



Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Σύστημα Συστάσεων για Τουριστικές Πληροφορίες με χρήση NoSQL Βάσης Δεδομένων

Διπλωματική Εργασία

ΚΩΝΣΤΑΝΤΙΝΟΥ ΤΣΟΥΜΑΝΗ

Επιβλέπων

Μιχαήλ Βασιλακόπουλος
Αναπληρωτής Καθηγητής

Βόλος, Σεπτέμβριος 2019



Πανεπιστήμιο Θεσσαλίας

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Σύστημα Συστάσεων για Τουριστικές Πληροφορίες με χρήση NoSQL Βάσης Δεδομένων

Διπλωματική Εργασία

ΚΩΝΣΤΑΝΤΙΝΟΥ ΤΣΟΥΜΑΝΗ

Επιτροπή επίβλεψης

Επιβλέπων

Μιχαήλ Βασιλακόπουλος
Αναπληρωτής Καθηγητής

Συνεπιβλέπουσα

Χαρίκλεια Τσαλαπάτα
Μέλος Ε.ΔΙ.Π.

Συνεπιβλέπουσα

Ελένη Τουσίδου
Μέλος Ε.ΔΙ.Π.

Βόλος, Σεπτέμβριος 2019



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

A Recommendation System for Touristic Information using NoSQL Database Technology

Diploma Thesis

KONSTANTINOS TSOUMANIS

Supervisor

Michael Vassilakopoulos
Associate Professor

Volos, September 2019

Περίληψη

Στην εποχή μας, πολλοί είναι οι άνθρωποι που βασίζονται σε διαδικτυακές υπηρεσίες για να σχεδιάσουν ένα ταξίδι. Οι περισσότερες από αυτές προκειμένου να βελτιώσουν την τουριστική εμπειρία, χρησιμοποιούν συστήματα συστάσεων, τα οποία προσφέρουν εξατομικευμένες πληροφορίες στους χρήστες τους. Στη συγκεκριμένη διπλωματική εργασία, παρουσιάζεται ένα σύστημα συστάσεων για καταλύματα που βασίζεται στην τεχνική του συλλογικού φιλτραρίσματος, μέσω ενός ειδικά σχεδιασμένου αλγορίθμου. Η εφαρμογή που δημιουργήθηκε για να φιλοξενήσει το σύστημα, αρχικά επιτρέπει στους χρήστες να επιλέξουν διάφορες παραμέτρους και αποθηκεύει τα στοιχεία που θεωρούνται απαραίτητα. Στη συνέχεια, με τη βοήθεια του αλγορίθμου συγκρίνει αυτά τα στοιχεία με αντίστοιχα των υπόλοιπων χρηστών της και εμφανίζει στην οθόνη του χρήστη τα προτεινόμενα καταλύματα σε ανάλογη σειρά της προβλεπόμενης από τον αλγόριθμο προτίμησης. Για την υλοποίηση της εφαρμογής επιλέχθηκε να δοκιμαστεί και μία νέα τεχνολογία στο χώρο του τουρισμού που αφορά τη βάση δεδομένων που εξυπηρετεί το σύστημα. Η βάση του συστήματος δεν είναι σχεσιακή όπως συμβαίνει στα περισσότερα συστήματα γύρω από τον τουρισμό αυτή τη στιγμή, αλλά τύπου NoSQL και συγκεκριμένα MongoDB. Οι NoSQL βάσεις δεδομένων φέρνουν μαζί τους πολλά νέα χαρακτηριστικά, τα οποία μπορούν να δώσουν λύσεις σε παραδοσιακά προβλήματα, αλλά και να διευκολύνουν το έργο των διαχειριστών της. Στη συνέχεια της εργασίας θα υπάρξει αναλυτική αιτιολόγηση αυτής της απόφασης και φυσικά, τελική αξιολόγηση της.

Λέξεις Κλειδιά

Τουρισμός, Σύστημα Συστάσεων, Μηχανική Μάθηση, MongoDB, Javascript

Abstract

Nowadays, there are many people who rely on online services to plan a trip. Most of those services use recommender systems to provide personalized information to their users to improve the touristic experience. In this thesis, we present a system of recommendations for accommodation, based on our collaborative filtering technique through a specially designed algorithm. The application that has been created to host the system, initially allows users to select various parameters and saves the data it considers necessary. The algorithm then compares these data with those of its other users and displays on the user's screen the proposed accommodation in a proportional order of preference provided by the algorithm. For the implementation of the application, it was chosen to test a new technology in the tourism sector, related to the database that serves the system. The database of the system is not relational as in most systems around tourism at the moment, but NoSQL type and MongoDB in particular. NoSQL databases are equipped with a lot of new features that can provide solutions to traditional problems and facilitate the work of its administrators. The upcoming thesis will provide a detailed justification for this decision and of course a final evaluation of it.

Keywords

Tourism, Recommender Systems, Machine Learning, MongoDB, Javascript

Αφιερωμένη στην οικογένεια μου για τη στήριξη στα χρόνια των σπουδών μου και στον παππού μου Θανάση.

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ. Βασιλακόπουλο για την επίβλεψη αυτής της διπλωματικής εργασίας και την πολύτιμη συμβολή του στην ολοκλήρωση της. Επιπλέον ευχαριστώ τους γονείς μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια σπουδών μου στο Πανεπιστήμιο Θεσσαλίας.

Πρόλογος

Η ζωή σε μία χώρα σαν την Ελλάδα, η οποία και στηρίζεται σημαντικά στον τομέα του τουρισμού, με έκανε να σκέφτομαι από μικρή ηλικία τρόπους με τους οποίους θα μπορούσα να συνεισφέρω στην ανάπτυξη του συγκεκριμένου χώρου. Η εισαγωγή μου τμήμα των Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών έκανε πιο ξεκάθαρο το πως θα μπορούσε να γίνει αυτό και έτσι, όταν έφτασε η στιγμή της απόφασης για το θέμα της διπλωματικής μου εργασίας, η επιλογή ήταν προφανής. Με τη συμβολή λοιπόν του καθηγητή μου, κ. Βασιλακόπουλου, συγκεκριμενοποίησα το αντικείμενο της εργασίας μου ώστε να ανταποκρίνεται στις τρέχουσες τεχνολογικές εξελίξεις, αλλά και να βελτιώνει την εμπειρία των μελλοντικών επισκεπτών της χώρας μας.

Περιεχόμενα

Περίληψη	i
Abstract	iii
Ευχαριστίες	vii
Πρόλογος	ix
Περιεχόμενα	xi
Κατάλογος σχημάτων	xiii
1 Εισαγωγή	1
1.1 Αντικείμενο της διπλωματικής	1
1.1.1 Συνεισφορά	1
1.2 Οργάνωση του τόμου	2
2 Συγγενικές εργασίες	3
2.1 Εισαγωγή	3
2.2 Ο τουρισμός στην εποχή του διαδικτύου	3
2.3 Συστήματα συστάσεων στον τουρισμό	4
2.4 Χρήση NoSQL βάσεων στα τουριστικά συστήματα	4
3 Θεωρητικό υπόβαθρο	7
3.1 Εισαγωγή	7
3.2 NoSQL Βάσεις Δεδομένων	7
3.3 MongoDB	8
3.4 Javascript	12
3.5 Μηχανική Μάθηση (Machine Learning)	12
3.5.1 Συστήματα Συστάσεων	14
4 Ο Τουρισμός στη ψηφιακή εποχή	17
4.1 Εισαγωγή	17
4.2 Το πρόβλημα	17

4.3	Ο τρόπος λειτουργίας σήμερα	17
5	Προτεινόμενος αλγόριθμος	19
5.1	Εισαγωγή	19
5.2	Ο αλγόριθμος	20
5.3	Ανάλυση του αλγορίθμου	21
5.3.1	Δεδομένα	21
6	Σενάριο χρήσης της web εφαρμογής	23
7	Τεχνικές λεπτομέρειες	29
7.1	Λεπτομέρειες υλοποίησης	29
7.1.1	Βασικές λειτουργίες	29
7.1.2	Εγγραφή / Σύνδεση	31
7.1.3	Αναζήτηση	32
7.1.4	Τελική κράτηση	33
7.1.5	Βάση δεδομένων	34
7.1.6	Front-End Προγραμματισμός	35
7.2	Πλατφόρμες και προγραμματιστικά εργαλεία	37
7.2.1	Node.js και Express.js	37
7.2.2	Navicat	38
7.2.3	Heroku	38
8	Επίλογος	39
8.1	Σύνοψη και συμπεράσματα	39
8.2	Μελλοντικές επεκτάσεις	40
I	Αρχεία της εφαρμογής	41
I.1	GitHub	41
	Βιβλιογραφία	43
	Συντομογραφίες	45
	Ορολογία - Γλωσσάρι	47

Κατάλογος σχημάτων

2.1	Sources of Information Used for Trip Planning	4
2.2	Annual increase in outbound US travel - 1996 compared to 2016.	5
3.1	MongoDB	9
3.2	MongoDB Architecture	9
3.3	Replication	10
3.4	Communication In A Sharded Cluster	12
3.5	Machine Learning	13
6.1	Homepage	23
6.2	Register	24
6.3	Login	24
6.4	First Login	25
6.5	Minimum Search Rating	25
6.6	Search Results	26
6.7	Search By Name	26
6.8	New Reservation	27
6.9	Access Denied	27
7.1	Cookies	29
7.2	Database Connection	29
7.3	Schemas	30
7.4	Extra Layer of Protection	30
7.5	Register	31
7.6	Login	31
7.7	Search	32
7.8	Book	33
7.9	HotelsCollection	34
7.10	UserCollection	34
7.11	Header	36
7.12	Footer	36

Κεφάλαιο 1

Εισαγωγή

Ο τομέας του τουρισμού εδώ και αρκετά χρόνια έχει μεταβεί πλήρως στην ψηφιακή εποχή και πλέον στηρίζεται σχεδόν αποκλειστικά στο διαδίκτυο. Αυτό γίνεται καθημερινά αντιληπτό αφού σε κάθε περιήγησή μας, θα έρθουμε σε επαφή με διαφημίσεις που είτε αφορούν χώρες και μνημεία, είτε τρόπους με τους οποίους μπορείς να οργανώσεις ένα ταξίδι. Παρ'όλα αυτά η απρόσωπη φύση του διαδικτύου πολλές φορές δημιουργεί κενό μεταξύ των προτιμήσεων των εν δυνάμει τουριστών και των υπηρεσιών που τους προσφέρονται.

1.1 Αντικείμενο της διπλωματικής

Η συγκεκριμένη διπλωματική εργασία έχει ως στόχο να γεφυρώσει το προαναφερθέν κενό και να εξατομικεύσει την παροχή υπηρεσιών στον τουριστικό τομέα, έτσι ώστε οι χρήστες να μπορούν εύκολα και άκοπα να επιλέγουν υπηρεσίες που θα είναι απολύτως ταιριαστές με τα ενδιαφέροντα τους. Το παραπάνω θα γίνει προσπάθεια να επιτευχθεί με ένα σύστημα συστάσεων βασισμένο στη συλλογική ανάλυση δεδομένων (collaborative based), που θα προέρχονται από τους χρήστες του χωρίς την παρεμβολή τρίτων παραγόντων.

Τα συστήματα συστάσεων που υπάρχουν αυτή τη στιγμή στον χώρο του τουρισμού, στηρίζονται κατά κόρον σε σχεσιακές βάσεις δεδομένων. Η διπλωματική εργασία που ακολουθεί, υιοθετήθηκε μία διαφορετική προσέγγιση και επιλέχθηκε να στηριχθεί αποκλειστικά σε βάση δεδομένων NoSQL. Ο στόχος δεν ήταν άλλος από το να δοκιμαστεί αυτή η σχετικά νέα τεχνολογία σε έναν τομέα που δεν έχει εφαρμοστεί ποτέ και να εντοπιστούν πιθανά θετικά ή αρνητικά στοιχεία που θα μπορούν να αξιοποιηθούν στο μέλλον.

1.1.1 Συνεισφορά

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Μελετήθηκαν παρόμοια συστήματα συστάσεων
2. Υλοποιήθηκε η εφαρμογή πάνω στην οποία στηρίζεται το εγχείρημα της συγκεκριμένης διπλωματικής

3. Υλοποιήθηκε ο αλγόριθμος που θα λύσει το πρόβλημα που εξετάζεται
4. Αξιολογήθηκε η αγορά εργασίας και βρέθηκε το κενό που προσπαθεί να καλύψει η εργασία

1.2 Οργάνωση του τόμου

Στο Κεφάλαιο 2 παρουσιάζονται εργασίες σχετικές με το αντικείμενο της διπλωματικής και τονίζονται οι διαφορές και τα κοινά σημεία τους. Το Κεφάλαιο 3 παραθέτει τις απαραίτητες πληροφορίες που χρειάζεται κάποιος για να μπορέσει να κατανοήσει το αντικείμενο της εργασίας. Στο Κεφάλαιο 4 αναπτύσσεται το πρόβλημα που συναντήθηκε και προσπαθεί να λύσει, ενώ το Κεφάλαιο 5 αναλύει τη λύση που τελικά προτείνεται. Το επόμενο (Κεφάλαιο 6) περιέχει ένα αναλυτικό σενάριο χρήσης της εφαρμογής και το Κεφάλαιο 7 παρουσιάζει όλες τις τεχνικές λεπτομέρειες που έχουν ενδιαφέρον. Η διπλωματική εργασία κλείνει με το Κεφάλαιο 8 που προσπαθεί να δώσει ιδέες για μελλοντική επέκταση της υπάρχουσας εργασίας.

Κεφάλαιο 2

Συγγενικές εργασίες

2.1 Εισαγωγή

Η επίδραση του ίντερνετ στον τομέα του τουρισμού έχει τραβήξει την προσοχή και για το λόγο αυτό τα τελευταία χρόνια βλέπουμε όλο και περισσότερες έρευνες και άρθρα γύρω από τη νέα κατάσταση που τείνει να δημιουργηθεί. Εκτός αυτών, στο κεφάλαιο αυτό θα αναλυθούν εργασίες που έχουν αφιερωθεί στα συστήματα συστάσεων και θα εξεταστεί ο τρόπος λειτουργίας τους στο χώρο του τουρισμού.

