NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATION

BSc THESIS

Building an accommodation rental application using Linked Geospatial Data

Anastasios G. Pantazopoulos

Supervisors: **Manolis Koumparakis**, Professor UoA
George Stamoulis, Ph.D. Candidate UoA

ATHENS

December 2022

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη εφαρμογής για ενοικίαση καταλυμάτων με τη χρήση Συνδεδεμένων Γεωχωρικών Δεδομένων (Linked Geospatial Data)

Αναστάσιος Γ. Πανταζόπουλος

Επιβλέποντες: **Μανώλης Κουμπαράκης**, Καθηγητής Ε.Κ.Π.Α.
Γεώργιος Σταμούλης, Υποψήφιος για Ph.D. Ε.Κ.Π.Α.

ΑΘΗΝΑ

Δεκέμβριος 2022

BSc THESIS

Building an accommodation rental application using Linked Geospatial Data

Anastasios G. Pantazopoulos
S.N.: 1115201500119

SUPERVISOR: **Manolis Koumparakis**, Professor UoA
George Stamoulis, Ph.D. Candidate UoA

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη εφαρμογής για ενοικίαση καταλυμάτων με τη χρήση Συνδεδεμένων
Γεωχωρικών Δεδομένων (Linked Geospatial Data)

Αναστάσιος Γ. Πανταζόπουλος
Α.Μ.: 1115201500119

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Μανώλης Κουμπαράκης**, Καθηγητής Ε.Κ.Π,Α
Γεώργιος Σταμούλης, Υποψήφιος για Ph.D. Ε.Κ.Π.Α.

ABSTRACT

The emergence of Airbnb is unquestionably one of the most significant and transformative recent developments within the worldwide tourism sector during the past decade. Nowadays, people from all over the world use travel and tourism-related platforms to search for accommodation, whereas they are highly interested in the points of interest (POIs) located nearby. In this thesis, a tourist application for searching for accommodation is presented, based on, among other criteria, their proximity to POIs of a specific type (e.g., Culture). The application uses geographic data (geospatial data), drawn from various open sources of Linked Data.

SUBJECT AREA: Linked geospatial data, Web development

KEYWORDS: Linked Data, Geospatial data, SPARQL, RDF, Point of Interest (POI), React, NodeJS

ΠΕΡΙΛΗΨΗ

Η εμφάνιση της εφαρμογής Airbnb αποτέλεσε αναμφίλεκτα μια από τις πιο σημαντικές και επιδραστικές αλλαγές στον παγκόσμιο τουριστικό κλάδο την περασμένη δεκαετία. Πλέον μέσω διάφορων τουριστικών πλατφορμών άνθρωποι από όλο τον κόσμο αναζητούν χώρους διαμονής, θέτοντας πολύ συχνά ως κριτήριο τα σημεία ενδιαφέροντος (POIs) που βρίσκονται κοντά σε αυτά. Στην παρούσα πτυχιακή εργασία παρουσιάζεται μια τουριστική εφαρμογή αναζήτησης καταλυμάτων, με γνώμονα μεταξύ άλλων την εγγύτητα τους σε σημεία ενδιαφέροντος συγκεκριμένου τύπου (πχ Πολιτισμός). Η εφαρμογή χρησιμοποιεί γεωγραφικά δεδομένα (geospatial data), που αντλήθηκαν από διάφορες ανοικτές πηγές συνδεδεμένων δεδομένων (Linked Data).

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Συνδεδεμένα γεωχωρικά δεδομένα, Κατασκευή και ανάπτυξη ιστοσελίδων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Συνδεδεμένα Δεδομένα, Γεωχωρικά δεδομένα, SPARQL, RDF, Σημείο ενδιαφέροντος (POI), React, NodeJS

AKNOWLEDGMENTS

CONTENTS

1.

LIST OF FIGURES

LIST OF TABLES

PREFACE

This thesis took place in Athens, Greece by undergraduate student Anastasios Pantazopoulos under the supervision of professor Manolis Koumparakis and PhD candidate Georgios Stamoulis. A desktop application was developed on a Linux machine using *Visual Studio Code* IDE. It is a *React* App with a *NodeJS* Backend, and it uses *Strabon* as a data store.

1. INTRODUCTION

As the world moves further into the digital epoch, online applications have gained an important role in how people travel and vacation. Thanks to them, billions of people from all over the world share their opinions about different places, accommodation and services. Tripadvisor for example had over 1 billion reviews in 2021[1].

The increased use of online media has brought about drastic changes in the way prospective clients choose their accommodation. The emergence of Airbnb is unquestionably one of the most crucial and transformative recent developments within the worldwide tourism sector during the past decade. Nowadays, people from all over the world use travel and tourism-related platforms to search for accommodation, whereas they are also highly interested in the points of interest (POIs) located nearby. POIs are digital proxies of real-world places, represented as geometric point entities, followed by a plethora of metadata (e.g., reviews, ratings, working hours) and usually divided in categories based on their nature (e.g., restaurants, stadiums, trains stations). POI categories play a prominent role in the service provision of all the above-mentioned online applications. Users often perform category-based search queries, such as "what is the best restaurant in this neighborhood?" or "what is the closest hotel to the train station?"[2][3].

This thesis was carried out taking into account the developments and needs mentioned above and consists of an application that enables users to search for accommodation and at the same time explore areas based on the possibilities they offer. In addition to the basic and necessary information about a room/house, the application displays many points of interest around, informing them about what it offers, the opening hours, the category of people it is aimed at, etc. Many users may want to find a house near a specific POI, like a famous monument, or a parking lot (especially if we are talking about densely populated areas without parking spaces), so the application gives the possibility to search near it.

2. BACKGROUND AND RELATED WORK

2.1. Linked Data

2.1.1. RDF, URI, SPARQL

The World Wide Web is based on the idea of creating hyperlinks between Web documents. These hyperlinks integrate all these web documents into a single global information space. Today, major Web data sources such as search engines (Google, Yahoo) and Web native companies (eBay and Amazon) provide access to their databases through Web APIs. Smaller organizations or even individuals are gradually doing the same, contributing to a vast collection of data. Across the web over 24,700 APIs have already been listed according to ProgrammableWeb[4].

The intelligent utilization of this enormous amount data has a major impact on our lives. Google and Yahoo for example consume structured data from websites to improve our search results. The famous online retailer Amazon uses an API to make their products available to third parties, and so they have established a referral program that pays a commission to referring websites. What is more, the appearance of Web APIs has led to a significant increase in small specialized applications (or mashups) that combine data from several sources.

