PalStay

Presented in partial fulfillment of the requirements for
Bachelor degree in Computer Engineering

Faculty of Engineering and Information Technology
Department of Computer Engineering

Academic Year: 2023-2024

Software Project

Student Name:
Ahmad Atout

Supervisors:
Dr. Emad Natsheh

Date: 29/01/2024

Acknowledgment

I extend my deepest gratitude to Dr. Emad Natsheh for his helpful guidance and insightful feedback throughout developing and designing this project and making it come to life. A special thanks to every individual who was there whenever needed gave me support or helped me through complex situations. I also acknowledge the faculty of Engineering and information for providing me with the necessary resources and inspiring learning environment. This journey has been both challenging and rewarding, and I'm always thankful for life experiences which give me opportunities to learn and grow.

Disclaimer

This report has been written by Ahmad Atout at the Computer Engineering department, Faculty of Engineering, An-Najah National University. The report was not altered nor corrected, only in cases of editorial corrections, it may contain language or contant errors. The views expressed in it together with any outcomes and recommendations are solely those of Ahmad Atout. An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

Contents

List of Figures

Nomenclature

API	Application Protocol Interface
HTTP	Hyper Text Transfer Protocol
JSON	JavaScript Object Notation
REST	Representational State Transfer
XML	eXtensible Markup Language

Abstract

The real estate rental landscape in Palestine is experiencing rapid evolution, there are a lot of property listings everywhere. However, to reach these listings is a big challenge, the communication channels between prospective renters and property hosts is not easy. In the past, the rental options were limited to hotels and resorts, the demand extends nowadays to single rooms within shared apartments.

Regarding that issue, specially in Palestine, we have developed PalStay, a pioneering application designed to simplify the rental process. PalStay provides an intuitive and user-friendly experience, enabling users to effortlessly refine their property searches with a single click. Users can filter their searches based on various criteria, including location, maps, pricing, and even pet-friendliness.

PalStay offers a dual functionality, allowing property owners to list a diverse range of accommodations, from individual rooms and apartments to houses, villas, and chalets, while allowing rentals to refine their searches to match their preferences, it also allows them to communicate with the hosts directly through our chatting system. It allows users to discuss details and address inquiries prior to finalizing their rental decisions.

Moreover, PalStay recognizes the importance of personalization in the rental experience. Users can add listings to their favorites. Using this feature, PalStay's recommendation system employs advanced algorithms to analyze users' favorites, providing suggestions based on their preferences. This personalized recommendation system significantly enhances the user's ability to discover properties that align with their unique tastes and requirements.

One of the features in PalStay is the rating system, which ensures user confidence and satisfaction. Previous rentals can share their experience and insights, which allows new rentals to read their feedback and review before deciding. It's different than the current method, which is through Facebook groups.

1 Introduction

1.1 Problem

Finding accommodation in Palestine is not a flexible thing, due to the lack of services that combine all the accommodation options in one place, whether for tourists to find a place to stay during their journey or locals who are looking for a temporary place to stay.
Especially with the living under occupation issue, many people find themselves stuck in other cities with nowhere to spend the night. In addition to that, there is no service that makes office spaces available for renting for a short or a long period of time.
so with PalStay, we were able to combine all of these real estate only with one click away.

1.2 Objectives

the main objective to develop such an application is to take the real estate business in Palestine to a higher level, making Palestine a new destination for investors to invest and expect successful results.
Furthermore, this application aims to streamline the property rental process, making it more efficient and accessible for both local residents and international clients. By harnessing advanced technology, we intend to offer a comprehensive and user-friendly platform that simplifies finding and renting properties, fostering a more dynamic and interconnected real estate market in Palestine.

1.3 Scope of The Work

PalStay is a mobile application which was built using verious software development technologies, for the Back-End, I used Nodejs development enviroment and Expressjs as the server Back-End application handeling all the endpoints.
The Client and the server communicate using REST architectural style that uses HTTP requests to carry the data in JSON/XML format.
The front-end was built using React Native, a framework for creating mobile apps using JavaScript, and Expo, a platform that simplifies React Native app development making it easier to test those apps during development on both IOS and Android.
For the database, I used mongoDb which is a noSql database service.
The website is for admin use only and it was built using React framework.

1.4 Significance of The Work

There might be other similar applications, but it was noticed that most lacked user-friendliness and people could not use easily, we are aiming to combine all the properties in Palestine whether it's an apartment, office or hotel room in one place, with many technologies such as map search, reccomendation system and

chat system that make the application really friendly and easy to use, which is not found in any application.

1.5 Organization of The Report

The body of the report will start with **the The First Chapter**, Introduction, providing a background for the research. **The Second Chapter**, Constraints and Earlier Coursework, that shows project constraints, and who they were overcame and solved. Previously taken courses that were helpful in building this application in any step of its development.

Followed by **The Third Chapter**, Literature Review, which establishes familiarity and understanding of the current research and includes any similar project.

Then **The Forth Chapter**, Methodology, that includes a systematic plan to resolve the problem, the process of building this application.

After comes **The Fifth Chapter**, Results and Discussion, to summarize the data collected, their statistical treatment and compare the results.

And finally **The Sixth Chapter**, Conclusion and Recommendation, will show the final project summary,with everything learnt in the journey of building it. Also, it'll introduce some of the features/subsystems that can be added in order to improve and enrich it.

2 Constraints and Earlier Course Work

2.1 Constraints

- This project was done during a period of great war against our people in Palestine. so working under all of this tension and stress watching devastating scenes every day was a huge challenge.
- The project was implemented by only one individual so building all these features and learning all the frameworks and technologies in one semester was a journey full of obstacles and time management challenges.
- Time Limit: It has been implemented in a period of time being 4 months, which is considered a challenging amount of time to build this whole project in both web application and mobile application. Choosing the idea, deciding features, searching the topic, taking courses to learn node.js, react native, dealing with APIs writing the full code and assembling everything into the model.