2.2 Ο τουρισμός στην εποχή του διαδικτύου

Οι καταναλωτές την τρέχουσα δεκαετία τείνουν στο να ξοδεύουν περισσότερα χρήματα στις ηλεκτρονικές αγορές προϊόντων ή υπηρεσιών. Έρευνες [12] έχουν δείξει πως αυτή η μετατόπιση του αγοραστικού κοινού, έχει οδηγήσει στον εκσυγχρονισμό παραδοσιακών εννοιών και τεχνικών του μάρκετινγκ όπως το Word Of Mouth(WOM). Πλέον οι καταναλωτές σε όλο τον κόσμο έχουν τη δυνατότητα να συνομιλούν μεταξύ τους και να ανταλλάσσουν απόψεις και πληροφορίες ακόμα και για πολύ συγκεκριμένα προϊόντα ή υπηρεσίες αλλά και να λαμβάνουν υπόψιν αξιολογήσεις ανθρώπων που έχουν πραγματοποιήσει μία αγορά πριν από τους ίδιους.

Όσον αφορά στον τομέα του τουρισμού, η ίδια έρευνα τονίζει τη σημαντικότητα των διαπροσωπικών επαφών υποστηρίζοντας πως ένας καταναλωτής μπορεί να καταλήξει άμεσα προκατειλημένος θετικά ή αρνητικά απέναντι σε έναν προορισμό απλά ακούγοντας μερικές θετικές ή αρνητικές εμπειρίες αντίστοιχα από άτομα που επισκέφτηκαν τον εν λόγω προορισμό. Σύμφωνα με άλλη έρευνα [5], το 89% των νέων που ερωτήθηκαν, απάντησαν πως η πρώτη τους επιλογή για πληροφορίες σχετικά με ένα ταξίδι είναι το ίντερνετ, ενώ μία επιπλέον έρευνα [13], δείχνει πως στις πρώτες θέσεις για τις πηγές από τις οποίες οι ταξιδιώτες αντλούν πληροφορίες για την οργάνωση ενός ταξιδιού είναι το ίντερνετ, το WOM και οι προηγούμενες εμπειρίες τους, πληροφορίες που όπως θα διαπιστωθεί στη συνέχεια έχουν αξιοποιηθεί στην υλοποίηση της προτεινόμενης λύσης αυτής της διπλωματικής.

Information Source	2007	2008	2009	2010	2011	2012	Change (2012- 2007)
	N=2,436	N=2,166	N=2,014	N=1,159	N=1,032	N=1,041	
	Pcnt	Pcnt	Pcnt	Pcnt	Pcnt	Pcnt	
Internet	85.3%	85.2%	85.1%	86.2%	87.7%	85.5%	0.2%
Previous experience	53.9%	54.0%	53.6%	52.4%	54.9%	52.7%	-1.2%
Word of mouth	42.7%	45.9%	41.3%	40.7%	43.3%	41.3%	-1.4%
Automotive clubs	24.8%	18.8%	20.5%	17.8%	20.1%	18.3%	-6.4%
Travel companies	23.2%	23.1%	20.9%	29.8%	32.3%	29.3%	6.1%
Travel guide books	22.0%	18.5%	19.7%	19.8%	23.3%	18.1%	-3.9%
Travel brochures	20.7%	19.9%	18.0%	19.5%	23.0%	20.1%	-0.6%
Magazines/newspapers	18.5%	16.8%	16.0%	20.4%	21.9%	22.2%	3.7%
Tourist offices	17.1%	13.9%	13.8%	13.4%	16.9%	14.6%	-2.5%
Travel Agents	14.5%	12.5%	10.7%	13.4%	18.0%	14.6%	0.2%
Travel documentaries	11.0%	10.7%	9.6%	9.8%	13.1%	13.2%	2.2%
TV	11.0%	11.6%	10.5%	15.5%	17.3%	17.9%	6.9%
Radio	3.3%	3.9%	3.1%	4.7%	4.1%	5.9%	2.6%
Movies	3.2%	3.7%	2.6%	4.5%	5.9%	7.9%	4.7%

Σχήμα 2.1: Sources of Information Used for Trip Planning
[13]

2.3 Συστήματα συστάσεων στον τουρισμό

Ο τουρισμός είναι ένας τομέας στον οποίο τα συστήματα συστάσεων έχουν κάνει την εμφάνισή τους εδώ και αρκετά χρόνια. Θεωρείται ένας αρκετά ενδιαφέρον χώρος καθώς έχει μεγάλο πεδίο δράσης και μπορεί να αξιοποιηθεί στην πρόταση τοποθεσιών, αξιοθεάτων, καταλυμάτων και πτήσεων έτσι ώστε να δώσει μία πλήρη εικόνα του ταξιδιού στον πελάτη.

Σε έρευνα που πραγματοποιήθηκε [6], παρατηρήθηκε μια αυξανόμενη τάση στην εκμετάλλευση των τεχνικών συλλογικού φίλτραρίσματος (Collaborative) από το 2012, κυρίως σε υβριδικά συστήματα. Συγκεκριμένα, από το 2008 έως το 2011 μόνο το 25% των συστημάτων χρησιμοποίησε αυτή τη μέθοδο, ενώ από το 2012 το ποσοστό αυξήθηκε στο 75%. Παρόλα αυτά, τα συστήματα συστάσεων που έχουν υιοθετήσει τη συγκεκριμένη τεχνική πάσχουν από το πρόβλημα του "κρύου ξεκινήματος" (cold start). Το πρόβλημα αυτό μπορεί να αναφέρεται είτε σε έναν νέο χρήστη, είτε σε προϊόντα που δεν έχουν λάβει αξιολόγηση. Άρθρα όπως αυτά του Huba Gaspar [4] και της InDataLabs [7] υποστηρίζουν πως το φίλτράρισμα βάσει περιεχομένου (content based) είναι η μέθοδος που δίνει λύση στο πρόβλημα. Αρχικά χρησιμοποιεί τα μεταδεδομένα των νέων προϊόντων κατά τη δημιουργία συστάσεων, ενώ η δράση των επισκεπτών είναι δευτερεύουσα για ορισμένο χρονικό διάστημα.

2.4 Χρήση NoSQL βάσεων στα τουριστικά συστήματα

Το γραφείο εθνικού τουρισμού της Αμερικής, παρουσίασε στοιχεία για τα ταξίδια των Αμερικανών από το 1996 μέχρι και το 2016 [1]. Συγκεκριμένα, αναφέρει πως το 1996 τα ταξίδια των Αμερικανών πολιτών εκτός της χώρας ανέρχονταν στα 27 εκατομμύρια, ενώ το 2016 έφτασαν τα 72,5 εκατομμύρια. Στο σχήμα που ακολουθεί, φαίνεται κατά πόσο αυξήθηκαν τα ταξίδια προς

συγκεκριμένους προορισμούς το 2016 σε σχέση με το 1996.



Σχήμα 2.2: Annual increase in outbound US travel - 1996 compared to 2016.

[1]

Επιπλέον, σύμφωνα με πρόσφατη έρευνα που πραγματοποίησε για το 2018 ο Παγκόσμιος Οργανισμός Τουρισμού (World Tourism Organization) [10], ο τουρισμός παγκοσμίως παρουσίασε αύξηση έως και 10% σε ορισμένες περιοχές του πλανήτη. Στην ίδια έρευνα επισημαίνεται πως οι προβλέψεις για το 2019 είναι ακόμα υψηλότερες, τονίζοντας παράλληλα πως θα καταγραφούν ιστορικά νούμερα.

Είναι, λοιπόν, καθολικά αποδεκτό πως οι άνθρωποι τα τελευταία χρόνια ταξιδεύουν περισσότερο από ποτέ και υπάρχει εξήγηση για αυτό. Οι πληροφορίες για άγνωστα -στο παρελθόν- μέρη έχουν πολλαπλασιαστεί και οι τιμές των πτήσεων έχουν μειωθεί κατά πολύ. Η κατάσταση αυτή έχει δημιουργήσει έναν πρωτοφανή όγκο δεδομένων στον τουριστικό τομέα και ευνοεί τη δοκιμή των NoSQL βάσεων που έχουν ως κυρίαρχο προτέρημα έναντι των SQL, την άριστη διαχείριση πολλών στοιχείων.

Κεφάλαιο 3

Θεωρητικό υπόβαθρο

3.1 Εισαγωγή

Για τη δημιουργία της Web Εφαρμογής που πραγματεύεται η εργασία, επιλέχθηκε η γλώσσα προγραμματισμού Javascript και συγκεκριμένα η πλατφόρμα ανάπτυξης λογισμικού Node.js. Για την πλευρά του server επιλέχθηκε το Expressjs, μια δομή που συμπληρώνει την προαναφερθείσα πλατφόρμα. Τα απαραίτητα δεδομένα θα αποθηκευτούν σε βάση δεδομένων της MongoDB, η οποία ανήκει στην κατηγορία των NoSQL βάσεων δεδομένων.

3.2 NoSQL Βάσεις Δεδομένων

Ο όρος NoSQL χρησιμοποιήθηκε από τον Carlo Strozzi το 1998 για να ονομάσει την ελαφριά σχεσιακή βάση δεδομένων του (Strozzi NoSQL) ανοικτού κώδικα που δεν εξέθετε την τυπική διαπαφή SQL, αλλά εξακολουθούσε να είναι σχεσιακή. Ο συγκεκριμένος όρος επαναπροσδιορίστηκε το 2009 οπότε και απέκτησε την έννοια που γνωρίζουμε όλοι σήμερα.

Η NoSQL, που αρχικά αναφερόταν σε μη SQL ή αλλιώς μη σχεσιακή, είναι μια βάση δεδομένων που παρέχει ένα μηχανισμό αποθήκευσης και ανάκτησης δεδομένων. Το παραδοσιακό σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων χρησιμοποιεί τη σύνταξη τύπου SQL για την αποθήκευση αλλά και την ανάκτηση δεδομένων από τις σχέσεις μεταξύ πινάκων που χρησιμοποιούνται στις συμβατικές βάσεις. Αντίθετα, τα δεδομένα στις NoSQL βάσεις μοντελοποιούνται με διαφορετικά μέσα. Ένα σύστημα βάσης δεδομένων NoSQL περιλαμβάνει ένα ευρύ φάσμα τεχνολογιών που μπορούν να αποθηκεύσουν δομημένα, ημι-δομημένα, αδόμητα και πολυμορφικά δεδομένα. Πρόκειται για ένα χαρακτηριστικό που ταιριάζει απόλυτα με τις τεχνικές ευέλικτης ανάπτυξης (Agile Development) που κυριαρχούν αυτή τη στιγμή στο χώρο του web development, αφού κάθε φορά που ολοκληρώνονται νέες λειτουργίες, το σχήμα της βάσης δεδομένων συνήθως πρέπει να αλλάξει. Αυτό σημαίνει πως αν χρησιμοποιούνταν το σχεσιακό μοντέλο, κάθε φορά που θα ήθελε ο προγραμματιστής να προσθέσει ένα νέο χαρακτηριστικό, θα έπρεπε να αλλάξει το σχήμα ολόκληρης της βάσης. Οι βάσεις δεδομένων NoSQL όμως, επιτρέπουν την πραγματοποίηση σημαντικών αλλαγών εφαρμογών σε πραγματικό χρόνο, χωρίς να υπάρχει ανησυχία για διακοπές λειτουργίας κάτι που σημαίνει ότι η ανάπτυξη είναι ταχύτερη, η ενσωμάτωση κώδικα

είναι πιο αξιόπιστη και απαιτείται λιγότερος χρόνος από τον διαχειριστή της εκάστοτε βάσης δεδομένων.

Τέτοιες βάσεις δεδομένων δημιουργήθηκαν στα τέλη της δεκαετίας του 1960, αλλά δεν έλαβαν το χαρακτηρισμό "NoSQL" μέχρι και την αύξηση της δημοτικότητας τους στις αρχές του 21ου αιώνα. Οι βάσεις δεδομένων NoSQL χρησιμοποιούνται στις μέρες μας σε web εφαρμογές πραγματικού χρόνου και εφαρμογές με μεγάλο όγκο δεδομένων και η χρήση τους αυξάνεται με την πάροδο των ετών. Τα συστήματα NoSQL ονομάζονται μερικές φορές και "όχι μόνο SQL" για να υπογραμμίσουν το γεγονός ότι ενδέχεται να υποστηρίζουν γλώσσες ερωτημάτων τύπου SQL.

Εταιρίες κολοσσοί, όπως η Google, στρέφονται ολοένα και περισσότερο στις NoSQL βάσεις δεδομένων. Αυτό είναι αποτέλεσμα ενός ακόμα πολύ σημαντικού πλεονεκτήματος των NoSQL βάσεων δεδομένων έναντι των συμβατικών, το οποίο είναι η οριζόντια κλιμάκωση (sharding). Η βάση κλιμακώνεται πλέον οριζόντια, προσθέτοντας διακομιστές αντί να συγκεντρώνει περισσότερη χωρητικότητα σε έναν μόνο. Οι προγραμματιστές από την πλευρά τους δεν χρειάζεται πλέον να κατασκευάζουν περίπλοκες, δαπανηρές πλατφόρμες για να υποστηρίξουν τις εφαρμογές τους και μπορούν να επικεντρωθούν στη σύνταξη κώδικα εφαρμογής. Οι βάσεις δεδομένων NoSQL, υποστηρίζουν συνήθως τον αυτόματο διαμοιρασμό (auto-sharding), πράγμα που σημαίνει ότι διανέμουν δεδομένα αυτόματα σε έναν αυθαίρετο αριθμό διακομιστών, χωρίς να χρειάζεται η εφαρμογή να γνωρίζει ακόμη και τη σύνθεση του συνόλου των διακομιστών. Τα δεδομένα και οι αναζητήσεις εξισορροπούνται αυτόματα σε όλους τους διακομιστές και όταν ένας από αυτούς παρουσιάσει βλάβη, μπορεί να αντικατασταθεί γρήγορα και με διαφάνεια χωρίς διακοπή της εφαρμογής.

Οι περισσότερες βάσεις δεδομένων NoSQL υποστηρίζουν επίσης την αυτόματη αναπαραγωγή (auto-replication), πράγμα που σημαίνει ότι έχετε υψηλή διαθεσιμότητα και αποκατάσταση μετά από καταστροφές χωρίς την προσθήκη ξεχωριστών εφαρμογών για τη διαχείριση αυτών των εργασιών.

