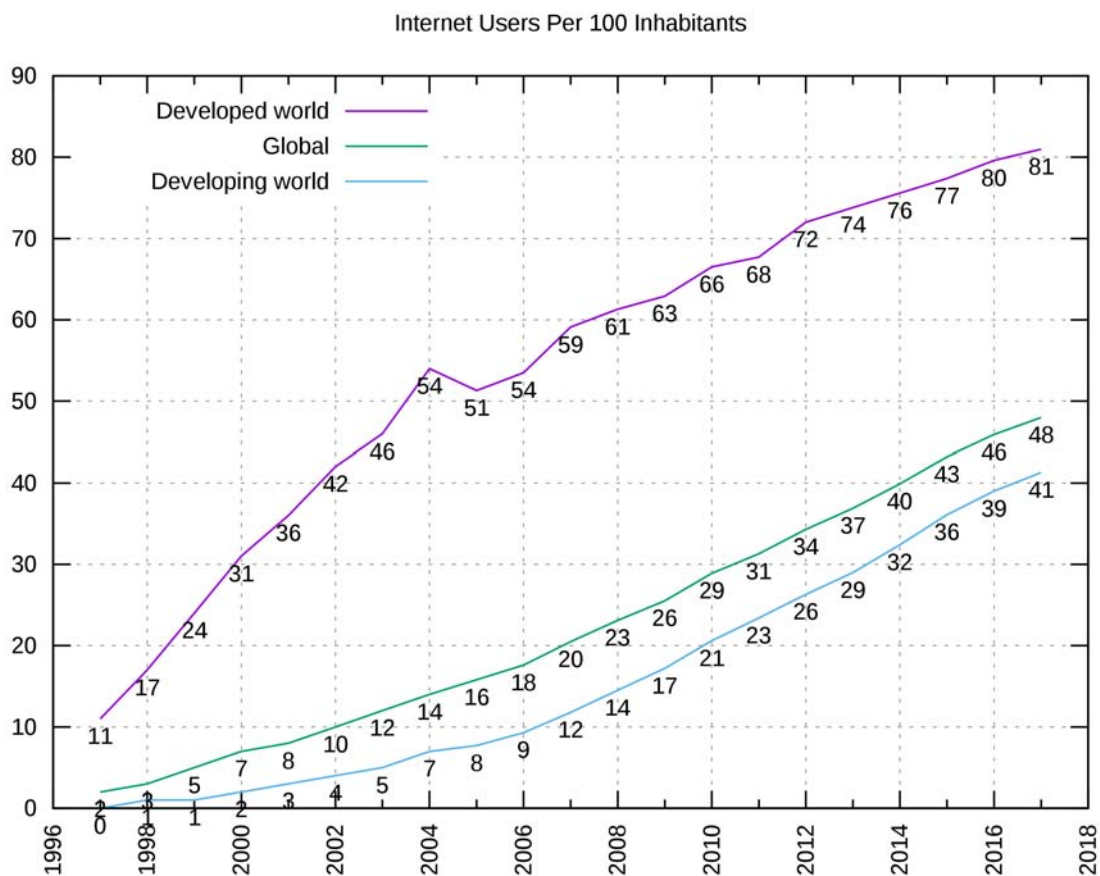
#### 1.2.2. Οφέλη

Ο άνθρωπος έχει μάθει να ζει και να κινείται ανέκαθεν σε κοινότητες. Καθώς αυτές εξελίσσονται σε ψηφιακές, αυτός μαθαίνει να τις αξιοποιεί προς όφελός του. Συγκεκριμένα, οι ψηφιακές κοινότητες έχουν δώσει την δυνατότητα σε πολλούς ανθρώπους να γνωρίσουν και να αλληλεπιδράσουν με άτομα που έχουν κοινά ενδιαφέροντα. Ως αποτέλεσμα γίνεται ευκολότερη η ανταλλαγή απόψεων

καταλήγοντας στην εξέλιξη του ανθρώπινου νου. Εκτός όμως από τον τομέα της διαβούλευσης οι ψηφιακές κοινότητες έχουν οφελήσει και στην ενημέρωση αλλά και στην εκπαίδευση πολλών ανθρώπων.

### 1.2.3 Εξέλιξη

Όπως είναι φυσιολογικό, η εξέλιξη του διαδικτύου είναι ραγδαία και μαζί του εξελίσσονται και οι ψηφιακές κοινότητες. Συγκεκριμένα 6 από τις 20 ιστοσελίδες με την περισσότερη επισκεψιμότητα στο διαδίκτυο βασίζονται σε κάποια ψηφιακή κοινότητα. Και αυτός ο αριθμός αναμένεται να αυξηθεί με την πάροδο του χρόνου καθώς το ποσοστό του παγκόσμιου πληθυσμού που έχει πρόσβαση στο διαδίκτυο αυξάνεται συνεχώς.[2]



Σχήμα 1.1: Γράφημα Ποσοστού Χρηστών Διαδικτύου ανά 100 κατοίκους.

Πηγή: [https://en.wikipedia.org/wiki/Global\\_Internet\\_usage](https://en.wikipedia.org/wiki/Global_Internet_usage)

### **1.3 Υπηρεσίες Ψηφιακών Κοινοτήτων**

Τα μέλη μίας ψηφιακής κοινότητας έχουν ως σημείο αναφοράς κάποιο κοινό ενδιαφέρον ή κάποιον κοινό στόχο. Έτσι προέκυψε και η ανάγκη για δημιουργία ψηφιακών υπηρεσιών οι οποίες θα βοηθούν τα μέλη της να αλληλεπιδράσουν ή να λύσουν κάποιο κοινό τους πρόβλημα.

#### **1.3.1 Υπηρεσίες Κοινωνικής Δικτύωσης**

Ο άνθρωπος, όντας φύσει κοινωνικό ον, προσπάθησε να κάνει την επικοινωνία όλο και πιο απλή και εύκολη. Έτσι, με την εξέλιξη του διαδικτύου, αναπτύχθηκαν και τα κοινωνικά δίκτυα που έδωσαν την δυνατότητα σε ανθρώπους από όλο τον κόσμο να επικοινωνούν μεταξύ τους.

Η εξάπλωση των υπηρεσιών αυτών ήταν ραγδαία, μετρώντας περισσότερους από 3,96 δισεκατομμύρια ενεργούς χρήστες παγκοσμίως.[3]

#### **1.3.2 Υπηρεσίες Εκμάθησης**

Το διαδίκτυο αποτελεί μία ανεξάντλητη πηγή πληροφοριών. Προκειμένου να οργανωθεί στον τομέα αυτό, δημιουργήθηκαν ψηφιακές πλατφόρμες που παρέχουν στους χρήστες τους την δυνατότητα της τηλεεκπαίδευσης.

Τα οφέλη των υπηρεσιών αυτών έγιναν άμεσα αντιληπτά και για αυτό όλο και περισσότεροι άνθρωποι τις επέλεξαν για την εκπαίδευσή τους. Συγκεκριμένα, το γεγονός ότι δεν δεσμεύουν συγκεκριμένο χρόνο από τον χρήστη σε συνδυασμό με το ότι μπορεί να τις χρησιμοποιήσει από την άνεση του σπιτιού του τις κατέστησαν πολύ γρήγορα δημοφιλείς.

## Κεφάλαιο 2

### Διαμοιρασμός Αυτοκινήτων

#### 2.1 Το Αυτοκίνητο

##### 2.1.1 Η ανάγκη για ταχεία μετακίνηση και η εξάπλωση του αυτοκινήτου

Το αυτοκίνητο είναι ένα εργαλείο που εισήχθη στις ζωές των ανθρώπων εν έτει 1886 και αποτελεί πλέον αναπόσπαστο κομμάτι της καθημερινότητας τους καθώς έλυσε ένα από τα μεγαλύτερα προβλήματα της ανθρωπότητας: την ταχεία μετακίνηση. Έτσι, με την εκβιομηχάνιση του όλο και περισσότεροι άνθρωποι αποκτούσαν αυτοκίνητα. Συγκεκριμένα, στον αναπτυσσόμενο κόσμο, πλέον, σε κάθε 1000 άτομα αντιστοιχούν 600 αυτοκίνητα.[4]

##### 2.1.2 Προβλήματα λόγω της τιμής

Το αυτοκίνητο λόγω του ότι είναι ένα αρκετά περίπλοκο όχημα δεν παρέχεται σε τιμή προσιτή για όλους. Δεν είναι λίγοι οι άνθρωποι οι οποίοι δεν μπορούν ή δεν θέλουν να πληρώσουν για να αποκτήσουν ένα αυτοκίνητο αφού δεν το έχουν συνεχή ανάγκη. Όπως είναι λογικό προέκυψε για αυτούς τους ανθρώπους η ανάγκη να έχουν πρόσβαση σε αυτοκίνητο μόνο όταν το χρειάζονται. Το πρόβλημα αυτό ήρθε να λύσει η ενοικίαση αυτοκινήτων.

#### 2.2 Ενοικίαση Αυτοκινήτων

##### 2.2.1 Ιστορία

Η ενοικίαση αυτοκινήτων έκανε τα πρώτα της ερασιτεχνικά βήματα το μακρινό 1904. Ωστόσο, το 1912 δημιουργήθηκε και η πρώτη εταιρία ενοικίασης αυτοκινήτων, στην Γερμανία. Όσον αφορά την Αμερική, λίγα χρόνια αργότερα, το 1916, ο Joe Saunders, ένας μεσίτης, αγόρασε ένα μεταχειρισμένο Model T Ford και ξεκίνησε να το νοικιάζει για 6 λεπτά του δολλαρίου για κάθε μίλι. Σε διάστημα ενός έτους είχε στην κατοχή του 18 Model T Ford και τα διέθετε για 10 λεπτά/μίλι καθώς τα άτομα που ήθελαν να νοικιάζουν αυτοκίνητα αυξάνονταν. Έτσι, όλο και περισσότερες εταιρίες ξεκίνησαν τα πρώτα τους βήματα στον τομέα αυτό εκείνη την περίοδο και πολλές από αυτές υπάρχουν μέχρι σήμερα.[5][6]



### 2.2.2 Αύξηση παροχών και τιμών

Όσο περνούσαν τα χρόνια άλλαζαν οι ανάγκες των πελατών των εταιριών ενοικίασης αυτοκινήτων. Για παράδειγμα, ζητούσαν την δυνατότητα να επιλέξουν από μία ποικιλία αυτοκινήτων ανάλογα με διάφορα κριτήρια όπως ο κυβισμός του, το χρώμα του ή η παλαιότητα του. Ταυτόχρονα, οι εταιρίες ξεκίνησαν να χρεώνουν υποχρεωτικά ασφάλεια για τα αυτοκίνητα τους, έτσι ώστε να μην επωμίζονται το κόστος των πιθανών ζημιών των οδηγών. Έτσι η αξία της ενοικίασης αυτοκινήτου άρχισε να ανεβαίνει αρκετά γρήγορα. Όσον αφορά τον τρόπο χρέωσης, σταδιακά άλλαξε σε χρέωση ανάλογα με το πλήθος των ημερών ενοικίασης και όχι ανάλογα με την διανυόμενη απόσταση ή τις ώρες χρήσης.

## 2.3 Διαμοιρασμός Αυτοκινήτων (Carsharing)

### 2.3.1 Τι είναι ο διαμοιρασμός αυτοκινήτων

Ο διαμοιρασμός αυτοκινήτων είναι ένα μοντέλο ενοικίασης που εκμεταλλεύεται τα αυτοκίνητα ιδιωτών όσο αυτοί δεν τα χρησιμοποιούν. Ως επί το πλείστον ο διαμοιρασμός αυτοκινήτων εξυπηρετεί ανθρώπους που επιθυμούν να ενοικιάσουν ένα αυτοκίνητο για λίγες μόνο ώρες. Ωστόσο, το πλήθος των ατόμων που επιλέγουν αυτόν τον τρόπο ενοικίασης αυτοκινήτου για μεγαλύτερα διαστήματα, αυξάνεται συνεχώς καθώς η διαφορά στο κόστος είναι αρκετά μεγάλη.