2.2 Earlier Course Work

- **Database**: This course helped in building the relational database, gave a good introduction on to how to implement action handling and helped in formatting the SQL queries and requests.
- **Web Application** : Gave an introduction to dealing with web applications and how to use MySQL database, HTML, CSS and javascript.
- **Advanced Software**: This course was very helpful as it gave the details about RESTful architecture, encouraged learning react.js and node.js as an outside activity. And introduced several concepts that are useful.
- **Distributed Opertaing Systems**: gave mea really good introduction on REST APIs.

- **Artificial Intellegence**: This course gave and introduction about dealing with neural networks and the basics of machine learning and artificial intelligence.
- **Computer Programming**: gave me a good inroduction on programming and solving problems.

3 Literature Review

Technology has evolved in many ways and interfered with our daily lives in almost every aspect, so using its features in the real estate business can be really helpful in many ways. So instead of using phone calls and connections to find a place to rent, we can use technology to do that for us and make us only one click away from finding the best place that fits our needs. it introduces many features that could make this process simpler in all aspects, starting from searching for the best place to reaching out to the owner to finally experiencing the best stay that fits all our needs.[8]

PalestineRealEstate Platform was stumbled on while searching. It allows users to find properties to rent, buy all across Palestine.[6] the application is very useful but not user-friendly and this discourages people from using it.

While searching for similar e-solutions, plenty have been developed to perform this. yet most of them lack a completely easy approach for renting or an easy approach for listing your properties, or a service that recommends to you what fits you.

So the search got wider to include machine learning. so we found a recommendation system engine .[4] that uses collaborative filtering .[1] for an e-commerce platform that we could use to give the user the best experience.

Now, after deciding the features of the application, the search for what technologies in the development process started. React Native was suggested to build a cross-platform application, and with its strong community, React Native was no doubt the best choice to start a cross-platform application from scratch..[3]

Since the details for the kind of database was not set just yet at the beginning, it was decided to choose REST architecture alongside Node.js in the backend. REST provides a great deal of flexibility. Data is not tied to resources or methods, so REST can handle multiple types of calls, return different data formats and even change structurally with the correct implementation of hypermedia. [7]

while reading the documentation for react native and similar projects done using it, react expo got recommended since, it is built on top of react native, but saves the effort for the developer to write separate code for android and IOS since it does it automatically. Expo is an open-source framework for apps that run natively on Android, iOS, and the web. Expo brings together the best of mobile and the web and enables many important features for building and scaling an app. .[9]

In our e-commerce application, we chose Node.js for its exceptional performance and scalability, which are crucial for handling the high traffic and dynamic content typical in online shopping platforms. Node.js's non-blocking I/O model

ensures efficient handling of simultaneous requests, maintaining fast response times even under heavy load. Its vast ecosystem of libraries, particularly for web development, allowed for rapid development and easy integration of various features. Moreover, being JavaScript-based, it streamlined our development process, as the same language could be used for both client-side and server-side scripting, enhancing team productivity and maintaining code consistency. [2]

We opted for MongoDB in our e-commerce application due to its flexibility and performance with large-scale data. MongoDB's schema-less nature allows us to easily adapt to the varying data requirements of e-commerce products and user profiles. Its ability to handle large volumes of data and high throughput makes it ideal for an application with potentially high traffic and diverse data. Additionally, MongoDB's powerful querying and real-time analytics enhance our application's responsiveness and user experience. [5]

4 Methodology

This chapter talks about the design of PalStay, the components, tools and languages that were used to build it .

4.1 Design

4.1.1 Architectural Design

In the first steps of designing PalStay, it was decided to make it scalable as such application is always on demand and there is always room for developing new features. So a good architectural styles needed to be followed that allows scalability.

The client-server architecture enables a separation of concerns, meaning, separating the client from the servers.

This enhances the scalability of the server components, and the portability of the user interface across diverse platforms. The picture below describes the RESTfull architecture:

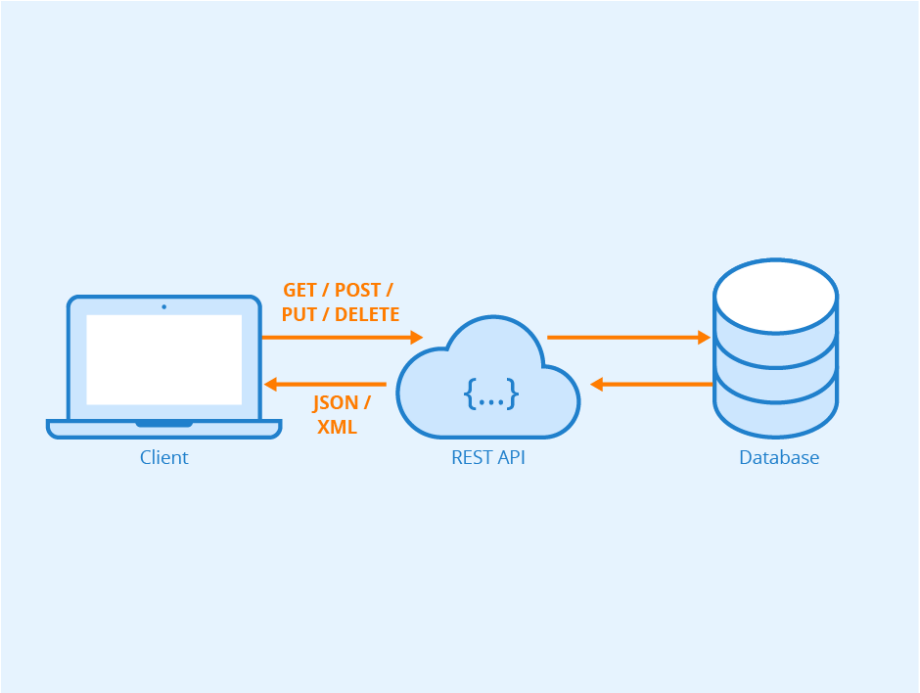


Figure 1: Rest API

It provide a great deal of flexibility. The user sends requests with:

· GET: Read only acces of the resource

· POST: Creates a new resource

· DELETE: delete resource

· PUT: Edit resource

And awaits the response from the server-side, which comes in json format. Since REST architecture was chosen, it was needed to choose a language for the client-side and the server-side. According to stack overflow Node.js and React where the most popular languages for year 2022, as shown in the figure below. So Node.js was used in building the server-side and react for the client-side.