Τέλος, πολλές τεχνολογίες των NoSQL βάσεων δεδομένων έχουν ενσωματωμένες άριστες δυνατότητες προσωρινής αποθήκευσης (caching), διατηρώντας όσο το δυνατόν περισσότερο τα δεδομένα που χρησιμοποιούνται συχνά στη μνήμη του συστήματος. Αυτό ουσιαστικά καταργεί την ανάγκη για ξεχωριστό επίπεδο προσωρινής αποθήκευσης, το οποίο θα χρειαζόταν και ξεχωριστή συντήρηση.

3.3 MongoDB

Πρόκειται για τύπο βάσης δεδομένων που στηρίζεται στις δομές των εγγράφων. Έχει κατηγοριοποιηθεί ως NoSQL βάση δεδομένων, κάτι που σημαίνει πως αγνοεί την παραδοσιακή δομή της σχεσιακής βάσης δεδομένων και χρησιμοποιεί κυρίως αρχεία που μοιάζουν πολύ σε αυτά της JSON μορφοποίησης τα οποία ονομάζονται BSON (Binary JSON). Έχει ιδιαίτερα απλή αρχιτεκτονική, χαρακτηριστικό που προσφέρει ευελεξία, ταχύτητα και ευκολία, ενώ μπορεί να χειριστεί είτε μέσω του ενσωματωμένου "κελύφους" (MongoShell), είτε μέσω τρίτων προγραμμάτων διαχείρισης βάσεων.

Κάθε καταχώρηση που γίνεται στη MongoDB θεωρείται ως ένα έγγραφο και τα πεδία στα έγγραφα της μπορούν να περιέχουν συστοιχίες τιμών ή άλλα εμφωλευμένα έγγραφα. Η MongoDB αποθηκεύει τα έγγραφα αυτά σε συλλογές που είναι ανάλογες με τους πίνακες σε σχεσιακές βάσεις δεδομένων. Στα RDBMS όλοι οι πίνακες, σε μια βάση δεδομένων, πρέπει να έχουν το ίδιο σχήμα, αλλά στη MongoDB δεν υπάρχει τέτοια απαίτηση και αυτός ο σχεδιασμός χωρίς σχήματα είναι μια καινοτομία που καθιστά την MongoDB την πιο χρησιμοποιημένη βάση δεδομένων NoSQL. Τα έγγραφα που είναι αποθηκευμένα σε μια συλλογή είναι απαραίτητο να έχουν και ένα μοναδικό πεδίο (id) το οποίο λειτουργεί ως πρωτεύον κλειδί (primary key). Τα έγγραφα σε μια συλλογή μπορούν να αποθηκευτούν είτε ως κανονικοποιημένα είτε ως ενσωματωμένα σε άλλο έγγραφο. Η βάση δεδομένων αποτελείται ουσιαστικά από ένα σύνολο συλλογών. Κάθε βάση δεδομένων έχει ένα ξεχωριστό σύνολο αρχείων και μπορεί να περιέχει μεγάλο αριθμό από συλλογές, ενώ μία ενιαία εγκατάσταση MongoDB μπορεί να έχει πολλές επιμέρους βάσεις δεδομένων.



Σχήμα 3.1: MongoDB



Σχήμα 3.2: MongoDB Architecture

Η MongoDB συνοδεύεται από κάποια ιδιαίτερα χαρακτηριστικά.

◇ **Ad hoc queries**

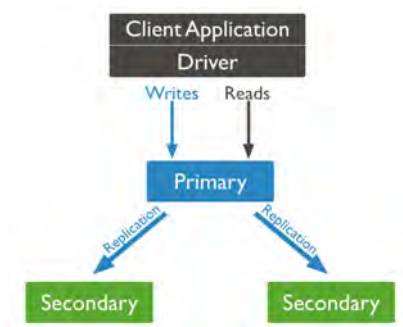
Η MongoDB υποστηρίζει αναζητήσεις πεδίων, εύρους αλλά και κανονικής έκφρασης. Τα ερωτήματα μπορούν να επιστρέψουν συγκεκριμένα πεδία των εγγράφων και να περιλαμβάνουν λειτουργίες JavaScript που ορίζονται από το χρήστη. Τα ερωτήματα μπορούν επίσης να ρυθμιστούν ώστε να επιστρέφουν ένα τυχαίο δείγμα αποτελεσμάτων ενός δεδομένου μεγέθους.

◇ **Ευρετήρια - Indexes**

Τα πεδία σε ένα έγγραφο MongoDB μπορούν να αποκτήσουν πρωτεύοντα ή δευτερεύοντα ευρετήρια ώστε να γίνεται πιο εύκολη η αναζήτηση δεδομένων. Τα ευρετήρια είναι ίσως η πιο χρήσιμη δυνατότητα που μπορεί να προσφέρει μια βάση δεδομένων στη σύγχρονη εποχή. Στις μέρες μας οι βάσεις έχουν συνήθως μεγάλο όγκο δεδομένων και αυτό κάνει τρομερά δύσκολη την αναζήτηση συγκεκριμένων δεδομένων από τους χρήστες και τους διαχειριστές τους. Το ευρετήριο λύνει ακριβώς αυτό το πρόβλημα, δίνοντας την δυνατότητα στη βάση να κρατήσει ένα αρχείο με πληροφορίες για τη θέση των δεδομένων ανάλογα φυσικά με τις ανάγκες τις εφαρμογής. Με απλά λόγια δημιουργεί μια σελίδα περιχομένων.

◇ **Αναπαραγωγή - Replication**

Η MongoDB δημιουργεί πολλαπλά αντίγραφα των δεδομένων σε διαφορετικούς διακομιστές με σκοπό να προστατεύει μια βάση δεδομένων από αποτυχία υλικού ή διακοπές υπηρεσιών, παρέχοντας με αυτόν τον τρόπο υψηλή διαθεσιμότητα. Ένα σετ αντιγράφων αποτελείται από δύο ή περισσότερα αντίγραφα των δεδομένων και κάθε μέλος της ομάδας αναπαραγωγής μπορεί να ενεργεί στο ρόλο πρωτογενούς ή δευτερογενούς αντιγράφου ανά πάσα στιγμή. Όλες οι εγγραφές και οι αναγνώσεις γίνονται στο κύριο αντίγραφο από προεπιλογή, ενώ τα δευτερεύοντα αντίγραφα διατηρούν ένα αντίγραφο των δεδομένων του πρωτογενούς χρησιμοποιώντας τη λειτουργία της ενσωματωμένης αναπαραγωγής. Όταν ένα πρωτότυπο αντίγραφο αποτύχει, το σύνολο αντιτύπων διεξάγει αυτόματα μια εκλογική διαδικασία για να προσδιορίσει ποιο δευτερεύον πρέπει να γίνει το κύριο. Τα δευτερεύοντα τμήματα μπορούν προαιρετικά να παρέχουν λειτουργίες ανάγνωσης.



Σχήμα 3.3: Replication

◇ Συνάθροιση - Aggregation

«Aggregations» είναι λειτουργίες που επεξεργάζονται τα αρχεία δεδομένων και επιστρέφουν τα υπολογιζόμενα αποτελέσματα, δηλαδή ένα είδος φίλτρου που χωρίζει τα δεδομένα. Η MongoDB χρησιμοποιεί συλλογές εγγράφων ως είσοδο με πιθανή έξοδο ενός ή περισσότερων εγγράφων. Είναι μια σειρά διαδικασιών που εκτελούνται σταδιακά και παρέχουν φίλτρα που λειτουργούν σαν μετασχηματισμοί εγγράφων, που τροποποιούν τη μορφή του εγγράφου εξόδου.

Το MapReduce είναι ένα ισχυρό και ευέλικτο εργαλείο που χρησιμοποιείται για τη συγκέντρωση δεδομένων. Μπορεί να λύσει προβλήματα που είναι πολύπλοκα διαλύοντας το σε τεμάχια τα οποία στέλνει σε διαφορετικά μηχανήματα και αφήνει σε κάθε μηχανή να λύσει το δικό του τμήμα του προβλήματος. Όταν όλα τα μηχανήματα έχουν τελειώσει, όλα τα κομμάτια της λύσης έχουν συγχωνευθεί πίσω σε μια πλήρη μορφή.

◇ Σύστημα αρχείων - GridFS

Μία MongoDB βάση μπορεί να χρησιμοποιηθεί και ως σύστημα αρχείων, το οποίο ονομάζεται GridFS, με δυνατότητες εξισορρόπησης φορτίου και αναπαραγωγής δεδομένων σε πολλαπλές μηχανές αποθήκευσης αρχείων.

Το GridFS είναι μια προδιαγραφή για την αποθήκευση και ανάκτηση αρχείων που υπερβαίνουν το όριο μεγέθους εγγράφου BSON 16MB. Αντί να αποθηκεύει ένα αρχείο σαν ένα ενιαίο έγγραφο, το GridFS διαιρεί ένα αρχείο σε τμήματα και αποθηκεύει κάθε τμήμα ως ξεχωριστό έγγραφο. Το GridFS χρησιμοποιεί δύο συλλογές για την αποθήκευση αρχείων. Μια συλλογή αποθηκεύει τα κομμάτια των αρχείων και τα άλλα αποθηκεύουν τα μεταδεδομένα των αρχείων. Όταν η εφαρμογή ζητάει από το GridFS ένα αρχείο, αυτό επανασυνθέτει τα κομμάτια ανάλογα με τις ανάγκες. Επίσης υπάρχει δυνατότητα πρόσβασης σε πληροφορίες από τυχαία τμήματα των αρχείων, κάτι που επιτρέπει την αναπαραγωγή πολυμέσων από ένα συγκεκριμένο σημείο.

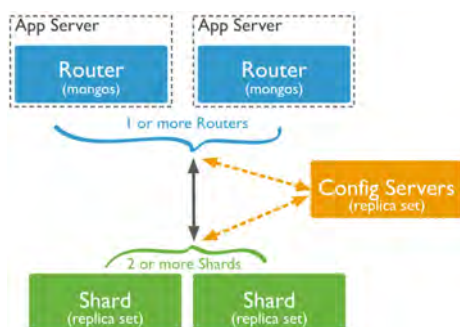
◇ Διαμοιρασμός - Sharding

Ο διαμοιρασμός είναι μια μέθοδος για τη διανομή δεδομένων σε πολλές μηχανές. Η MongoDB τη χρησιμοποιεί για να υποστηρίξει την ανάπτυξη σχεδίων με πολύ μεγάλα σύνολα δεδομένων και λειτουργίες υψηλής απόδοσης.

Τα συστήματα βάσεων δεδομένων με μεγάλα σύνολα δεδομένων ή εφαρμογές υψηλής απόδοσης μπορούν να δημιουργήσουν προβλήματα στη λειτουργία ενός μόνο διακομιστή. Για παράδειγμα, τα πολλά ταυτόχρονα queries μπορούν να εξαντλήσουν την χωρητικότητα του CPU του διακομιστή. Η MongoDB υποστηρίζει δύο μεθόδους για την αντιμετώπιση του προβλήματος: την κατακόρυφη και την οριζόντια κλιμάκωση.

Η κατακόρυφη κλιμάκωση συνεπάγεται την αύξηση της χωρητικότητας ενός μόνο διακομιστή, όπως η χρήση πιο ισχυρής CPU, η προσθήκη περισσότερης μνήμης RAM ή η αύξηση του χώρου αποθήκευσης. Η οριζόντια κλίμακα περιλαμβάνει τη διαίρεση του συνόλου δεδομένων του συστήματος και τη φόρτωση τους σε πολλούς διακομιστές. Ενώ η συνολική

ταχύτητα ή η χωρητικότητα ενός μηχανήματος μπορεί να μην είναι υψηλή, κάθε μηχανήμα χειρίζεται ένα υποσύνολο του συνολικού φόρτου, παρέχοντας ενδεχομένως καλύτερη απόδοση από ένα μόνο διακομιστή υψηλής ταχύτητας και υψηλής χωρητικότητας.



Σχήμα 3.4: Communication In A Sharded Cluster

3.4 Javascript

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

Η JavaScript είναι μια γλώσσα σεναρίων που βασίζεται στα πρωτότυπα (prototype-based), είναι δυναμική, με ασθενείς τύπους και έχει συναρτήσεις ως αντικείμενα πρώτης τάξης. Η σύνταξή της είναι επηρεασμένη από τη C. Η JavaScript αντιγράφει πολλά ονόματα και συμβάσεις ονοματοδοσίας από τη Java, αλλά γενικά οι δύο αυτές γλώσσες δε σχετίζονται και έχουν πολύ διαφορετική σημασιολογία. Οι βασικές αρχές σχεδιασμού της JavaScript προέρχονται από τις γλώσσες προγραμματισμού Self και Scheme. Είναι γλώσσα βασισμένη σε διαφορετικά προγραμματιστικά παραδείγματα (multi-paradigm), υποστηρίζοντας αντικειμενοστραφές, προστακτικό και συναρτησιακό στυλ προγραμματισμού.

Η JavaScript χρησιμοποιείται και σε εφαρμογές εκτός ιστοσελίδων — τέτοια παραδείγματα είναι τα έγγραφα PDF, οι εξειδικευμένοι φυλλομετρητές (site-specific browsers) και οι μικρές εφαρμογές της επιφάνειας εργασίας (desktop widgets). Οι νεότερες εικονικές μηχανές και πλαίσια ανάπτυξης για JavaScript (όπως το Node.js) έχουν επίσης κάνει τη JavaScript πιο δημοφιλή για την ανάπτυξη εφαρμογών Ιστού στην πλευρά του διακομιστή (server-side).