However, in contrast to the classic Web, which is organized on a specific set of standards - Uniform Resource Identifiers (URIs), the Hypertext Transfer Protocol (HTTP), and the Hypertext Markup Language (HTML) - different Web APIs use different mechanisms to identify, access and represent retrieved data. Because most Web APIs do not assign globally unique identifiers to data items, it is not always possible to set hyperlinks between data items provided by different APIs. As a result, Web APIs divide the Web into separate data groups, and mashup developers are forced to choose a specific set of data sources for their application. In other words, they can't take advantage of all the data available on the Web[5][6].

The term Linked Data refers to a set of best practices for publishing and interlinking structured data on the Web. This set, aimed at solving the aforementioned issue, was outlined by Tim Berners-Lee in his Web architecture note "Linked Data" and has become known as the "Linked Data principles":

  i. Use URIs as names for things.

  ii. Use HTTP URIs, so that people can look up those names.

  iii. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)

  iv. Include links to other URIs, so that they can discover more things.

RDF

The need to adopt a single data model for representing structured data led to the *RDF* (Resource Description Framework) model, a graph-based model through which entity descriptions are constructed.

Descriptions are represented as a number of triples. The three parts of each triple are called subject, predicate and object. A triplet imitates the parts of a simple sentence:

Giannis   has Brother   Thanasis

Giannis   is   basketball player

Giannis   national team   Greece

Subject   Predicate   Object

URI

A Uniform Resource Identifier (URI) is a unique sequence of characters that identifies things. Linked data use HTTP URIs not only to identify Web documents (as the classic Web does), but also to identify real-world objects and abstract concepts.

SPARQL

SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware[7]. Most forms of SPARQL query contain a set of triple patterns, which are like RDF triples except that each of the subject, predicate and object may be a variable. Example:

Data:

<http://example.org/player/Giannis> <http://fiba.com/national_team> "Greece" .

SPARQL Query:

```
SELECT ?team

WHERE

{

  <http://example.org/player/Giannis> <http://fiba.com/national_team> ?team .

}
```

Query Result:

```
  Team
"Greece"
```

2.1.2. Linked Data Tools

Geotriples

GeoTriples is an open-source tool that generates and processes extended R2RML and RML mappings that transform geospatial data from many input formats into RDF. GeoTriples allows the transformation of data stored in raw files (shapefiles, CSV, KML, XML, GML and GeoJSON) and spatially-enabled RDBMS (Relational Database

Management Systems) (PostGIS and MonetDB) into RDF graphs using well-known vocabularies like GeoSPARQL and stSPARQL, but without being tightly coupled to a specific vocabulary[8].

Strabon

Strabon is a geospatial DBMS (Database Management System) for storing and querying geospatial data that changes over time. It implements the data model stRDF, the query language stSPARQL and the respective part of the OGC (Open Geospatial Consortium) standard GeoSPARQL. The query language stSPARQL is an extension of the query language SPARQL that supports spatiotemporal data[9].

2.2. Web development

React (also known as React.js or ReactJS), is a free front-end JavaScript library developed by Facebook. It can be used as a base in the development of single-page, mobile, or server-rendered applications with frameworks. Today React is broadly adopted across the developer communities for building user interfaces, mostly because of its simplicity and interactivity[10].

React is a component-based tool, which means that a React application consists of encapsulated components, each of which manages its own state. The general idea of React components is similar to that of functions. Functions in programming are usually called by taking some input parameters and after executing they produce an output. Components work in a similar way: props (properties) get passed to the component (similar to function parameters) whereas state is managed within the component (similar to variables declared within a function). The output is a description of a user interface. Components can be reused or contain other components, just like functions[11].

Node.js (also known as NodeJS) is a free and open source server environment that runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.).

Although it is a relatively new environment, it has become quite popular due to some of its innovative features. It can process multiple functions and events in parallel, in other words it uses asynchronous programming, unlike PHP and ASP. Also, NodeJS uses JavaScript on the server, so because a single language is used throughout the whole application, the development process is much easier[12].

2.3. Open knowledge bases / data sources

The **DBpedia** project, initiated in 2007, is focused on offering structured content from the information created in Wikipedia. Through DBpedia users are allowed to semantically query relationships and properties of Wikipedia resources, including links to other related datasets. Data is accessed using SPARQL.

For this project, OpenLink Virtuoso SPARQL Query Editor, a public SPARQL endpoint (http://dbpedia.org/sparql) was used in order to download data from the DBpedia dataset[13].

OpenStreetMap (OSM) is a well-known source of openly available volunteered geographic information. With its database including almost 7.5 billion data points (nodes), contributed by approximately 1.8 million users as of March 2022, it is perhaps the most accomplished example of a crowdsourced geoinformation project and of the concept of volunteered geographic information[14].

The **Overpass API** is a read-only API that serves up custom selected parts of the OSM map data. **Overpass turbo** (overpass-turbo.eu) is an open-source web based data mining tool for OpenStreetMap. With it, users can run queries in Overpass API in order to retrieve and download data. They can also see the results on an interactive map. Overpass turbo was used in this project to compose a dataset with thousands of OSM POIs.
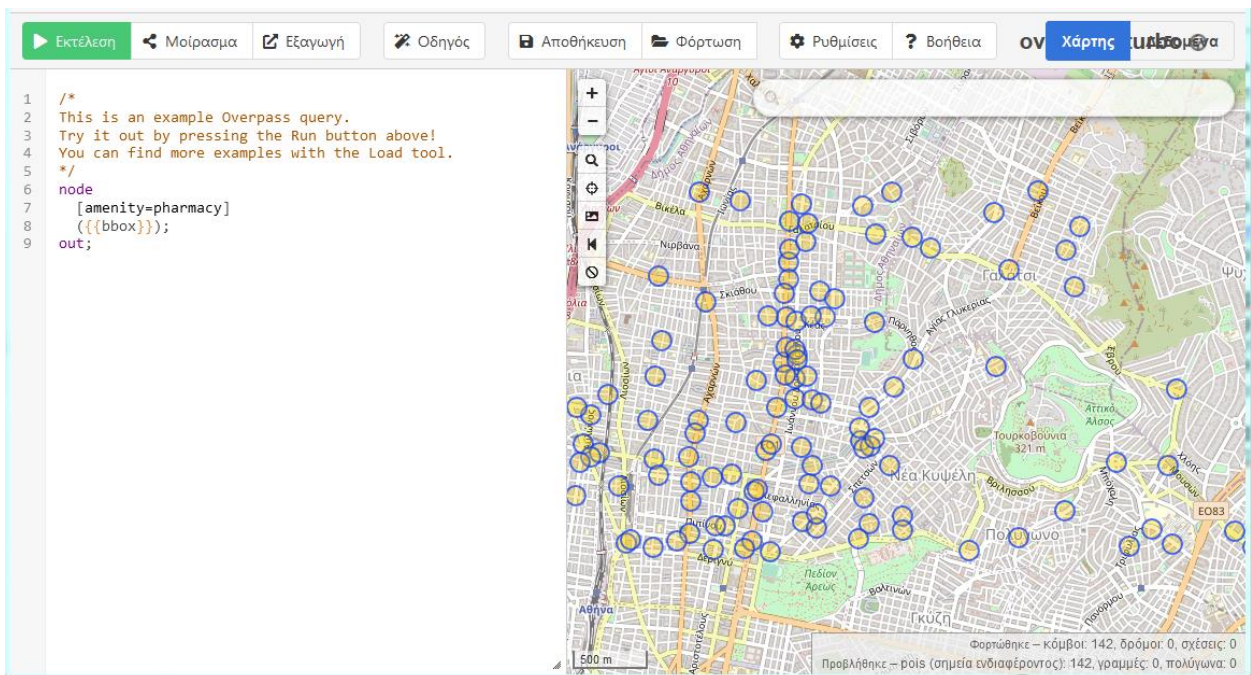