4.1.2 PalStay DataBase Schema

The database used for PalStay application was Mongodb and the collections that were made are;

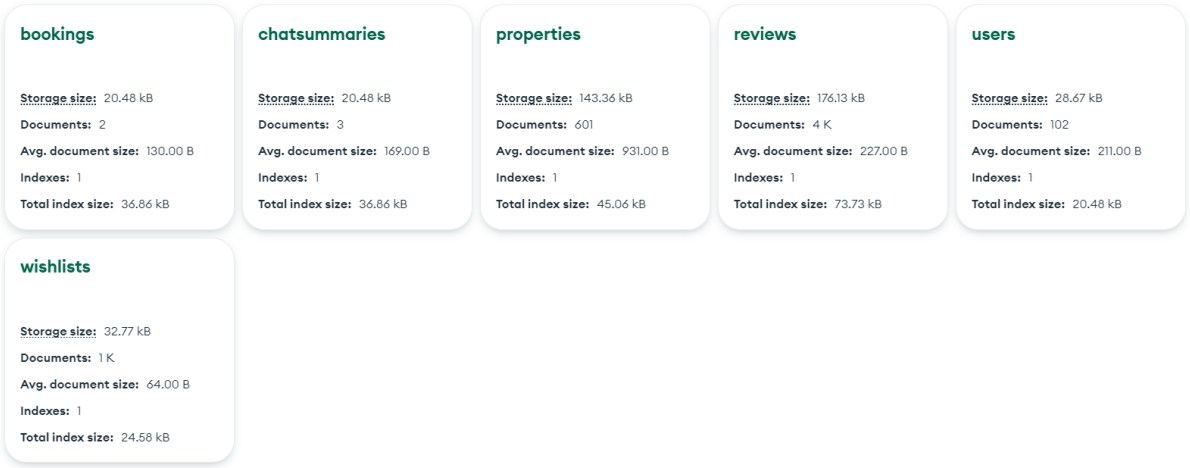


Figure 2: database collections

- **users**: for storing the users of the application


```
_id: ObjectId('65b222bd20758ce5be2d078a')
phone: "00970593117349"
username: "wdailey0"
email: "jduchart0@earthlink.net"
password: "$2b$10$jUi5K1WTS8xsnEaBD70IZO4nMhbrqBL6O1CaiM1/K4LCfIVwawD9i"
role: "user"
▸ wishlist: Array (empty)
```


