



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**An application for booking in London, using OSM data and the Schema. org
ontology**

Γιάννης Ν. Γιαννακίδης

Επιβλέπωντας: Κουμπάρακης Μανόλης, Καθηγητής ΕΚΠΑ

Συνεπιβλέπωντας: Σταμούλης Γεώργιος, Υποψήφιος Διδάκτορας ΕΚΠΑ

ΑΘΗΝΑ
ΙΟΥΝΙΟΣ 2020

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

An application for booking in London, using OSM data and the Schema. org ontology

Γιάννης Ν. Γιαννακίδης

A. M. : 1115201500025

ΕΠΙΒΛΕΠΟΝΤΑΣ: **Κουμπάρακης Μανόλης, Καθηγητής ΕΚΠΑ**

ΣΥΝΕΠΙΒΛΕΠΟΝΤΑΣ: **Σταμούλης Γεώργιος, Υποψήφιος Διδάκτορας ΕΚΠΑ**

ΠΕΡΙΛΗΨΗ

Την εποχή που διανύουμε στην Ελλάδα αλλά και στον κόσμο, ένας από τους πιο σημαντικούς παράγοντες για την οικονομία είναι ο τουρισμός και τα πολλά ωφέλη αυτού. Πολλές φορές ως τουρίστες του εσωτερικού ή του εξωτερικού για λόγους ψυχαγωγίας, εκπαίδευσης ή εργασίας έχουμε όλοι μας βρεθεί αντιμέτωποι με το άγνωστο. Οι πληροφορίες που υπάρχουν την τωρινή εποχή στο διαδίκτυο είναι πάρα πολλές και υπεραρκετές για να μάθουμε πληροφορίες για μια καινούργια περιοχή που επισκεπτόμαστε για ένα σύντομο ή αλλά και μεγαλύτερο χρονικό διάστημα. Το πρόβλημα της ύπαρξης μεγάλου όγκου πληροφοριών είναι πως είναι διάσπαρτες σε πολλούς ιστοχώρους σε όλο το διαδίκτυο και υπάρχει δυσκολία στην εύρεση πληροφοριών καθώς είναι αρκετά χρονοβόρο.

Έχοντας τα παραπάνω ως δεδομένα, η πτυχιακή αυτή έχει ως κύρια ιδέα την διασύνδεση αυτών των πληροφοριών και την παρουσίασή τους ως οργανωμένες και συνδεδεμένες μεταξύ τους ως μία οντολογία σε έναν ιστοχώρο. Το όνομα αυτού του ιστοχώρου είναι London Bookings και μπορεί ο αναγνώστης να τον βρει στον προσωπικό μου λογαριασμό στο GitHub [1]. Όπως μπορεί εύκολα να διακρίνει κάποιος και από τον τίτλο η διαδικτυακή εφαρμογή αφορά έναν τουριστικό οδηγό με επίκεντρο το Λονδίνο. Η γλώσσα που χρησιμοποιήθηκε για την υλοποίηση του back-end (οπίσθιο μέρος) της εφαρμογής ήταν η Java μαζί με το Strabon [2] που αποτελεί ένα SPARQL endpoint (τελικό σημείο) για την διασύνδεση των πληροφοριών από το διαδίκτυο και την αποθήκευση τους ενώ στο front-end (μετωπιαίο μέρος) χρησιμοποιήθηκαν HTML, CSS, Javascript και κάποιες επιπρόσθετες βιβλιοθήκες της τελευταίας.

Η εφαρμογή αυτή μπορεί να χρησιμοποιηθεί από ένα ευρύ πλήθος κόσμου και ο χρήστης έχει την δυνατότητα να αναζητήσει πληροφορίες αρχικά για ξενοδοχεία του Λονδίνου καθώς και για μουσεία, εστιατόρια, πάρκα, καζίνο, νοσοκομεία αλλά και κινηματογράφους. Ένας χρήστης που έχει λογαριασμό έχει την δυνατότητα να κάνει κράτηση σε ένα ξενοδοχείο καθώς και να δει τις κρατήσεις του. Στις κρατήσεις του περιληπτικά μπορεί να αξιολογήσει τις παλιές και να ακυρώσει τις μελλοντικές.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Ανάπτυξη διαδικτυακής εφαρμογής, διασύνδεση δεδομένων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Java, SPARQL endpoint

ABSTRACT

In today's age in Greece and all over the world, one of the most important factors in the economy is tourism and its benefits. Many times as domestic or foreign tourists for entertainment, education or work reasons we were all in the position of the unknown. The plethora of information in today's age on the internet for learning information for a new location that we are visiting for a short or bigger amount of time is too much. The problem of the existence of plethora of information is that they are scattered in many websites and it is difficult and time consuming to find them.

With all that in mind, this thesis has as main idea to interlink those information and represent them as organised and connected to each other as one ontology in a website. The name of this website is London Bookings and the reader can find it to my personal Github account . As anyone can see with ease, this website is about a tourist guide with London as it's center. Java was used as the backend language with Strabon as a SPARQL endpoint for interlinking information from the internet and storing them and in the front-end HTML, CSS, Javascript and some extra libraries of the latter were used.

This application can be used by a wide range of people and the user has the ability to search information for hotels from London and for museums, restaurants, parks, casinos, hospitals and cinemas. If a user has an account he has the ability to make a reservation to a hotel and check his reservations. Briefly, in his reservations he can rate the past ones and cancel the future ones.

SUBJECT AREA: Web application , interlinking information

KEYWORDS: Java, SPARQL endpoint

*Η παρούσα πτυχιακή είναι αφιερωμένη στους γονείς μου και στην αδερφή μου καθώς και
I στους φίλους μου που ήταν δίπλα μου σε κάθε εύκολη και δύσκολη στιγμή.*

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τους επιτηρητές μου, τον κύριο Κουμπάρáκη για την άμεση ανταπόκριση του στην εκκίνηση της πτυχιάκης και στην ολοκλήρωση της καθώς και τον κύριο Σταμούλη που με βοήθουσε καθόλη την διάρκεια της εκπόνησης της με πολύ κατατοπιστικές πληροφορίες.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ.....	8
1. ΕΙΣΑΓΩΓΗ.....	10
2. ΕΡΕΥΝΑ.....	11
2. 1 Linked data.....	11
2. 2 RDF.....	12
2. 3 SPARQL.....	12
2. 4 Schema. org.....	13
3. Εφαρμογή.....	14
3. 1 Ρόλος εφαρμογής.....	14
3. 2 Αρχιτεκτονική εφαρμογής.....	15
3. 3 Datasets.....	17
3. 4 Features.....	20
4. Μελλοντικές επεκτάσεις.....	30
5. Παραρτήματα.....	30
5. 1 SPARQL queries.....	30
5. 2 Κώδικας.....	36
6. ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ.....	38
7. ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ.....	40
8. ΑΝΑΦΟΡΕΣ.....	41

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Αποτέλεσμα.....	σελ.13
Εικόνα 2: Αρχιτεκτονική εφαρμογής.....	σελ.15
Εικόνα 3: Οντολογία.....	σελ.18
Εικόνα 4: N-triples.....	σελ.19
Εικόνα 5: Αρχική.....	σελ.21
Εικόνα 6: Αρχική συνέχεια.....	σελ.21
Εικόνα 7: Αρχική σύνδεση.....	σελ.22
Εικόνα 8: Sign-up.....	σελ.23
Εικόνα 9: Επεξεργασία προφίλ.....	σελ.24
Εικόνα 10: Ξενοδοχεία.....	σελ.25
Εικόνα 11: Κράτηση.....	σελ.26
Εικόνα 12: Παλιά κράτηση.....	σελ.27
Εικόνα 13: Μελλοντική κράτηση.....	σελ.27
Εικόνα 14: Εστιατόρια.....	σελ.28
Εικόνα 15: Νοσοκομεία.....	σελ.29
Εικόνα 16: Query Αρχικής σελίδας.....	σελ.31
Εικόνα 17: Σελίδα ξενοδοχείων.....	σελ.31
Εικόνα 18: Ταξινόμηση.....	σελ.32
Εικόνα 19: Αναζήτηση με όνομα.....	σελ.33
Εικόνα 20: Αναζήτηση κουζίνας.....	σελ.33
Εικόνα 21: Τύποι κουζίνας.....	σελ.34
Εικόνα 22: Ώρα opening.....	σελ.34
Εικόνα 23: Κατοικίδια.....	σελ.35
Εικόνα 24: Αξιολόγηση.....	σελ.36
Εικόνα 25: HTTP request.....	σελ.36
Εικόνα 26: SQL query.....	σελ.37

1. ΕΙΣΑΓΩΓΗ

Στην συγχρονή εποχή όλοι μας έχουμε πρόσβαση μέσω του κινητού μας τηλεφώνου ή μέσω του προσωπικού μας υπολογιστή στο διαδίκτυο. Υπάρχουν πλέον πάρα πολλές εφαρμογές για όλες τις ανάγκες μας αλλά όταν επισκεπτόμαστε έναν προορισμό για ένα σύντομο χρονικό διάστημα και χρειαζόμαστε άμεσα κάποιες πληροφορίες, το να κατεβάσουμε μια εφαρμογή δεν είναι πολλές φορές αποδοτικό. Για αυτόν τον λόγο μια διαδικτυακή εφαρμογή αποτελεί έναν γρήγορο τρόπο να βρούμε πληροφορίες για ένα εστιατόριο π. χ. και στον συγκεκριμένο ιστοχώρο όλες οι πληροφορίες είναι διαθέσιμες για τον χρήστη χωρίς να χρειαστεί να δημιουργηθεί κάποιος λογαριασμός.

2. ΕΡΕΥΝΑ

2.1 Linked data

Τα linked data (συνδεδεμένα δεδομένα) είναι δομημένα δεδομένα που είναι συνδεδεμένα με άλλα δεδομένα προκειμένου να είναι περισσότερο χρήσιμα μέ την χρήση semantic queries (σημασιολογικά ερωτήματα) . Χρησιμοποιούνται κυρίως σε διαδικτυακές τεχνολογίες όπως HTTP παρά την γνωστή χρήση των δεδομένων για διαδικτυακές σελίδες με ανθρώπους ως αναγνώστες. Είναι δομημένα με τέτοιον τρόπο έτσι ώστε να μπορούν να διαβαστούν μέσω μίας αυτοματοποιημένης διαδικασίας απο υπολογιστές. Ένας βασικός λόγος ύπαρξης τους είναι η απόπειρα του να γίνει ολόκληρο το διαδίκτυο μια παγκόσμια βάση δεδομένων. Οι βασικές αρχές είναι οι εξής:

- Χρήση URIs για να ονοματιστούν τα αντικείμενα
- Χρήση HTTP URIs για να μπορούν να αναζητηθούν αυτά τα αντικείμενα
- Προβολή πληροφοριών για το τι συμβολίζει το κάθε όνομα όταν αναζητηθεί μέσω πχ RDF
- Αναφορά σε άλλα αντικείμενα με την χρήση των HTTP URI ονομάτων τους όταν δημοσιευτούν στο διαδίκτυο

Η κεντρική ιδέα πίσω απο αυτές τις αρχές έχει δύο όψεις. Η πρώτη είναι να χρησιμοποιούνται πρότυπα για την αναπαράσταση και την πρόσβαση των δεδομένων στο διαδίκτυο. Η δεύτερη είναι οι αρχές διαδίδουν την ύπαρξη υπερσυνδέσμων μεταξύ των δεδομένων απο διαφορετικές πηγές. Αυτοί οι υπερσύνδεσμοι έχουν την δυνατότητα να ενώσουν όλα τα linked data σε έναν παγκόσμιο γράφο δεδομένο με παρόμοιο τρόπο όπως το διαδίκτυο ενώνει τις HTML σελίδες σε έναν παγκόσμιο χώρο πληροφοριών.