### 2.3.2 Πρώτα βήματα

Η προαναφερθείσα αύξηση τιμών ήταν η αρχική αφορμή για να επέλθει μία ριζική αλλαγή στον τρόπο που οι άνθρωποι ενοικίαζαν αυτοκίνητα. Σε συνδυασμό με το γεγονός ότι δεν ήταν εφικτή η γρήγορη ενοικίαση ενός οχήματος για λίγες μόνο ώρες, αλλά και με την συνειδητοποίηση ότι υπάρχουν αρκετά αμάξια τα οποία είναι αχρησιμοποίητα από τους ιδιοκτήτες τους, προέκυψε η ιδέα του διαμοιρασμού αυτοκινήτου με τα πρώτα της βήματα να γίνονται το 1948 στην Ελβετία. Ωστόσο, δεν ήταν μέχρι και τις αρχές του 21ου αιώνα που η ιδέα αυτή έκανε μεγάλη πρόοδο και εξαπλώθηκε σε ένα μεγάλο μέρος του κόσμου.[7]

### 2.3.3 Είδη

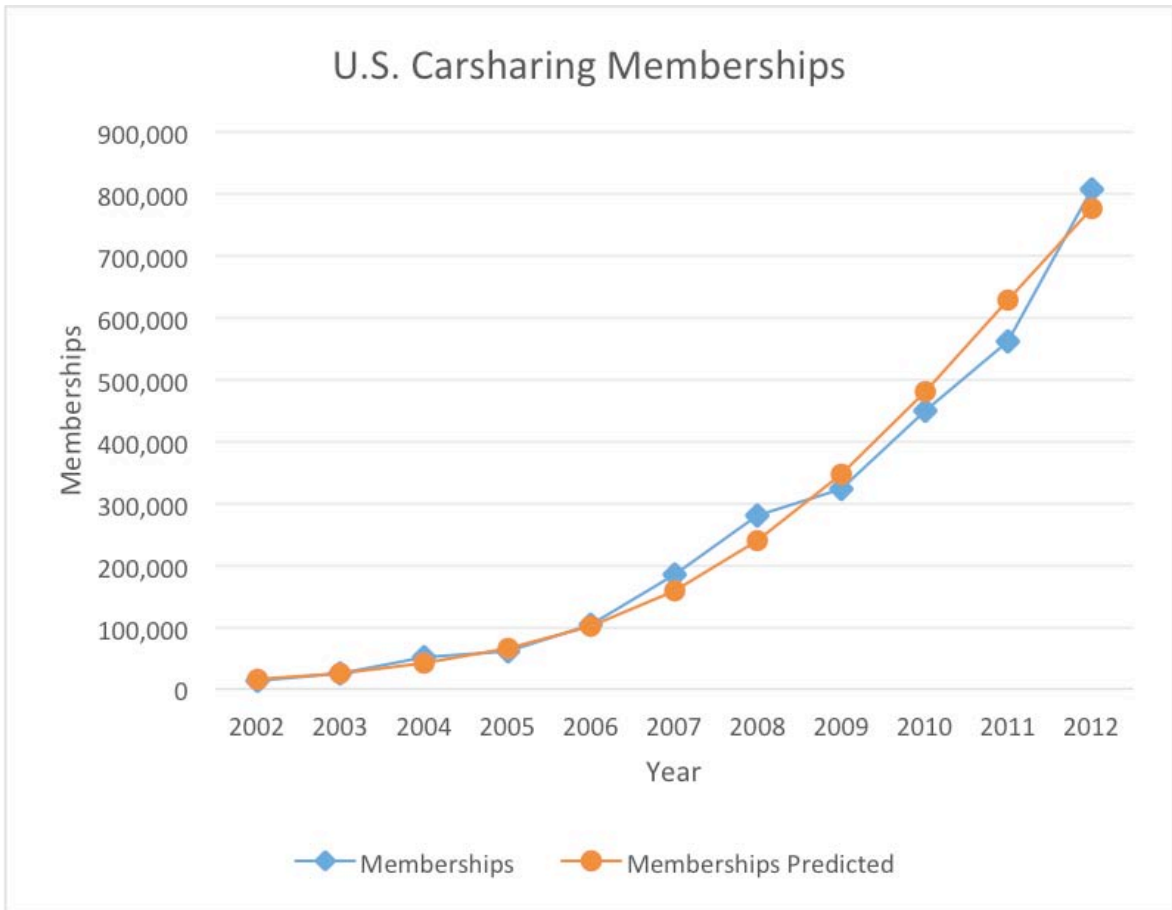
Οι ανάγκες των χρηστών του διαμοιρασμού αυτοκινήτων διέφεραν αισθητά κι έτσι προκειμένου να εξυπηρετηθούν όσο περισσότεροι άνθρωποι ήταν δυνατό ξεκίνησε η διαφοροποίηση του τρόπου χρήσης αυτού του μοντέλου.

Αρχικά ο ενοικιαστής ήταν υποχρεωμένος να επιστρέψει το όχημα που χρησιμοποίησε ακριβώς εκεί από όπου το παρέλαβε καθώς ο ιδιοκτήτης του θα έπρεπε να χρεώσει τον πελάτη με βάση την εκάστοτε τιμολογιακή πολιτική στο τέλος της ενοικίασης.

Όσο όμως το μοντέλο αυτό διαδιδόταν, όπως είναι αναμενόμενο, ξεκίνησαν να εισχωρούν σε αυτό εταιρίες που αναλάμβαναν την διαχείριση των οχημάτων χρεώνοντας μία προμήθεια για κάθε ενοικίαση που ολοκληρωνόταν μέσω αυτών. Έτσι, με την εξάπλωση των εταιριών αυτών σε πολλές περιοχές ταυτόχρονα, εμφανίστηκε και μία άλλη δυνατότητα διαμοιρασμού αυτοκινήτου· η ολοκλήρωση της διαδρομής να γίνεται σε άλλο σημείο εξυπηρέτησης από αυτό που έγινε η παραλαβή με την επιστροφή του αυτοκινήτου στον ιδιοκτήτη του να είναι ευθύνη της εταιρίας διαχείρισης.

### 2.3.4 Εξάπλωση

Χρόνο με το χρόνο ο διαμοιρασμός αυτοκινήτου εξαπλώνεται σε περισσότερες χώρες. Πέρα από αυτό όμως, πλέον, οι μεγάλες εταιρίες ενοικίασης αυτοκινήτων δημιούργησαν κι αυτές τις δικές τους εφαρμογές για διαμοιρασμό αυτοκινήτου εκμεταλλευόμενες την ήδη υπάρχουσα εμπειρία τους στον κλάδο και τον ταχέως αυξανόμενο αριθμό χρηστών διαμοιρασμού αυτοκινήτων.



Σχήμα 2.1: Γράφημα Χρηστών Διαμοιρασμού Αυτοκινήτων στις ΗΠΑ

Πηγή: <https://en.wikipedia.org/wiki/Carsharing#History>

## Κεφάλαιο 3

### Recommendation Algorithm

#### 3.1 Ορισμός

Recommendation Algorithm είναι ένας αλγόριθμος ο οποίος έχει σκοπό να προβλέπει την επιλογή του χρήστη σε κάποια διαδικτυακή εφαρμογή. Συγκεκριμένα, ο λόγος που δημιουργήθηκαν τέτοιου τύπου αλγόριθμοι είναι οι εφαρμογές αυτές να μπορούν να προτείνουν στον χρήστη, με κριτήριο τις επιλογές που είχε κάνει σε προηγούμενες χρήσεις της εφαρμογής, αποτελέσματα που πιθανόν να ήθελε να δει.

#### 3.2 Γνωστές εφαρμογές Recommendation Algorithm

Σχεδόν όλες οι εφαρμογές που χρησιμοποιεί ο κόσμος στην καθημερινότητά του αξιοποιούν κάποιου είδους recommendation algorithm για να προτείνουν περιεχόμενο στον χρήστη.

##### 3.2.1 Κοινωνικά δίκτυα

Ένα μεγάλο μέρος των πλατφορμών κοινωνικής δικτύωσης χρησιμοποιούν recommendation algorithms προκειμένου είτε να προτείνουν στον χρήστη άλλους χρήστες με τους οποίους πιθανόν να επιθυμεί να δικτυωθεί είτε για να τον προτρέψουν να αναγνώσει περιεχόμενο που ίσως να τον ενδιαφέρει.

Στην πρώτη περίπτωση θα προταθούν άτομα με τα οποία είτε έχει κοινές συνδέσεις είτε βρίσκονται σε πολύ κοντινή περιοχή, οπότε είναι και πιο λογικό να γνωρίζονται από το να προτεινόταν άτομα χωρίς κανένα κριτήριο.

Στην δεύτερη περίπτωση θα προταθούν, για παράδειγμα, άρθρα ή ειδήσεις σχετικά με θέματα τα οποία επιλέγει να διαβάσει πιο συχνά ο χρήστης.

### 3.2.2 Υπηρεσίες streaming

Ίσως η πιο συχνή χρήση recommendation algorithm να γίνεται στις εφαρμογές που παρέχουν στον χρήστη την δυνατότητα streaming είτε για μουσική είτε για ταινίες και τηλεοπτικές σειρές.

Στην περίπτωση της μουσικής εφαρμογές όπως το Spotify προτείνουν στον χρήστη τραγούδια κρίνοντας από αρκετές παραμέτρους. Οι παράμετροι αυτές μπορεί να είναι είτε το είδος της μουσικής που επιλέγει ο χρήστης πιο συχνά, είτε καλλιτέχνες είτε οι χρονολογίες των τραγουδιών που ακούει. Συνήθως τα αποτελέσματα προκύπτουν από όλες αυτές τις παραμέτρους ταυτόχρονα δίνοντας βέβαια άλλη βαρύτητα στην καθεμία.

Στην περίπτωση των ταινιών και των τηλεοπτικών σειρών η λογική είναι αρκετά παρόμοια. Οι παράμετροι, αυτήν την φορά, είναι το είδος, οι ηθοποιοί που παίρνουν μέρος και όπως και πριν η χρονολογία.

### 3.2.3 Διαδικτυακά καταστήματα

Ένας αρκετά ανερχόμενος τομέας του διαδικτύου είναι οι διαδικτυακές αγορές. Έτσι δεν θα μπορούσαν να λείπουν τέτοιας μορφής αλγόριθμοι από τις εφαρμογές αυτές οι οποίοι είναι διαμορφωμένοι έτσι ώστε να προτείνουν στον χρήστη προϊόντα τα οποία πιθανόν να επιθυμεί να αγοράσει. Οπότε, με παραμέτρους όπως προηγούμενες αγορές από την ίδια ή και άλλη εφαρμογή, ή ακόμα και αναζητήσεις στο διαδίκτυο επιτυγχάνεται η πρόταση προϊόντων σχετικών με τις προτιμήσεις του και τις επιθυμίες του.