Figure 3: users document

- **properties**: for storing the properties listed

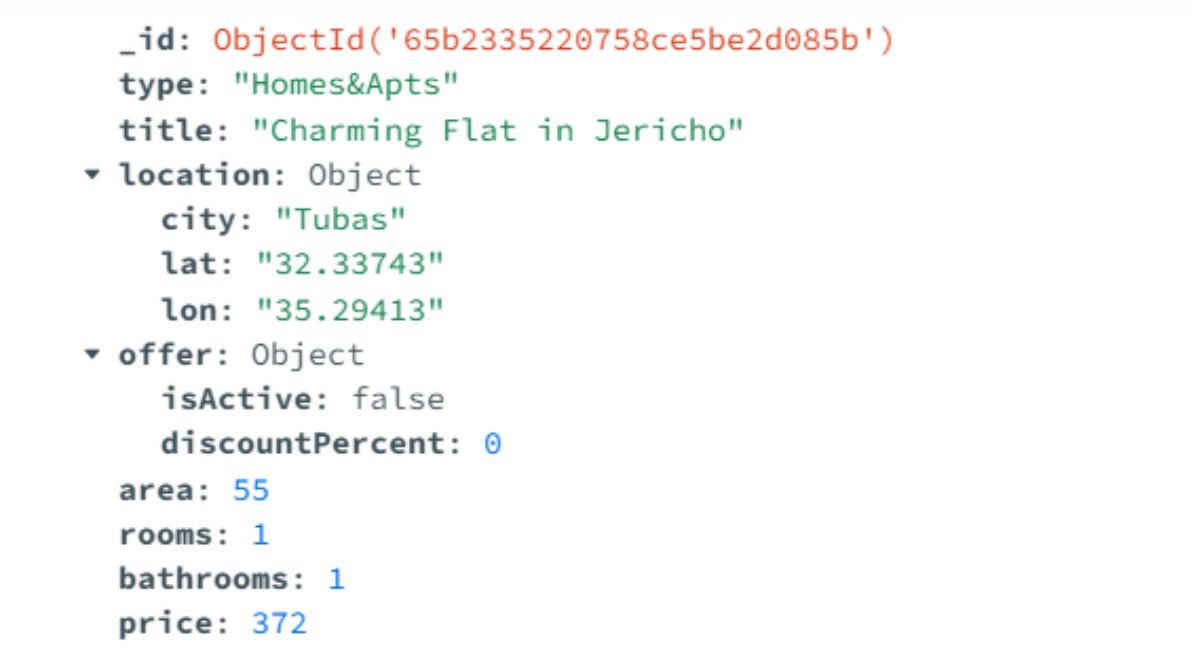
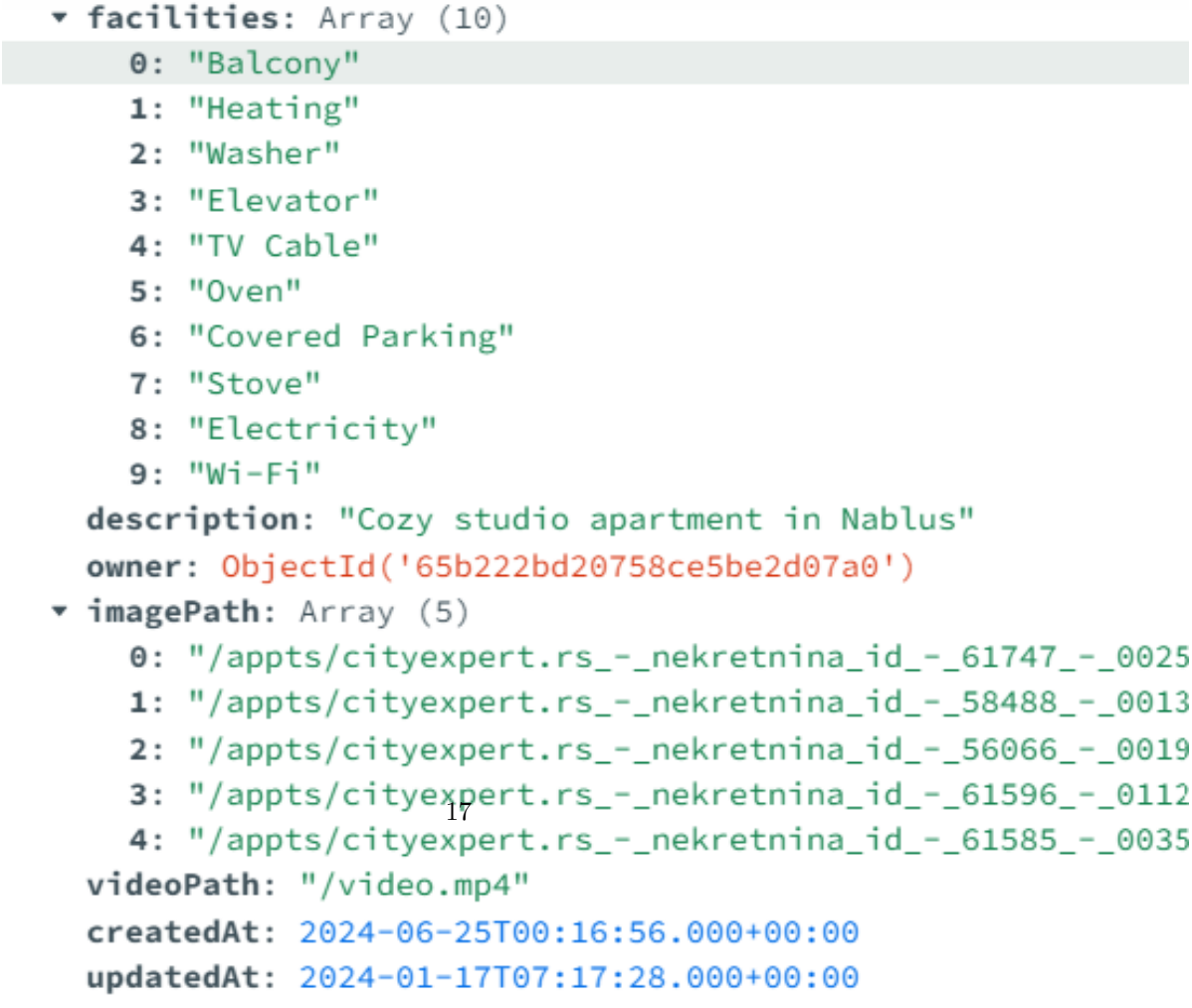


Figure 4: property document



· **reviews document**: for storing the reviews of tproperties

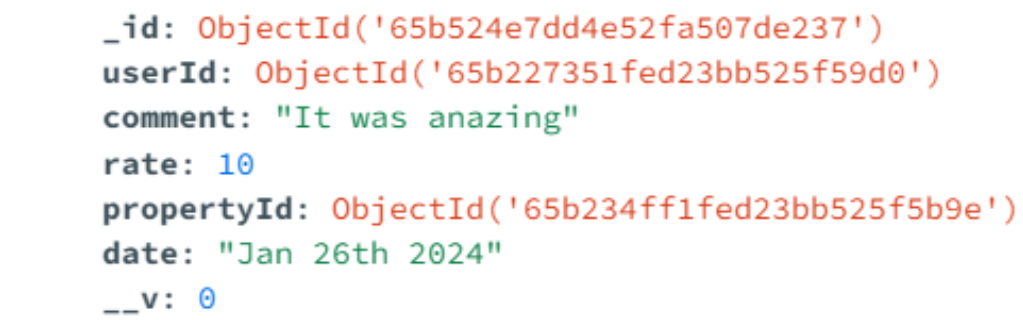


Figure 6: reviews document

· **wishlists document**: for storing the wishlit of each user

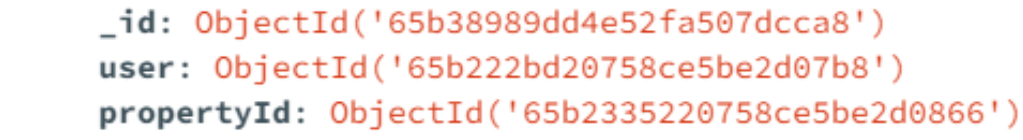


Figure 7: wishlist document

· **booking document**: for storing all the bookings requests and valid ones

Figure 8: booking document

· **chat summaries document** : to store all the chat summaries between users

Figure 9: chat summaries document

4.2 Tools Programming Languages APIs Technologies

4.2.1 Programming languages and frameworks

1-**Mobile Application**

- **React Native**: The mobile application was built using React native because there is no need to create separate codes for different platforms (Android and iOS), 90be reused between the two platforms, React Native supports "live reloading" which allows developers to see the changes that made to the code in another live preview window at the same time. Because of the real-time feedback, it gave developers a significant advantage, in addition to, React native gives Strong performance for mobile environments, Time and cost-effective, Flexibility, Comprehensive libraries.
- **Node JS** : The back-end for both the mobile application and website was built with Node js and express. Node JS makes a good back-end for applications, providing server-less architecture and real-time communication support. And express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

2-**Web Application**

- **Reactjs**: The admin side website was built using React js it's an amazing JavaScript framework that has made the development process much easier. it provides quality assurance React js interface, and easy to adopt.