Για να μπορεί να γίνει πραγματικότητα η δημιουργία ενός διαδικτύου απο linked data είναι αρκετά σημαντικό να υπάρχει μεγάλος όγκος δεδομένων διαθέσιμος στο διαδίκτυο καθώς και να μπορεί να είναι προσβάσιμος και διαχειρίσιμος. Επίσης διαθέσιμη πρέπει να είναι η σχέση ανάμεσα σε αυτά τα δεδομένα για να μπορούν να μετατραπούν σε κάποιο κοινή μορφή πχ RDF.

2.2 RDF

Το RDF [3] αποτελεί ένα από τα βασικά μοντέλα στο διαδίκτυο για ανταλλαγή δεδομένων. Διαθέτει πολλά χαρακτηριστικά προκειμένου να γίνεται πιο εύκολη η διαδικασία συγχώνευσης των δεδομένων ακόμα και αν η δομή τους διαφέρει. Επίσης, προσφέρει την δυνατότητα της εξέλιξης της δομής των δεδομένων κατά το πέρασμα του χρόνου με τις ελάχιστες δυνατές αλλαγές στους παραλείπτες αυτών των δεδομένων. Τα δεδομένα αποθηκεύονται σε μορφή triples (τριάδες) “subject-predicate-object” (υποκείμενο-κατηγορούμενο-θέμα). Το μοντέλο RDF μπορεί και να θεωρηθεί και ως ένας πίνακας με 3 στήλες. Η πρώτη στήλη για τα subject , η δεύτερη για τα predicate και η τρίτη για τα object. Το predicate δηλώνει χαρακτηριστικά ή πτυχές και εκφράζει την σχέση ανάμεσα σε subject και object. Για παράδειγμα στην πρόταση “Ο ουρανός έχει χρώμα μπλε” σε μορφή triple τότε το subject είναι το “Ο ουρανός” το predicate είναι το “έχει χρώμα” και το object είναι το “μπλε”. Το RDF αποτελεί ένα αφηρημένο μοντέλο με πολλές διαφορετικές μορφές αρχείων οπότε η κωδικοποίηση του διαφέρει ανάλογα με την μορφή του αρχείου.

Ένα σύνολο από RDF triples εγγενώς αντιπροσωπεύει έναν κατευθυνόμενο γράφο. Στην θεωρία αυτό κάνει το RDF ένα μοντέλο δεδομένων που ταιριάζει καλύτερα σε αναπαράσταση πληροφορίας σε σχέση με άλλα μοντέλα για οντολογίες αλλά στην πράξη αυτό δεν ισχύει καθώς συνήθως αποθηκεύεται σε κάποια σχεσιακή βάση δεδομένων.

2.3 SPARQL

Η SPARQL αποτελεί μία σημασιολογική query language για δεδομένα αποθηκευμένα σε RDF format. Χρησιμοποιείται για την ανάκτηση των δεδομένων καθώς και για πιθανή διαχείριση τους με διάφορους τρόπους. Η SPARQL προσφέρει ένα πλήρες σύνολο διαδικασιών προκειμένου να κάνει query κάποιο RDF triple. Οι βασικές παραλλαγές των query είναι το SELECT query για να γίνεται ανάκτηση τιμών από ένα SPARQL endpoint. Το CONSTRUCT query για να γίνεται ανάκτηση τιμών και μετατροπή σε RDF. Το ASK query όπου επιστρέφεται ως απάντηση True/False και το DESCRIBE query για να γίνεται ανάκτηση ενός RDF γράφου. Το SPARQL endpoint που αναφέρθηκε προηγουμένως αποτελεί ένα σημείο παρουσίας σε ένα δίκτυο HTTP όπου έχει την ικανότητα να λαμβάνει και να διαχειρίζεται SPARQL Protocol αιτήματα. Το SPARQL Protocol είναι ένα πρωτόκολλο βασισμένο σε HTTP για να εκτελούνται διαδικασίες σε SPARQL μέσω του SPARQL endpoint όπως για παράδειγμα ένα SELECT query.

Η SPARQL μπορεί να χρησιμοποιηθεί για να εκφράσει ερωτήματα ανάμεσα σε ποικίλες πηγές δεδομένων. Επίσης έχει την δυνατότητα για να εξετάσει μοτίβα γράφων μαζί με τις συνδέσεις και αποσυνδέσεις τους.

Ένα χαρακτηριστικό παράδειγμα ενός SPARQL query για τα δεδομένα σε μορφή triples είναι

DATA:

Subject	Predicate	Object
<http://example. org/book/book1>	<http://purl. org/dc/elements/1. 1/title>	"SPARQL Tutorial" .

QUERY:

SELECT ?title

WHERE

```
{  
    <http://example. org/book/book1> <<http://purl. org/dc/elements/1. 1/title> ?title .  
}
```

Όπου έχει το εξής αποτέλεσμα

title
"SPARQL Tutorial"

Εικόνα 1: Αποτέλεσμα

Μια επέκταση της SPARQL είναι η GeoSPARQL. Η GeoSPARQL αποτελεί ένα στάνταρ για την αναπαράσταση και την εκτέλεση ερωτημάτων σε γεωχωρικά συνδεδεμένα δεδομένα για το σημασιολογικό διαδίκτυο από το OGC. Ο ορισμός μιας μικρής οντολογίας βασισμένη στις αρχές του OGC αποσκοπεί στο να παρέχει μια βάση ανταλλαγής για γεωχωρικά δεδομένα RDF που υποστηρίζει ποιοτική καθώς και ποσοτική συλλογιστική χώρου και για να κάνει ερωτήματα σε μια βάση δεδομένων SPARQL.

2.4 Schema. org

Το Schema. org [4] αποτελεί μια συλλογική διαδικασία με σκοπό την δημιουργία, συντήρηση και προώθηση δομημένων δεδομένων στο διαδίκτυο, σε ιστοσελίδες, σε μηνύματα ηλεκτρονικής αλληλογραφίας αλλά και όχι μόνο. Το λεξιλόγιο του Schema. org μπορεί να χρησιμοποιηθεί με ποικίλες κωδικοποιήσεις που έχουν την δυνατότητα να καλύπτουν οντότητες, σχέσεις μεταξύ οντοτήτων καθώς και δράσεις και μπορούν με ευκολία να επεκταθούν. Ένα κοινό λεξιλόγιο κάνει πιο απλουστευμένη την διαδικασία

απόφασης για ένα σχήμα και αποτελεί μια απόπειρα να υπάρχει μια κοινή συλλογή σχημάτων στο διαδίκτυο.

Υπάρχει ένα τεράστιο πλήθος αντικειμένων που μπορούν να σημανθούν μέσω του Schema. org σε μία ιστοσελίδα. Μερικά παραδείγματα είναι

- Article (άρθρο)
- Movie (ταινία)
- Recipe (συνταγή)
- LocalBusiness (τοπική επιχείρηση)

Ένα παράδειγμα σήμανσης πληροφοριών για μία ταινία χρησιμοποιώντας σχήματα του Schema. org και microdata (μικροδεδομένα) που αποτελεί μια προδιαγραφή του HTML για να εμφωλεύονται μεταδεδομένα.

```
<div itemscope itemtype=http:"schema. org/Movie">
```

```
<h1 itemprop="name">Avatar</h1>
```

```
</div>
```

Το παραπάνω σχήμα επισημαίνει την ταινία “Avatar” χρησιμοποιώντας το το σχήμα Movie του Schema. org .