3.5 Μηχανική Μάθηση (Machine Learning)

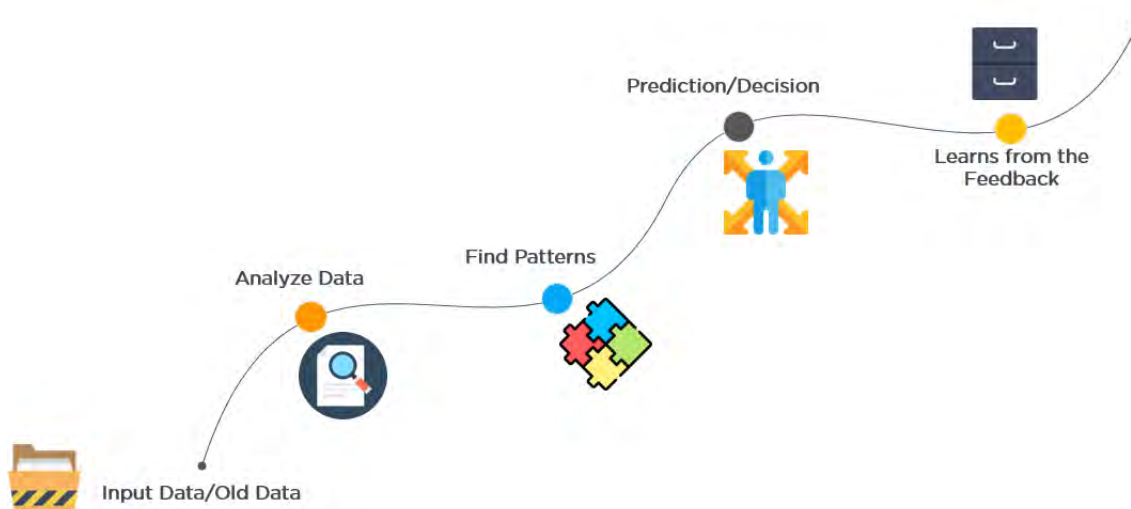
Η μηχανική μάθηση είναι μια εφαρμογή της τεχνητής νοημοσύνης (Artificial Intelligence, ή AI) που παρέχει στα συστήματα τη δυνατότητα να μαθαίνουν και να βελτιώνουν αυτόματα από την εμπειρία χωρίς να έχουν προγραμματιστεί ρητά. Η μηχανική μάθηση επικεντρώνεται στην ανάπτυξη προγραμμάτων ηλεκτρονικών υπολογιστών που μπορούν να έχουν πρόσβαση στα δεδομένα και να τα χρησιμοποιούν για να μάθουν για τον εαυτό τους.

Η διαδικασία μάθησης ξεκινά με παρατηρήσεις ή δεδομένα, όπως παραδείγματα, άμεση εμπειρία ή διδασκαλία, προκειμένου να αναζητήσουμε μοτίβα στα δεδομένα και να λάβουμε καλύτερες αποφάσεις στο μέλλον βάσει των παραδειγμάτων που παρέχουμε. Ο πρωταρχικός στόχος είναι να επιτρέπεται στους υπολογιστές να μαθαίνουν αυτόματα χωρίς ανθρώπινη παρέμβαση ή βοήθεια και να προσαρμόζουν ανάλογα τις ενέργειες. Οι αλγόριθμοι μηχανικής μάθησης συχνά κατηγοριοποιούνται ως εποπτευόμενοι ή μη επιτηρούμενοι.

Οι εποπτευόμενοι αλγόριθμοι μηχανικής μάθησης μπορούν να εφαρμόσουν όσα έχουν μαθευτεί στο παρελθόν σε νέα δεδομένα χρησιμοποιώντας επισημασμένα παραδείγματα για να προβλέψουν μελλοντικά γεγονότα. Ξεκινώντας από την ανάλυση ενός γνωστού συνόλου δεδομένων κατάρτισης, ο αλγόριθμος εκμάθησης παράγει μια συναγόμενη συνάρτηση για να κάνει προβλέψεις σχετικά με τις τιμές εξόδου. Το σύστημα είναι σε θέση να παρέχει στόχους για κάθε νέα είσοδο μετά από επαρκή εκπαίδευση. Ο αλγόριθμος εκμάθησης μπορεί επίσης να συγκρίνει την έξοδο του με τη σωστή προβλεπόμενη έξοδο και να βρει λάθη για να τροποποιήσει ανάλογα το μοντέλο.

Αντίθετα, οι μη εποπτευόμενοι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται όταν οι πληροφορίες που χρησιμοποιούνται για την εκπαίδευση δεν ταξινομούνται ούτε επισημαίνονται. Οι μη εποπτευόμενες μελέτες μάθησης στοχεύουν στο πώς μπορούν τα συστήματα να συναγάγουν μια συνάρτηση για να περιγράψουν μια κρυμμένη δομή από μη επισημασμένα δεδομένα. Το σύστημα δεν υπολογίζει τη σωστή έξοδο, αλλά διερευνά τα δεδομένα και μπορεί να αντλεί συμπεράσματα από σύνολα δεδομένων για να περιγράψει κρυφές δομές από μη επισημασμένα δεδομένα.

Η μηχανική μάθηση επιτρέπει την ανάλυση τεράστιας ποσότητας δεδομένων. Παρόλο που παρέχει γενικά ταχύτερα και ακριβέστερα αποτελέσματα, προκειμένου να εντοπίσει επικερδείς ευκαιρίες ή επικίνδυνα ρίσκα, μπορεί επίσης να απαιτήσει επιπλέον χρόνο και πόρους για να πραγματοποιηθεί σωστά η εκπαίδευση. Ο συνδυασμός της μηχανικής μάθησης με τις AI τεχνολογίες μπορεί να καταστήσει ακόμη πιο αποτελεσματική την επεξεργασία μεγάλων όγκων πληροφοριών.



Σχήμα 3.5: Machine Learning

3.5.1 Συστήματα Συστάσεων

Ένα σύστημα συστάσεων (Recommender System, ή αλλιώς RS) είναι ένα συγκεκριμένο είδος προσαρμοστικού φίλτρου πληροφοριών. Είναι μια τεχνική που προσπαθεί να δώσει στον χρήστη τις πληροφορίες που τον ενδιαφέρουν. Ο κεντρικός άξονας ενός RS είναι ο όρος "εξατομίκευση". Ο στόχος της εξατομίκευσης είναι να παρέχει στους χρήστες αυτό που χρειάζονται χωρίς να το ζητάνε ρητά. Το σύστημα μπορεί να συμβάλλει σε αυτό που ο χρήστης απαιτεί όχι μόνο σύμφωνα με τις πληροφορίες που παρέχει αρχικά, αλλά και συγκρίνοντας το προφίλ του με άλλους χρήστες που έχουν παρόμοιο και λαμβάνοντας υπόψη τα δημογραφικά στοιχεία του (δημογραφικό προφίλ). Με λίγα λόγια, ένα RS προσφέρει τη δυνατότητα προσαρμογής των διαθέσιμων πληροφοριών στο Διαδίκτυο χάρη σε ένα φίλτρο. Έτσι, το RS θα παρέχει μια ποσότητα πληροφοριών εύκολη στη διαχείριση, προσαρμοσμένη στις ανάγκες του χρήστη, στις προτιμήσεις και πάνω από όλα στα συμφέροντα. Το αποτέλεσμα της διαδικασίας συστάσεως είναι ένα σύνολο στοιχείων που παρέχονται στον χρήστη με τον τρόπο που περιγράφεται στη διεπαφή.

Η διαδικασία εξατομίκευσης είναι χωρισμένη σε τρία στάδια:

- ◇ **Συμπλήρωση πληροφοριών που αφορούν τους χρήστες:** Συλλογή πληροφοριών σχετικά με τους χρήστες προκειμένου να τους γνωρίσουν καλύτερα. Αυτή η πληροφορία αποθηκεύεται στο προφίλ χρήστη.
- ◇ **Προτεινόμενα στοιχεία:** Προτείνει στον χρήστη στοιχεία με βάση τη γνώση που έχει αποκτήσει το σύστημα.
- ◇ **Βαθμός ικανοποίησης:** Μετρά την επίδραση της εξατομίκευσης χρησιμοποιώντας τη γνώμη των χρηστών σχετικά με τα προτεινόμενα στοιχεία. Αυτό το βήμα χρησιμοποιείται ως ανατροφοδότηση από το πρώτο, αφού λαμβάνει υπόψη τη γνώμη του χρήστη και έτσι το σύστημα τείνει να τον γνωρίζει καλύτερα.

Τα συστήματα συστάσεων χωρίζονται στις εξής κατηγορίες:

- Συλλογικά (Collaboratives)
- Δημογραφικά (Demographic)
- Βασισμένα στη γνώση
- Υβρίδια που συνδυάζουν στοιχεία από τα προηγούμενα

Αυτά που θα μας απασχολήσουν στην συγκεκριμένη διπλωματική είναι τα συλλογικά που είναι τα πιο διαδεδομένα, πιο ευρέως εφαρμοζόμενα και αποτελούν την πιο ώριμη τεχνολογία που διατίθεται στην αγορά. Τα συλλογικά συστήματα λοιπόν, συναθροίζουν αξιολογήσεις ή συστάσεις αντικειμένων, αναγνωρίζουν κοινά στοιχεία μεταξύ των χρηστών βάσει των αξιολογήσεών τους και δημιουργούν νέες συστάσεις βάσει συγκρίσεων μεταξύ χρηστών. Η μεγαλύτερη δύναμη αυτών των τεχνικών είναι ότι είναι εντελώς ανεξάρτητες από οποιαδήποτε μηχανικά αναγνώσιμη αναπαράσταση των αντικειμένων που συνιστώνται και δουλεύουν καλά για πολύπλοκα αντικείμενα. Το συλλογικό φίλτράρισμα βασίζεται στην υπόθεση ότι οι άνθρωποι που συμφώνησαν στο

παρελθόν θα συμφωνήσουν στο μέλλον και ότι θα τους αρέσουν παρόμοια αντικείμενα που τους άρεσαν στο παρελθόν.

Κεφάλαιο 4

Ο Τουρισμός στη ψηφιακή εποχή

4.1 Εισαγωγή

Από τις αρχές του 21ου αιώνα που η τεχνολογία έχει μπει στην καθημερινότητα κάθε νοικοκυριού, ολοένα και περισσότερες ανάγκες του ανθρώπου έχουν ψηφιοποιηθεί και μπορούν να ικανοποιηθούν από μία ηλεκτρονική υπηρεσία μέσω ενός ηλεκτρονικού υπολογιστή.

Ο τομέας του τουρισμού δε θα μπορούσε να αποτελεί εξαίρεση. Ειδικότερα τα τελευταία χρόνια, παρατηρείται μεγάλη άνοδος στη χρήση υπηρεσιών για διαδικτυακές κρατήσεις καταλυμάτων. Όπως γίνεται αντιληπτό, αυτό σηματοδοτεί τη σταδιακή απομάκρυνση των ταξιδιωτών από τα παραδοσιακά ταξιδιωτικά γραφεία αλλά και τον ανθρώπινο παράγοντα. Η ανάγκη των μεγάλων εταιριών που δραστηριοποιούνται στον χώρο της παροχής τουριστικών υπηρεσιών για κέρδος, έφερε νέα προβλήματα για τα οποία οι υποψήφιοι ταξιδιώτες καλούνται να βρουν λύση.

4.2 Το πρόβλημα

Η απουσία της ανθρώπινης επαφής μπορεί αρχικά να δείχνει θετική, καθώς η αναζήτηση του επιθυμητού καταλύματος γίνεται πιο άμεση και γρήγορη, αλλά ταυτόχρονα δημιουργεί απόλυτη εξάρτηση του ενδιαφερόμενου από τον τρόπο που θα επιλέξει η εκάστοτε υπηρεσία να του εμφανίσει τις διαθέσιμες επιλογές. Οι συγκεκριμένες υπηρεσίες όμως συνήθως λειτουργούν με απώτερο σκοπό τη διασφάλιση του μεγαλύτερου δυνατού κέρδους, αφήνοντας σε δεύτερη μοίρα την απόλυτη ικανοποίηση του πελάτη.

4.3 Ο τρόπος λειτουργίας σήμερα

Μία κύρια πηγή εσόδων για αυτού του είδους τις εταιρίες είναι οι συναλλαγές με τους ιδιοκτήτες των καταλυμάτων που βρίσκονται αναρτημένα στην πλατφόρμα τους. Αυτές οι συναλλαγές περιλαμβάνουν αρχικά ένα συμφωνημένο ποσοστό από κάθε κράτηση που γίνεται στο κατάλυμα μέσω της πλατφόρμας τους και επεκτείνονται στα διάφορα πλεονεκτήματα που μπορεί να απολαύσει ο ιδιοκτήτης και αφορούν στην καλύτερη κατάταξη του καταλύματος στις αναζητήσεις των

χρηστών, ή ακόμα και στην ένταξη τους σε ειδικά προγράμματα μέσω των οποίων τα αντίστοιχα καταλύματα θα προτείνονται στο χρήστη κατά τη διάρκεια της περιήγησης του στη σελίδα.

Συνεπώς οι αλγόριθμοι οι οποίοι είναι υπεύθυνοι για την εμφάνιση των αποτελεσμάτων μίας αναζήτησης, λαμβάνουν υπόψιν στοιχεία που δεν έχουν καμία σχέση με τον δυνητικό πελάτη, αλλά κυρίως με το ποιός ιδιοκτήτης έχει προσφέρει περισσότερα χρήματα. Ως αλγόριθμος ορίζεται μια πεπερασμένη σειρά ενεργειών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, που στοχεύουν στην επίλυση ενός προβλήματος. Πιο απλά, αλγόριθμο ονομάζουμε μία σειρά από εντολές που έχουν αρχή και τέλος, είναι σαφείς και έχουν ως σκοπό την επίλυση κάποιου προβλήματος. Στην περίπτωση μας ο αλγόριθμος θα συνδυάσει τις αρχικές επιλογές του χρήστη (π.χ. αριθμός ατόμων και τόπος) με τις προαναφερθείσες συναλλαγές μεταξύ παρόχου και ιδιοκτήτη και θα επιλύσει το "πρόβλημα" της αναζήτησης.

Κεφάλαιο 5

Προτεινόμενος αλγόριθμος

5.1 Εισαγωγή

Η λύση που προτείνεται στη συγκεκριμένη διπλωματική εργασία είναι η εφαρμογή ενός αλγορίθμου που θα βασίζεται στη μηχανική μάθηση. Θα χρησιμοποιεί τα δεδομένα που θα παράγουν οι χρήστες της web εφαρμογής που έχει δημιουργηθεί και κάθε φορά θα βελτιώνει την ποιότητα της εμφάνισης των αποτελεσμάτων. Ως επακόλουθο της τεχνικής της μηχανικής μάθησης, όσο περισσότερα δεδομένα χρηστών υπάρχουν, τόσο καλύτερα αποτελέσματα θα παράγει. Ο στόχος είναι αποκλειστικά η διευκόλυνση και ικανοποίηση του πελάτη και ο αλγόριθμος θα είναι πλήρως απαλλαγμένος από ενδιάμεσα φίλτρα οικονομικής φύσεως.