## 3.3 Έλεγχος αποτελεσματικότητας

Κάθε αλγόριθμος πρέπει να κριθεί για την αποτελεσματικότητά του. Στον τομέα των recommendation algorithms οι βασικοί τρόποι να ελεγχθεί το κατά πόσο οι προτάσεις είναι εύστοχες είναι 2· μελέτες χρηστών και διαδικτυακές εκτιμήσεις.[8]

### 3.3.1 Μελέτες χρηστών

Οι μελέτες χρηστών λειτουργούν με τον εξής τρόπο. Παρέχονται σε κάποιους ανθρώπους προτάσεις που έχουν προκύψει από αλγόριθμους που λειτουργούν

με διαφορετικές παραμέτρους και στην συνέχεια κρίνουν κατά πόσο ήταν εύστοχες. Ωστόσο αυτός ο τρόπος ελέγχου των αλγορίθμων έχει το μειονέκτημα ότι το δείγμα είναι αναγκαστικά μικρό και συχνά όχι αντιπροσωπευτικό.

### 3.3.2 Διαδικτυακές εκτιμήσεις

Στην περίπτωση αυτή τα αποτελέσματα είναι αρκετά πιο ακριβή καθώς κρίνονται οι επιλογές των χρηστών στις εφαρμογές που χρησιμοποιούν recommendation algorithms. Δηλαδή, ελέγχεται κατά πόσο ένας χρήστης θα επιλέξει την πρόταση που του εμφανίζεται κάθε φορά. Έτσι, μπορεί να κριθεί κατά πόσο είναι εύστοχες οι προτάσεις.

## 3.4 Γιατί χρησιμοποιούνται Recommendation Algorithms στην εφαρμογή;

Οι εφαρμογές που κάνουν χρήση recommendation algorithms βελτιώνουν αισθητά την εμπειρία του χρήστη. Αυτό συμβαίνει διότι το περιεχόμενο είναι προσαρμοσμένο κατά μεγάλο βαθμό στις προτιμήσεις και τις εμπειρίες του. Για τον λόγο αυτό κρίθηκε σωστή η επιλογή του να χρησιμοποιηθεί μία μορφή recommendation algorithm στην εφαρμογή η οποία εμφανίζει τα αποτελέσματα με τέτοια σειρά έτσι ώστε τα πρώτα να σχετίζονται με τις προηγούμενες ενοικιάσεις του χρήστη περισσότερο από ότι τα επόμενα.

## 3.5 Πώς χρησιμοποιείται Recommendation Algorithm στην εφαρμογή;

Στην ανάπτυξη της εφαρμογής δημιουργήθηκε ένας προσαρμοσμένος recommendation algorithm με σκοπό να ληφθούν υπ' όψιν οι παράμετροι οι οποίες συνήθως επηρεάζουν την απόφαση του χρήστη στην ενοικίαση αυτόκινήτου. Πιο συγκεκριμένα, ο recommendation algorithm αυτός λαμβάνει ως επί το πλείστον υπ' όψιν τα κυβικά των προηγούμενων ενοικιάσεων του χρήστη μιας που είναι ο νούμερο ένα παράγοντας επιλογής αυτοκινήτου ενοικίασης.

Για κάθε χρήστη υπολογίζεται ο μέσος όρος των κυβικών που ενοικιάζει και με χρήση αυτού τα αποτελέσματα εμφανίζονται από τα πιο σχετικά στα λιγότερο. Δηλαδή πρώτο θα εμφανιστεί το αποτέλεσμα που είναι πιο κοντά στον μέσο όρο

αυτό. Βεβαίως, το πεδίο αυτό αλλάζει τιμή με κάθε καινούρια ενοικίαση. Στην συνέχεια σε περίπτωση που υπάρχει ισοβαθμία στον παράγοντα αυτόν κρίνεται η τιμή εμφανίζοντας πρώτο το φθηνότερο αυτοκίνητο. Τέλος, τελευταίος παράγοντας που λαμβάνεται υπόψιν στην διάταξη των αντικειμένων είναι η χρονιά τους. Στην περίπτωση, δηλαδή, ισοβαθμίας των δύο προηγούμενων παραγόντων θα εμφανιστεί πρώτα το πιο καινούριο αυτοκίνητο.

## Κεφάλαιο 4

### Περιγραφή της Εφαρμογής

#### 4.1 Παρόμοιες εφαρμογές

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο η ιδέα του διαμοιρασμού αυτοκινήτου υπάρχει πολλά χρόνια και έχει υλοποιηθεί από πολλές εταιρίες. Επομένως, υπάρχει ένας μεγάλος αριθμός ανταγωνιστικών εφαρμογών παγκοσμίως. Παρ' όλα αυτά στην Ελλάδα είναι λίγο διαφορετική η κατάσταση καθώς οι ενεργές εφαρμογές που παρέχουν τον διαμοιρασμό αυτοκινήτων είναι μόνο μία. Η εφαρμογή αυτή ονομάζεται ridemind και έχει παρουσία μέχρι στιγμής μόνο στην Ελλάδα.

##### 4.1.1 Σύγκριση εφαρμογών

Η υπάρχουσα εφαρμογή διαφέρει σε μερικούς τομείς με την εφαρμογή που διαμορφώθηκε στα πλαίσια της εργασίας αυτής. Καταρχάς, το ridemind παρέχει μικτή ασφάλεια, σε συνεργασία με την Allianz, σε κάθε ενοικίαση κάτι το οποίο δεν συμβαίνει στην εφαρμογή της εργασίας καθώς αποτελεί μόνον τον διαμεσολαβητή μεταξύ του ιδιοκτήτη του κάθε αυτοκινήτου και του ενοικιαστή. Για τον ίδιο λόγο προκύπτει και άλλη μία διαφορά που είναι ότι δεν κρατάται προμήθεια από κάθε ενοικίαση καθώς η εφαρμογή αυτή δεν έχει, ακόμα, επιχειρηματική υπόσταση. Ως αποτέλεσμα των 2 παραπάνω διαφορών προκύπτει και μία ακόμα που είναι η σχετικά μεγάλη διαφορά στην τιμή κάθε ενοικίασης αφού προστίθενται επιπλέον χρεώσεις στην περίπτωση του ridemind.

Η ουσιαστικότερη, όμως, διαφορά είναι ότι η εφαρμογή της εργασίας αυτής διαθέτει recommendation algorithm όπως αναφέρθηκε στο προηγούμενο κεφάλαιο προκειμένου να εμφανίζονται εξατομικευμένα αποτελέσματα αυτοκινήτων σε κάθε αναζήτηση του χρήστη λαμβάνοντας υπόψιν προηγούμενες επιλογές.



## 4.2 Σχεδιασμός εφαρμογής

### 4.2.1 Σκοπός εφαρμογής

Ο σκοπός για την υλοποίηση μίας εφαρμογής διαμοιρασμού αυτοκινήτου με χρήση recommendation algorithm είναι η διευκόλυνση των ατόμων που τους ενδιέφερε η διαδικασία αυτή αλλά δεν νιώθαν αρκετά οικεία με την ήδη υπάρχουσα υλοποίησή της ή δεν γνώριζαν ότι υπάρχει τέτοια δυνατότητα καθώς στην Ελλάδα δεν είναι όσο διαδεδομένη θα έπρεπε.

### 4.2.2 Απαιτήσεις εφαρμογής

Μία τέτοια εφαρμογή θα πρέπει να διέπεται από κάποια πρότυπα. Συγκεκριμένα πρέπει να παρέχεται εύκολη διεπαφή χρήστη (user interface), κάτι το οποίο είναι αρκετά απαιτητικό καθώς υπάρχουν δύο διαφορετικά είδη χρηστών που πρέπει να εξυπηρετηθούν εξίσου καλά· αυτός που επιθυμεί να ενοικιάσει αυτοκίνητο και αυτός που διαθέτει το αυτοκίνητό του προς ενοικίαση.

Η προαναφερθείσα διεπαφή χρήστη, λοιπόν, προβλέπει δύο ξεχωριστές εμπειρίες ανάλογα με το είδος χρήστη που είναι συνδεδεμένος κάθε φορά κι αυτό διότι κάθε χρήστης χρειάζεται να του εμφανίζονται εντελώς διαφορετικές πληροφορίες από τον άλλον αλλά να του παρέχονται και διαφορετικές δυνατότητες μέσα στην εφαρμογή.

Φυσικά, η πιο σημαντική απαίτηση της εφαρμογής είναι και αυτή που την ξεχωρίζει από τον ανταγωνισμό· το recommendation algorithm που χρησιμοποιείται στους πιθανούς ενοικιαστές κατά την διαδικασία της αναζήτησης του κατάλληλου αυτοκινήτου.

### 4.2.3 Εργαλεία που χρησιμοποιήθηκαν

Η υλοποίηση μίας ολοκληρωμένης web εφαρμογής απαιτεί την χρήση πολλών και διαφορετικών εργαλείων.

Ένα απαραίτητο εργαλείο είναι το περιβάλλον σύνταξης του κώδικα της εφαρμογής. Για τον σκοπό αυτό επιλέχθηκε η χρήση του sourceLair IDE. Το sourceLair IDE είναι ένα online IDE(Integrated Development Enviroment), δηλαδή

ένας διαδικτυακός ιστότοπος που παρέχει την δυνατότητα για επεξεργασία κώδικα, και αποθήκευση των αλλαγών. Το πιο σημαντικό είναι ότι το sourceLair IDE παρέχει υπηρεσία για άμεση απεικόνιση του κώδικα σε προσωρινό url το οποίο λειτουργεί όσο λειτουργεί και ο server της εφαρμογής έτσι ώστε να ελέγχονται άμεσα και εύκολα οι αλλαγές στον κώδικα.

Οι γλώσσες προγραμματισμού που επιλέχθηκαν για την ανάπτυξη της εφαρμογής είναι HTML, CSS, JavaScript και Python(Django framework). Η HTML, η CSS και η Javascript είναι υπεύθυνες για το κομμάτι της εφαρμογής που είναι ορατό στον χρήστη (FrontEnd). Πιο συγκεκριμένα, η HTML ορίζει τις θέσεις των αντικειμένων στην σελίδα, η CSS αλλάζει το εμφανισιακό κομμάτι, κυρίως, των αντικειμένων και η JavaScript θέτει την λειτουργικότητά τους.

Όσον αφορά το κομμάτι της εφαρμογής που δεν είναι ορατό στον χρήστη (BackEnd) επιλέχθηκε η χρήση Python και συγκεκριμένα του framework της Django που χρησιμοποιείται για την ανάπτυξη πλήρως λειτουργικού server. Ένα μεγάλο πλεονέκτημα του Django Framework είναι το γεγονός πως δεν απαιτεί γνώσεις ξεχωριστής γλώσσας για την ανάπτυξη βάσης δεδομένων καθώς διαθέτει το ίδιο εντολές που διαμορφώνουν ολοκληρωμένη βάση δεδομένων με χρήση της τεχνολογίας SQLite.

Επίσης, μία μεγάλη διευκόλυνση είναι το Django Admin Page που είναι μία σελίδα διαχείρισης της βάσης δεδομένων όπου είναι ορατά όλα τα μοντέλα της αλλά και όλοι οι πίνακες με τα περιεχόμενά τους ενώ παρέχεται και η δυνατότητα επεργασίας τους.



Εικόνα 4.1 - Django Administration Page

## 4.3 Επεξήγηση κώδικα

### 4.3.1 Εγγραφές μοντέλων

#### 4.3.1.1 Εγγραφή ενοικιαστή

Πριν γίνει δυνατή η εγγραφή ενός καινούριου ενοικιαστή είναι υποχρεωτικό να δημιουργηθεί το μοντέλο του στην βάση δεδομένων.