Figure 1: Overpass Turbo, a friendly interface to the Overpass API

Inside Airbnb is a project that provides and visualizes scraped data on the property rental marketplace company Airbnb. The application uses Inside Airbnb datasets that contain data related to accommodations.

3. APPLICATION

3.1. Role of the App

The name of the application is "bnt" ("Book and Travel"). The application is aimed at travelers who search for accommodation of any type (rooms, apartments, etc.) to rent with short-term lease.

The application provides thousands of accommodation, with detailed information such as photo, host information, amenities, maximum people who can be accommodated, beds, etc. It is possible to search lodgings by area and other criteria.

The purpose of the application and characteristic that differentiates it from other similar applications is to focus on the POIs in the accommodation area. Users can identify and see on a map POIs by category.
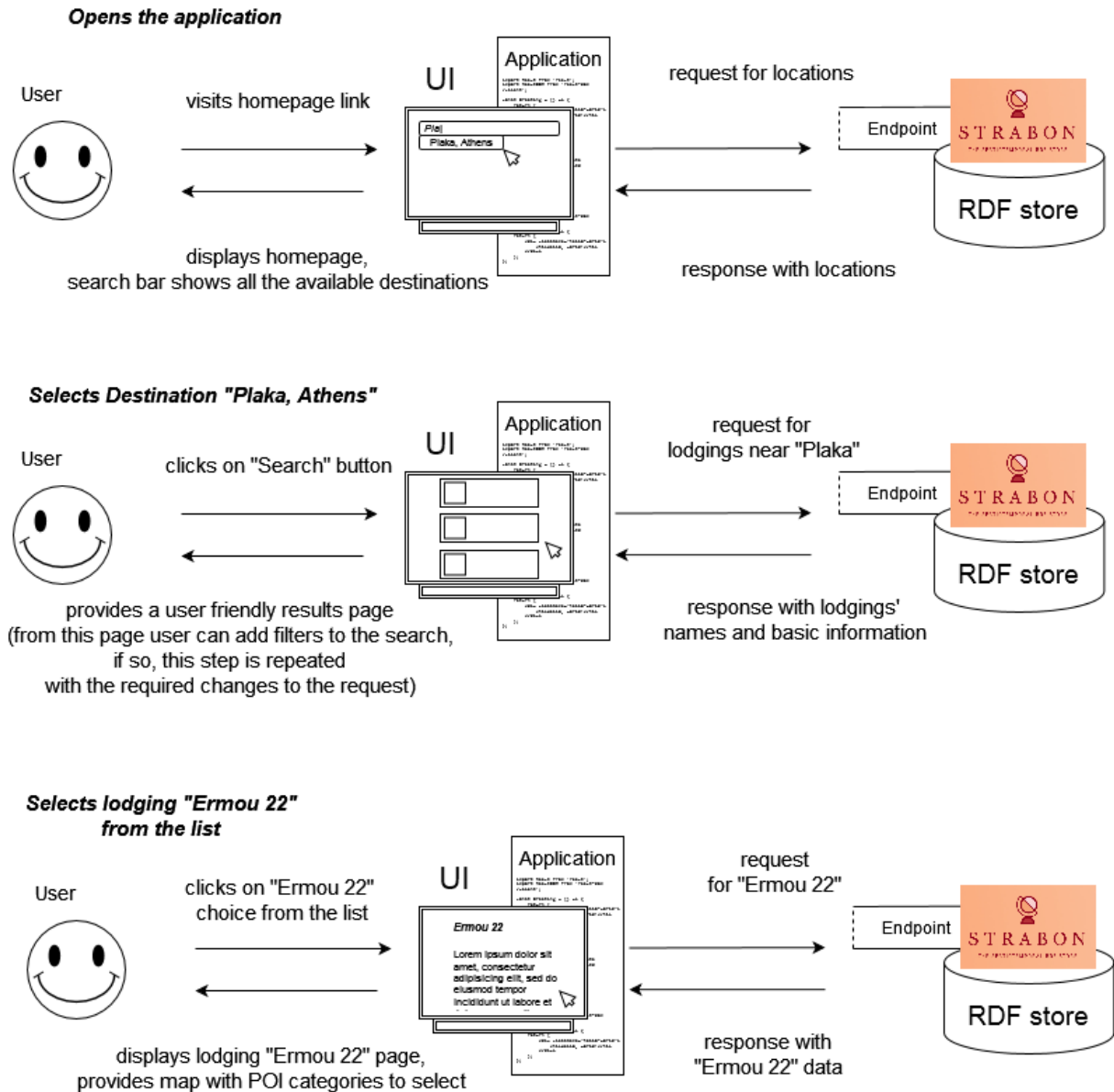
3.2. Architecture