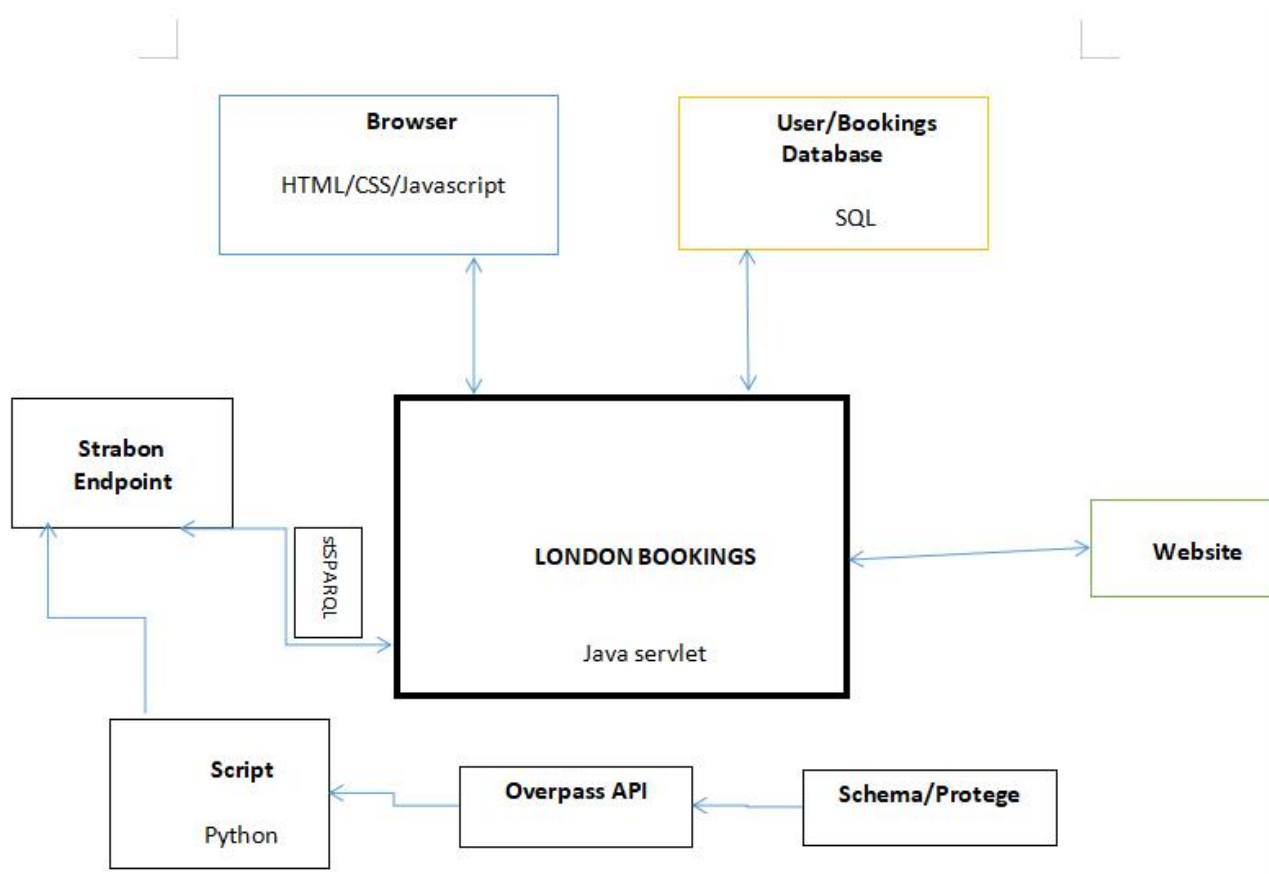
3. Εφαρμογή

3.1 Ρόλος εφαρμογής

Ο στόχος της εφαρμογής είναι να αποτελεί έναν γρήγορο τουριστικό οδηγό του Λονδίνου με επίκεντρο τα ξενοδοχεία. Αποτελεί μια οργανωμένη αναπαράσταση διαφόρων τουριστικών αξιοθέατων σε μία ιστοσελίδα ώστε ένας χρήστης να έχει άμεση πρόσβαση σε οτιδήποτε αναζητεί. Πιο συγκεκριμένα ένας χρήστης μπορεί να αναζητήσει πληροφορίες για ξενοδοχεία, μουσεία, εστιατόρια, πάρκα, κινηματογράφους, καζίνο καθώς και νοσοκομεία σε όλο το Λονδίνο. Επιπλέον, έχει την δυνατότητα με δημιουργία λογαριασμού να πραγματοποιήσει μία κράτηση στο ξενοδοχείο της αρεσκείας του καθώς και να διαχειριστεί τις κρατήσεις του.

3.2 Αρχιτεκτονική εφαρμογής

Μια εικόνα υψηλού επιπέδου για την αρχιτεκτονική της εφαρμογής είναι η ακόλουθη



Εικόνα 2: Αρχιτεκτονική Εφαρμογής

Μέσω του Schema. org πήρα κλάσσεις που με ενδιέφεραν προκειμένου να φτιάξω μία οντολογία που θα περιγράφει την πληροφορία που θέλω να πάρω από τα dataset. Τα δεδομένα τα ανέκτησα από το Overpass API του OpenStreetMap [5] .

Το OpenStreetMap αποτελεί ένα έργο για την δημιουργία ενός χάρτη ολόκληρου του κόσμου και περιέχει γεωγραφικά δεδομένα από όλο τον κόσμο. Το Overpass API αποτελεί μια ιστοσελίδα στην οποία μπορούμε να κατεβάσουμε κάποια από τα δεδομένα που μας ενδιαφέρουν που υπάρχουν στο OpenStreetMap, για παράδειγμα για τις λίμνες στην Πολωνία, σε πολλές διαφορετικές μορφές αρχείων της επιλογής μας.

Στην συνέχεια μέσω ενός script σε python τα έκανα μετατροπή σε RDF χρησιμοποιώντας τα URIs της οντολογίας που έφτιαξα με βάση το Schema. org. Συγκεκριμένα τα μετέτρεψα σε N-Triples[6] για την αποθήκευσή τους.

Στην συνέχεια τα δεδομένα αυτά τα αποθήκευσα χρησιμοποιώντας το Strabon ως SPARQL endpoint. Μέσω του endpoint εκτελώ διάφορα SPARQL queries για να αναπαραστήσω τα δεδομένα μου στον χρήστη.

Το Strabon αποτελεί ένα μέσο για να αποθηκεύονται δεδομένα RDF. Μπορεί να χρησιμοποιηθεί για να αποθηκεύονται συνδεδεμένα γεωχωρικά δεδομένα που αλλάζουν κατά το πέρασμα του χρόνου και να εκτελούνται πολλαπλά ερωτήματα πάνω σε αυτά τα δεδομένα. Επίσης χρησιμοποιείται για να εκτελούνται ερωτήματα και σε δεδομένα που δεν αλλάζουν. Αποτελεί έναν από τους πιο αποδοτικούς τρόπους αποθήκευσης χωροχρονικών RDF που υπάρχουν διαθέσιμη αυτήν τη στιγμή.

Το SPARQL endpoint είναι ένα σημείο στο οποίο εκτελούνται τα SPARQL queries πάνω στην βάση που έχουν αποθηκευτεί τα δεδομένα RDF. Επιστρέφει αποτελέσματα για αυτά τα δεδομένα.

Το περιβάλλον που αναπτύχθηκε η εφαρμογή είναι το Eclipse [7] και σαν server χρησιμοποιήθηκε ο Tomcat[8] .

Το Eclipse αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης για τον προγραμματισμό. Χρησιμοποιείται κυρίως σε εφαρμογές Java αλλά υποστηρίζει πολλές ακόμα γλώσσες προγραμματισμού.

Ο Tomcat αποτελεί ένα περιβάλλον διακομιστή ιστού στο οποίο μπορεί να εκτελεστεί κώδικας γραμμένος σε Java.

Οι προγραμματιστικές γλώσσες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής είναι Java για το back-end σε συνδυασμό με SQL καθώς και SPARQL. Στο front-end χρησιμοποιήθηκαν HTML, CSS, Javascript και κάποιες βιβλιοθήκες της.

Για την αποθήκευση των δεδομένων του χρήστη και για τις πληροφορίες για τις κρατήσεις χρησιμοποιήθηκε η PostgreSQL [9] για να δημιουργηθεί μια βάση δεδομένων. Τα δεδομένα αποθηκεύονται μέσω Java και SQL στην βάση δεδομένων.

Η επικοινωνία μεταξύ της εφαρμογής και των αποθηκευμένων δεδομένων στο endpoint έγινε μέσω ενός protocol [10] . Το πρωτόκολλο αυτό είναι βασισμένο σε HTTP για να εκτελούνται διαδικασίες SPARQL σε δεδομένα μέσω ενός SPARQL endpoint.

Πιο αναλυτικά έγινε χρήση HTTP get request (αίτημα απόκτησης) με τα SPARQL queries σαν παράμετρο και παίρνουμε ως response (ανταπόκριση) τα αποτελέσματα του query. Στην συνέχεια γίνεται ένα parse (ανάλυση) των αποτελεσμάτων μέσω Java στο back-end προκειμένου να παρουσιαστούν τα δεδομένα αυτά στον χρήστη.

3.3 Datasets

Το βασικό στοιχείο της εφαρμογής είναι τα ξενοδοχεία. Για τα ξενοδοχεία οι πληροφορίες που αφορούν τον ενδιαφερόμενο ποικίλουν. Με αυτό το σκεπτικό επέλεξα κάποιες βασικές κατηγορίες πληροφοριών για να δημιουργήσω την βάση δεδομένων μου. Πληροφορίες όπως το όνομα του ξενοδοχείου, η διεύθυνση του και η τοποθεσία του στον χάρτη καθώς και το τηλέφωνο του ξενοδοχείου είναι κάποιες από τις βασικές που περιγράφονται και αναλυτικότερα στην συνέχεια. Όμως για την επιλογή κάποιου ξενοδοχείου για διαμονή τα κριτήρια του ενδιαφερόμενου συνήθως είναι περισσότερο απαιτητικά. Για αυτόν τον λόγο πήρα και πληροφορίες από το OpenStreetMap που αφορούν την δυνατή ώρα άφιξης και αναχώρησης από το ξενοδοχείο καθώς και αν επιτρέπονται κατοικίδια που είναι αρκετά σημαντικό κριτήριο ειδικά για οικογένειες. Τέλος, πήρα και την πληροφορία που αφορά την βαθμολογία του ξενοδοχείου που ουσιαστικά αποτελεί ίσως το σημαντικότερο κριτήριο επιλογής καθώς και ένα μέτρο σύγκρισης.

Τα υπόλοιπα μέρη που περιλαμβάνει η βάση δεδομένων είναι μέρη ψυχαγωγίας που είναι το Καζίνο, τα Πάρκα , οι Κινηματογράφοι και τα Μουσεία καθώς και Εστιατόρια. Τέλος, περιλαμβάνει και τα Νοσοκομεία για περιπτώσεις κάποιου ατυχές

γεγονότος. Για αυτά τα μέρη έχω πάρει παρόμοιες βασικές λεπτομέρειες όπως και στα ξενοδοχεία που περιγράφονται στην συνέχεια.

Για τα δεδομένα που πήρα απο το OpenStreetMap δημιούργησα μία οντολογία μέσω της εφαρμογής Protege [11] και μέσω των URIs του Schema. org.

Αρχικά η ιεραρχία των κλάσεων είναι η ακόλουθη:



Εικόνα 3: Οντολογία

Για τα ξενοδοχεία τα properties που επέλεξα να έχω πληροφορίες για το καθένα με βάση της ονομασίες των URI από το Schema. org είναι τα εξής:

- Address (διεύθυνση)
- CheckinTime (ώρα άφιξης)
- CheckoutTime (ώρα αναχώρησης)
- Description (περιγραφή)
- Email (διεύθυνση ηλεκτρονικού ταχυδρομείου)
- hasGeometry (γεωγραφικές συντεταγμένες)
- Name (όνομα ξενοδοχείου)
- numberOfRooms (πλήθος δωματιών)
- petsAllowed (αν επιτρέπονται κατοικίδια ζώα κατά την παραμονή)
- Telephone (τηλέφωνο)
- Url (σύνδεσμος διαδικτυακού ιστότοπου)

- starRating (αξιολόγηση ξενοδοχείου)

Για τα υπόλοιπα μέρη δηλαδή για Πάρκα, Καζίνο, Κινηματογράφους, Μουσεία, Νοσοκομεία καθώς και Εστιατόρια τα properties τους είναι τα :

- Address
- Description
- Email
- hasGeometry
- Name
- openingHours (ώρα που ανοίγει η κάθε επιχείρηση)
- Telephone
- Url

Επιπλέον για τα εστιατόρια έχω προσθέσει και το property servesCuisine που περιγράφει τον τύπο κουζίνας που σερβίρει το εκάστοτε εστιατόριο.