4.2.2 Tools

- **GitHub**: It is a code hosting platform for version control and collaboration. It was used to share the code between the team members to work together on the project from anywhere. there were three repositories, one for the Front-End, one for the Back-End, and the third one for the react js admin side.

- **Visual Studio Code**: Used to develop both the client-side, mobile and web, and the server-side applications.

- **Expo Go**: Expo is an open-source platform for making universal native apps for Android, iOS, and the web with JavaScript and React. It was used to test the application features in the development process.

- **Postman**: Postman is an API platform for building and using APIs. Postman simplifies each step of the API life-cycle and streamlines collaboration so you can create better APIs—faster. It was used to test the endpoint created in node. js before sending them to the client-side

- **Mongodb Compass**: used to manage the DataBase and have an overview of it while developing

- **Mockaroo**: Used to generate Mock Data to fill the application with working data.

- **Collaborative filtering reccommendation engine**: used to recommend properties for the user based on the likings of the closest neighbor.

4.2.3 APIs

- **Firebase Chat API Integration**: Integrated Firebase's chat API to enable real-time communication between users within the application. This feature enhanced user engagement by allowing seamless messaging functionality.

- **Firebase Notification Service Integration**: Implemented Firebase's notification service to deliver timely updates and alerts to users. This ensured that users remained informed about important events, such as new messages or updates to their listings, enhancing overall user experience and interaction with the platform.

4.3 Implementation

4.3.1 Frameworks

- **React Native**: Expo was used for the mobile application, it's a free and open source framework that's used to build React native apps. It comes with a lot of native APIs out of the box for both iOS and Android. That makes the developers job of adding native features to the app fairly easy. It also provides a number of built-in Core Components ready to use in the app. some useful components was used in the application (ActivityIndicator, Alert, Animated, Dimensions, Modal, RefreshControl).

- **React js**: The website was created for the admin side, The admin can observe the number of users, any changes that happen in users information, add and remove users. Admin can also perform CRUD operation. with a very nice GUI.

- **Node js**: The server-side for both the admin side and the client side was implemented using node. js with express framework. The back-end provides all the endpoints needed to dynamically show the information for the client. It also handles the requests the user sends and send the right responses.

4.3.2 Back-end implementation

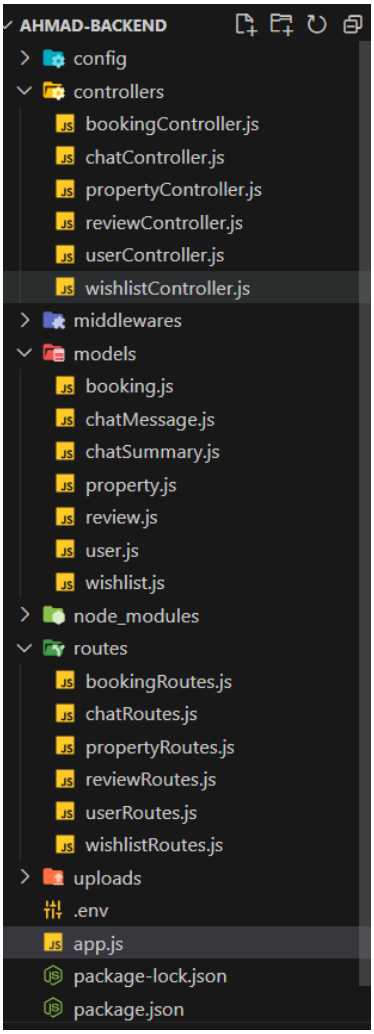


Figure 10: Back-end implementation

4.3.3 Application features

- **Authorization and authentication**: users can sign up and sign in with a secure way, the application has 2 roles: Admin And user, The admin is the only user who can log in to the web Application and view the admin panel to manage the application

- **Listing properties**: users can list their properties to make them available for people to look up and rent

- **renting properteis**: Users can look for the best fit properties on the application and decides which one suits them best and rent them

- **search filters**: users can search for the best fit property to rent using their personal prefrences using the search feature that provides good prefrences metrics such as number of rooms of the facilites available, and the duration they want to spend in the property or seach by the city.

- **search by maps**: users can look up for properties using map view to find the best property that matches their desired location.

- **renting properteis**: Users can look for the best fit properties on the application and decides which one suits them best and rent them

- **wishlist**: users have the ability so save the properties they liked in the wishlist section to make it easier for them to rent the best option

- **making offers**: users can make a deal offer on their listed properties for people to enjoy.

- **recommended properties**: based on the own likings of the user in the wish list, the application will recommend similar properteis to them that they may like.

- **make book request**: users can make a booking request to the owner for him to evaluate ut before accepting.

- **review system**: when a user finishes his stay at a property he can then give a rating of his stay with a review comment, all of this reviews will appear in each property details and an avarage rating of the property will appear too

- **Chatting system**: user can start a conversation with the owner to discuss matters related to the property or property renting

- **review system**: when a user finishes his stay at a property he can then give a rating of his stay with a review comment, all of this reviews will appear in each property details and an avarage rating of the property will appear too

- **notification system**: the application will send push notifications for the users when they recieve a message, recieve a booking request, receive an acceptance of their booking request

- **review system**: when a user finishes his stay at a property he can then give a rating of his stay with a review comment, all of this reviews will appear in each property details and an avarage rating of the property will appear too

- **review system**: when a user finishes his stay at a property he can then give a rating of his stay with a review comment, all of this reviews will appear in each property details and an avarage rating of the property will appear too

4.3.4 Mobile Application

- **Welcome Screen**: The welcome screen is the first screen the appears for the user, its user friendly and it has the application logo, application name, a log in button, and a sign up button