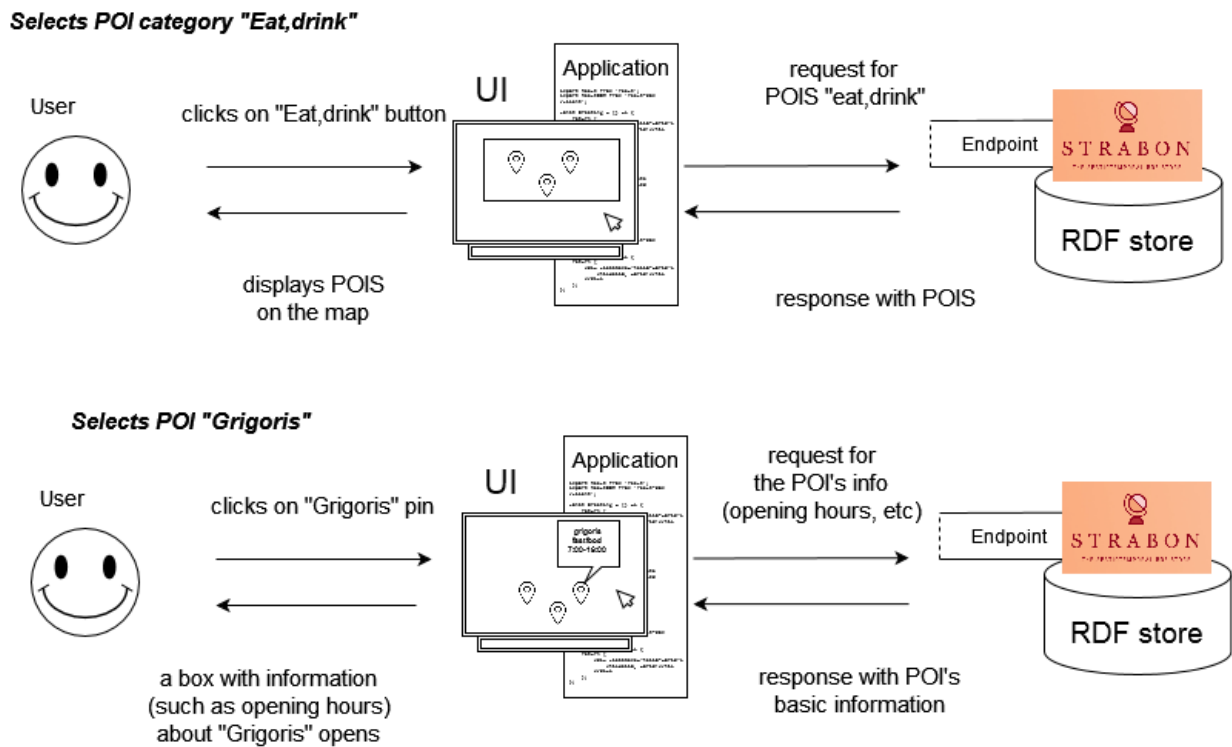**Figure 2: Application architecture (1/2)**

Figure 3: Application architecture (2/2)

3.3. Technologies

This paragraph details all the important technologies used for the implementation. The application code is divided into two folders, the client folder that implements the frontend part and the server for the backend.

3.3.1. Frontend

Navbar

Navbar is displayed in both the homepage and the subpages. It includes three fields. In the first, all the available locations for search are displayed in a drop-down menu. The package react-select, a flexible Select Input control with autocomplete[15] was used. In the second field, with the help of the react-date-range package the user can choose the date range he is interested in[16]. Finally, the user can choose the number of guests, before pressing the Search button.

Results page

It mainly contains two components: Filtersbox and List.

Filtersbox

This box displays all the filters that can be applied to the search. The react-select package was used. The filters are stored in the page's state, and from there they are used by the "List" component to make the appropriate queries to the backend.

List

In this component, the list of search results is displayed.

Axios was used to retrieve the data. Axios is a popular library, which is mainly used to send asynchronous HTTP requests to endpoints[17]. The queries use as parameters the requirements defined by the navbar and filtersbox components and have a structure similar to the following:

```
const result = await axios.get(`/results/${destination}?guests=${numofguests}&min=

${minPrice}&max=${maxPrice}&rtype=${roomType}&amenities=${amenities}`);
```
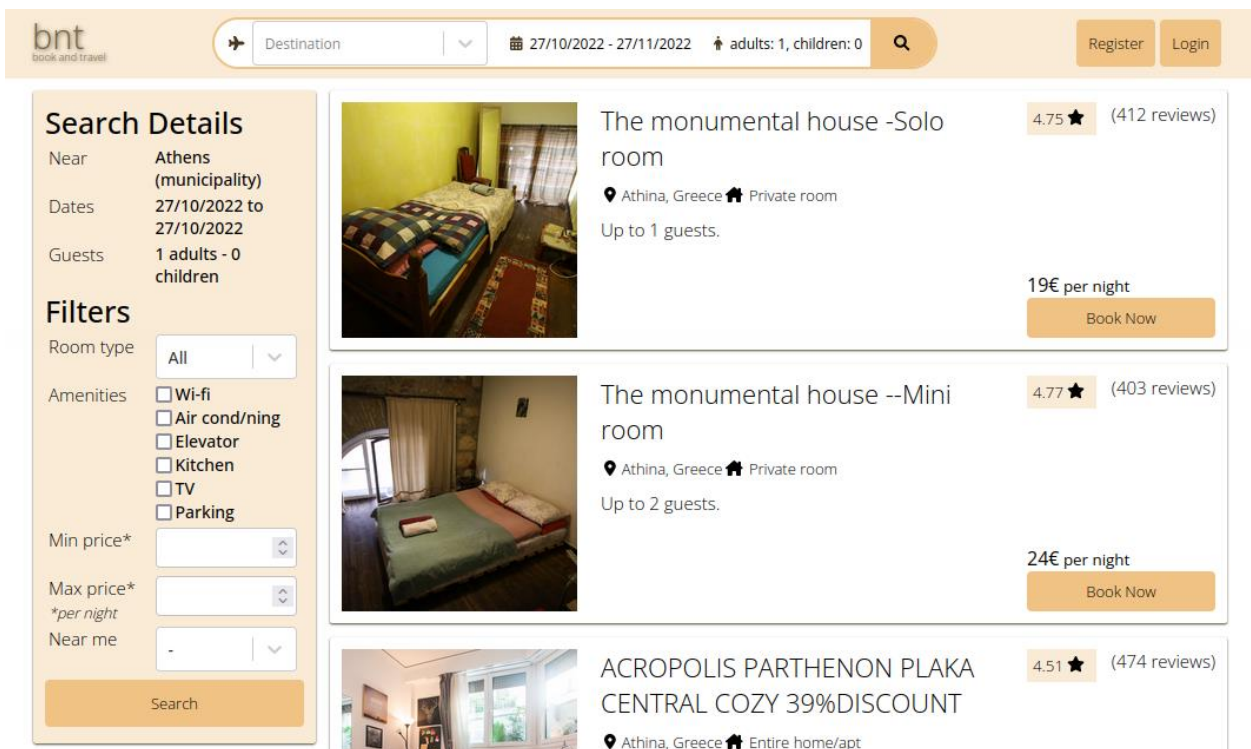

Figure 4: Navbar (top), Filtersbox (left), List (center)

To print the results returned by the executed query, the slice() and array() functions were used. To use these two functions, you have to attach each one of them to the array you want to iterate over. The slice() function is used to cut a part of an array (used in the List for pagination) and the map() function lets you manipulate the items in an array by iterating and accessing individual items (used to print the information of each lodging).