Μετέτρεψα τα δεδομένα από το OpenStreetMap μέσω ενός python script σε μορφή N-triples για την αποθήκευσή τους στο endpoint. Ένα παράδειγμα ενός αρχείου είναι:

```
@prefix schema: <https://schema.org/> .
@prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#> .

_:100 schema:Place "Restaurant" .
_:100 schema:name "Napa" .
_:100 geo:lat 51.4924822 .
_:100 geo:lon -0.2780734 .
_:100 schema:openingHours 10 .
```

Εικόνα 4: N-Triples

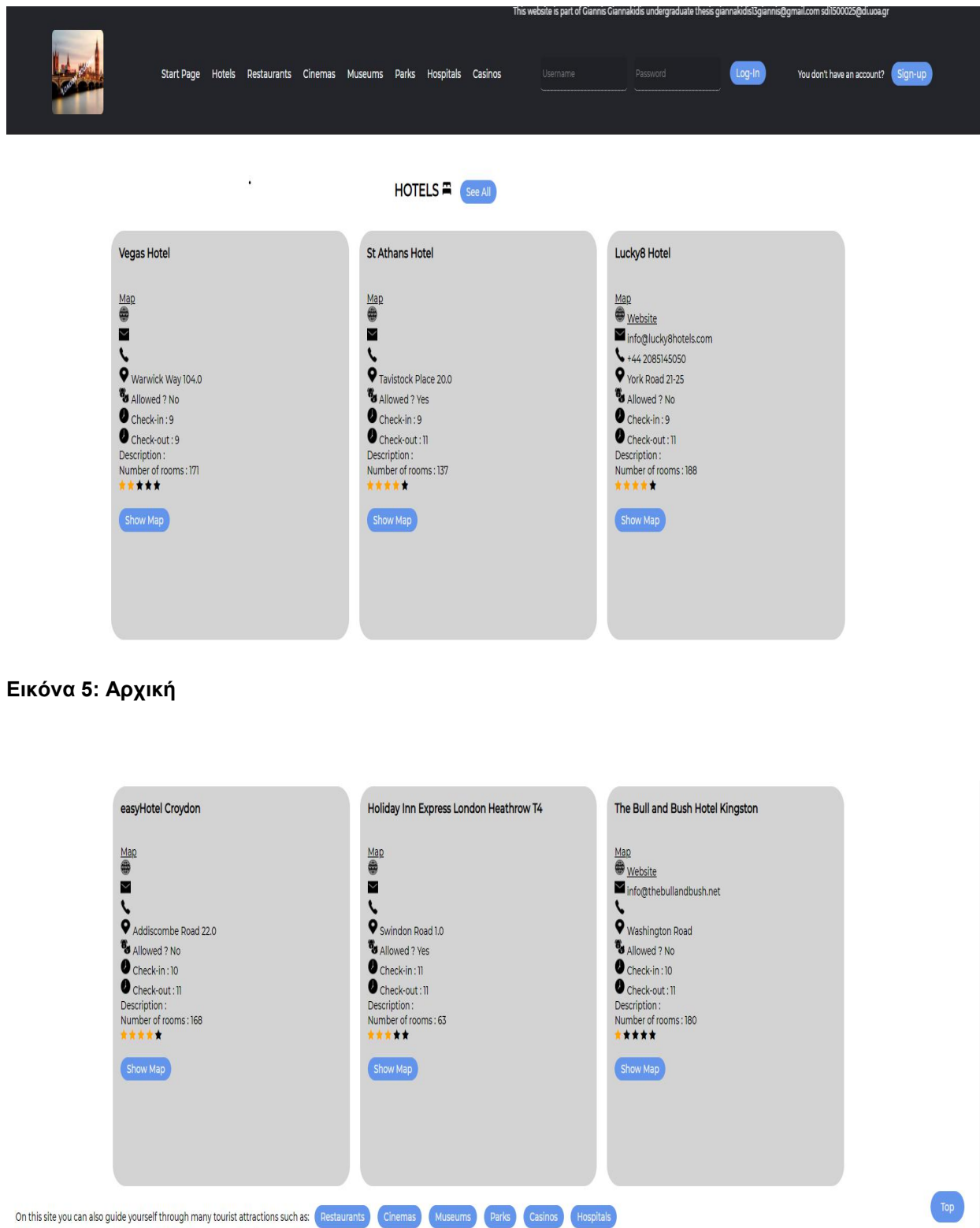
Όπου χρησιμοποιούνται τα prefixes (προθέματα) του Schema. org για τα URIs και του w3 [12] για τις γεωγραφικές συντεταγμένες. Στην φωτογραφία έχουμε ένα μέρος(schema:Place) τύπου εστιατορίου, με όνομα (schema:name) Napa που ανοίγει (schema:openingHours) στις 10. Όπως αναφέρθηκε και σε ένα προηγούμενο σημείο η RDF αναπαράσταση χρησιμοποιεί την μορφή “subject-predicate-object”. Στο

παράδειγμα το `_:100` αποτελεί έναν μοναδικό “κωδικό” για την αναπαράσταση ενός μέρους και αποτελεί το `subject`. Το `schema:Place` το `predicate` και τέλος το “Restaurant” το `object`. Σε αυτά τα δεδομένα εκτελούνται στην συνέχεια τα SPARQL queries για να γίνει ανάκτηση των πληροφοριών που αναζητά ο εκάστοτε χρήστης.

3.4 Features

- Αρχική σελίδα

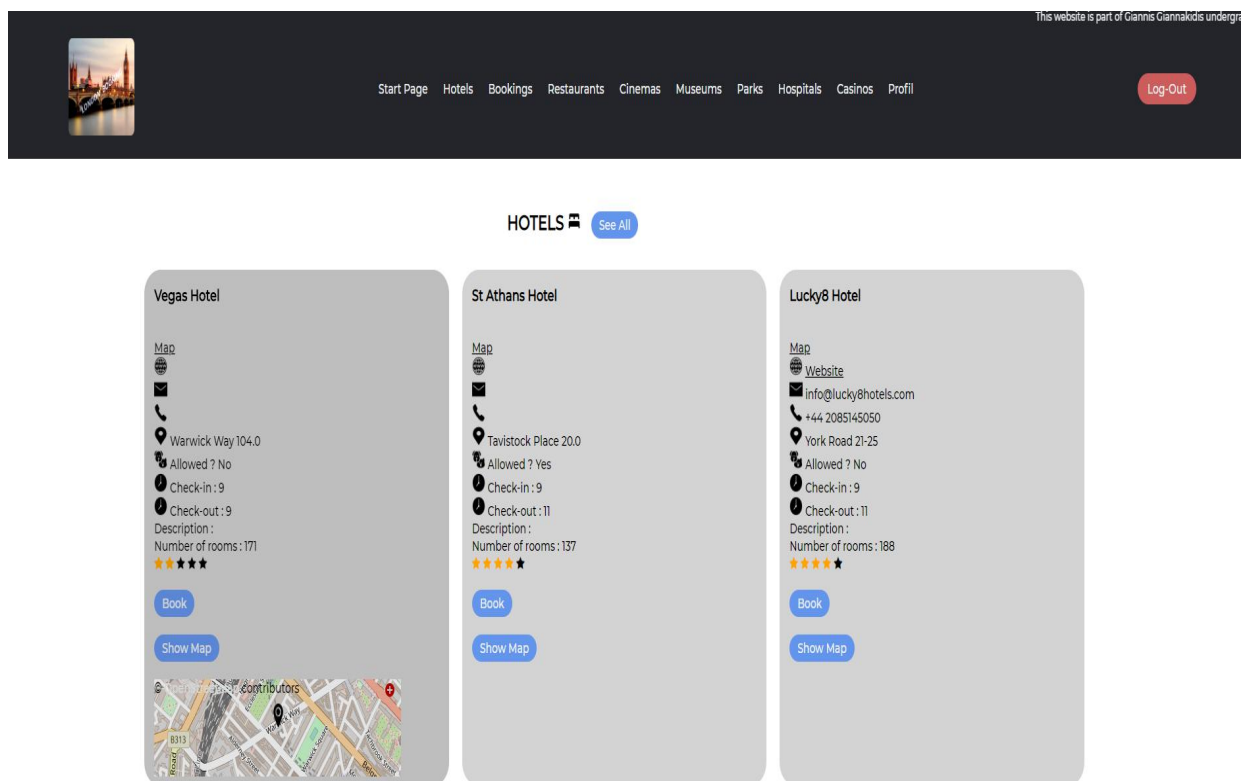
Ο χρήστης όταν εισέρχεται στον ιστοχώρο, εμφανίζεται η αρχική σελίδα. Στην αρχική σελίδα υπάρχει η μπάρα πλοήγησης για όλες τις κατηγορίες που μπορεί ο χρήστης να πλοηγηθεί. Στην μπάρα πλοήγησης υπάρχει επίσης και η φόρμα σύνδεσης του χρήστη με τα στοιχεία του λογαριασμού του καθώς και επιλογή δημιουργίας λογαριασμού. Επιπλέον, εμφανίζονται και ενδεικτικές πληροφορίες για 6 τυχαία επιλεγμένα ξενοδοχεία. Όπως και στις υπόλοιπες ιστοσελίδες έτσι και στην αρχική εμφανίζονται διάφορες πληροφορίες για κάθε μέρος καθώς και επίσης χάρτης μέσω του `openstreetmap` για την τοποθεσία κάθε μέρους αλλά και `link` (σύνδεσμος) για έναν μεγαλύτερο χάρτη.