Figure 11: Welcome Screen

- **Log in Screen**: this screen is for the user to log in to the application

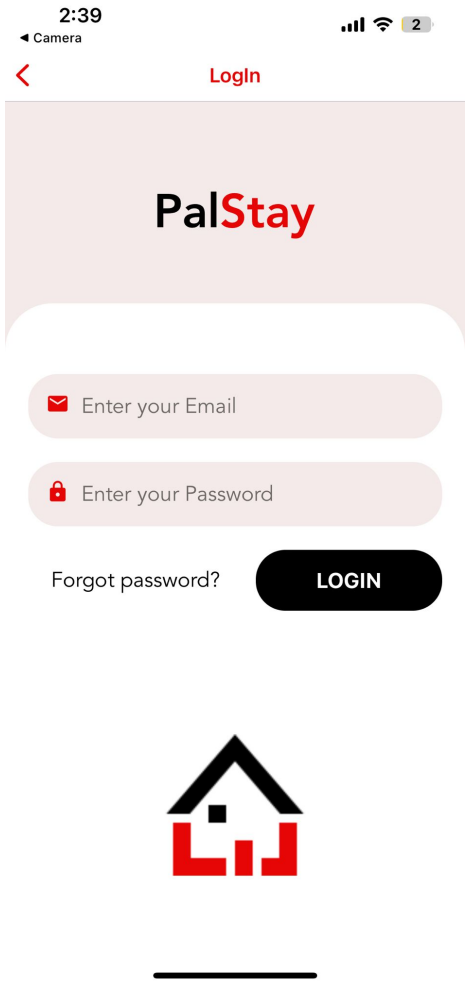


Figure 12: login Screen

- **Sign up Screen**: this screen is for the user to sign up to the application

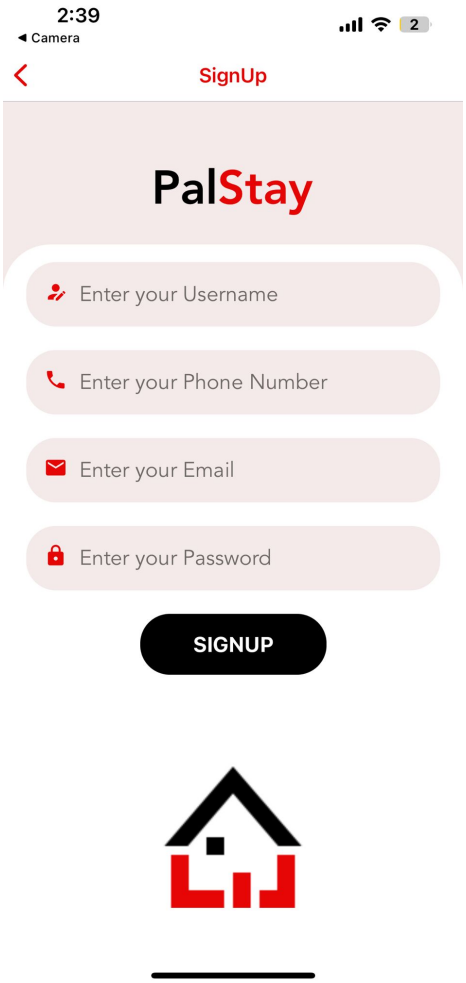


Figure 13: Sign up Screen

- **home Screen**: this is the home screen which contains 3 categoty buttons (homesapts, offices, hotels) when u press on one of them it takes you to the filter search screen. we also display special offers in the home page as well as the for you slider which display recommended properties for you.

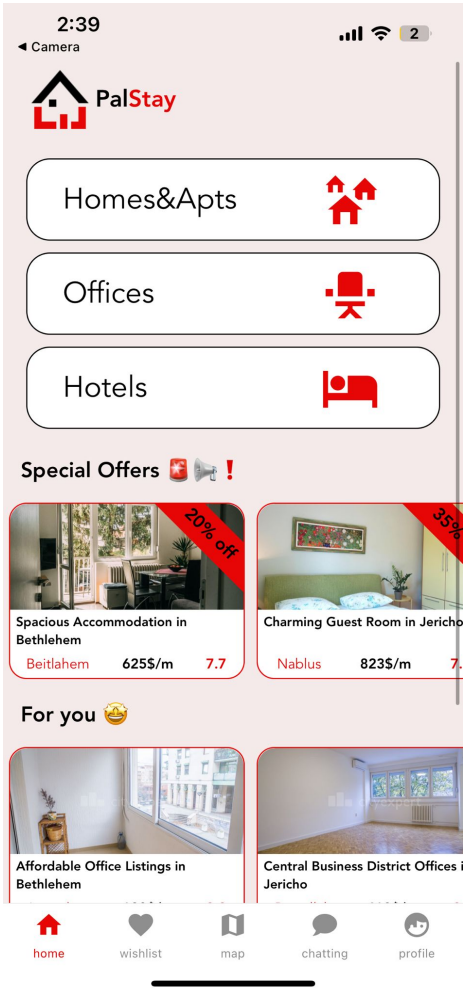


Figure 14: home Screen

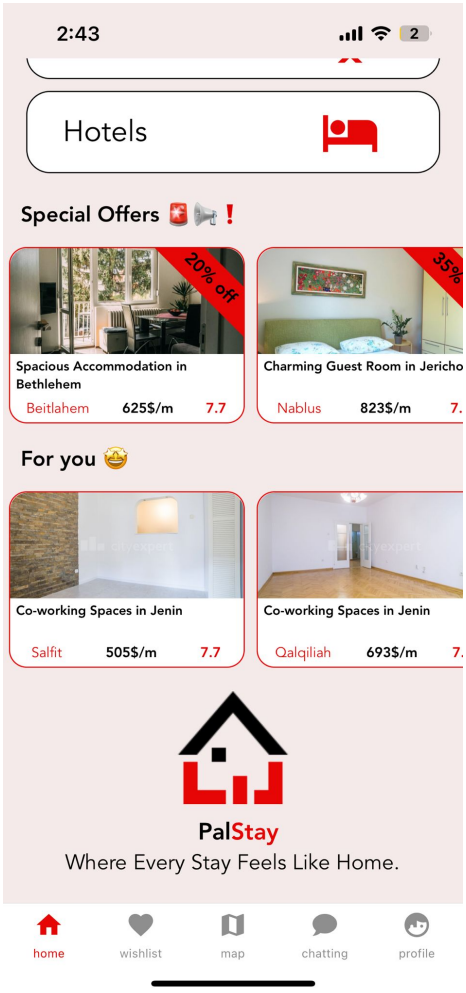


Figure 15: home Screen

- **search filter Screen**: is this pages u can choose filters based on your own preferences and you enter the date range of your stay to see available properties on these dates, and you also select the facilities you would like to be available during your stay