Ο τρόπος με τον οποίο δημιουργείται το μοντέλο αυτό είναι:

```
class LesseeProfile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE,)

    driving_license = models.ImageField(upload_to='driving_licenses', blank=True)
    average_cc = models.FloatField(blank=True, default=0)
    cars_count = models.IntegerField(blank=True, default=0)
    def __str__(self):
        return self.user.username
```

Εικόνα 4.2 - Δημιουργία μοντέλου ενοικιαστή

Η πρώτη γραμμή κώδικα ορίζει το όνομα του μοντέλου. Η επόμενη διευκρινίζει πως το μοντέλο αυτό θα είναι επέκταση του ήδη υπάρχοντος μοντέλου χρήστη που διαθέτει το Django. Στην συνέχεια πρέπει να προστεθούν τα επιπλέον πεδία που θα διαθέτει το μοντέλο αυτό, δηλαδή δίπλωμα οδήγησής του καθώς και δύο πεδία που θα χρειαστούν για την εφαρμογή του recommendation algorithm· το πλήθος των ενοικιασμένων αυτοκινητών και ο μέσος όρος των κυβικών τους.

Στην συνέχεια θα πρέπει να δημιουργηθεί η κατάλληλη φόρμα εγγραφής η οποία θα περιλαμβάνει τα πεδία του μοντέλου.

```
class LesseeForm(forms.ModelForm):
    class Meta():
        model = LesseeProfile
        fields = ('driving_license', 'average_cc', 'cars_count')
        widgets = {'average_cc': forms.HiddenInput(), 'cars_count': forms.HiddenInput}
```

Εικόνα 4.3 - Δημιουργία φόρμας εγγραφής ενοικιαστή

Στην πρώτη γραμμή κώδικα, όπως ίσχυε και στο μοντέλο, δίνουμε όνομα στην φόρμα αυτή. Στην συνέχεια προσθέτουμε τα πεδία που δεν ανήκουν στην υπάρχουσα φόρμα εγγραφής χρήστη η οποία φαίνεται στην επόμενη εικόνα.

```

class UserForm(forms.ModelForm):
    password = forms.CharField(widget=forms.PasswordInput())

    class Meta():
        model = User
        fields = ('username', 'email', 'first_name', 'last_name', 'password',)

```

Εικόνα 4.4 - Δημιουργία φόρμας εγγραφής χρήστη

Η φόρμα εγγραφής χρήστη περιλαμβάνει τα πεδία που φαίνονται στην τελευταία γραμμή.

Τα επιπλέον πεδία για την εγγραφή του ενοικιαστή είναι το δίπλωμα οδήγησης και τα δύο πεδία για την λειτουργία του recommendation algorithm τα οποία δεν πρέπει να είναι προσβάσιμα από την φόρμα για αυτό και θέτονται ως κρυφά.

Πλέον, είναι δυνατόν να δημιουργηθεί η συνάρτηση η οποία δημιουργεί τον χρήστη και τον ενθέτει στην βάση δεδομένων.

```

if request.method == 'POST':
    user_form = UserForm(data=request.POST)
    lessee_form = LesseeForm(data=request.POST)

    if user_form.is_valid() and lessee_form.is_valid():
        user = user_form.save()
        user.set_password(user.password)
        user.save()

        lessee = lessee_form.save(commit=False)
        lessee.user = user
        if 'driving_license' in request.FILES:
            lessee.driving_license = request.FILES['driving_license']

        lessee.save()

        registered = True
    else:
        print(user_form.errors, lessee_form.errors)

```

Εικόνα 4.5 - Λήψη στοιχείων εγγραφής ενοικιαστή

Σε αυτό το κομμάτι του κώδικα αφού γίνει η αποστολή των στοιχείων της φόρμας στον server, η συνάρτηση αυτή τα αποθηκεύει σε προσωρινές μεταβλητές (user\_form, lessee\_form), ελέγχει την εγκυρότητα των δοθέντων στοιχείων και εφόσον είναι έγκυρα αποθηκεύεται το περιεχόμενο της φόρμας στα σωστά σημεία

της βάσης. Σε περίπτωση, βέβαια, που δεν είναι έγκυρα τα στοιχεία αυτά τότε εκτυπώνεται μήνυμα λάθους στον server όπως φαίνεται στην τελευταία γραμμή τον παραπάνω κώδικα.

```
else:  
    user_form = UserForm()  
    lessee_form = LesseeForm()
```

Εικόνα 4.6 - Αποστολή κενής φόρμας εγγραφής ενοικιαστή

Στο κομμάτι αυτό δημιουργούνται οι κενές φόρμες που πρέπει να εμφανιστούν όταν φορτώσει η σελίδα εγγραφής ενοικιαστή και αφού συμπληρωθούν και γίνει υποβολή εκτελείται ο κώδικας που εξηγήθηκε προηγουμένως.

```
return render(request, 'myCarApp/lessee_registration.html',  
              {'user_form':user_form,  
               'lessee_form':lessee_form,  
               'registered':registered})
```

Εικόνα 4.7 - Επιστροφή απαραίτητων στοιχείων στην σελίδα εγγραφής ενοικιαστή

Τέλος, αφού ολοκληρωθεί κάποια από τις προηγούμενες δύο επιλογές, η συνάρτηση επιστρέφει στην σελίδα τις φόρμες, συμπληρωμένες στην πρώτη περίπτωση και άδειες στην δεύτερη, και την μεταβλητή `registered` η οποία είναι αρχικοποιημένη ως `False` και παίρνει την τιμή `True` εάν έγινε εγγραφή του χρήστη.

#### 4.3.1.2 Εγγραφή ενοικιάζοντος

Όπως ίσχυε και στην εγγραφή του ενοικιαστή πρέπει καταρχάς να δημιουργηθεί το μοντέλο του χρήστη αυτού στην βάση δεδομένων.

```
class LessorProfile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE,)
    def __str__(self):
        return self.user.username
```

Εικόνα 4.8 - Δημιουργία μοντέλου ενοικιάζοντος

Στην περίπτωση του ενοικιάζοντος, το μοντέλο δεν χρειάζεται κάποιο επιπλέον πεδίο από αυτά που περιέχονται στο υπάρχων μοντέλο του χρήστη. Έτσι, το μόνο που χρειάζεται να κάνει ο κώδικας είναι να ορίσει το όνομα του μοντέλου και να αντιστοιχήσει τα πεδία του με αυτά του μοντέλου του χρήστη.

Μόλις ολοκληρωθεί η διαδικασία αυτή, επαναλαμβάνοντας τα ίδια βήματα με την εγγραφή ενοικιαστή, πρέπει να δημιουργηθεί η σωστή φόρμα.

```
class LessorForm(forms.ModelForm):
    class Meta():
        model = LessorProfile
```

Εικόνα 4.9 - Δημιουργία φόρμας εγγραφής ενοικιάζοντος

Όπως έγινε και στο καινούριο μοντέλο, έτσι και στην καινούρια φόρμα δεν χρειάζονται επιπλέον πεδία από αυτά που παρέχονται. Έτσι μπορεί να δημιουργηθεί, πλέον, η συνάρτηση που αναλαμβάνει την εγγραφή του ενοικιάζοντος στην βάση δεδομένων.

```
registered=False

if request.method == 'POST':
    user_form = UserForm(data=request.POST)

    if user_form.is_valid():
        user = user_form.save()
        user.set_password(user.password)
        user.save()
        lessor = LessorProfile()
        lessor.user = user
        lessor.save()
        registered = True
    else:
        print(user_form.errors)
```

Εικόνα 4.10 - Λήψη στοιχείων εγγραφής ενοικιάζοντος

Σε αυτό το κομμάτι του κώδικα, εφόσον έχει γίνει υποβολή των στοιχείων από τον χρήστη, η συνάρτηση τα λαμβάνει και ελέγχει την εγκυρότητα τους. Στην περίπτωση που είναι έγκυρα τα αποθηκεύει στο σωστό μοντέλο της βάσης. Ωστόσο, εάν δεν είναι θα εκτυπωθεί μήνυμα λάθους στον server.

```
else:  
    user_form = UserForm()
```

Εικόνα 4.11 - Αποστολή κενής φόρμας εγγραφής ενοικιάζοντος

Η συνέχεια του κώδικα εκτελείται όταν φορτώνει η σελίδα της εγγραφής χωρίς να έχουν δοθεί ακόμα στοιχεία οπότε πρέπει να αποσταλεί η κενή φόρμα και αφού συμπληρωθεί θα επαναφορτωθεί η σελίδα και θα λειτουργήσει, ουσιαστικά, το κομμάτι του κώδικα που περιγράφεται στην εικόνα 4.10.

Τέλος, η συνάρτηση αυτή κλείνει με την επιστροφή της φόρμας είτε άδειας είτε συμπληρωμένης αλλά και της μεταβλητής `registered` η οποία είναι αρχικοποιημένη ως `False` και με την επιτυχημένη εγγραφή του χρήστη αλλάζει σε `True`.

```
return render(request, 'myCarApp/lessor_registration.html',  
              {'user_form': user_form,  
               'registered': registered})
```

Εικόνα 4.12 - Επιστροφή απαραίτητων στοιχείων στην σελίδα εγγραφής ενοικιάζοντος

#### 4.3.1.3 Εγγραφή καινούριου αυτοκινήτου

Ένα ακόμα μοντέλο που χρειάζεται να κάνει εγγραφές αντικειμένων στην βάση είναι το αυτοκίνητο. Συγκεκριμένα, στον ενοικιάζοντα δίνεται η δυνατότητα να καταχωρεί αυτοκίνητα στην βάση παρέχοντας κάποιες πληροφορίες για αυτά.

Το αυτοκίνητο είναι ένα αρκετά πιο πολύπλοκο μοντέλο από τα δύο μοντέλα χρηστών κι αυτό διότι δεν βασίζεται σε κάποιο έτοιμο μοντέλο της Django και επίσης περιλαμβάνει αρκετά διαφορετικά πεδία.

```
class Car(models.Model):
    brand = models.CharField(max_length = 30, choices=BRAND_CHOICES)
    model = models.CharField(max_length = 30)
    CC = models.IntegerField()
    year = models.IntegerField(choices=YEAR_CHOICES, default=2020)
    color = models.CharField(max_length = 10, choices=COLOR_CHOICES)
    transmission = models.CharField(max_length = 10, choices=TRANSMISSION_CHOICES)
    fuelType = models.CharField(max_length = 10, choices=FUEL_CHOICES)
    firstAvailableDay = models.DateField()
    lastAvailableDay = models.DateField()
    pricePerDay = models.FloatField()
    photo = models.ImageField(upload_to='car_photos', blank=True)
```

Εικόνα 4.13 - Δημιουργία μοντέλου αυτοκινήτου

Συγκεκριμένα είναι απαραίτητο για κάθε αμάξι να δίνεται η μάρκα και το μοντέλο του καθώς και κάποια βασικά χαρακτηριστικά του· τα κυβικά, το έτος της αγοράς, το χρώμα του, το είδος του κιβωτίου ταχυτήτων αλλά και το καύσιμο που καταναλώνεται.

Πέραν, όμως, από τα βιομηχανικά χαρακτηριστικά του αυτοκινήτου χρειάζεται ο ενοικιάζων να καταχωρήσει σε ποιο διάστημα θα είναι διαθέσιμο το αμάξι του προς ενοικίαση, την τιμή στην οποία θέλει να το διαθέσει, την ημέρα, καθώς πρέπει να καταχωρήσει και μία φωτογραφία του.