Εικόνα 5: Αρχική

Εικόνα 6: Αρχική συνέχεια

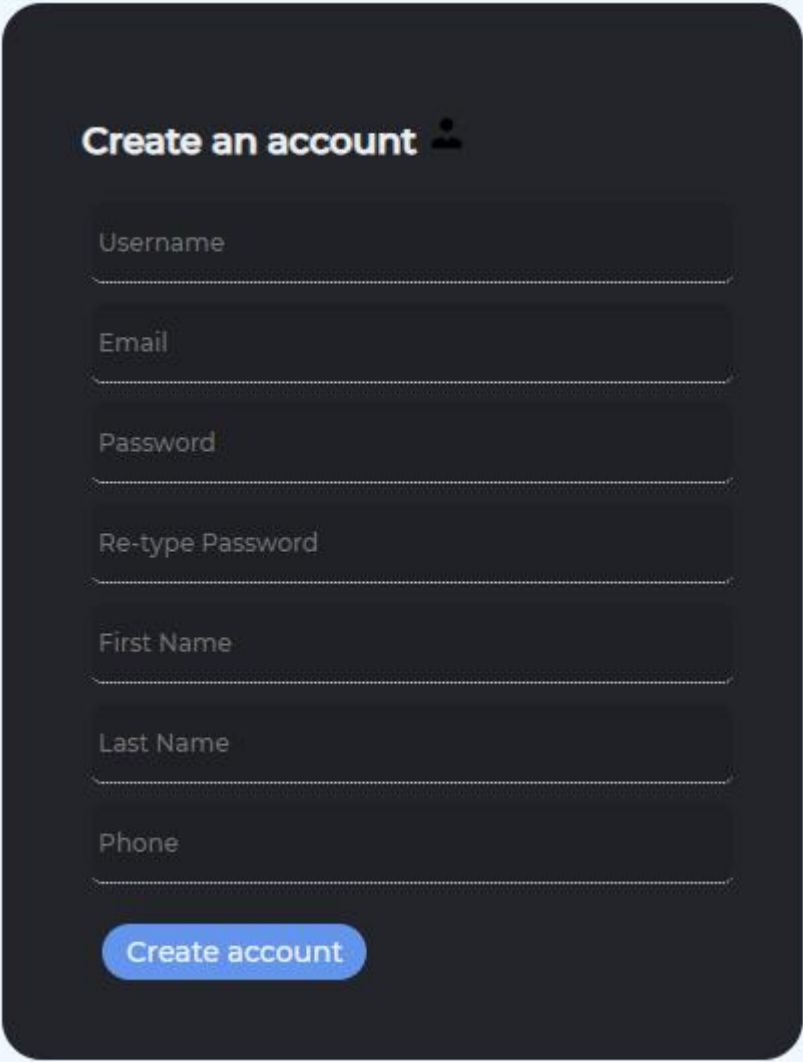
Αν ένας χρήστης συνδεθεί με τα στοιχεία του λογαριασμού του εμφανίζεται η επιλογή αποσύνδεσης καθώς και η επιλογή κράτησης σε κάποιο από τα ξενοδοχεία της επιλογής του.




Εικόνα 7: Αρχική σύνδεση

- Sign-up χρήστη

Σε περίπτωση που ένας χρήστης δεν διαθέτει λογαριασμό, του δίνεται η δυνατότητα να δημιουργήσει έναν. Τα στοιχεία που χρειάζεται να εισάγει είναι ένα username (όνομα χρήστη), ένα email (διεύθυνση ηλεκτρονικού ταχυδρομείου), ένα password (κωδικός) καθώς και να το επαναπληκτρολογήσει για επιβεβαίωση καθώς και το μικρό του όνομα, το επίθετο του και το κινητό του τηλέφωνο.



Create an account 

Username

Email

Password

Re-type Password

First Name

Last Name

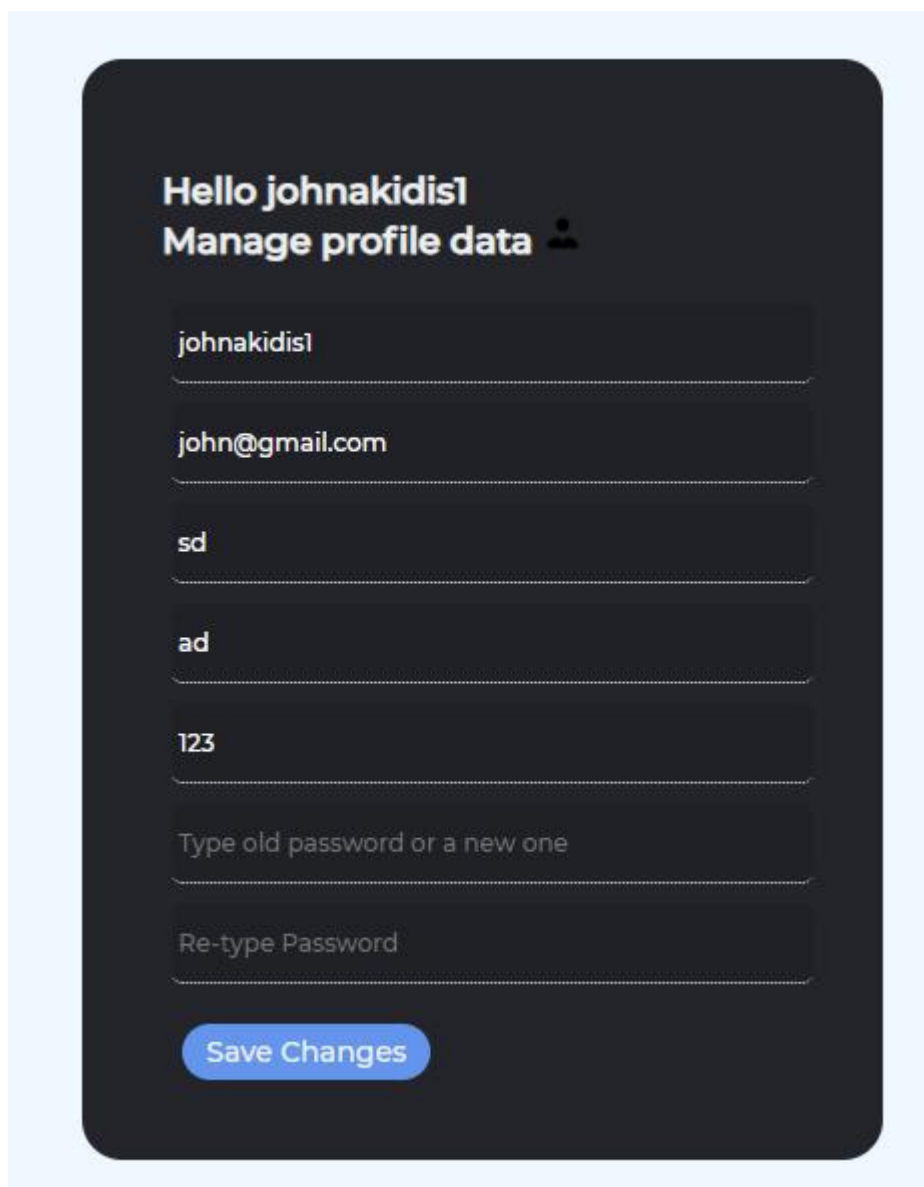
Phone

Create account

Εικόνα 8: Sign-up

- Επεξεργασία προφίλ

Ένας χρήστης, εφόσον έχει συνδεθεί μπορεί να επεξεργαστεί όλα τα στοιχεία του λογαριασμού του.



The image shows a dark-themed user profile management interface. At the top, it says "Hello johnakidis1" and "Manage profile data" with a user icon. Below this are several input fields containing the following text: "johnakidis1", "john@gmail.com", "sd", "ad", "123", "Type old password or a new one", and "Re-type Password". At the bottom is a blue button labeled "Save Changes".

Εικόνα 9: Επεξεργασία προφίλ

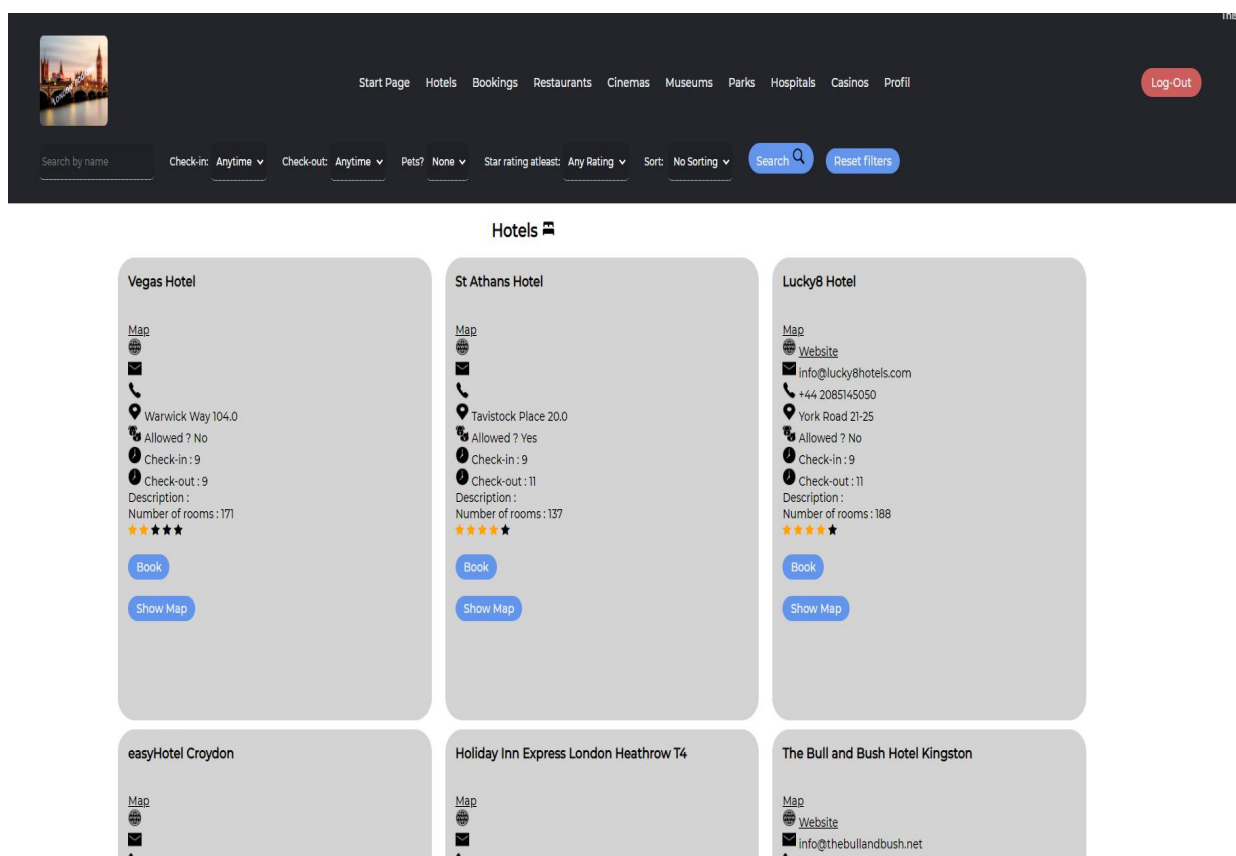
- Ξενοδοχεία

Στην σελίδα των ξενοδοχείων ο χρήστης μπορεί να πλοηγηθεί σε όλα τα ξενοδοχεία του dataset και να δει αναλυτικές πληροφορίες για το καθένα καθώς και έναν χάρτη για το κάθε ένα. Επιπλέον, αν είναι συνδεδεμένος μπορεί να κάνει μία κράτηση σε ένα ξενοδοχείο.