Sample code:

```
const Lodgings = () => {

  const data = [
    { id: 1, name: "Plaka hotel room", price: 100 },
    { id: 2, name: "Luxury Kolonaki private room", price: 120 },
    { id: 3, name: "Cozy apt near Acropolis", price: 90 },
  ];


return (
<div className="list">
{data.map((lodging) => (
<div className="lodgingContainer">
<h1>{lodging.name}</h1> <br />
                        Price: {lodging.price} euros
</div>
    ))}
</div>
);
};
```

Lodging

In this page/component, the detailed presentation of a lodging is displayed. In addition to the basic information (name, reviews, amenities etc.) an interactive map with the POIs around the lodging is displayed. Some markers are displayed on the map around a central larger marker that identifies the location of the lodging. To implement this interactive map, in addition to the axios library (sending queries to the backend) and the map function (to display the returned results), the "react-leaflet" library was used. Leaflet is an open source mapping library that utilizes OpenStreetMap. Although Google Maps and MapBox are more popular, both require accounts to use them and a paid subscription for certain features, while, on the other hand, Leaflet is a free, lightweight and easy to use map tool. React Leaflet provides React components for Leaflet maps [18].

Sample code:

```
const position = [38.0028291, 23.7408652]


render(

<MapContainer center={position} zoom={17} scrollWheelZoom={false}>

<TileLayer                              attribution='&copy;                              <a
href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'

url="https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png"

/>

<Marker position = {position}>

                <Popup>

    <b>Grigoris</b> <br /> Fast food <br />Mo-Sat 7:00-15:00

</Popup>

</Marker>

</MapContainer>

)
```
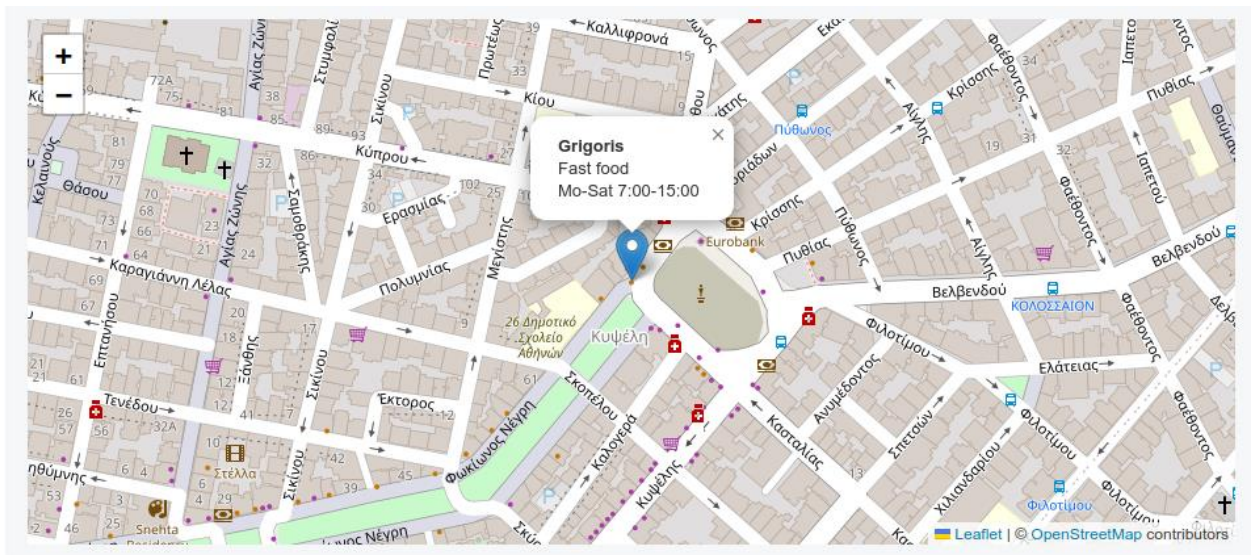
Result:

Figure 5: React Leaflet: Creating a basic map

3.3.2. Backend

NodeJS, Express

The backend application was built with NodeJS and EspressJS ().

Express is a NodeJS framework that provides several useful functions. It is commonly used in web applications as it make it easier for programmers to create APIs.

A basic API with Express, is presented in the code below.

```
//app.js

import express from "express"
const app = express()

app.listen(4000, () => {
console.log('server is listening on port 4000')
})

app.get('/api/pois', (req, res) => {
    res.json([
        { name: 'Acropolis', likes: 800 },
        { name: 'Omonia Square', likes: 650 },
        { name: 'Syntagma Square', likes: 750 }
    ])
})
```

In this code, the app.listen() will create an HTTP server and read incoming requests.

The app.get() defines a URL path that returns a JSON list containing POIs.

Strabon

In this application, all the data was stored in an RDF server created with Strabon. Strabon is a spatiotemporal RDF store. You can use it to store linked geospatial data.

Via Strabon Endpoint you can pose queries using two popular extensions of SPARQL (GeoSPARQL and stSPARQL). You can also store data from RDF files or make updates.
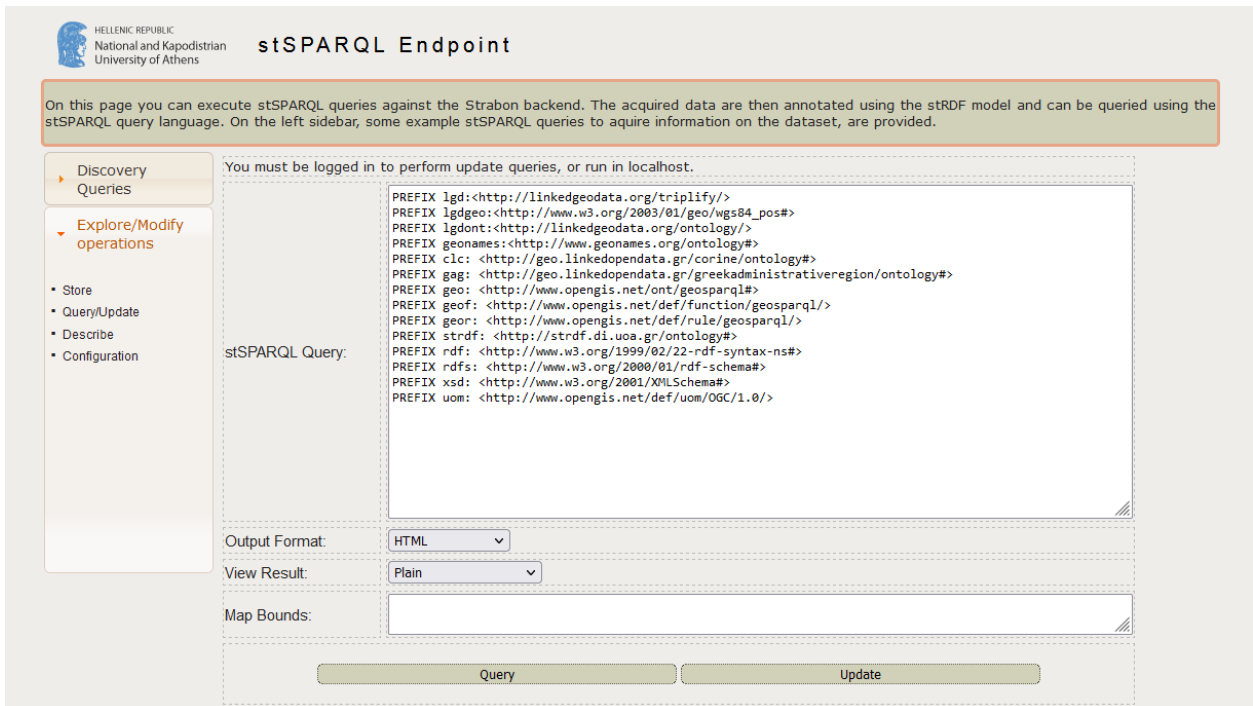