5.2 Ο αλγόριθμος

SEARCH ALGORITHM

```
function train(items , query , rating){
  const visit_weight = 0.01;
  const query_weight = 0.02;
  const rating_weight = 0.02;
  const match_weight = 0.05;

  if ( items !== undefined && items.length ){
    for(let i = 0; i < items.length; i++){
      if ( items[i].visits.length ){
        items[i].score += (items[i].visits.length * visit_weight);
        let matchQuery = false;
        let matchRating = false;

        for(let j = 0; j < items[i].visits.length; j++){
          if ( items[i].visits[j].query == query ){
            items[i].score += query_weight;
            matchQuery = true;
          }

          if ( rating > 0 && items[i].visits[j].rating == rating ){
            items[i].score += rating_weight;
            matchRating = true;
          }

          if ( matchQuery && matchRating ){
            items[i].score += match_weight;
          }
        }
      }
    }
  }
  return items;
}
```


5.3 Ανάλυση του αλγορίθμου

Κάθε φορά που ένας χρήστης χρησιμοποιεί την εφαρμογή, αποθηκεύονται κάποια στοιχεία, τα οποία **αν και μόνον αν** γίνει κράτηση σε ένα κατάλυμα διατίθενται στον παραπάνω αλγόριθμο για την εκπαίδευση του.

Κάθε στοιχείο διαθέτει ένα "βάρος" ανάλογο της σημαντικότητας του. Το στοιχείο "query" αναφέρεται στον τρόπο με τον οποίο έκανε την αναζήτηση ο χρήστης, δηλαδή ποια ή ποιες λέξεις χρησιμοποίησε. Αντίστοιχα, το "rating" σχετίζεται με την ελάχιστη βαθμολογία που επέλεξε ως φίλτρο ο χρήστης για την αναζήτηση που τελικά οδήγησε σε κράτηση. Το "visit" αναφέρεται στον συνολικό αριθμό των κρατήσεων που έχουν γίνει στο κατάλυμα που εν τέλει επέλεξε ο χρήστης.

Τέλος, καταλήγουμε στο σημαντικότερο στοιχείο που είναι το "match". Δόθηκε ιδιαίτερη βαρύτητα στο στοιχείο αυτό καθώς θεωρήθηκε πως αν υπάρχει πλήρης ταύτιση των δύο προαναφερθέντων στοιχείων, ο συνδυασμός τους είναι ικανός να δώσει πολύ χρήσιμες πληροφορίες στον αλγόριθμο για το πόσο κοντά είναι πιθανόν να βρίσκονται τα ενδιαφέροντα των χρηστών.

Ακριβώς στην επόμενη γραμμή ξεκινά το κυρίως κομμάτι του αλγορίθμου στο οποίο γίνονται διαδοχικοί έλεγχοι με στόχο να προκύψει το κατάλληλο "score" για κάθε κατάλυμα. Αρχικά, ελέγχουμε αν υπάρχουν επισκέψεις στο συγκεκριμένο κατάλυμα και αν υπάρξει θετική απάντηση, πολλαπλασιάζουμε τον αριθμό τους με το βάρος του "visit" θέτοντας το αποτέλεσμα στο πεδίο "score". Σημαντικό είναι να τεθούν ως "false" οι δύο μεταβλητές που αφορούν τα στοιχεία "query" και "rating" ώστε να υπάρξει σωστή διαχείριση τους στο κριτήριο του "match".

Εν συνεχεία για κάθε μία κράτηση γίνεται ο έλεγχος των κριτηρίων "query" και "rating". Αν η αναζήτηση με την οποία είχε γίνει η συγκεκριμένη κράτηση ταιριάζει με αυτήν που γίνεται αυτή τη στιγμή, το βάρος του "query" προστίθεται στο υπάρχον "score". Με την ίδια λογική χειρίζεται και το βάρος του "rating" στο επόμενο κομμάτι. Στο τέλος κάθε λογικής συνθήκης, αλλάζουμε την αντίστοιχη τιμή boolean σε "true".

Στην περίπτωση που και οι δύο μεταβλητές έχουν την τιμή "true" μετά το πέρας των λογικών συνθηκών, εφαρμόζεται το τελευταίο κριτήριο όπου το βάρος του "match" προστίθεται και αυτό στο πεδίο "score". Στην αναζήτηση που πραγματοποίησε ο χρήστης θα εμφανιστεί πρώτο το κατάλυμα που συγκέντρωσε το μεγαλύτερο score και θα ακολουθήσουν τα υπόλοιπα σε φθίνουσα σειρά.

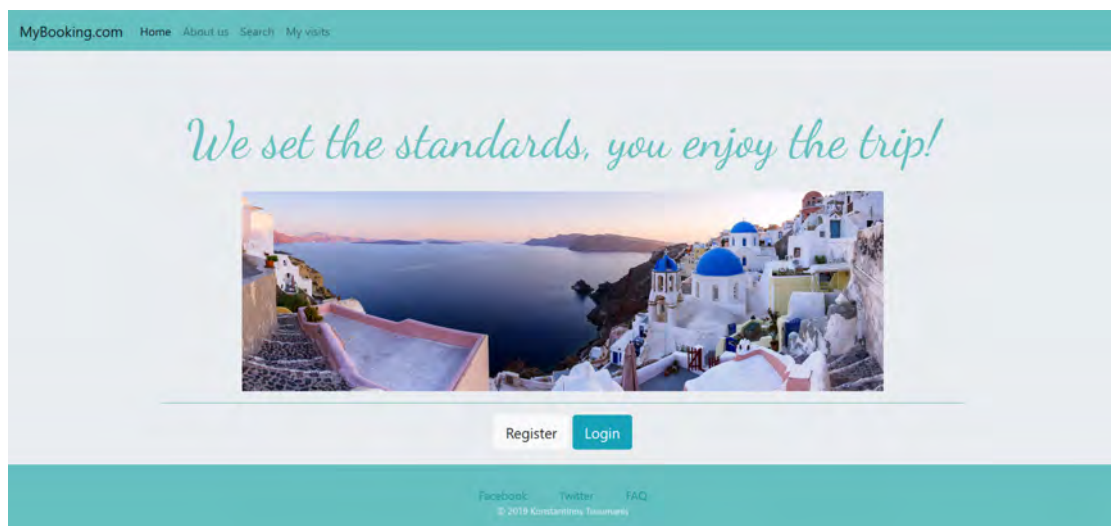
5.3.1 Δεδομένα

Προκειμένου να ελεγχθεί η λειτουργία και η αξιοπιστία του αλγορίθμου μας, χρησιμοποιήθηκαν δεδομένα από την τουριστική υπηρεσία TripAdvisor για καταλύματα τεσσάρων δημοφιλών περιοχών της Ελλάδος: Αθήνα, Κρήτη, Σαντορίνη και Μύκονος.

Κεφάλαιο 6

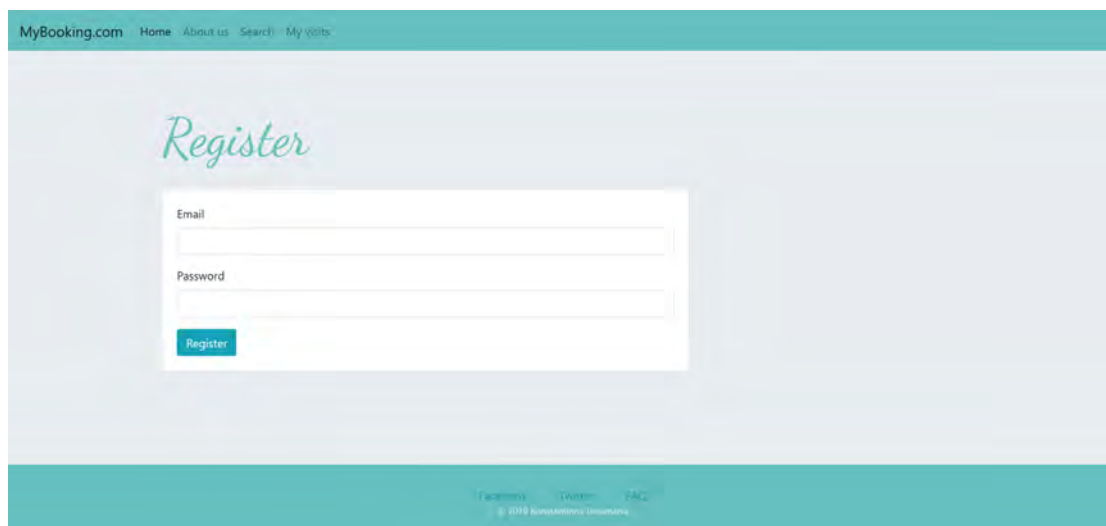
Σενάριο χρήσης της web εφαρμογής

Ο χρήστης ανοίγοντας τη σελίδα της εφαρμογής θα έρθει σε μία πρώτη επαφή με τη δομή της. Αρχικά θα του ζητηθεί να κάνει εγγραφή ή στην περίπτωση που έχει ήδη δημιουργήσει έναν λογαριασμό στο παρελθόν, να εισέλθει στο σύστημα με τα στοιχεία του.

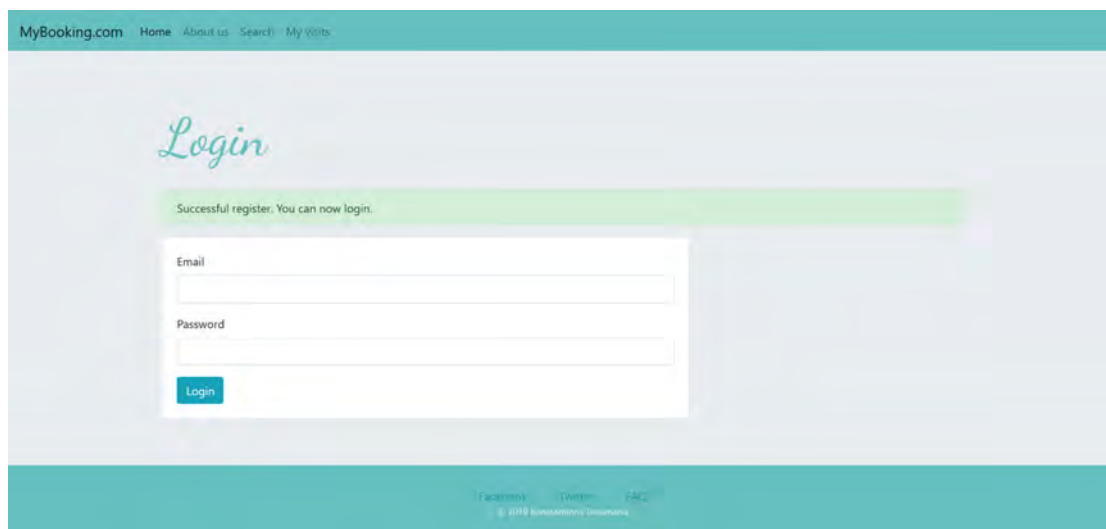


Σχήμα 6.1: Homepage

Στη συνέχεια, ανάλογα με την κατηγορία στην οποία ανήκει ο χρήστης, θα βρεθεί σε μία από τις παρακάτω σελίδες στις οποίες θα πρέπει να πληκτρολογήσει τα στοιχεία του. Αν πρόκειται για νέο χρήστη, αφού εγγραφεί θα ανακατευθυνθεί αυτόματα στην αντίστοιχη σελίδα ώστε να εισέλθει στο σύστημα με τα στοιχεία που καταχώρησε προηγουμένως.



Σχήμα 6.2: Register



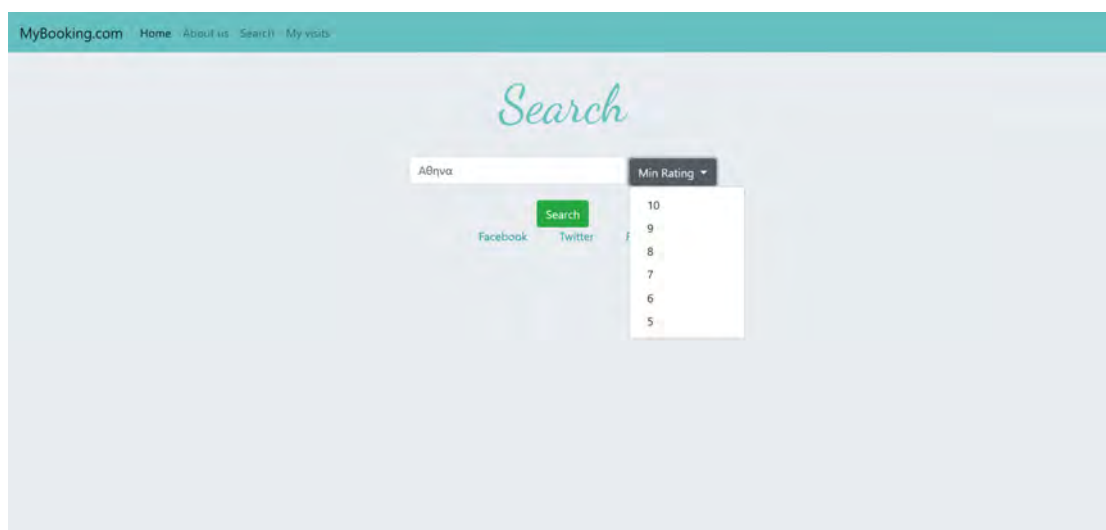
Σχήμα 6.3: Login

Κάθε φορά που κάποιος χρήστης εισέρχεται στο σύστημα, θα βλέπει το ιστορικό των καταλυμάτων που έχει κλείσει. Φυσικά αν είναι νέος χρήστης ή δεν έχει κάνει κράτηση θα βλέπει το ακόλουθο μήνυμα.



Σχήμα 6.4: First Login

Το επόμενο βήμα για το χρήστη είναι η αναζήτηση του καταλύματος. Επιλέγοντας την καρτέλα "Search" (στο πάνω αριστερά μέρος της σελίδας) θα μπορέσει να επιλέξει το κάτω όριο του μέσου όρου βαθμολογίας των καταλυμάτων που θα συμπεριληφθούν στην προτεινόμενη λίστα και φυσικά την τοποθεσία που επιθυμεί να επισκεφτεί.



Σχήμα 6.5: Minimum Search Rating

Το αποτέλεσμα μίας πετυχημένης αναζήτησης παρουσιάζεται στο σχήμα 6.6. Ο χρήστης πλέον βλέπει μία λίστα καταλυμάτων στην οποία υπάρχουν σε φθίνουσα κατάταξη τα καταλύματα που ταιριάζουν περισσότερο στο προφίλ του.