Βεβαίως, ορισμένα από τα προαναφερθέντα πεδία του μοντέλου του αυτοκινήτου δεν μπορούν να συμπληρωθούν αυθαίρετα αλλά αντιθέτως παρέχονται προκαθορισμένες επιλογές από τις οποίες μπορεί να διαλέξει οι οποίες φαίνονται στις επόμενες εικόνες.

```
TRANSMISSION_CHOICES = [('MANUAL', 'Manual'), ('AUTO', 'Auto'),]
FUEL_CHOICES = [('GASOLINE', 'Gasoline'), ('DIESEL', 'Diesel'), ('GAS', 'Gas'), ('ELECTRIC', 'Electric'), ('HYBRID', 'Hybrid'),]
YEAR_CHOICES = [(i, i) for i in range(1990, 2020)]
```

Εικόνα 4.14 - Επιλογές για το κιβώτιο ταχυτήτων, τον τύπο καυσίμου και την χρονολογία



```
BRAND_CHOICES = [('ABARTH', 'Abarth'),('ALFA ROMEO', 'Alfa Romeo'),('AUDI', 'Audi'),('BMW', 'BMW'),('CHEVROLET', 'Chevrolet'),('CITROEN', 'Citroen'),('DAIHATSU', 'Daihatsu'),('FIAT', 'Fiat'),('HONDA', 'Honda'), ('HYUNDAI', 'Hyundai'),('JEEP', 'Jeep'),('LANCIA', 'Lancia'),('LAND ROVER', 'Land Rover'),('MAZDA', 'Mazda'), ('MERCEDES', 'Mercedes'),('MINI', 'Mini'),('MITUBISHI', 'Mitsubishi'),('NISSAN', 'Nissan'),('OPEL', 'Opel'), ('PEUGEOT', 'Peugeot'),('RENAULT', 'Renault'),('SEAT', 'Seat'),('SKODA', 'Skoda'),('SMART', 'Smart'), ('SUZUKI', 'Suzuki'),('TOYOTA', 'Toyota'),('VOLKSWAGEN', 'Volkswagen'),('VOLVO', 'Volvo'),]
```

Εικόνα 4.15 - Επιλογές για την μάρκα του αυτοκινήτου

```
COLOR_CHOICES = [('BLACK', 'Black'),('BLUE', 'Blue'),('BROWN', 'Brown'),('GOLD', 'Gold'),('GREY', 'Grey'), ('RED', 'Red'),('SILVER', 'Silver'),('WHITE', 'White'),('YELLOW', 'Yellow'),]
```

Εικόνα 4.16 - Επιλογές για το χρώμα του αυτοκινήτου

Αφού ολοκληρωθεί η δημιουργία του μοντέλου του αυτοκινήτου μπορεί να δημιουργηθεί και η κατάλληλη φόρμα εγγραφής καινούριου αντικειμένου στην βάση δεδομένων.

```
class CarRegistrationForm(forms.ModelForm):  
    class Meta():  
        model = Car  
        fields = ('brand', 'model', 'CC', 'year', 'color', 'transmission', 'fuelType', 'firstAvailableDay', 'lastAvailableDay', 'pricePerDay', 'photo')  
        widgets = {  
            'firstAvailableDay': forms.SelectDateWidget(),  
            'lastAvailableDay': forms.SelectDateWidget(),  
        }
```

Εικόνα 4.17 - Δημιουργία φόρμας εγγραφής αυτοκινήτου

Στην παραπάνω εικόνα φαίνεται ο κώδικας που δημιουργεί την φόρμα εγγραφής η οποία περιλαμβάνει τα πεδία που περιλαμβάνονται στο fields. Στο widgets δίνεται η μορφή που θα έχουν τα πεδία που ορίζουν το διάστημα στο οποίο είναι διαθέσιμο το αυτοκίνητο. Δηλαδή, διευκρινίζεται ότι οι ημερομηνίες θα επιλέγονται με τον τρόπο που επιβάλλει το συγκεκριμένο widget.

Μόλις ολοκληρωθεί και η διαμόρφωση της φόρμας, το επόμενο βήμα είναι η δημιουργία της συνάρτησης η οποία θα λάβει τα στοιχεία του καινούριου αντικειμένου και θα τα καταχωρήσει στην βάση δεδομένων.

```
@login_required  
def car_registration(request):  
    car_is_registered = False  
    if request.method == 'POST':  
        car_registration_form = CarRegistrationForm(data=request.POST)  
  
        if car_registration_form.is_valid():  
            car = car_registration_form.save()  
            car.save()  
            car_is_registered = True  
        else:  
            print(car_registration_form.errors)
```

Εικόνα 4.18 - Λήψη στοιχείων καινούριου αυτοκινήτου

Καταρχάς στον κώδικα χρησιμοποιείται το token @login\_required το οποίο διευκρινίζει ότι για να είναι δυνατή η εκτέλεση της συνάρτησης αυτής θα πρέπει να είναι συνδεδεμένος κάποιος χρήστης. Φυσικά ο λόγος που υπάρχει αυτό το token είναι για να αποφευχθούν πιθανά λάθη. Η συνάρτηση αυτή εάν ληφθεί υποβολή στοιχείων από τον χρήστη τα ελέγχει και εφόσον είναι έγκυρα δημιουργεί το καινούριο αντικείμενο με τα στοιχεία αυτά , το αποθηκεύει και το καταχωρεί στην βάση δεδομένων. Σε αντίθετη περίπτωση εκτυπώνεται μήνυμα λάθους στον sever.

Εφόσον δεν ληφθεί υποβολή στοιχείων, θα πρέπει να δημιουργηθεί η κενή φόρμα την οποία πρέπει να συμπληρώσει ο χρήστης με τα χαρακτηριστικά του αυτοκινήτου του.

```
else:  
    car_registration_form = CarRegistrationForm()
```

Εικόνα 4.19 - Δημιουργία κενής φόρμας καταχώρησης αυτοκινήτου

Στο τέλος, η συνάρτηση αυτή, επιστρέφει την κατάλληλη φόρμα, τα στοιχεία του χρήστη καθώς και την μεταβλητή car\_is\_registered η οποία είναι αρχικοποιημένη ως False και παίρνει την τιμή True με το που γίνει επιτυχής δημιουργία του καινούριου αντικειμένου.

```
context={'username': request.user.username, 'type':type, 'title':"Car Registration",  
'car_registration_form':car_registration_form, 'car_is_registered':car_is_registered}  
return render(request, 'myCarApp/car_registration.html', context)
```

Εικόνα 4.20 - Επιστροφή συνάρτησης δημιουργίας αντικειμένου

#### 4.3.2 Σύνδεση χρήστη

Αφού ένας χρήστης είναι επιτυχώς καταχωρημένος στην βάση δεδομένων του δίνεται η δυνατότητα να συνδεθεί στην εφαρμογή. Για να υλοποιηθεί η λειτουργία αυτή πρέπει να δημιουργηθεί μία συνάρτηση η οποία ελέγχει αν τα στοιχεία που δόθηκαν από τον χρήστη είναι έγκυρα και αντιστοιχούν σε κάποιο αντικείμενο.

```

def user_login(request):
    if request.method == 'POST':
        username = request.POST.get('username')
        password = request.POST.get('password')

        user = authenticate(username=username, password=password)

        if user:
            if user.is_active:
                login(request, user)
                return HttpResponseRedirect(reverse('index'))
            else:
                return HttpResponse("Your account is not active.")
        else:
            print("Someone tried to login and failed.")
            print("They used username: {} and password: {}".format(username, password))
            return HttpResponseRedirect(reverse('login'))

    else:
        return render(request, 'myCarApp/login.html', {})

```

Εικόνα 4.21 - Συνάρτηση σύνδεσης χρήστη

Εάν τα στοιχεία είναι σωστά και ο χρήστης στον οποίο αντιστοιχούν είναι ενεργός τότε ανακατευθύνεται στην αρχική σελίδα. Σε περίπτωση που τα στοιχεία είναι σωστά αλλά ο χρήστης δεν είναι ενεργός εμφανίζεται κατάλληλο μήνυμα. Στην περίπτωση, δε, που τα στοιχεία που δόθηκαν δεν αντιστοιχούν σε κάποιο αντικείμενο, εμφανίζεται στον server μήνυμα με τα στοιχεία που δόθηκαν και ο χρήστης ανακατευθύνεται και πάλι στην σελίδα σύνδεσης.

#### 4.3.3 Αποσύνδεση χρήστη

Μία εξίσου βασική λειτουργία με την σύνδεση χρήστη είναι η αποσύνδεση του. Για την υλοποίηση της χρειάζεται επίσης μία συνάρτηση η οποία την αναλαμβάνει.

```

@login_required
def user_logout(request):
    logout(request)
    return HttpResponseRedirect(reverse('index'))

```

Εικόνα 4.22 - Συνάρτηση αποσύνδεσης χρήστη

Το Django διαθέτει την εντολή `logout(request)`, η οποία κάνει από μόνη της όλες τις απαραίτητες ενέργειες για να γίνει επιτυχής αποσύνδεση χρήστη. Αφού ολοκληρωθεί αυτό, γίνεται ανακατεύθυνση στην αρχική σελίδα της εφαρμογής. Επίσης, όπως φαίνεται στην εικόνα για να λειτουργήσει η συνάρτηση αυτή

χρησιμοποιείται το token `@login_required` έτσι ώστε να διασφαλίζεται ότι είναι κάποιος συνδεδεμένος όταν καλείται αποφεύγοντας πιθανά λάθη.

#### 4.3.4 Αναζήτηση αυτοκινήτου

Απαραίτητη για την εφαρμογή είναι μία σελίδα στην οποία θα μπορεί ο μελλοντικός ενοικιαστής να επιλέξει το αυτοκίνητο που επιθυμεί. Φυσικά, για μεγαλύτερη διευκόλυνση του χρήστη είναι ιδανικό η σελίδα αυτή να του παρέχει την δυνατότητα να εφαρμόσει διάφορα φίλτρα στην αναζήτησή του έτσι ώστε να δει μόνο τα αποτελέσματα που τον ενδιαφέρουν. Ως αποτέλεσμα πρέπει να δημιουργηθεί μία συνάρτηση η οποία θα εφαρμόζει τα φίλτρα που επιλέχθηκαν.

```
@login_required
def find_car(request):
    searchResult = Car.objects.all()
    if request.method == 'POST':
        firstDay = request.POST.get('firstDay')
        lastDay = request.POST.get('lastDay')
        CC = request.POST.get('CC')
        year = request.POST.get('Year')
        fuelType = request.POST.get('fuelType')
        pricePerDay = request.POST.get('pricePerDay')
        Transmission = request.POST.get('Transmission')
```

Εικόνα 4.23 - Λήψη φίλτρων από τον χρήστη

Αρχικά, όπως φαίνεται στην εικόνα, για να λειτουργήσει η εν λόγω συνάρτηση θα πρέπει να είναι συνδεδεμένος κάποιος χρήστης κάτι το οποίο διασφαλίζεται με το token `@login_required` που έχει αναφερθεί και προηγουμένως. Στην συνέχεια δημιουργείται το `querySet` το οποίο θα φιλτραριστεί ανάλογα με τις επιλογές του χρήστη και στην αρχή περιέχει όλα τα αυτοκίνητα της βάσης δεδομένων.

Εφόσον ο χρήστης έχει υποβάλλει φίλτρα αυτά λαμβάνονται μέσω της μεθόδου `POST` και αποθηκεύονται σε κατάλληλες μεταβλητές όπως φαίνεται στον κώδικα.