Στην μπάρα πλοήγησης έχει προστεθεί και η επιλογή της αναζήτησης ξενοδοχείου όπου ο χρήστης μπορεί να αναζητήσει όλα τα ξενοδοχεία με βάση το όνομα, την ώρα του check-in (ώρα άφιξης), την ώρα του check-out (ώρα αποχώρησης), το αν επιτρέπονται ή όχι κατοικίδια ζώα, την αξιολόγηση τους καθώς και να ταξινομήσει αλφαβητικά τα αποτελέσματα.

Για τις αναζητήσεις αλλά και την εμφάνιση των αντίστοιχων ιστοσελίδων εκτελούνται τα αντίστοιχα queries που περιγράφηκαν στην ενότητα 5.

Στην εικόνα που ακολουθεί φαίνεται το αρχικό κομμάτι της ιστοσελίδας των ξενοδοχείων, στην συνέχεια της ιστοσελίδας ακολουθούν παρόμοιες πληροφορίες με το ίδιο μοτίβο για τα υπόλοιπα ξενοδοχεία.



Εικόνα 10: Ξενοδοχεία

- Κράτηση

Ένας συνδεδεμένος χρήστης μπορεί να επιλέξει ένα ξενοδοχείο της αρεσκείας του προκειμένου να δημιουργήσει μία κράτηση. Όταν επιλέξει ένα ξενοδοχείο, τότε

μεταφέρεται σε μια καινούργια ιστοσελίδα όπου εμφανίζονται οι πληροφορίες του ξενοδοχείου που επέλεξε. Μαζί με τις πληροφορίες εμφανίζεται και μια φόρμα όπου ο χρήστης συμπληρώνει το πλήθος των ατόμων της κράτησης καθώς και την ημερομηνία άφιξης και ολοκλήρωσης της κράτησης. Τέλος του δίνεται η επιλογή να επιστρέψει στην σελίδα που βρισκόταν πριν.

St Athans Hotel

Map

🕒 Check-in : 9


🕒 Check-out : 11

👤 Allowed ? Yes

Number of rooms : 137

📍 Tavistock Place 20.0

★★★★★



Booking information

Number of people

Starting date

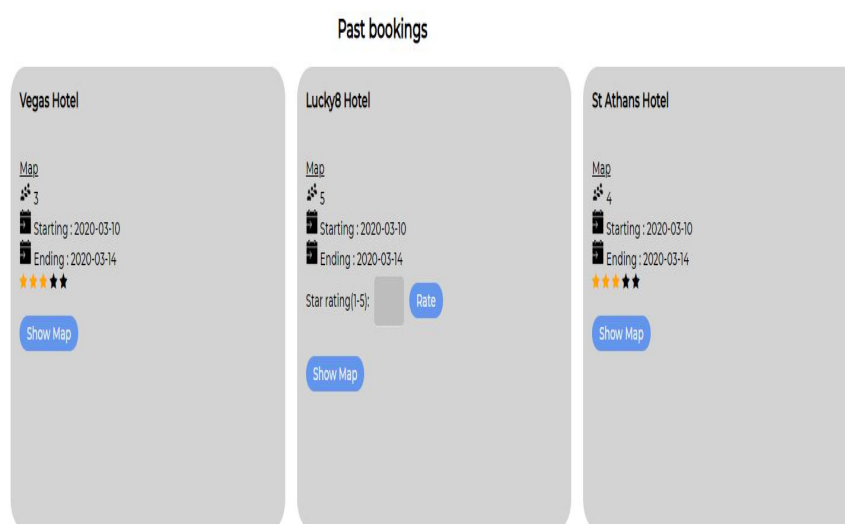
Ending date

[Book](#) [Go Back](#)

Εικόνα 11: Κράτηση

- Προηγούμενες κρατήσεις

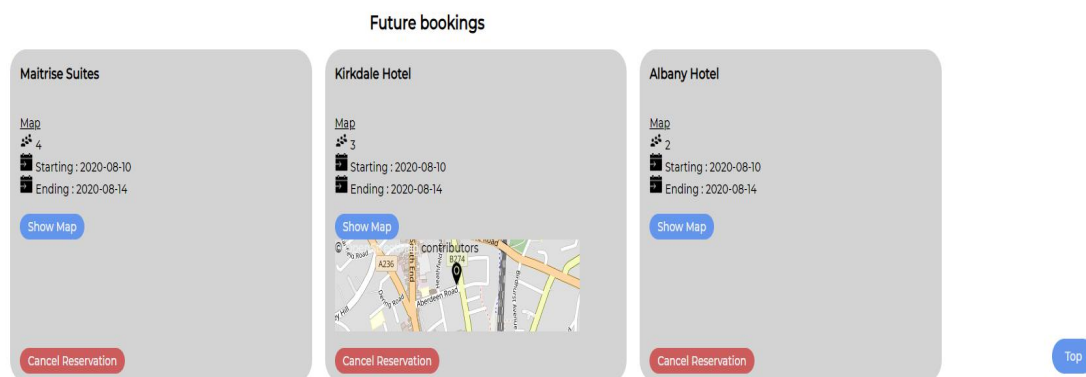
Ο χρήστης εφόσον είναι συνδεδεμένος μπορεί να δει όλες τις κρατήσεις του. Οι κρατήσεις ανάλογα με την ημερομηνία χωρίζονται σε παλιές, τωρινές και μελλοντικές. Στις παλιές κρατήσεις ο χρήστης μπορεί να δει τις πληροφορίες για τις κράτησεις καθώς και να αξιολογήσει την κράτηση του.



Εικόνα 12: Παλιά κράτηση

- Μελλοντικές κρατήσεις

Στις μελλοντικές του κρατήσεις, ο χρήστης μπορεί να δει όλες τις πληροφορίες της κράτησης καθώς και να ακυρώσει κάποια κράτηση της επιλογής του

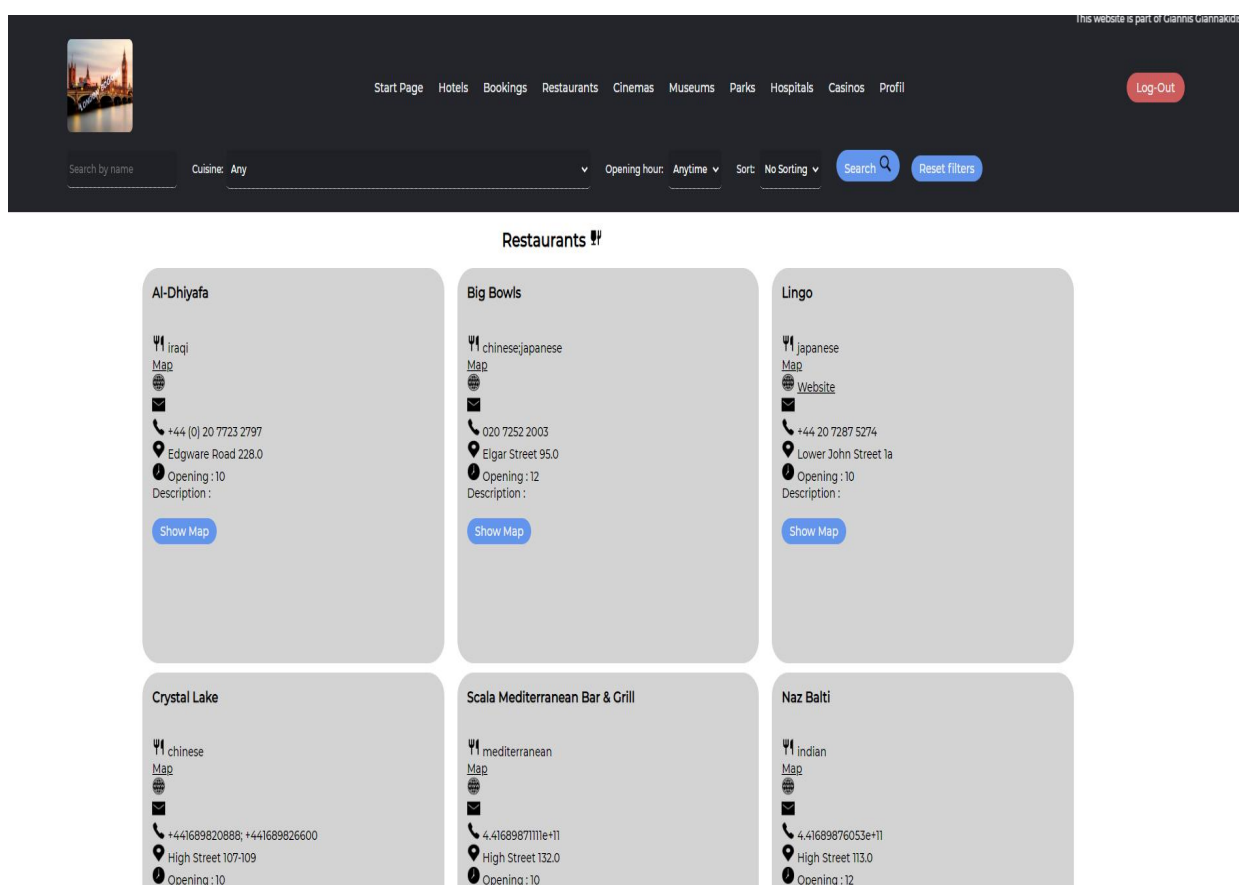


Εικόνα 13: Μελλοντική κράτηση

- Εστιατόρια

Στην σελίδα των εστιατορίων, αντίστοιχα με την σελίδα των ξενοδοχείων ο χρήστης μπορεί να πλοηγηθεί σε όλα τα εστιατόρια του dataset. Μπορεί να προβεί σε αντίστοιχες αναζητήσεις καθώς και να αναζητήσει εστιατόρια με βάση τον τύπο κουζίνας που σερβίρουν.

Και σε αυτήν την ιστοσελίδα χρησιμοποιούνται αντίστοιχα κάποια queries που περιγράφηκαν στην ενότητα 5.