The screenshot shows the search results for 'Αθήνα' on MyBooking.com. The search criteria are 'Αθήνα' and 'Min Rating: 8'. The results are displayed in a table with the following data:

Hotel Name	Location	Rating	Reviews count	Total Score	Price	Book
Ξενοδοχείο Ηλέκτρα Παλάς - Αθήνα	Ναυάρχου Νικοδήμου 18-20 Αθήνα 105 57, ΑΘΗΝΑ	9	1	1.57	371 €	Book
Agistri Island Dream	Αθήνα 18010, ΑΘΗΝΑ	9	16	1.26	0 €	Book
Exarcheia House	4 Ζωσίμου Αθήνα, ΑΘΗΝΑ	10	2	1.14	0 €	Book
Welcommon Hostel	4 Kapodistriou Αθήνα, ΑΘΗΝΑ	9	4	1.14	61 €	Book

Σχήμα 6.6: Search Results

Αν για οποιοδήποτε λόγο ο χρήστης επιλέξει να πληκτρολογήσει το όνομα ενός καταλύματος, η αναζήτηση θα λειτουργήσει κανονικά. Η εφαρμογή χρησιμοποιεί την τεχνική του "Full-Text Search" η οποία συγκρίνει κάθε λέξη της αιτούμενης αναζήτησης με κάθε λέξη μέσα στη βάση δεδομένων.

The screenshot shows the search results for 'ηλέκτρα' on MyBooking.com. The search criteria are 'ηλέκτρα' and 'Min Rating: 8'. The results are displayed in a table with the following data:

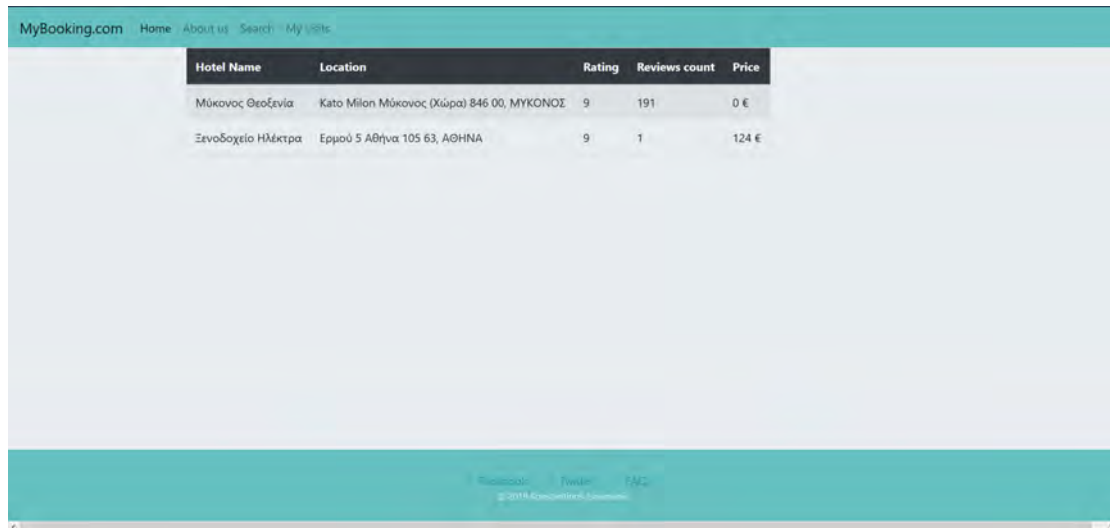
Hotel Name	Location	Rating	Reviews count	Total Score	Price	Book
Ξενοδοχείο Ηλέκτρα	Ερμού 5 Αθήνα 105 63, ΑΘΗΝΑ	9	1	0.76	124 €	Book
Ξενοδοχείο Ηλέκτρα Παλάς - Αθήνα	Ναυάρχου Νικοδήμου 18-20 Αθήνα 105 57, ΑΘΗΝΑ	9	1	0.64	371 €	Book

At the bottom of the page, there are links for Facebook, Twitter, and FAQ.

Σχήμα 6.7: Search By Name

Εκτός από την τοποθεσία, το όνομα του καταλύματος και την τιμή ανά βραδιά, είναι ορατά στον χρήστη ο μέσος όρος βαθμολογίας του, ο αριθμός των κριτικών που καθόρισε αυτό το μέσο όρο και το κουμπί μέσω του οποίου μπορεί να γίνει η κράτηση. Η επιβεβαίωση της κράτησης θα γίνει με αντίστοιχο μήνυμα στην οθόνη του πελάτη.

Αν ξαναεπισκεφτεί τη σελίδα με τις επισκέψεις του, ο χρήστης θα δει ότι έχει προστεθεί η κράτηση που μόλις έγινε στο ιστορικό του.



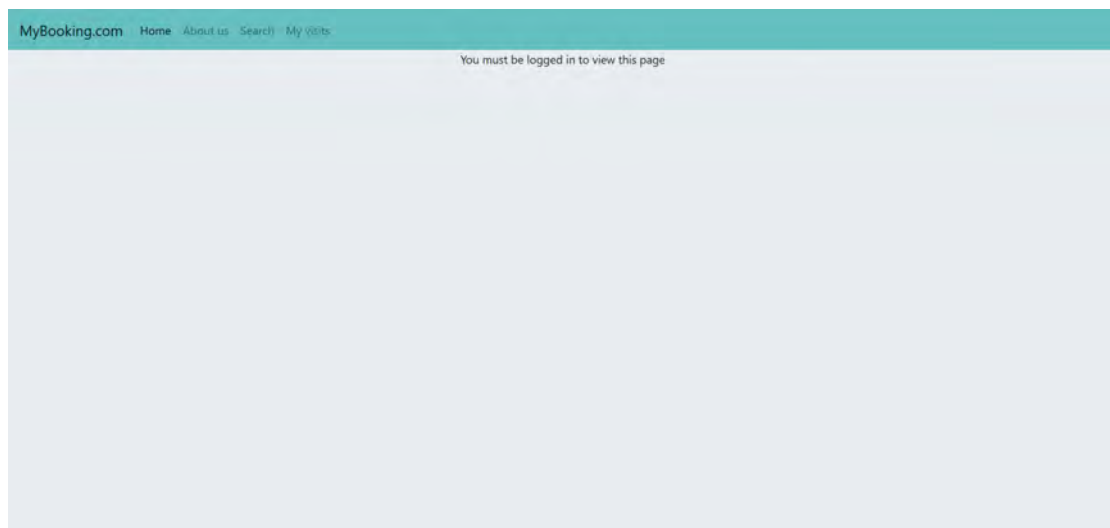
The screenshot shows the MyBooking.com website interface. At the top, there is a navigation bar with links for Home, About us, Search, and My visits. Below this is a table with the following data:

Hotel Name	Location	Rating	Reviews count	Price
Μύκονος Θεοξενία	Kato Milon Μύκονος (Χώρα) 846 00, ΜΥΚΟΝΟΣ	9	191	0 €
Ξενοδοχείο Ηλέκτρα	Ερμού 5 Αθήνα 105 63, ΑΘΗΝΑ	9	1	124 €

At the bottom of the page, there is a footer with the text "© 2018. All rights reserved." and a small logo.

Σχήμα 6.8: New Reservation

Οι καρτέλες "Search" και "My Visits" εμφανίζουν περιεχόμενο μόνο αν υπάρχει ενεργή σύνδεση κάποιου λογαριασμού. Σε αντίθετη περίπτωση εμφανίζεται το παρακάτω μήνυμα.



Σχήμα 6.9: Access Denied

Κεφάλαιο 7

Τεχνικές λεπτομέρειες

Στο κεφάλαιο αυτό θα ακολουθήσει ανάλυση των τεχνικών λεπτομερειών που παρουσιάζουν ενδιαφέρον και περιγραφή των εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση της.

7.1 Λεπτομέρειες υλοποίησης

7.1.1 Βασικές λειτουργίες

Το κύριο αρχείο της εφαρμογής είναι το "app.js" και σε αυτό συναντώνται οι περισσότερες λειτουργίες της. Αρχικά, για κάθε χρήστη δημιουργείται μία νέα συνεδρία (session) μέσω της οποίας διατηρούνται τα απαραίτητα cookies που καθιστούν ξεχωριστό τον κάθε χρήστη.

```
11 //USES
12 app.use(session({
13   secret: 'machine-learning',
14   resave: true,
15   saveUninitialized: false,
16   cookie: {
17     expires: 600000
18   }
19 }));
```

Σχήμα 7.1: Cookies

Συνεχίζοντας, συναντάται μία λογική συνθήκη η οποία επιλέγει τη βάση που θα γίνει η σύνδεση. Επιλέχθηκε αυτός ο τρόπος ώστε να υπάρχει η δυνατότητα δοκιμής της εφαρμογής τοπικά πριν γίνει το "ανέβασμα" της στο διαδίκτυο.

```
27 //LOCAL
28 if ( platform == 'win32' ){
29   mongoose.connect("mongodb://localhost:27017/test", {useNewUrlParser: true});
30 }
31 //HEROKU
32 else{
33   mongoose.connect("mongodb://kostis:kostis77@ds117178.mlab.com:17178/heroku_1vfsrfgbc", {useNewUrlParser: true});
34 }
```

Σχήμα 7.2: Database Connection

Ακολουθώς υπάρχουν τα τρία βασικά σχήματα που χρησιμοποιούνται στην εφαρμογή. Το πρώτο αφορά τον χρήστη και αποτελείται από τα πεδία "email" και "password", το δεύτερο αφορά τα καταλύματα και ουσιαστικά τις επισκέψεις που έχουν δεχθεί. Τα πεδία του είναι τα "user id", "query" και "rating" και αποθηκεύουν το id των χρηστών, την λέξη ή τις λέξεις που χρησιμοποιήσαν στην αναζήτηση τους, η οποία τελικά κατέληξε σε κράτηση και το ελάχιστο όριο βαθμολογίας που είχαν επιλέξει αντίστοιχα.

```
36 •const userSchema = new mongoose.Schema({
37   email: String,
38   password: String
39 });
40
41 •const visitSchema = new mongoose.Schema({
42   user_id: String,
43   query: String,
44   rating: Number,
45 });
46
47 •const hotelSchema = new mongoose.Schema({
48   id: String,
49   name: String,
50   location: String,
51   rating: Number,
52   price: Number,
53   reviews_count: Number,
54   score: Number,
55   visits: [visitSchema]
56 });
```

Σχήμα 7.3: Schemas

Κάθε φορά που κρίνεται απαραίτητο χρησιμοποιείται η συνάρτηση "requiresLogin" ώστε να επιβεβαιώνεται πως ο χρήστης έχει δικαίωμα να περιηγηθεί στο συγκεκριμένο κομμάτι της εφαρμογής. Η συνάρτηση ελέγχει αν υπάρχει συνεδρία και το "user id" του χρήστη της συνεδρίας.

```
94 //AUTHENTICATE MIDDLEWARE
95 function requiresLogin(req, res, next) {
96   //console.log(req.session.user_sid);
97   if ( req.session && req.session.user_sid ){
98     return next();
99   } else {
100     res.render("message", {message : 'You must be logged in to view this page'});
101   }
102 }
```

Σχήμα 7.4: Extra Layer of Protection

7.1.2 Εγγραφή / Σύνδεση

Το επόμενο κομμάτι περιέχει 2 μεθόδους "post" του Node.js οι οποίες είναι υπεύθυνες για την εγγραφή και τη σύνδεση των χρηστών. Για την ασφάλεια των χρηστών έχει γίνει κρυπτογράφηση των κωδικών των χρηστών μέσω του γνωστού MD5 αλγορίθμου κατακερματισμού. Προσθέτουμε έτσι ένα ακόμα επίπεδο ασφάλειας των προσωπικών δεδομένων των χρηστών, καθώς ακόμα κι αν υπάρξει κενό ασφαλείας, ο "εισβολέας" θα βλέπει μόνο το hash που θα έχει αποθηκευτεί στη βάση μας και όχι το πραγματικό password που έχει επιλέξει ο εκάστοτε χρήστης.

```
100 //REGISTER
101 app.post("/register", function(req, res) {
102   • const username = req.body.username;
103   • const User = mongoose.model("User", userSchema);
104   User.count( {email:username}, function(err, foundUser) {
105     if (err) {
106       console.error(err);
107       res.render("message", {message : 'An error occured'});
108     } else {
109       if (foundUser) {
110         res.render("message", {message : 'Please choose another email'});
111       } else{
112         • const password = md5(req.body.password).toUpperCase();
113         • const newUser = new User({email: username, password: password});
114         newUser.save(function(err) {
115           if (err) {
116             console.log(err);
117             res.render("message", {message : 'An error occured'});
118           } else {
119             req.session.user_register = true;
120             res.redirect("login");
121           }
122         });
123       }
124     }
125   });
126 });
127 });
```

Σχήμα 7.5: Register

```
128 //LOGIN
129 app.post("/login", function(req, res) {
130   • const username = req.body.username;
131   • const password = md5(req.body.password);
132   • const User = mongoose.model("User", userSchema);
133   //console.log(password);
134   User.findOne(
135     {email: username}, function(err, foundUser) {
136     if (err) {
137       console.error(err);
138       res.render("message", {message : 'An error occured'});
139     } else {
140       if (foundUser) {
141         if (foundUser.password == password.toUpperCase()) {
142           req.session.user_sid = foundUser.id;
143           //console.log(req.session.user_sid);
144           res.redirect("visits");
145         } else{
146           res.render("message", {message : 'Invalid login'});
147         }
148       }
149     } else{
150       res.render("message", {message : 'Invalid login'});
151     }
152   }
153 });
154 });
```