Ο χρήστης προκειμένου να υποβληθεί αίτημα για αναζήτηση αυτοκινήτου είναι υποχρεωμένος να καταχωρήσει τις ημερομηνίες τις οποίες επιθυμεί να ενοικιάσει αυτοκίνητο. Έτσι αυτό είναι το πρώτο φίλτρο το οποίο εφαρμόζεται, υποχρεωτικά.

```
searchResult = Car.objects.filter(firstAvailableDay__lte=firstDay)
searchResult = searchResult.filter(lastAvailableDay__gte=lastDay)
```

Εικόνα 4.24 - Εφαρμογή φίλτρου ημερομηνιών

Στην συνέχεια ελέγχονται τα υπόλοιπα πεδία στα οποία μπορεί να έχει καταχωρήσει κάποιο φίλτρο ο χρήστης.

```
if CC == '500-1000':
    print("500-1000")
    searchResult = searchResult.filter(CC__gte=500).filter(CC__lte=1000)
elif CC == '1000-1500':
    print("1000-1500")
    searchResult = searchResult.filter(CC__gte=1000).filter(CC__lte=1500)
elif CC == '1500-2000':
    print("1500-2000")
    searchResult = searchResult.filter(CC__gte=1500).filter(CC__lte=2000)
elif CC == '2000+':
    print("2000+")
    searchResult = searchResult.filter(CC__gt=2000)
```

Εικόνα 4.25 - Εφαρμογή φίλτρου κυβικών

Όπως φαίνεται στην εικόνα ανάλογα με τα διαστήματα κυβικών που διατίθενται στον χρήστη για επιλογή εφαρμόζονται τα κατάλληλα φίλτρα στο queryset έτσι ώστε να εξαλειφθούν αντικείμενα που δεν πληρούν τις προϋποθέσεις. Με τον ίδιο τρόπο εφαρμόζονται και τα υπόλοιπα φίλτρα τα οποία περιλαμβάνουν την χρονιά αγοράς του αυτοκινήτου, τον τύπο του καυσίμου, την τιμή ανά ημέρα καθώς και τον τύπο του κιβώτιου ταχυτήτων.

Σε περίπτωση που η εφαρμογή των φίλτρων έχει ως αποτέλεσμα να επιστραφεί άδειο queryset, δηλαδή δεν υπάρχει αυτοκίνητο στην βάση δεδομένων που να πληρεί όλες τις επιθυμίες του χρήστη, εμφανίζεται κατάλληλο μήνυμα.

```
if(not searchResult):
    message = "Your search returned no results"
```

Εικόνα 4.26 - Έλεγχος άδειου queryset

Όπως είναι λογικό, στην συνάρτηση αυτή θα εφαρμοστεί και ο recommendation algorithm. Αφού εφαρμοστούν στο querySet τα φίλτρα που επέλεξε ο χρήστης θα πρέπει να στοιχεία του να αναδιαταχθούν ανάλογα με τον μέσο όρο των κυβικών των αυτοκινήτων που έχει νοικιάσει ο χρήστης. Σε αυτό βοηθάει το πεδίο που έχει προστεθεί στο μοντέλο του χρήστη που έχει αποθηκευμένη ανά πάσα στιγμή την τιμή αυτή.

```
searchResult = searchResult.annotate(  
    averagegap = Func(ExpressionWrapper(F('CC') - currentAverageCc, output_field=FloatField()), function='ABS')  
)
```

Εικόνα 4.27 - Δημιουργία προσωρινού πεδίου στο μοντέλο του αυτοκινήτου

Ο κώδικας που φαίνεται στην παραπάνω εικόνα δημιουργεί ένα προσωρινό πεδίο που σχετίζεται με το μοντέλο του αυτοκινήτου και με τον εκάστοτε συνδεδεμένο χρήστη. Στην μεταβλητή currentAverageCc υπάρχει ο μέσος όρος των κυβικών από τις ενοικιάσεις του συνδεδεμένου χρήστη. Στην συνέχεια υπολογίζεται η διαφορά της τιμής αυτής με τα κυβικά του κάθε αυτοκινήτου που υπάρχει στο querySet.

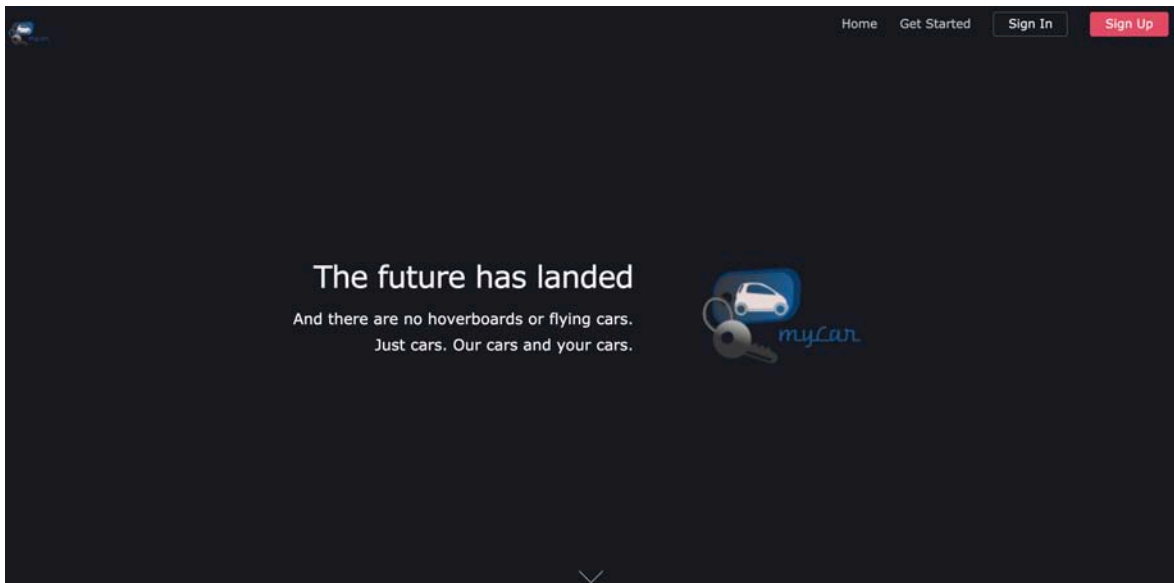
```
searchResult = searchResult.order_by('averagegap', 'pricePerDay', '-year')
```

Εικόνα 4.28 - Αναδιάταξη των αποτελεσμάτων

Τέλος, τα αντικείμενα του querySet αναδιατίθενται με την σειρά προτεραιότητας που περιγράφηκε στο προηγούμενο κεφάλαιο. Δηλαδή, ο βασικός παράγοντας είναι η διαφορά των κυβικών του κάθε αυτοκινήτου από τον μέσο όρο των κυβικών των αυτοκινήτων που έχει ενοικιάσει ο χρήστης. Από εκεί και πέρα αν υπάρχει ισοβαθμία στον παράγοντα αυτό λαμβάνεται υπόψιν η τιμή του κάθε αυτοκινήτου και τέλος η χρονολογία του.

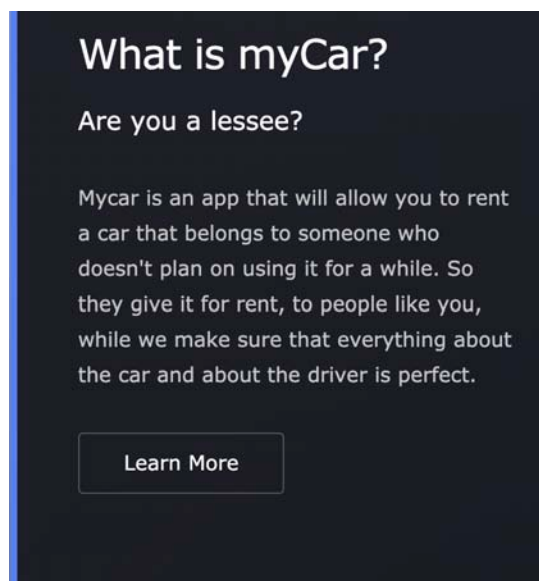
#### 4.4 Γενική παρουσίαση εφαρμογής

Η αρχική σελίδα της εφαρμογής περιλαμβάνει ως επί το πλείστον γενικές πληροφορίες σχετικά με την εφαρμογή και λεπτομέρειες για τον τρόπο που λειτουργεί. Επιπλέον, δίνει την δυνατότητα στον χρήστη να εγγραφεί ή να συνδεθεί.



Εικόνα 4.29 - 1ο μέρος αρχικής σελίδας

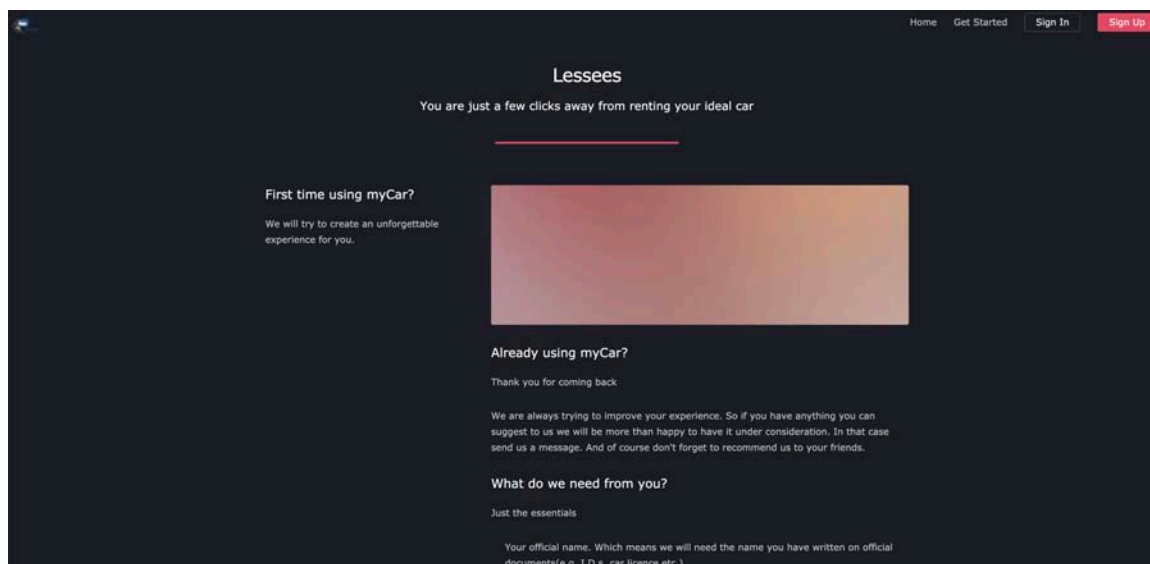
Κατεβαίνοντας πιο κάτω στην αρχική σελίδα εμφανίζονται γενικές πληροφορίες όπως φαίνεται στην επόμενη εικόνα.



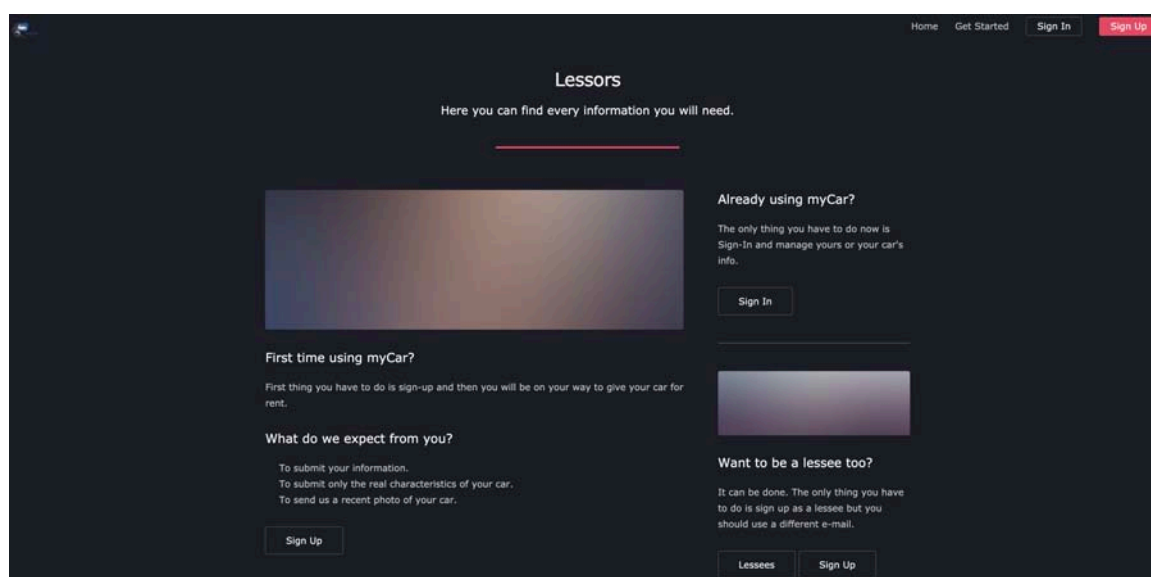
Εικόνα 4.30 - Γενικές πληροφορίες για τον ενοικιαστή

Όσο για τις επιλογές που υπάρχουν στην μπάρα της αρχικής σελίδας ο χρήστης μπορεί είτε να δει περισσότερες πληροφορίες για την εφαρμογή επιλέγοντας το 'Get Started', μπορεί να μεταφερθεί στην σελίδα σύνδεσης ή να κάνει εγγραφή διαλέγοντας εάν επιθυμεί να είναι ενοικιαστής ή ενοικιάζων.