Figure 6: Strabon Interface

Queries made to the Strabon endpoint must be of the form http://localhost:9999/Strabon/Query?query=<query>. All queries coming from the frontend are encoded in the SPARQL language in the backend. The encodeURIComponent() function is then called with the string containing the SPARQL query as a parameter. encodeURIComponent() is a JavaScript function that encodes a URI by replacing each instance of certain characters by one, two, three, or four escape sequences representing the UTF-8 encoding of the character. For example the following SPARQL query is listed.

```html
<!DOCTYPE html>
<html>
<body>

<h2>The encoded query:</h2>

<p id="encoded"></p>

<script>
let sparqlQuery = `select ?POI ?label where
                  {
                      ?poi rdf:type :poi .
                      ?poi :hasLabel ?label
                  }`;
let endpointUrl = 'http://localhost:9999
/Strabon/'
let encoded = endpointUrl + "Query?query=" +
encodeURIComponent(sparqlQuery);

document.getElementById("encoded").innerHTML =
encoded;
</script>

</body>
</html>
```

The encoded query:

http://localhost:9999/Strabon/Query?query=select
%20%3FPOI%20%3Flabel%20where%20%0A
%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%7B%20%0A%20%20
%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%20
%3Fpoi%20rdf%3Atype%20%3Apoi%20.%0A%20
%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%20%20%20%20%3Fpoi
%20%3AhasLabel%20%3Flabel%0A%20%20%20
%20%20%20%20%20%20%20%20%20%20%20%20
%20%20%20%20%7D

Figure 7: The encodeURIComponent() function: From a SPARQL query to an HTTP API request

The backend, after making a get request (with the axios get() function), retrieves the data and sends it to the frontend, in JSON format.

Figure 8: Application connection with Strabon
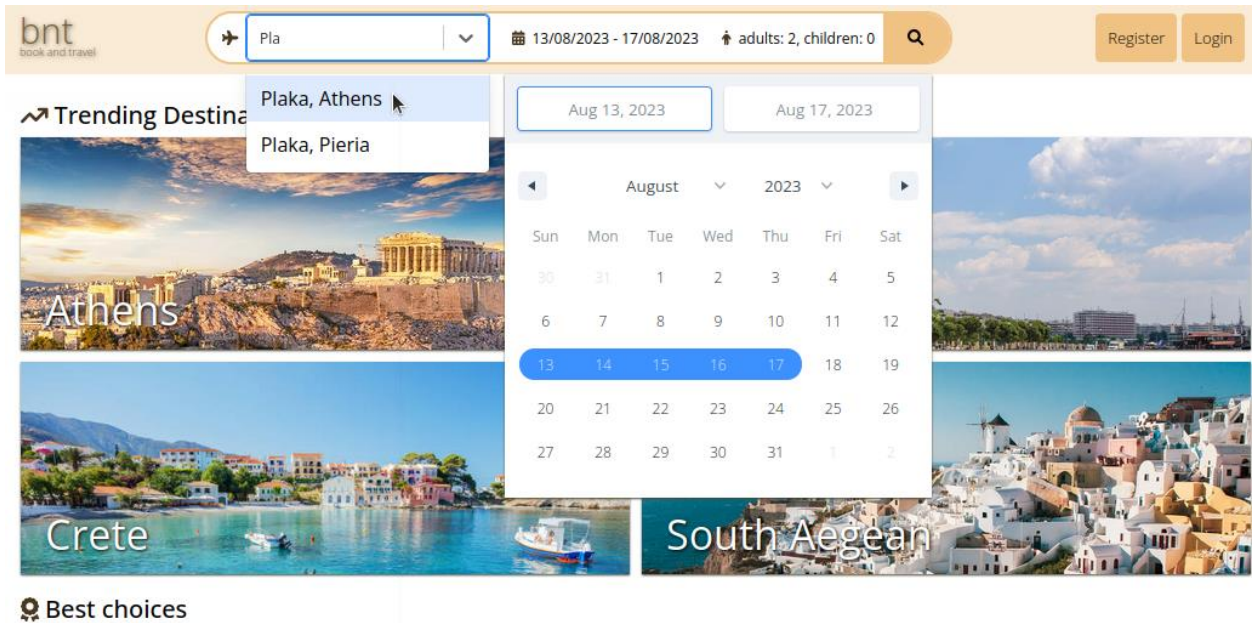
3.4. User Interface and features

Figure 9: Navigation bar

The navigation bar contains a search bar, which enables users to search for destination, set up check-in and check-out dates, and choose the number of the people who will stay in the lodging.
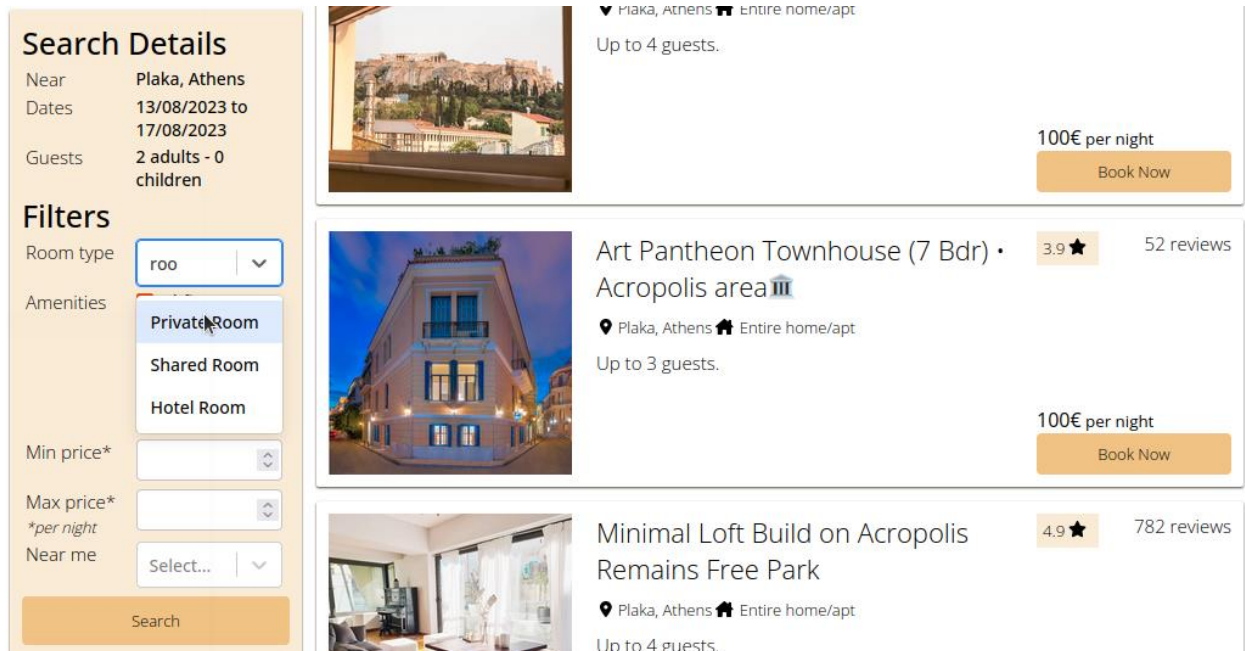