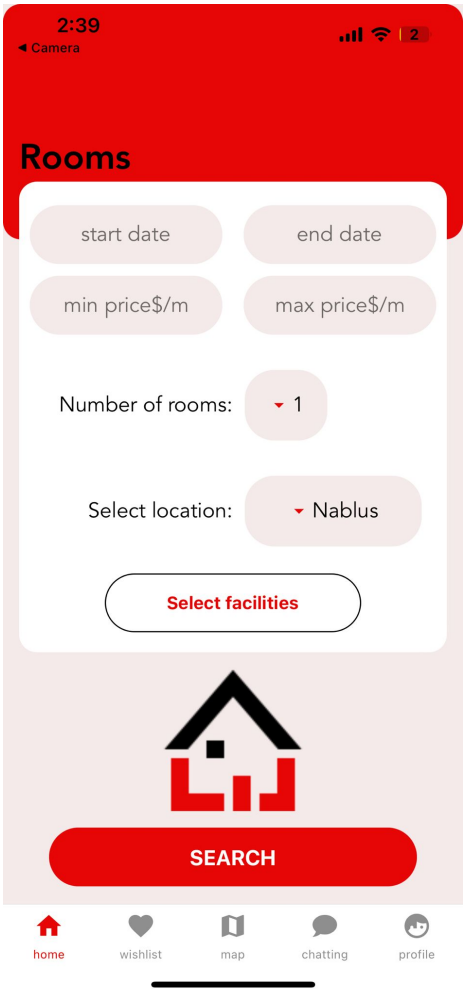


Figure 16: Search filter Screen

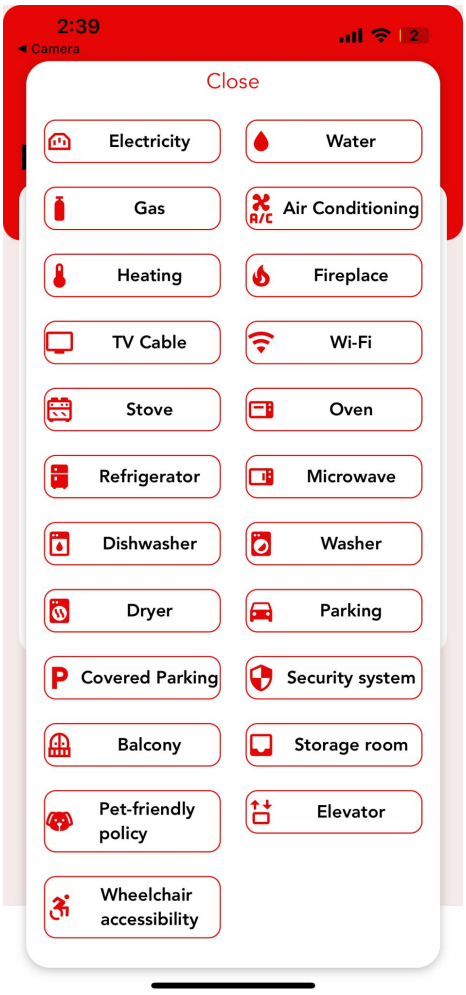


Figure 17: pick facilities

- **Search result Screen**: this screen displays the results of your search

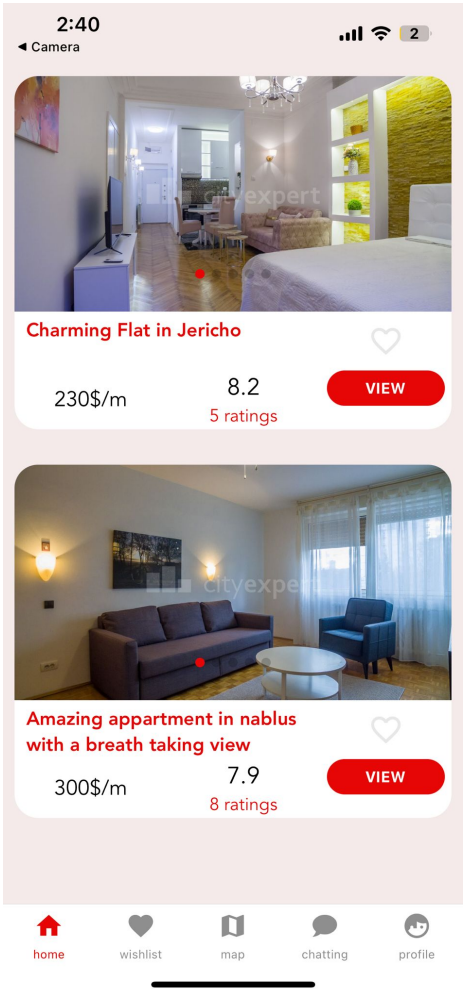


Figure 18: Search results Screen

- **Property Details Screen**:this screen to view the details about the property with options to open a video, view the property reviews, make a booking and contact the owner or view the owner profile screen.