Σχήμα 7.6: Login

7.1.3 Αναζήτηση

Ίσως το σημαντικότερο κομμάτι της εφαρμογής. Όπως έχει επισημανθεί σε προηγούμενο σημείο της εργασίας, για την αναζήτηση χρησιμοποιείται η τεχνική "full text search" η οποία και εφαρμόζεται πάνω σε ευρετήριο που έχει δημιουργηθεί για να διευκολυνθεί η διαδικασία. Το ευρετήριο έχει δημιουργηθεί πάνω στα πεδία "name" και "location". Αφού τρέξει το query πάνω στο μοντέλο του ξενοδοχείου, εφαρμόζεται ο βασικός αλγόριθμος της εφαρμογής που είναι υπεύθυνος για την "μηχανική εκπαίδευση" και στη συνέχεια γίνεται η ταξινόμηση των καταλυμάτων σύμφωνα με τις επιλεγμένες προτιμήσεις.

```
163 //HOTEL SEARCH
164 ● app.post('/search', (req, res) => {
165
166 ● let query = req.body.query;
167 ● let rating = 0;
168 if ( req.body.rating ){
169   rating = parseInt(req.body.rating);
170 }
171 console.log(query, rating);
172
173 ● let Hotel = mongoose.model("Hotel", hotelSchema);
174
175 //FULL TEXT SEARCH
176 //hotelSchema.index({"$**":"text"});
177 hotelSchema.index({
178   name: 'text',
179   location: 'text',
180 });
181
182 Hotel.find({ rating: { $gte: rating }, $text: { $search: query }},
183   {score: {$meta: "textScore"}}, function(err, items) {
184     //console.log(items);
185     items = train(items, query, rating);
186     res.render("search", {items: items, query: query, 'rating': rating});
187   })
188 //SORT
189 .sort({ score:{$meta:"textScore"}, 'rating': 'desc', 'reviews_count': 'desc',
190   'price': 'asc', 'name': 'asc' });
191
192 });
```

Σχήμα 7.7: Search

7.1.4 Τελική κράτηση

Για την κράτηση σε ένα κατάλυμα, αποθηκεύονται τα δεδομένα που μας ενδιαφέρουν σε τοπικές μεταβλητές και ακολούθως εισάγονται στον πίνακα "visits" και συγκεκριμένα στο έγγραφο του επιλεγμένου καταλύματος με την εντολή "push", όπως παρατηρούμε στο σχήμα 7.8. Σε περίπτωση επιτυχούς κράτησης ο χρήστης θα δει στην οθόνη του αντίστοιχο μήνυμα επιβεβαίωσης.

```
225 //BOOK HOTEL
226 app.get('/book', requiresLogin, function(req, res, next) {
227   if ( req.query.id == undefined ){
228     next();
229   } else{
230     var user_id = req.session.user_sid;
231     let hotel_id = req.query.id;
232     let query = req.query.query;
233     let rating = req.query.rating;
234     console.log(hotel_id, user_id);
235     const Hotel = mongoose.model("Hotel", hotelSchema);
236     Hotel.findOneAndUpdate(
237       { id: hotel_id},
238       {$push: {'visits': { query:query, rating: rating, user_id: user_id } }},
239       {upsert:true},
240       function(err, doc){
241         if (err){
242           console.error(err);
243           res.render("message", {message : 'An error occurred'});
244         }
245         //return res.send("successfully saved");
246         return res.redirect('confirm');
247       });
248   }
249 });
```

Σχήμα 7.8: Book

7.1.5 Βάση δεδομένων

Μετά από ανάλυση του συγκεκριμένου χώρου αποφασίστηκε να στηριχθεί όλο το project σε NoSQL βάση δεδομένων όπως η MongoDB. Στη βάση μας υπάρχουν δύο συλλογές (collections). Η μία ονομάζεται "hotels", αφορά τα καταλύματα και αποτελείται από έγγραφα με πεδία "id", "όνομα", "περιοχή", "τιμή", "αξιολόγηση", "αριθμός αξιολογήσεων" και το πεδίο "visits". Το τελευταίο πεδίο είναι ένας πίνακας που περιέχει εμφωλευμένα έγγραφα για κάθε επιβεβαιωμένη κράτηση του εν λόγω καταλύματος. Στα έγγραφα αυτά αποθηκεύονται στοιχεία όπως το id του χρήστη, η αναζήτηση που έκανε και κατέληξε σε κράτηση και το κατώτατο όριο που έθεσε στην αναζήτηση του. Η άλλη συλλογή αφορά τους χρήστες που δημιουργούν λογαριασμό. Τα πεδία των εγγράφων της έχουν τρία πεδία: Το id του χρήστη, το email του και τον κωδικό του. Το τελευταίο περιέχει το κωδικοποιημένο hash που έχει δημιουργηθεί και όχι τον πραγματικό κωδικό που επέλεξε ο χρήστης.

```

69420 db.getCollection("hotels").insert([ {
69421     _id: ObjectId("5d35a013de7100001e007329"),
69422     name: "Anamisi Mykonos",
69423     location: "Ag Paraskevis Mykonos Μύκονος (Χώρα) 84600, ΜΥΚΟΝΟΣ",
69424     price: NumberInt("0"),
69425     rating: NumberInt("1"),
69426     "reviews_count": NumberInt("0"),
69427     visits: [
69428         {
69429             "user_id": "",
69430             query: "",
69431             rating: NumberInt("0")
69432         }
69433     ]
69434 } ]);

```

Σχήμα 7.9: HotelsCollection

```

69760 db.getCollection("users").insert([ {
69761     _id: ObjectId("5d2a2cc61e90d8286c007705"),
69762     email: "kostas@mail.com",
69763     password: "736ABDC3F4089DDAB24EDAE080C6787E"
69764 } ]);

```

Σχήμα 7.10: UserCollection

Ευρετήριο

Για τη διευκόλυνση της αναζήτησης δημιουργήθηκε ένα ευρετήριο στα πεδία "όνομα" και "περιοχή" με την εντολή:

```
db.hotels.createIndex({name:"text", location:"text"});
```

7.1.6 Front-End Προγραμματισμός

Αν και το λειτουργικό κομμάτι της εφαρμογής (back-end) υλοποιήθηκε με χρήση της γλώσσας προγραμματισμού Javascript, το οπτικό αποτέλεσμα (front-end) θα ήταν τελείως διαφορετικό χωρίς την χρησιμοποίηση γλωσσών όπως η HTML, η CSS αλλά και εργαλείων σαν το Bootstrap. Το τελευταίο είναι ένα εργαλείο ανοιχτού κώδικα που βελτιώνει και διευκολύνει τον front-end προγραμματισμό των εφαρμογών συνδυάζοντας τις γλώσσες HTML, CSS, και Javascript.

Header και Footer

Για να γίνει πιο εύκολος ο προγραμματισμός της εφαρμογής και των διαφορετικών σελίδων που έπρεπε να διαμορφωθούν ώστε να καταστεί πλήρως λειτουργική, δημιουργήθηκαν δύο αρχεία που θα περιέχουν στατικές πληροφορίες και θα προσκολλώνται σε κάθε σελίδα.

Στο αρχείο "header", αρχικά συνδέουμε τις σελίδες με το αρχείο που περιέχει τον κώδικα CSS ο οποίος είναι υπεύθυνος για τη μορφοποίηση των σελίδων αλλά και με έναν CDN server που προσφέρει το εργαλείο bootstrap. Επιλέχθηκε να συνδεθεί με αυτόν τον τρόπο ώστε να μην επιβαρυνθεί το σύστημα μας τοπικά και ταυτόχρονα να είναι πάντα διαθέσιμο σε περίπτωση σύνδεσης στο διαδίκτυο. Συνεχίζοντας το κομμάτι του "body", υπάρχει η μπάρα πλοήγησης (navigation bar) η οποία αποτελεί αναπόσπαστο κομμάτι κάθε σύγχρονου site και βελτιώνει την εμπειρία χρήσης του προσφέροντας άμεση πρόσβαση στην αναζήτηση, και το ιστορικό κρατήσεων.

Στο αρχείο "footer" υπάρχει η σύνδεση της Javascript και links προς τα κοινωνικά μέσα της εφαρμογής και τις πιο συχνές ερωτήσεις που μπορεί να έχει ο μέσος χρήστης της.


```

1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3
4 <head>
5   <meta charset="utf-8">
6   <title>Make your trip better!</title>
7   <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"
8     integrity="sha384-UHRTZLI+pbxthCwp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE00s/"
9     crossorigin="anonymous">
10  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
11    crossorigin="anonymous">
12  <link rel="stylesheet" href="css/styles.css">
13  <link href="https://fonts.googleapis.com/css?family=Dancing+Script" rel="stylesheet">
14
15 </head>
16
17 <body>
18 <nav class="navbar navbar-expand-lg navbar-light" style="background-color: #66BF6F;">
19   <a class="navbar-brand" href="/">MyBooking.com</a>
20   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
21     aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
22     <span class="navbar-toggler-icon"></span>
23   </button>
24
25   <div class="collapse navbar-collapse" id="navbarSupportedContent">
26     <ul class="navbar-nav mr-auto">
27       <li class="nav-item active">
28         <a class="nav-link" href="/">Home <span class="sr-only">(current)</span></a>
29       </li>
30       <li class="nav-item">
31         <a class="nav-link" href="about">About us</a>
32       </li>
33       <li class="nav-item">
34         <a class="nav-link" href="search">Search</a>
35       </li>

```

Σχήμα 7.11: Header

```

1
2 <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" crossorigin="anonymous"></script>
3 <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js" crossorigin="anonymous"></script>
4 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" crossorigin="anonymous"></script>
5
6 </body>
7
8 <footer>
9   <div class="bottom-container centered">
10    <a class="footer-link" href="https://www.linkedin.com/">Facebook</a>
11    <a class="footer-link" href="https://twitter.com/">Twitter</a>
12    <a class="footer-link" href="https://www.appbrewery.co/">FAQ</a>
13    <p class="copyright">© 2019 Konstantinos Tsoumanis</p>
14  </div>
15 </footer>
16
17 </html>

```

Σχήμα 7.12: Footer

7.2 Πλατφόρμες και προγραμματιστικά εργαλεία

7.2.1 Node.js και Express.js

Το Node.js είναι μια πλατφόρμα ανάπτυξης λογισμικού, χτισμένη σε περιβάλλον Javascript. Στόχος του είναι να παρέχει ένα εύκολο τρόπο δημιουργίας κλιμακωτών διαδικτυακών εφαρμογών. Σε αντίθεση με τα περισσότερα σύγχρονα περιβάλλοντα ανάπτυξης εφαρμογών δικτύων, μία διεργασία node δεν στηρίζεται στην πολυνηματικότητα αλλά σε ένα μοντέλο ασύγχρονης επικοινωνίας εισόδου/εξόδου.

Το Node χαρακτηρίζεται από την έμφαση στην ασύγχρονη επικοινωνία μεταξύ των υπολογιστικών πόρων. Αυτό επιτυγχάνεται με την χρήση συμβάντων (events) που προσφέρει η Javascript και ονομάζονται callbacks. Για παράδειγμα όταν ένας περιηγητής ιστού φορτώσει πλήρως ένα αρχείο, ένας χρήστης πατάει κάποιο κουμπί, ολοκληρώνεται ένα αίτημα AJAX, τα συμβάντα αυτά πυροδοτούν ένα συγκεκριμένο callback. Αυτό με την σειρά του επιτρέπει την ροή του κώδικα χωρίς να αφήνει ανενεργό τον επεξεργαστή προκειμένου να εκτελεστεί μια λειτουργία, όπως μια επιτυχής ανάγνωση αρχείου από τον δίσκο.

Ως ένα ασύγχρονο σύστημα βασισμένο στο περιβάλλον της Javascript, το Node.js έχει σχεδιαστεί για τη δημιουργία επεκτάσιμων εφαρμογών δικτύου στις οποίες πολλές συνδέσεις μπορούν να αντιμετωπιστούν ταυτόχρονα. Σε κάθε σύνδεση η κλήση ενεργοποιείται, αλλά αν δεν υπάρχει εργασία ο κόμβος θα "κοιμηθεί".

Αυτό έρχεται σε αντίθεση με το πιο συνηθισμένο μοντέλο κοινής χρήσης του σήμερα, όπου χρησιμοποιούνται τα νήματα του λειτουργικού συστήματος. Η δικτύωση με βάση τα νήματα είναι σχετικά αναποτελεσματική και πολύ δύσκολη στη χρήση. Επιπλέον, οι χρήστες του Node δεν έχουν ανησυχίες για περιπτώσεις αδιεξόδων, καθώς δεν υπάρχουν locks. Σχεδόν καμία λειτουργία στο Node δεν εκτελεί απευθείας I/O, οπότε η διαδικασία δεν μπλοκάρει ποτέ. Επειδή τίποτα δεν εμποδίζει, τα κλιμακωτά συστήματα είναι πολύ λογικό να αναπτυχθούν με τη χρήση του συγκεκριμένου εργαλείου.

Το Node.js είναι παρόμοιο στο σχεδιασμό και επηρεάζεται από συστήματα όπως το Event Machine της Ruby ή το Twisted της Python. Το Node πηγαίνει το μοντέλο συμβάντος (event) λίγα βήματα παρακάτω. Παρουσιάζει έναν βρόχο συμβάντος ως δομή runtime αντί για βιβλιοθήκη. Εισέρχεται στον κύκλο συμβάντων μετά την εκτέλεση της δέσμης ενεργειών εισαγωγής και εξέρχεται από τον βρόχο συμβάντος όταν δεν υπάρχουν άλλες κλήσεις προς εκτέλεση.

Παρόλο που το Node έχει σχεδιαστεί δίχως να χρησιμοποιεί νήματα, δεν σημαίνει ότι δεν μπορεί να υπάρξει εκμετάλλευση πιθανών πολλαπλών πυρήνων. Διαδικασίες - παιδιά μπορούν να δημιουργηθούν και έχουν σχεδιαστεί έτσι ώστε να επικοινωνούν εύκολα. Όντας χτισμένο πάνω στο ίδιο interface, δίνει τη δυνατότητα να μοιραστούν τα sockets μεταξύ των διεργασιών, έτσι ώστε να ενεργοποιηθεί η εξισορρόπηση φορτίου στους πυρήνες.

Τέλος το Express.js, ή απλά το Express, είναι ένα πλαίσιο διαδικτυακής εφαρμογής για το Node.js, το οποίο κυκλοφορεί ως ελεύθερο και ανοιχτό λογισμικό με άδεια MIT. Έχει σχεδιαστεί για την κατασκευή εφαρμογών ιστού και API και ουσιαστικά αποτελεί το συμπληρωματικό πρότυπο του Node.js.