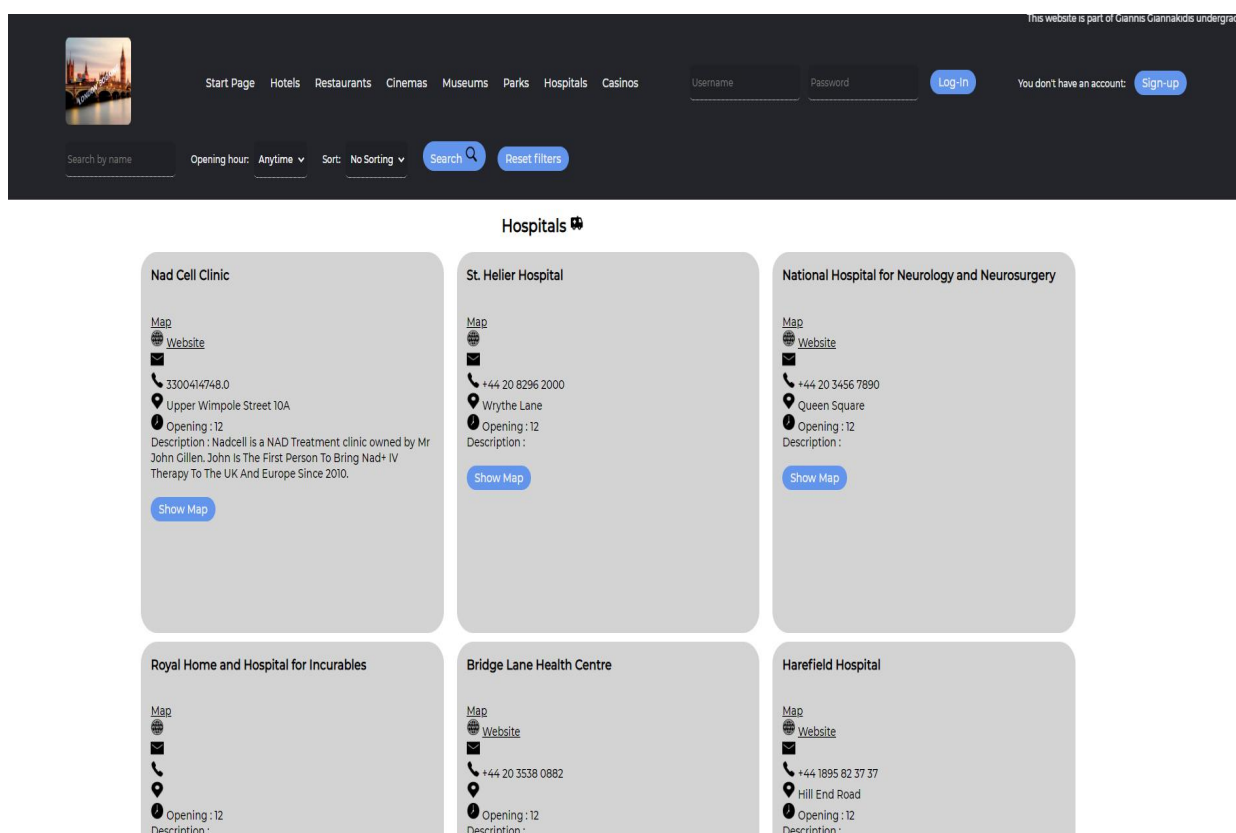
Εικόνα 14: Εστιατόρια

- Υπόλοιπες ιστοσελίδες

Αντίστοιχα με τα εστιατόρια και τα ξενοδοχεία έχουν υλοποιηθεί οι σελίδες για τα καζίνο, τα μουσεία, τα πάρκα, τους κινηματογράφους και τα νοσοκομεία. Ο χρήστης μπορεί να πλοηγηθεί και να δει πληροφορίες για καθένα από αυτά τα μέρη που

υπάρχουν στο dataset και να δει έναν χάρτη με την τοποθεσία του μέρους. Μπορεί να κάνει παρόμοιες αναζητήσεις στις οποίες και εδώ εκτελούνται τα sparql queries που έχουν ήδη περιγραφεί στην αντίστοιχη περίπτωση.

Στην εικόνα που ακολουθεί φαίνεται ένα παράδειγμα της αρχής της ιστοσελίδας των νοσοκομείων, όπου η συνέχεια της είναι πληροφορίες και για τα υπόλοιπα νοσοκομεία. Αντίστοιχη υλοποίηση έχει πραγματοποιηθεί και στις ιστοσελίδες για τα υπόλοιπα μέρη του ιστοχώρου.



Εικόνα 15: Νοσοκομεία

4. ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

Η εφαρμογή έχει πολλούς τομείς που θα μπορούσε να επεκταθεί. Συγκεκριμένα τρεις.

Ο πρώτος είναι με περισσότερα τουριστικές κατηγορίες αξιοθέατων. Για παράδειγμα για γυμναστήρια ή για μέρη βραδινής διασκέδασης

Κατα δεύτερον θα μπορούσε να επεκταθεί γεωγραφικά. Πιο συγκεκριμένα να μην αφορά μόνο το Λονδίνο αλλά και άλλες πολείς ή/και χώρες ώστε να υπάρχουν συγκεντρωμένες πληροφορίες για πολλούς τουριστικούς και μη προορισμούς ανά τον κόσμο.

Τέλος θα μπορούσε να προστεθεί η δυνατότητα να γίνεται κράτηση και σε εστιατόρια ή και στους κινηματογράφους με παρόμοι τρόπο που γίνεται και στα ξενοδοχεία.

5. Παραρτήματα

5.1 SPARQL queries

- Χρήση

Για τις διάφορες ιστοσελίδες του ιστοχώρου ανάλογα με την κάθε μία εκτελούνται πολλά διαφορετικά queries προς το endpoint για την πληροφορία που θέλουμε να αναπαραστήσουμε. Για παράδειγμα θα χρησιμοποιηθεί ένα query για να αναπαραστήσουμε όλα τα εστιατόρια και τις πληροφορίες τους. Θα χρησιμοποιηθεί ένα διαφορετικό query για όλα τα εστιατόρια που έχουν Ιταλική κουζίνα και το όνομα τους ξεκινάει από την συμβολοσειρά "Α".

- Query αρχικής σελίδας

Στην φωτογραφία που ακολουθεί φαίνεται ένα sparql query που εκτελείται στο endpoint προκειμένου να παρουσιαστούν 6 ξενοδοχεία από το dataset, στην αρχική σελίδα της διαδικτυακής εφαρμογής.

```
121 PREFIX schema: <https://schema.org/>
122 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
123 SELECT ?name ?lat ?lon ?url ?email ?telephone ?address ?description ?checkinTime ?checkoutTime ?starRating ?numberOfRooms ?petsAllowed
124 WHERE {
125   ?x schema:Place "Hotel" .
126   ?x schema:name ?name .
127   ?x geo:lat ?lat .
128   ?x geo:lon ?lon .
129   ?x schema:checkinTime ?checkinTime .
130   ?x schema:checkoutTime ?checkoutTime .
131   ?x schema:starRating ?starRating .
132   ?x schema:petsAllowed ?petsAllowed .
133   ?x schema:numberOfRooms ?numberOfRooms .
134   OPTIONAL { ?x schema:url ?url } .
135   OPTIONAL { ?x schema:email ?email } .
136   OPTIONAL { ?x schema:telephone ?telephone } .
137   OPTIONAL { ?x schema:address ?address } .
138   OPTIONAL { ?x schema:description ?description } .
139 }
140 LIMIT 6
```

Εικόνα 16: Query αρχικής σελίδας

- Query ξενοδοχείων

Στην φωτογραφία που ακολουθεί φαίνεται ένα query για την σελίδα που παρουσιάζονται όλα τα ξενοδοχεία από το dataset.

```
304 PREFIX schema: <https://schema.org/>
305 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
306 SELECT ?name ?lat ?lon ?url ?email ?telephone ?address ?description ?checkinTime ?checkoutTime ?starRating ?numberOfRooms ?petsAllowed
307 WHERE {
308   ?x schema:Place "Hotel" .
309   ?x schema:name ?name .
310   ?x geo:lat ?lat .
311   ?x geo:lon ?lon .
312   ?x schema:checkinTime ?checkinTime .
313   ?x schema:checkoutTime ?checkoutTime .
314   ?x schema:starRating ?starRating .
315   ?x schema:petsAllowed ?petsAllowed .
316   ?x schema:numberOfRooms ?numberOfRooms .
317   OPTIONAL { ?x schema:url ?url } .
318   OPTIONAL { ?x schema:email ?email } .
319   OPTIONAL { ?x schema:telephone ?telephone } .
320   OPTIONAL { ?x schema:address ?address } .
321   OPTIONAL { ?x schema:description ?description } .
322 }
```

Εικόνα 17: Σελίδα ξενοδοχείων

- Αλφαβητική ταξινόμηση

Στην φωτογραφία που ακολουθεί φαίνεται το φιλτράρισμα των αποτελεσμάτων για τους κινηματογράφους με βάση την αλφαβητική σειρά του ονόματος τους.

```
207 PREFIX schema: <https://schema.org/>
208 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
209 SELECT ?name ?lat ?lon ?url ?email ?telephone ?address ?description ?openingHours
210 WHERE {
211   ?x schema:Place "Cinema" .
212   ?x schema:name ?name .
213   ?x geo:lat ?lat .
214   ?x geo:lon ?lon .
215   OPTIONAL { ?x schema:url ?url } .
216   OPTIONAL { ?x schema:email ?email } .
217   OPTIONAL { ?x schema:telephone ?telephone } .
218   OPTIONAL { ?x schema:address ?address } .
219   OPTIONAL { ?x schema:description ?description } .
220   OPTIONAL { ?x schema:openingHours ?openingHours } .
221 } ORDER BY ASC(?name)
222
```

Εκόνα 18: Ταξινόμηση

- Αναζήτηση με βάση το όνομα

Στην φωτογραφία που ακολουθεί φαίνεται ένα query για την αναζήτηση όλων των κινηματογράφων με βάση μια συμβολοσειρά που έχει δώσει ο χρήστης.


```
174 PREFIX schema: <https://schema.org/>
175 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
176 SELECT ?name ?lat ?lon ?url ?email ?telephone ?address ?description ?openingHours
177 WHERE {
178   ?x schema:Place "Cinema" .
179   ?x schema:name ?name .
180   ?x geo:lat ?lat .
181   ?x geo:lon ?lon .
182   OPTIONAL { ?x schema:url ?url } .
183   OPTIONAL { ?x schema:email ?email } .
184   OPTIONAL { ?x schema:telephone ?telephone } .
185   OPTIONAL { ?x schema:address ?address } .
186   OPTIONAL { ?x schema:description ?description } .
187   OPTIONAL { ?x schema:openingHours ?openingHours } .
188   FILTER regex(str(?name),".*name_var.*","i") .
189 }
```

Εικόνα 19: Αναζήτηση με όνομα

- Αναζήτηση με βάση την κατηγορία φαγητού

Στην εικόνα που ακολουθεί φαίνεται η εμφάνιση όλων των ξενοδοχείων του endpoint με βάση την κατηγορία της κουζίνας που διαθέτουν.

```
290 PREFIX schema: <https://schema.org/>
291 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
292 SELECT ?name ?servesCuisine ?lat ?lon ?url ?email ?telephone ?address ?description ?openingHours
293 WHERE {
294   ?x schema:Place "Restaurant" .
295   ?x schema:name ?name .
296   ?x geo:lat ?lat .
297   ?x geo:lon ?lon .
298   ?x schema:servesCuisine ?servesCuisine .
299   OPTIONAL { ?x schema:url ?url } .
300   OPTIONAL { ?x schema:email ?email } .
301   OPTIONAL { ?x schema:telephone ?telephone } .
302   OPTIONAL { ?x schema:address ?address } .
303   OPTIONAL { ?x schema:description ?description } .
304   OPTIONAL { ?x schema:openingHours ?openingHours } .
305   FILTER regex(str(?servesCuisine),".*category_var.*","i")
306 }
```