Εφόσον επιλέξει το Get Started θα εμφανιστούν δύο υποεπιλογές οι οποίες θα τον ανακατευθύνουν στην σελίδα πληροφοριών για ενοικιαστή ή ενοικιάζων.



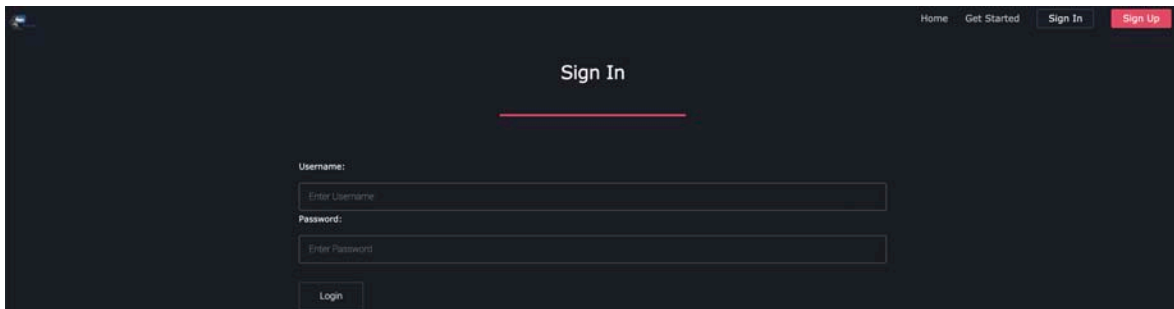
Εικόνα 4.31 - Σελίδα πληροφοριών ενοικιαστή



Εικόνα 4.32 - Σελίδα πληροφοριών ενοικιάζοντος

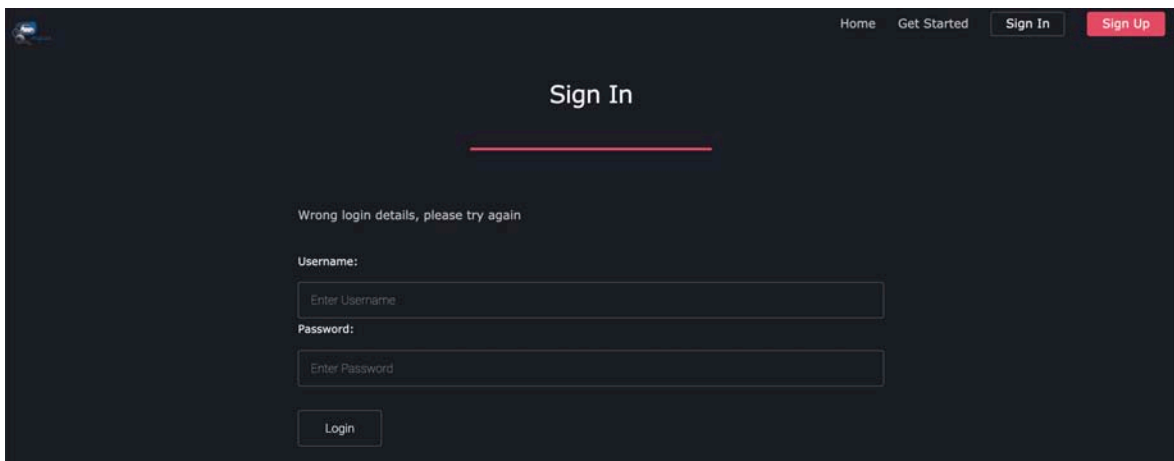
Στην επιλογή του Sign In ο χρήστης θα ανακατευθυνθεί στην σελίδα σύνδεσης στην οποία, εφόσον είναι ενεργός χρήστης της εφαρμογής, μπορεί να συνδεθεί χρησιμοποιώντας τα στοιχεία που είχε καταχωρήσει κατά την εγγραφή του.





Εικόνα 4.33 - Σελίδα σύνδεσης χρήστη

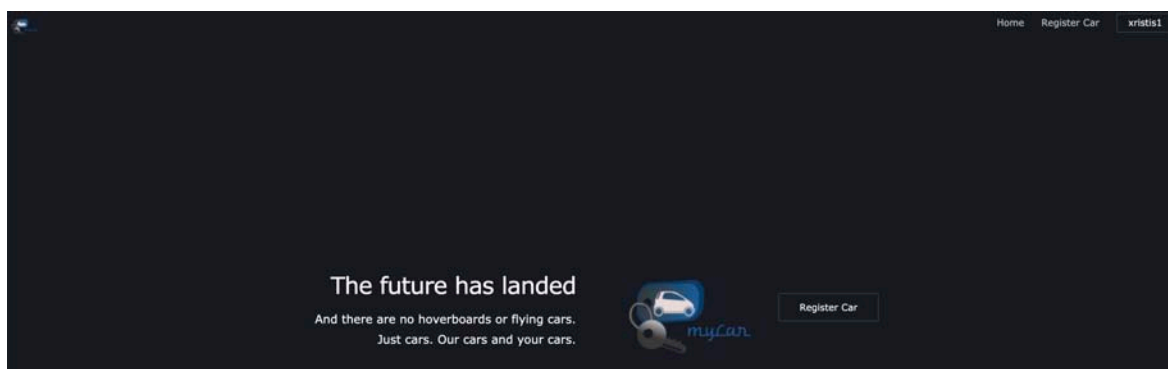
Εάν τα στοιχεία καταχώρησης του χρήστη δεν αντιστοιχούν σε κάποιο αντικείμενο της βάσης τότε ανακατευθύνεται στην σελίδα σύνδεσης για να δοκιμάσει να κάνει εκ νέου σύνδεση ενώ εμφανίζεται και κατάλληλο μήνυμα



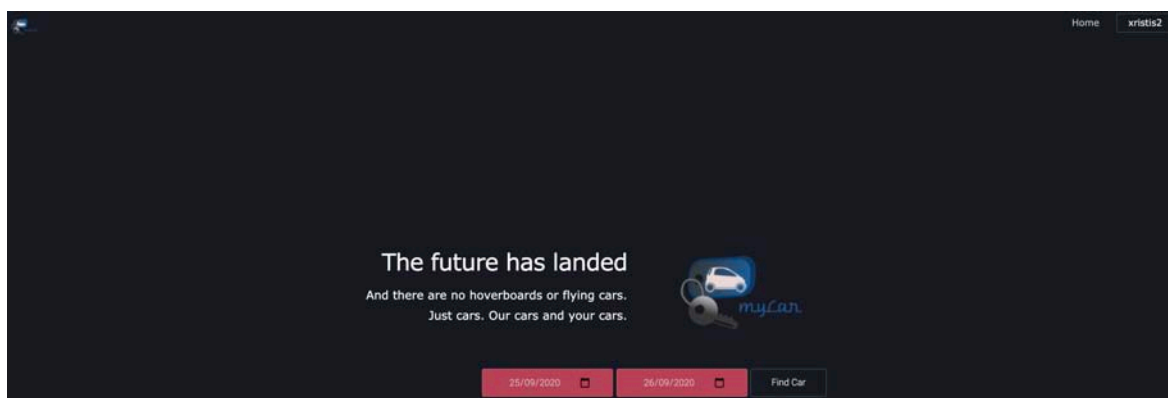
Εικόνα 4.34 - Σελίδα σύνδεσης χρήστη μετά από αποτυχημένη προσπάθεια σύνδεσης

Εάν γίνει επιτυχής σύνδεση θα μεταφερθεί στην αρχική σελίδα η οποία όμως θα του δίνει κάποιες επιπλέον δυνατότητες ανάλογα με το είδος του(ενοικιαστής ή ενοικιάζων).

Συγκεκριμένα, ο ενοικιάζων έχει την δυνατότητα να μεταφερθεί στην σελίδα καταχώρησης νέου αυτοκινήτου ενώ ο ενοικιαστής στην σελίδα αναζήτησης αυτοκινήτου για συγκεκριμένες ημερομηνίες.



Εικόνα 4.35 - Αρχική σελίδα ενοικιάζοντος



Εικόνα 4.36 - Αρχική σελίδα ενοικιαστή

Και οι δύο χρήστες μπορούν να δουν το username τους πάνω δεξιά και πλησιάζοντας τον κέρσορά τους εκεί θα εμφανιστεί υποεπιλογή η οποία τους δίνει την δυνατότητα να αποσυνδεθούν και ανακατευθυνθούν στην αρχική σελίδα που δείχνει η εικόνα 4.29.

Όσον αφορά την εγγραφή του χρήστη υπάρχουν δύο διαφορετικές σελίδες ανάλογα με το είδος του χρήστη που θα εγγραφεί.

Register Here if you want to be a lessee

Username:  
  
Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Email address:

First name:

Last name:

Password:

Driving license:  
 Επιλογή οργάνου δεν επιλέχθηκε... ανένα οργάνο

Εικόνα 4.37 - Φόρμα εγγραφής ενοικιαστή

Register Here if you want to be a lessor

Username:  
  
Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Email address:

First name:

Last name:

Password:

Εικόνα 4.38 - Φόρμα εγγραφής ενοικιάζοντος

Εφόσον ο χρήστης συνδεθεί ως ενοικιάζων επιτυχώς μπορεί πλέον να καταχωρήσει ένα αυτοκίνητο προς ενοικίαση επιλέγοντας το Register Car. Στην περίπτωση αυτή θα πρέπει να συμπληρώσει μία φόρμα η οποία ζητάει κάποια βασικά στοιχεία του αυτοκινήτου τις ημερομηνίες που θα είναι διαθέσιμο μία φωτογραφία του καθώς και την τιμή του ανά ημέρα ενοικίασης

### Register Your Car

Brand:

Model:

CC:

Year:

Color:

Transmission:

FuelType:

FirstAvailableDay:

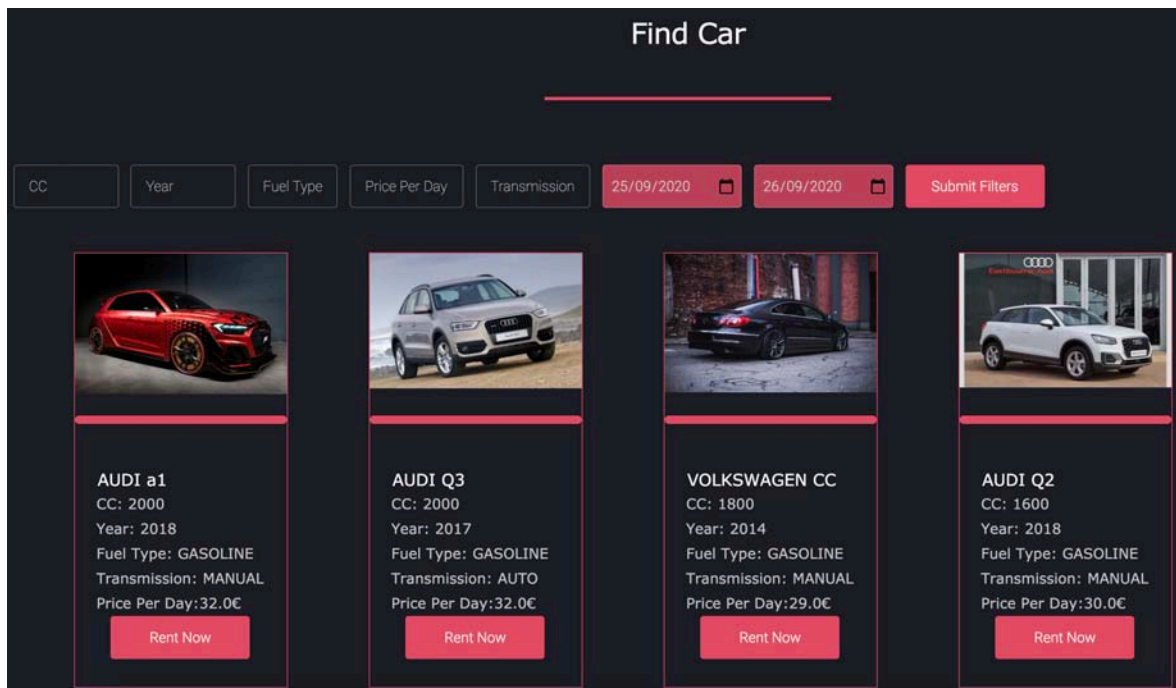
LastAvailableDay:

PricePerDay:

Photo:  Μην επιλέγεται κανένα αρχείο

Εικόνα 4.39 - Φόρμα καταχώρησης αυτοκινήτου

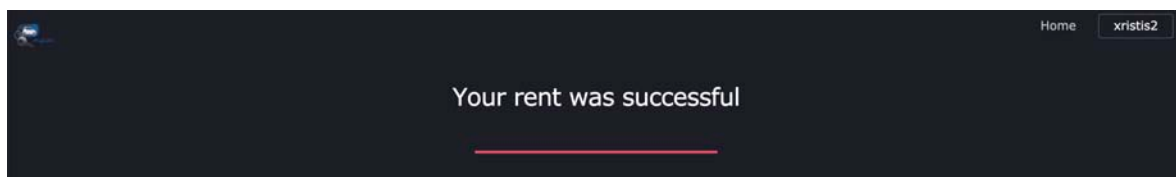
Στην περίπτωση που έχει συνδεθεί επιτυχώς χρήστης ως ενοικιαστής μπορεί να κάνει αναζήτηση για αυτοκίνητο δίνοντας τις ημερομηνίες που το χρειάζεται. Έτσι θα εμφανιστεί μία σελίδα η οποία θα περιλαμβάνει τα αυτοκίνητα που είναι διαθέσιμα στις ημερομηνίες αυτές. Στην σελίδα αυτή ο χρήστης θα μπορέσει να καταχωρήσει και τα φίλτρα που επιθυμεί στην αναζήτησή του.



Εικόνα 4.40 - Σελίδα αναζήτησης αυτοκινήτων

Ο χρήστης μπορεί να επιλέξει όσα φίλτρα επιθυμεί από αυτά που παρέχονται και στο τέλος επιλέγοντας Submit Filters γίνεται εκ νέου αναζήτηση.

Εάν εν τελεί καταλήξει στην ενοικίαση κάποιου αυτοκινήτου επιλέγοντας το Rent Now τότε εμφανίζεται μήνυμα επιτυχημένης ενοικίασης.



Εικόνα 4.41 - Μήνυμα επιτυχημένης ενοικίασης

## 4.5 Επίπεδο της εφαρμογής

Η εφαρμογή που παρουσιάζεται είναι υλοποιημένη σε έναν πολύ μεγάλο βαθμό, ωστόσο δεν είναι ολοκληρωμένη. Αυτό συμβαίνει διότι εφόσον δημιουργήθηκε στα πλαίσια διπλωματικής εργασίας δεν έχει επιχειρηματική υπόσταση. Επομένως δεν υπάρχει τρόπος να υλοποιηθούν τυχόν πληρωμές που θα έπρεπε να γίνονται στα πλαίσια μίας ολοκληρωμένης εφαρμογής.

Σκοπός της εργασίας αυτής είναι να δείξει με ποιον τρόπο λειτουργούν κάποια βασικά σημεία μίας εφαρμογής διαμοιρασμού αυτοκινήτου καθώς και να επιδείξει με ποιον τρόπο γίνεται να εφαρμοστούν recommendation algorithms σε τέτοιου τύπου εφαρμογές

## 4.6 Χρήση και δοκιμή της εφαρμογής

Το να είναι μία εφαρμογή στο διαδίκτυο και προσβάσιμη από όλους είναι δυστυχώς μία κοστοβόρα διαδικασία. Επομένως δεν είναι δυνατόν να βρεθεί σε κάποιον ιστότοπο.

Όμως, όλος ο κώδικας της εφαρμογής μπορεί να βρεθεί από οποιονδήποτε στο <https://github.com/irokas/myCarApp>. Μέσω του GitHub μπορούν όλοι να κατεβάσουν τα αρχεία που δημιουργήθηκαν και εφόσον διαθέτουν τα απαραίτητα εργαλεία να το δουν να λειτουργεί τοπικά, στον δικό τους υπολογιστή.

Τα εργαλεία που θα χρειαστούν για κάτι τέτοιο θα είναι ένα τερματικό, python3 αλλά και ένας φυλλομετρητής.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Wikipedia - Communities:  
<https://en.wikipedia.org/wiki/Community>
2. Wikipedia - Online Communities:  
[https://en.wikipedia.org/wiki/Online\\_community](https://en.wikipedia.org/wiki/Online_community)
3. Smart Insights - Global social media research summary July 2020:  
<https://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/>
4. Quora:  
<https://www.quora.com/How-many-people-own-a-car-in-the-world-compared-to-the-number-of-people-that-live-on-this-Earth>
5. Wikipedia - Car rental:  
[https://en.wikipedia.org/wiki/Car\\_rental](https://en.wikipedia.org/wiki/Car_rental)
6. Automotive Fleet - Car Renting... Its Development... And Future  
<https://www.automotive-fleet.com/147063/car-renting-its-development-and-future>
7. Wikipedia - Car sharing:  
<https://en.wikipedia.org/wiki/Carsharing>
8. Wikipedia - Recommender System  
[https://en.wikipedia.org/wiki/Recommender\\_system](https://en.wikipedia.org/wiki/Recommender_system)

## **ΠΑΡΑΡΤΗΜΑΤΑ**



## ΠΑΡΑΡΤΗΜΑ Α

### ΛΙΣΤΑ ΓΡΑΦΗΜΑΤΩΝ

Σχήμα 1.1: Γράφημα Ποσοστού Χρηστών Διαδικτύου ανά 100 κατοίκους. xxii

Πηγή: [https://en.wikipedia.org/wiki/Global\\_Internet\\_usage](https://en.wikipedia.org/wiki/Global_Internet_usage) xxii

Σχήμα 2.1: Γράφημα Χρηστών Διαμοιρασμού Αυτοκινήτων στις ΗΠΑ xxvii

Πηγή: <https://en.wikipedia.org/wiki/Carsharing#History> xxvii

## ΠΑΡΑΡΤΗΜΑ Β

### ΛΙΣΤΑ ΕΙΚΟΝΩΝ

- Εικόνα 4.1 - Django Administration Page*
- Εικόνα 4.2 - Δημιουργία μοντέλου ενοικιαστή*
- Εικόνα 4.3 - Δημιουργία φόρμας εγγραφής ενοικιαστή*
- Εικόνα 4.4 - Δημιουργία φόρμας εγγραφής χρήστη*
- Εικόνα 4.5 - Λήψη στοιχείων εγγραφής ενοικιαστή*
- Εικόνα 4.6 - Αποστολή κενής φόρμας εγγραφής ενοικιαστή*
- Εικόνα 4.7 - Επιστροφή απαραίτητων στοιχείων στην σελίδα εγγραφής ενοικιαστή*
- Εικόνα 4.8 - Δημιουργία μοντέλου ενοικιάζοντος*
- Εικόνα 4.9 - Δημιουργία φόρμας εγγραφής ενοικιάζοντος*
- Εικόνα 4.10 - Λήψη στοιχείων εγγραφής ενοικιάζοντος*
- Εικόνα 4.11 - Αποστολή κενής φόρμας εγγραφής ενοικιάζοντος*
- Εικόνα 4.12 - Επιστροφή απαραίτητων στοιχείων στην σελίδα εγγραφής ενοικιάζοντος*
- Εικόνα 4.13 - Δημιουργία μοντέλου αυτοκινήτου*
- Εικόνα 4.14 - Επιλογές για το κιβώτιο ταχυτήτων, τον τύπο καύσιμου και την χρονολογία*
- Εικόνα 4.15 - Επιλογές για την μάρκα του αυτοκινήτου*
- Εικόνα 4.16 - Επιλογές για το χρώμα του αυτοκινήτου*
- Εικόνα 4.17 - Δημιουργία φόρμας εγγραφής αυτοκινήτου*
- Εικόνα 4.18 - Λήψη στοιχείων καινούριου αυτοκινήτου*
- Εικόνα 4.19 - Δημιουργία κενής φόρμας καταχώρισης αυτοκινήτου*
- Εικόνα 4.20 Επιστροφή συνάρτησης δημιουργίας αντικειμένου*
- Εικόνα 4.21 - Συνάρτηση σύνδεσης χρήστη*
- Εικόνα 4.22 - Συνάρτηση αποσύνδεσης χρήστη*
- Εικόνα 4.23 - Λήψη φίλτρων από τον χρήστη*
- Εικόνα 4.24 - Εφαρμογή φίλτρου ημερομηνιών*
- Εικόνα 4.25 - Εφαρμογή φίλτρου κυβικών*
- Εικόνα 4.26 - Έλεγχος άδειου queryset*
- Εικόνα 4.27 - Δημιουργία προσωρινού πεδίου στο μοντέλο του αυτοκινήτου*
- Εικόνα 4.28 - Αναδιάταξη των αποτελεσμάτων*
- Εικόνα 4.29 - 1ο μέρος αρχικής σελίδας*
- Εικόνα 4.30 - Γενικές πληροφορίες για τον ενοικιαστή*

***Εικόνα 4.31 - Σελίδα πληροφοριών ενοικιαστή***

***Εικόνα 4.32 - Σελίδα πληροφοριών ενοικιάζοντος***

***Εικόνα 4.33 - Σελίδα σύνδεσης χρήστη***

***Εικόνα 4.34 - Σελίδα σύνδεσης χρήστη μετά από αποτυχημένη προσπάθεια σύνδεσης***

***Εικόνα 4.35 - Αρχική σελίδα ενοικιάζοντος***

***Εικόνα 4.36 - Αρχική σελίδα ενοικιαστή***

***Εικόνα 4.37 - Φόρμα εγγραφής ενοικιαστή***

***Εικόνα 4.38 - Φόρμα εγγραφής ενοικιάζοντος***

***Εικόνα 4.39 - Φόρμα καταχώρησης αυτοκινήτου***

***Εικόνα 4.40 - Σελίδα αναζήτησης αυτοκινήτων***

***Εικόνα 4.41 - Μήνυμα επιτυχημένης ενοικίασης***