7.2.2 Navicat

Στο κομμάτι των δεδομένων εμφανίστηκε η πιο σοβαρή πρόκληση για την εκπόνηση αυτής της εργασίας. Οι εταιρίες κολοσσοί του τομέα που μας απασχολεί έχουν αποθηκευμένα τα δεδομένα τους σε σχεσιακές (SQL) βάσεις, ενώ ο στόχος της εργασίας ήταν να υλοποιηθεί με τη χρήση NoSQL βάσης. Μετά από έρευνα διαπιστώθηκε πως η καλύτερη επιλογή για την επίλυση της ήταν το πρόγραμμα "Navicat", που δίνει τη δυνατότητα μετατροπής μίας SQL βάσης δεδομένων σε NoSQL και ειδικότερα σε MongoDB. Πρόκειται για ένα πολύ χρήσιμο εργαλείο που προορίζεται για καθαρά επαγγελματική χρήση, ταυτόχρονα όμως παρέχει τη δυνατότητα δοκιμαστικής χρησιμοποίησης του για 30 ημέρες. Έχει λάβει σημαντικά βραβεία για την αποτελεσματικότητα και τη χρησιμότητα των υπηρεσιών του, ενώ σε αυτό στηρίζονται και πάρα πολλές εταιρίες ανά τον κόσμο.

7.2.3 Heroku

Το Heroku είναι μια cloud πλατφόρμα ως υπηρεσία (PaaS) που υποστηρίζει πολλές γλώσσες προγραμματισμού. Ως μια από τις πρώτες cloud πλατφόρμες, το Heroku ξεκίνησε να αναπτύσσεται από τον Ιούνιο του 2007, όταν και υποστήριζε μόνο τη γλώσσα προγραμματισμού Ruby. Πλέον υποστηρίζει γλώσσες όπως Java, Node.js, Scala, Clojure, Python, PHP και Go. Το Heroku είναι ουσιαστικά μία υπηρεσία που απευθύνεται σε developers, προσφέροντας το κατάλληλο περιβάλλον για άμεση και πλήρη φιλοξενία web εφαρμογών. Μέσω του εργαλείου αυτού η εφαρμογή της εργασίας αυτής θα μπορεί να είναι επισκέψιμη από κάθε πλευρά του κόσμου μέσω ενός URL. Το Heroku επιλέχθηκε όμως κυρίως λόγω της ευελιξίας που προσφέρει στον developer για άμεσες αλλαγές, αλλά και του οικοσυστήματος υπηρεσιών (add-ons) που παρέχει.

Πρόσθετο mLab (Add-on)

Για να ανέβει η βάση δεδομένων της εφαρμογής στο Heroku, έπρεπε να γίνει η προσθήκη του βοηθητικού εργαλείου mLab στο project που δημιουργήθηκε στην πλατφόρμα. Μέσω του mLab ο διαχειριστής της βάσης μπορεί να εκτελεί διάφορες ενέργειες, όπως να κρατάει αντίγραφα της βάσης για περιπτώσεις βλαβών και να επιβλέπει τη λειτουργία της μέσω των στατιστικών που του παρέχει το πρόσθετο.

Κεφάλαιο 8

Επίλογος

8.1 Σύνοψη και συμπεράσματα

Η συγκεκριμένη διπλωματική λαμβάνοντας σοβαρά υπόψιν τις συνθήκες που επικρατούν αυτή τη στιγμή σε παγκόσμια κλίμακα, προσπάθησε να βοηθήσει τους ανθρώπους που σχεδιάζουν τα ταξίδια μέσω του διαδικτύου. Πάρθηκε η απόφαση να κινηθεί σε αντίθετη κατεύθυνση από το γενικότερο κλίμα μαζοποίησης των ταξιδιωτών και πρότεινε ένα σύστημα που βοηθάει τον κάθε χρήστη του να λάβει πιο εύκολα αποφάσεις που ταιριάζουν στην αισθητική του.

Πιο συγκεκριμένα, ο αλγόριθμος μηχανικής μάθησης που αναπτύχθηκε αναλαμβάνει το δύσκολο αυτό έργο. Στοιχεία που δίνονται από τον χρήστη σε πραγματικό χρόνο συνδυάζονται και συγκρίνονται με αντίστοιχα προηγούμενων χρηστών ώστε να προκύψει μία εντελώς εξατομικευμένη πρόταση. Ο χρήστης δεν περιορίζεται από το σύστημα και έχει πάντοτε τη δυνατότητα να μην επιλέξει την πιο "ταιριαστή" επιλογή από την προτεινόμενη λίστα καταλυμάτων, ενώ ταυτόχρονα έχει ενημέρωση για βασικές πληροφορίες κάθε καταλύματος.

Τέλος, το εγχείρημα της χρήσης μίας τεχνολογίας που δεν είχε χρησιμοποιηθεί στο παρελθόν για το συγκεκριμένο σκοπό, απέφερε ιδιαίτερα χρήσιμα συμπεράσματα. Η χρησιμοποίηση βάσης δεδομένων NoSQL στη χρονική περίοδο που εκοπηθήκε η εργασία αποδείχθηκε πρόκληση, καθώς το γενικότερο περιβάλλον του τουρισμού είναι βασισμένο στη λογική των SQL βάσεων δεδομένων. Αυτό έκανε δύσκολη την αξιοποίηση υπάρχοντων δεδομένων και οδήγησε στην ανάπτυξη νέων μοντέλων δεδομένων, προκειμένου να αναπαραστηθούν κατάλληλα οι σχέσεις μεταξύ των οντοτήτων. Παρ' όλα αυτά, υπάρχουν δεδομένα πλεονεκτήματα που αποδεικνύουν ότι οι NoSQL βάσεις έχουν τη δυνατότητα να πρωταγωνιστήσουν τα επόμενα χρόνια έναντι των σχεσιακών. Ο συνεχώς αυξανόμενος ρυθμός αποθήκευσης δεδομένων που αφορούν τους πελάτες, έχοντας ως στόχο την εξατομίκευση των παροχών υπηρεσιών, συνετέλεσε στη δημιουργία τεράστιου όγκου δεδομένων στον τομέα του τουρισμού. Οι NoSQL βάσεις είναι εξοπλισμένες με τεχνολογίες που μπορούν να διευκολύνουν τη διαχείριση μεγάλης ποσότητας δεδομένων, ενώ ταυτόχρονα προσφέρει ευελιξία με δυνατότητα αλλαγής των σχημάτων δεδομένων δίχως να απαιτεί την εφαρμογή της αλλαγής σε ολόκληρη τη βάση δεδομένων.

8.2 Μελλοντικές επεκτάσεις

Ο αλγόριθμος που δημιουργήθηκε και κυρίως η λογική που αναπτύχθηκε, μπορεί να εφαρμοστεί και σε άλλους τομείς του τουρισμού. Μία κατεύθυνση στην οποία θα μπορούσαν να κινηθούν μελλοντικές προσπάθειες είναι η βοήθεια στην επιλογή προορισμού για τους ταξιδιώτες. Θα ήταν πραγματικά ενδιαφέρον να ζητούνται από τον χρήστη τα ενδιαφέροντα του και αφού περαστούν στον αλγόριθμο να παρουσιάζεται μια λίστα με τους προτεινόμενους προορισμούς, προσφέροντας έτσι μία ολοκληρωμένη ταξιδιωτική εμπειρία.

Παράρτημα I

Αρχεία της εφαρμογής

Τα αρχεία που δημιουργήθηκαν για την εφαρμογή της διπλωματικής εργασίας, επιλέχθηκε να βρίσκονται συνεχώς διαθέσιμα στο διαδίκτυο. Με αυτόν τον τρόπο, όποιος διαβάσει την εργασία θα μπορεί να ελέγχει άμεσα τον τρόπο με τον οποίο έχουν δομηθεί τα διάφορα στάδια και να επεξεργαστεί τον κώδικα της. Σε αυτό το κομμάτι, φάνηκε χρήσιμη η γνωστή πλατφόρμα του GitHub που φιλοξενεί όλα τα απαραίτητα αρχεία.

I.1 GitHub

Όλοι οι ενδιαφερόμενοι μπορούν να έχουν πρόσβαση στα αρχεία της εφαρμογής, επισκεπτόμενοι το `github.com` και συγκεκριμένα το αποθετήριο της στο `https://github.com/tsoumk/Thesis`.

Ο φάκελος `”php”`, περιέχει τρία αρχεία της μορφής `”.php”`: `config.php`, `functions.php` και `tripadvisor.php`. Τα αρχεία αυτά βοήθησαν στην ανάκτηση δεδομένων από το σάιτ της υπηρεσίας `tripadvisor`, τα οποία βοήθησαν στον αληθοφανή τρόπο λειτουργίας του συστήματος.

Ο επόμενος φάκελος (`public`), περιέχει τους υποφακέλους `”css”` και `”images”` που φιλοξενούν τον κώδικα CSS που χρειάστηκε η εφαρμογή για το στυλιστικό μέρος της και τις εικόνες που χρησιμοποιήθηκαν αντίστοιχα.

Ακολουθεί ο φάκελος `”views”`, στον οποίο περιλαμβάνονται αρχεία της μορφής `”.ejs”` και αφορούν τις διάφορες σελίδες που συναντάει ο χρήστης κατά την περιήγηση του στην εφαρμογή. Στον υποφάκελο `”partials”` υπάρχουν τα αρχεία που είναι υπεύθυνα για την εμφάνιση του `”header”` και του `”footer”` σε κάθε σελίδα.

Το αρχείο `”.gitignore”` είναι ένα προαιρετικό αρχείο που όμως προτείνεται για μεγαλύτερη ασφάλεια του κώδικα. Σε αυτό το αρχείο καθορίστηκε ποιες πληροφορίες θα παραβλεφθούν κατά το ανέβασμα των αρχείων στο `”GitHub”`.

Το `”app.js”` αποτελεί το βασικό αρχείο της εργασίας και δίνει πληροφορίες όπως το πως συνδεόμαστε με τη βάση δεδομένων ή το πως θα γίνει η τελική σύσταση προς τον χρήστη.

Τα δύο αρχεία που ακολουθούν (`package-lock.json` και `package.json`) είναι δύο αυτόματα παραγόμενα αρχεία που περιλαμβάνουν βασικές πληροφορίες του Node.js για π.χ. πακέτα που έχουν εγκατασταθεί κατά τον προγραμματισμό.

Βιβλιογραφία

- [1] ChampionTraveler. Are People Traveling More? 2017. <https://championtraveler.com/news/here-is-how-much-u-s-outbound-travel-has-increased-around-the-world-the-last-20-years/>.
- [2] Laurent Charlin. Supervised and Active Learning for Recommender Systems. Μεταπτυχιακή διπλωματική εργασία, Computer Science Department in University of Toronto, 2014. <https://pdfs.semanticscholar.org/9ef9/9b04b7defafd60ba0bb38a1c4bbd45de2ee0.pdf>.
- [3] Konstantinos Mastakas Grammati Pantziou Damianos Gavalas, Charalampos Konstantopoulos. Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, 39:319–333, 2014. <https://doi.org/10.1016/j.jnca.2013.04.006>.
- [4] Huba Gaspar. The Cold Start Problem for Recommender Systems. 2015. <https://www.yuspify.com/blog/cold-start-problem-recommender-systems/>.
- [5] Panagiota Dionysopoulou Ioanna Bizirgianni. The Influence of Tourist Trends of Youth Tourism through Social Media (SM) & Information and Communication Technologies (ICTs). *Procedia - Social and Behavioral Sciences*, 73:652–660, 2013. <https://doi.org/10.1016/j.sbspro.2013.02.102>.
- [6] Aida Valls Joan Borràs, Antonio Moreno. Intelligent tourism recommender systems: A survey. *Procedia - Social and Behavioral Sciences*, 41(16):7370–7389, 2014. <https://doi.org/10.1016/j.eswa.2014.06.007>.
- [7] InData Labs. Approaching the Cold Start Problem in Recommender Systems. 2016. <https://medium.com/@InDataLabs/approaching-the-cold-start-problem-in-recommender-systems-e225e0084970>.
- [8] Stanley Loh, Fabiana Lorenzi, Ramiro Saldana Garin και Daniel Lichtnow. A Tourism Recommender System Based on Collaboration and Text Analysis. *J. of IT & Tourism*, 6:157–165, 2003.
- [9] I.García A.Garrido E.Onaindia L.Sebastiá, D.Yuste. A highly interactive tourism recommender system for multi-day trips. Στο *Proceedings of the Workshop on Tourism Recommender System*, σελίδες 1–10, Vienna, Austria, 2015. The ACM Conference Series. <http://deim.urv.cat/~itaka/workshops/recsys2015/Files/ProceedingsTourRS15.pdf>.
- [10] World Tourism Organization. International Tourist Arrivals Reach 1.4 billion Two Years Ahead of Forecasts. 2019. <http://www2.unwto.org/press-release/2019-01-21/international-tourist-arrivals-reach-14-billion-two-years-ahead-forecasts>.

-
- [11] Emili Roger Ciurana Simó. Development of a Tourism recommender system. Μεταπτυχιακή διπλωματική εργασία, Universitat Politècnica de Catalunya, 2012. <https://upcommons.upc.edu/bitstream/handle/2099.1/16444/Emili%20Ciurana%20Simo%20MIA.pdf>.
- [12] Bing Pan Stephen W.Litvin, Ronald E.Goldsmith. Electronic word-of-mouth in hospitality and tourism management. *Tourism Management*, 29(3):458–468, 2008. <https://doi.org/10.1016/j.tourman.2007.05.011>.
- [13] Zheng Xiang, Wang Dan, Joseph O’Leary και D.R. Fesenmaier. Adapting to the Internet: Trends in Travelers’ Use of the Web for Trip Planning. *Journal of Travel Research*, 54, 2014. <https://doi.org/10.1177/0047287514522883>.

Συντομογραφίες

CDN	Content Delivery Network
RDBMS	relational database management system
CPU	Central Processing Unit
FS	File System
RAM	Random access memory
AI	Artificial Intelligence
RS	Recommender Systems
WOM	Word Of Mouth

Ορολογία - Γλωσσάρι

Ελληνικός όρος

Αναζήτηση

Αναζήτηση Πλήρους Κειμένου

Από στόμα σε στόμα

Βαθμολογία

Διαμοιρασμός

Δομή

Επικεφαλίδα

Ευρετήριο

Ταίριασμα

Ταυτότητα

Τοποθεσία

Συνάθροιση

Φιλτράρισμα

Υποσέλιδο

Αγγλικός όρος

Query

Full-Text-Search

Word Of Mouth

Rating

Sharding

Framework

Header

Index

Match

Id

Location

Aggregation

Filtering

Footer