Figure 10: Results page

After clicking on the search button, user is driven to the results page. There, on the left, there is a box in which the user sees the search details he has just selected, and adds or removes filters. Filters that can be applied are "amenities" (Wifi, Parking, Air-condition system and elevator), "room type" (Entire home/apt, Private room), "price" (minimum, maximum). In the central part of the page, accommodation that meets the specified

conditions, paginated and sorted by price, are available. For each accommodation, information such as its title, price, type of accommodation, the city in which it is located in, its average rating and the number of its reviews are displayed. Through this page the user chooses the accommodation he/she is interested in.

Figure 11: Lodging page. Image and basic information.

On the lodging page, additional information is available. Specifically, the detailed description, the number of beds, the number of people who can be accommodated, the amenities it provides are displayed.
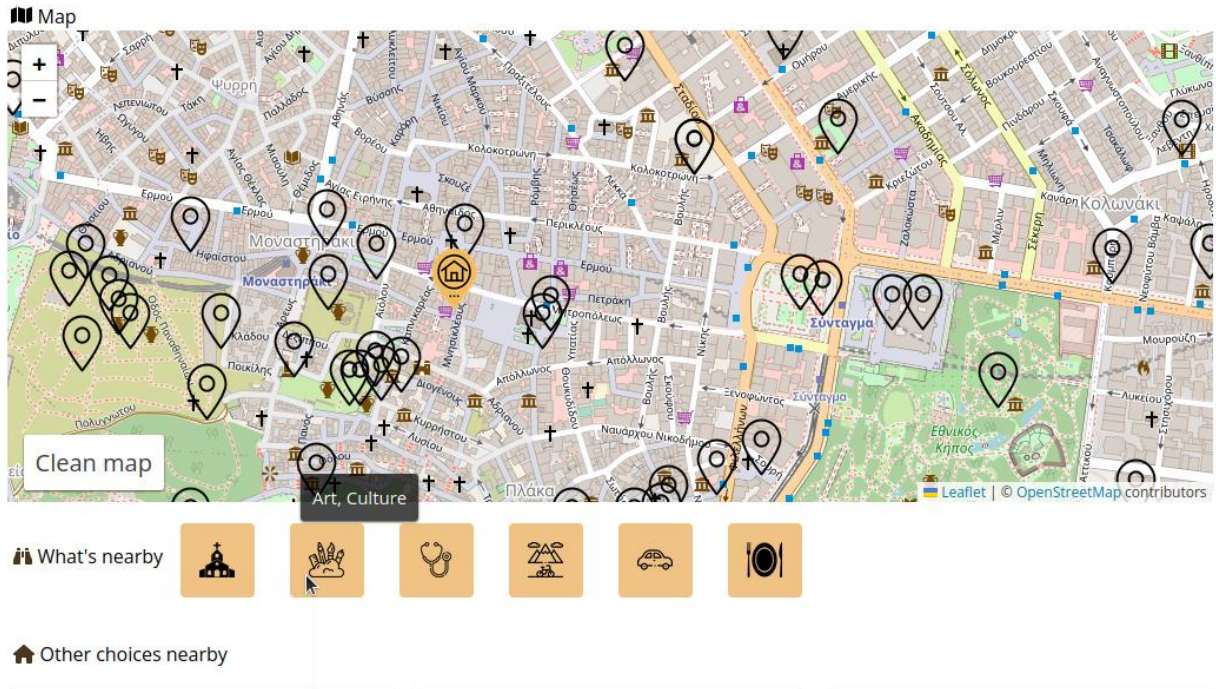
Figure 12: Lodging page. Selecting POI category.

In addition, this page has a map displaying the exact location of the accommodation, in which users can also display various POIs depending on the category they choose. For every POI presented on the map, users can read information about it and conduct a search for lodgings around it. If they do so, they will go back to a results page, in which they will also see the distance from the aforementioned point, for every result. In this case, the results are sorted by the distance from the POI.
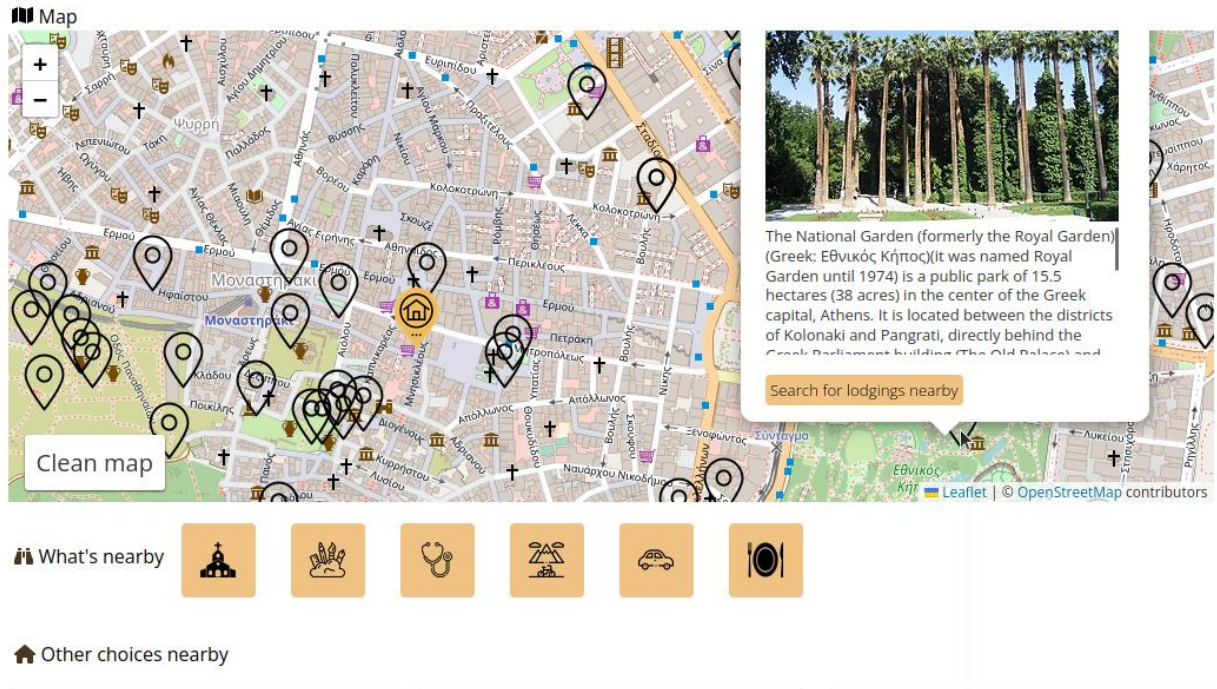
Figure 13: Lodging page. Popup with POI information.