Εικόνα 20: Αναζήτηση κουζίνας

- Αναζήτηση όλων των τύπων κουζίνας

Στην εικόνα που ακολουθεί φαίνεται η αναζήτηση όλων των διαθέσιμων τύπων κουζίνας στα εστιατόρια που διαθέτουμε στο dataset.

```
15 PREFIX schema: <https://schema.org/>
16 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
17 SELECT DISTINCT ?servesCuisine
18 WHERE{
19   ?x schema:servesCuisine ?servesCuisine.
20 }
```

Εικόνα 21: Τύποι κουζίνας

- Ώρα ανοίγματος

Στην εικόνα που ακολουθεί φαίνεται η αναζήτηση των κινηματογράφων με βάση την ώρα που ανοίγουν.

```
191 PREFIX schema: <https://schema.org/>
192 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
193 SELECT ?name ?lat ?lon ?url ?email ?telephone ?address ?description ?openingHours
194 WHERE {
195   ?x schema:Place "Cinema" .
196   ?x schema:name ?name .
197   ?x geo:lat ?lat .
198   ?x geo:lon ?lon .
199   OPTIONAL { ?x schema:url ?url } .
200   OPTIONAL { ?x schema:email ?email } .
201   OPTIONAL { ?x schema:telephone ?telephone } .
202   OPTIONAL { ?x schema:address ?address } .
203   OPTIONAL { ?x schema:description ?description } .
204   OPTIONAL { ?x schema:openingHours ?openingHours } .
205   FILTER ( ?openingHours = hour_var ) .
206 }
```

Εικόνα 22: Ώρα opening

- Κατοικίδια

Στην φωτογραφία που ακολουθεί, επιστρέφονται όλα τα ξενοδοχεία ανάλογα με το κριτήριο του αν δέχονται ή όχι κατοικίδια ζώα.

```
424 PREFIX schema: <https://schema.org/>
425 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
426 SELECT ?name ?lat ?lon ?url ?email ?telephone ?address ?description ?checkinTime ?checkoutTime ?starRating ?numberOfRooms ?petsAllowed
427 WHERE {
428   ?x schema:Place "Hotel" .
429   ?x schema:name ?name .
430   ?x geo:lat ?lat .
431   ?x geo:lon ?lon .
432   ?x schema:checkinTime ?checkinTime .
433   ?x schema:checkoutTime ?checkoutTime .
434   ?x schema:starRating ?starRating .
435   ?x schema:petsAllowed ?petsAllowed .
436   ?x schema:numberOfRooms ?numberOfRooms .
437   OPTIONAL { ?x schema:url ?url } .
438   OPTIONAL { ?x schema:email ?email } .
439   OPTIONAL { ?x schema:telephone ?telephone } .
440   OPTIONAL { ?x schema:address ?address } .
441   OPTIONAL { ?x schema:description ?description } .
442   FILTER regex(str(?petsAllowed),pets_var) .
443 }
```

Εικόνα 23: Κατοικίδια

- Αξιολόγηση ξενοδοχείων

Στην εικόνα που ακολουθεί, επιστρέφονται όλα τα ξενοδοχεία φιλτραρισμένα με βάση την αξιολόγηση τους.

```
448 PREFIX schema: <https://schema.org/>
449 PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
450 SELECT ?name ?lat ?lon ?url ?email ?telephone ?address ?description ?checkinTime ?checkoutTime ?starRating ?numberOfRooms ?petsAllowed
451 WHERE {
452   ?x schema:Place "Hotel" .
453   ?x schema:name ?name .
454   ?x geo:lat ?lat .
455   ?x geo:lon ?lon .
456   ?x schema:checkinTime ?checkinTime .
457   ?x schema:checkoutTime ?checkoutTime .
458   ?x schema:starRating ?starRating .
459   ?x schema:petsAllowed ?petsAllowed .
460   ?x schema:numberOfRooms ?numberOfRooms .
461   OPTIONAL { ?x schema:url ?url } .
462   OPTIONAL { ?x schema:email ?email } .
463   OPTIONAL { ?x schema:telephone ?telephone } .
464   OPTIONAL { ?x schema:address ?address } .
465   OPTIONAL { ?x schema:description ?description } .
466   FILTER ( ?starRating >= _star_var ) .
467 }
```

Εικόνα 24: Αξιολόγηση

5.2 Κώδικας

- Αρχική σελίδα καζίνο

Παράδειγμα HTTP Get request για όλα τα καζίνο για την αναπαράσταση τους.

```
22 @Override
23 protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
24     try {
25         String qr="PREFIX schema: <https://schema.org/> \r\n" +
26             "PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>\r\n" +
27             "\r\n" +
28             "SELECT ?name (SAMPLE(?lat) as ?lat) (SAMPLE(?lon) as ?lon) (SAMPLE(?url) as ?url) (SAMPLE(?email) as ?email) (SAMPLE(?telephone) as ?telephone) (SAMPLE(?address) as ?address) (SAMPLE(?description) as ?description) (SAMPLE(?openingHours) as ?openingHours) \r\n" +
29             "WHERE {\r\n" +
30             "  ?x schema:Place \"Casino\" .\r\n" +
31             "  ?x schema:name ?name .\r\n" +
32             "  ?x geo:lat ?lat .\r\n" +
33             "  ?x geo:lon ?lon .\r\n" +
34             "  OPTIONAL { ?x schema:url ?url } .\r\n" +
35             "  OPTIONAL { ?x schema:email ?email } .\r\n" +
36             "  OPTIONAL { ?x schema:telephone ?telephone } .\r\n" +
37             "  OPTIONAL { ?x schema:address ?address } .\r\n" +
38             "  OPTIONAL { ?x schema:description ?description } .\r\n" +
39             "  OPTIONAL { ?x schema:openingHours ?openingHours } .\r\n" +
40             "} GROUP BY ?name\r\n";
41         URL url=new URL("http://localhost:8090/strabon-endpoint-3.3.2-SNAPSHOT/Query?query="+URLEncoder.encode(qr,StandardCharsets.UTF_8.toString()));
42         URLConnection con=url.openConnection();
43         con.setConnectTimeout(5000);
44         con.setReadTimeout(5000);
```

Εικόνα 25: HTTP request

- Παράδειγμα εισαγωγής στοιχείων χρήστη

Στο παράδειγμα που ακολουθεί εισάγονται τα στοιχεία ενός καινούργιου χρήστη στην βάση δεδομένων καθώς και γίνεται κρυπτογράφηση του password για την προστασία των δεδομένων του

```
java.sql.Connection conn=DriverManager.getConnection("jdbc:postgresql://localhost:5432/postgres","postgres","admin");
Statement statement=conn.createStatement();
ResultSet rs=statement.executeQuery("SELECT * FROM users WHERE username='"+uname+"'");
if(rs.next()) {
    request.getSession().setAttribute("error", "used");
    conn.close();
    response.sendRedirect("signup.jsp");
}
else {
    if(!password.equals(password2)) {
        request.getSession().setAttribute("error", "pmatch");
        conn.close();
        response.sendRedirect("signup.jsp");
    }
    else {
        //hash
        MessageDigest md=MessageDigest.getInstance("MD5");
        byte[] messageDigest=md.digest(password.getBytes());
        BigInteger no=new BigInteger(1,messageDigest);
        String newpas=no.toString(16);
        while(newpas.length()<32) {
            newpas="0"+newpas;
        }
        statement.executeUpdate("INSERT INTO users (username,password,email,name,surname,phone) VALUES ('"+uname+"','"+newpas+"','"+email+"','"+name+"','"+lastname+"','"+phone+"')");
        conn.close();
        session.setAttribute("user", uname);
    }
}
```

Εικόνα 26: SQL query

6. ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός όρος
Dataset	Σύνολο δεδομένων
Classes	Κλάσεις
Properties	Ιδιότητες
Queries	Ερωτήματα
Back-end	Οπίσθιο μέρος
Front-end	Μπροστινό μέρος
Get request	Αίτημα απόκτησης
Response	Ανταπόκριση
Endpoint	Τελικό σημείο
Link	Σύνδεσμος
Username	Όνομα χρήστη
Password	Κωδικός
E-mail	Διεύθυνση ηλεκτρονικού ταχυδρομείου
Check-in	Ωρα άφιξης
Check-out	Ωρα αποχώρησης
Parse	Ανάλυση
Prefixes	Προθέματα
Linked data	Συνδεδεμένα δεδομένα
Semantic queries	Σημασιολογικά ερωτήματα

Movie	Ταινία
Article	Άρθρο
Recipe	Συνταγή
LocalBusiness	Τοπική επιχείρηση
Microdata	Μικροδεδομένα

7. ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

SPARQL	SPARQL Protocol and RDF Query Language
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
RDF	Resource Description Framework
API	Application Programming Interface
URI	Uniform Resource Identifier
HTTP	Hypertext Transfer Protocol
SQL	Structured Query Language
OGC	Open Geospatial Consortium

8. ΑΝΑΦΟΡΕΣ

[1] Giannis N. Giannakidis “London Bookings”

<https://github.com/johnakidis/London-Bookings> [Προσπελάστηκε 20/06/2020]

[2] Strabon. The most powerful spatiotemporal RDF Store!

<http://www.strabon.di.uoa.gr/> {Προσπελάστηκε 10/05/2020}

[3] RDF is a standard model for data interchange on the Web

<https://www.w3.org/RDF/> [Προσπελάστηκε 15/02/2020]

[4] Schema is a collaborative activity for structured data on the internet

<https://schema.org/> [Προσπελάστηκε 20/12/2019]

[5] OpenStreetMap is a map of the world

<https://www.openstreetmap.org/> [Προσπελάστηκε 10/12/2019]

[6] N-Triples is a line-based, plain text format for encoding an RDF graph.

<https://www.w3.org/TR/n-triples/> [Προσπελάστηκε 25/02/2020]

[7] <https://www.eclipse.org/> [Προσπελάστηκε 10/05/2020]

[8] <http://tomcat.apache.org/> [Προσπελάστηκε 10/05/2020]

[9] <https://www.postgresql.org/> [Προσπελάστηκε 10/05/2020]

[10] <https://www.w3.org/TR/sparql11-protocol/> [Προσπελάστηκε 20/05/2020]

[11] Protege a free open-source ontology editor

<https://protege.stanford.edu/> [Προσπελάστηκε 05/02/2020]