Finally, at the bottom of the page users can have a look at a list with lodgings near the current one.

4. CONCLUSIONS – FUTURE WORK

The preparation of this thesis was the occasion for me to study on the Open Linked Data and Sparql language and to get in touch with tools such as Strabon and Geotriples, and data sources such as OpenStreetMap, DBpedia and others.

What is more, it developed my skills in terms of web development as I worked with the React and Nodejs environments.

In the future, the application could be enriched and extended to many areas. The POIs could be divided in more categories. More data about the POIs could be used whereas new ones (such as kiosks, hotels, beaches, shops, shopping malls and outdoor markets, internet/gaming cafés, public toilets, water supplies etc.) could be added. In addition, the destination search bar could be improved by suggesting appropriate locations and recognizing typographical errors. For example, "ejarhia" must be converted to "Exarchia, Athens". Also, it would be useful to provide a "group by" option in the filters box. Finally, the addition of Login and Register functionalities is useful so that users can configure their profile, add or remove lodgings, and change existing ones, exchange messages, add or read ratings and reviews.

5. APPENDICES

5.1. Code

You can find the application's code at the link below:
https://drive.google.com/drive/folders/1QhdHY9QJdYxDYIEg5BEL7EF9wGynTkpS?usp=share_link

TABLE OF TERMINOLOGY

Ξενόγλωσσος όρος	Ελληνικός Όρος
Point of Interest (POI)	Σημείο ενδιαφέροντος
Linked data	Συνδεδεμένα Δεδομένα
Geospatial	Γεωχωρικός
Web development	Ανάπτυξη ιστοσελίδων

Table 1: Terminology

ABBREVIATIONS - ACRONYMS

IDE	Integrated Development Environment
POI(s)	Point(s) Of Interest
URI(s)	Uniform Resource Identifier(s)
HTTP	Hypertext Transfer Protocol
HTML	Hypertext Markup Language
PHP	Hypertext Preprocessor
ASP	Active Server Pages
OSM	OpenStreetMap
RDF	Resource Description Framework
UI(s)	User Interface(s)
ΕΚΠΑ	Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Table 2: Abbreviations - Acronyms

REFERENCES

[1] Statista.com - Total number of user reviews and opinions on Tripadvisor worldwide from 2014 to 2021; https://www.statista.com/statistics/684862/tripadvisor-number-of-reviews/ [Retrieved 12/09/2022]

[2] V. Milias, A. Psyllidis, Assessing the influence of point-of-interest features on the classification of place categories, Computers, Environment and Urban Systems, Vol. 86, 2021, 101597, ISSN 0198-9715, https://doi.org/10.1016/j.compenvurbsys.2021.101597 [Retrieved 19/08/2022]

[3] D. Guttentag, "Progress on Airbnb: a literature review", Journal of Hospitality and Tourism Technology, Vol. 10 No. 4, 2019, pp. 814-844. https://doi.org/10.1108/JHTT-08-2018-0075 [Retrieved 19/08/2022]

[4] API Directory, Programmable Web; https://www.programmableweb.com/content/search [Retrieved 11/06/2022]

[5] C. Bizer, "The Emerging Web of Linked Data", IEEE Intelligent Systems, vol. 24, no. 5, pp. 87-92, Sept.-Oct. 2009, doi: 10.1109/MIS.2009.102.

[6] T. Heath, C. Bizer, "Linked Data: Evolving the Web into a Global Data Space" (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool.

[7] SPARQL Query Language for RDF, W3C Recommendation, 15 January 2008; https://www.w3.org/TR/rdf-sparql-query/ [Retrieved 11/06/2022]

[8] Kostis Kyzirakos, Dimitrianos Savva, Ioannis Vlachopoulos, Alexandros Vasileiou, Nikolaos Karalis, Manolis Koubarakis, Stefan Manegold, GeoTriples: Transforming geospatial data into RDF graphs using R2RML and RML mappings, Journal of Web Semantics, Vol. 52–53, 2018, Pages 16-32, ISSN 1570-8268, https://doi.org/10.1016/j.websem.2018.08.003.

[9] Kostis Kyzirakos, Manos Karpathiotakis, Konstantina Bereta, George Garbis, Charalampos Nikolaou, Panayiotis Smeros, Stella Giannakopoulou, Kallirroi Dogani, and Manolis Koubarakis. *The Spatiotemporal RDF Store Strabon.* In the 13th International Symposium on Spatial and Temporal Databases (SSTD 2013). Munich, Germany, August 21-23, 2013.

[10] React - A JavaScript library for building user interfaces, https://reactjs.org/ [Retrieved 07/04/2022]

[11] Component State - React, https://reactjs.org/docs/faq-state.html [Retrieved 08/04/2022]

[12] Node.js Introduction, W3 Schools Online Web Tutorials; https://www.w3schools.com/nodejs/nodejs_intro.asp [Retrieved 02/05/2022]

[13] Online Access – DBPedia ;http://wikidata.dbpedia.org/OnlineAccess#1.1%20Public%20SPARQL%20Endpoint [Retrieved 02/07/2022]

[14] Grinberger, A. Yair, Marco Minghini, Levente Juhász, Godwin Yeboah, and Peter Mooney. "OSM Science—The Academic Study of the OpenStreetMap Project, Data, Contributors, Community, and Applications" *ISPRS International Journal of Geo-Information, 2022,* 11, no. 4: 230; https://doi.org/10.3390/ijgi11040230 [Retrieved 09/10/2022]

[15] react select Library Official Page; https://react-select.com/home [Retrieved 20/02/2022]

[16] react-date-range Readme; https://www.npmjs.com/package/react-date-range [Retrieved 22/02/2022]

[17] axios Docs - https://axios-http.com/docs/intro [Retrieved 01/01/2022]

[18] React Leaflet - React components for Leaflet maps; https://react-leaflet.js.org/ [Retrieved 05/07/2022]