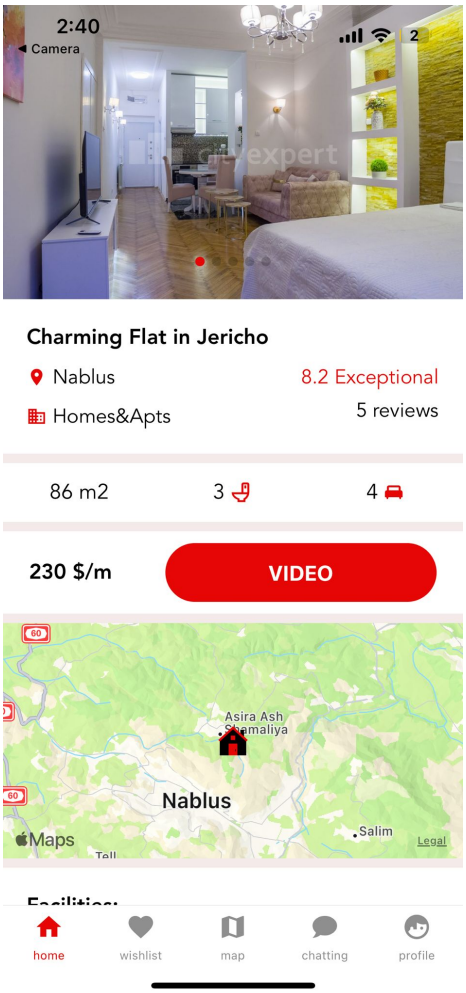


Figure 19: property details Screen

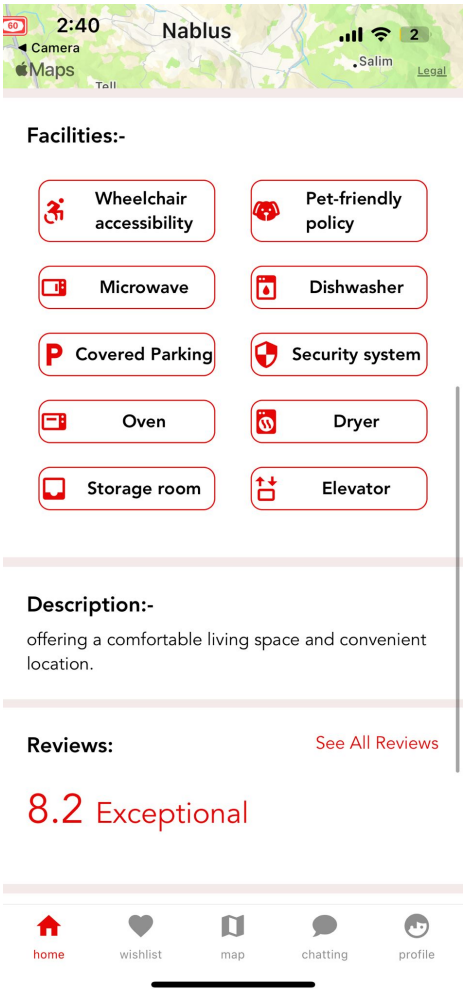


Figure 20: property details Screen

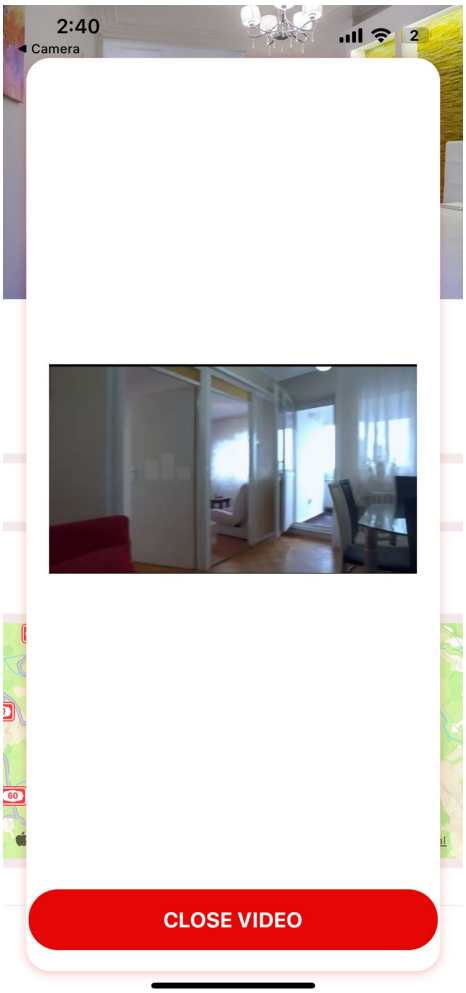


Figure 21: property video Screen

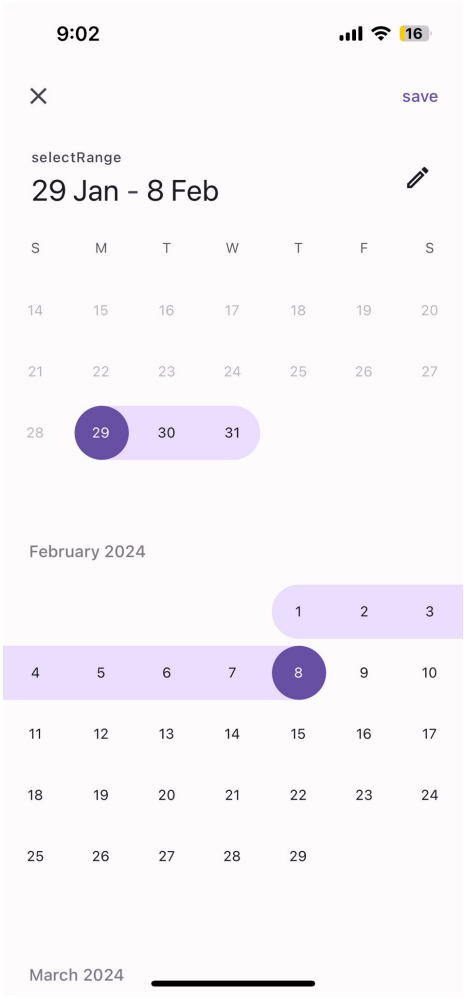


Figure 22: Make booking Screen

- **Reviews Screen**: this screen is to display the reviews on this property

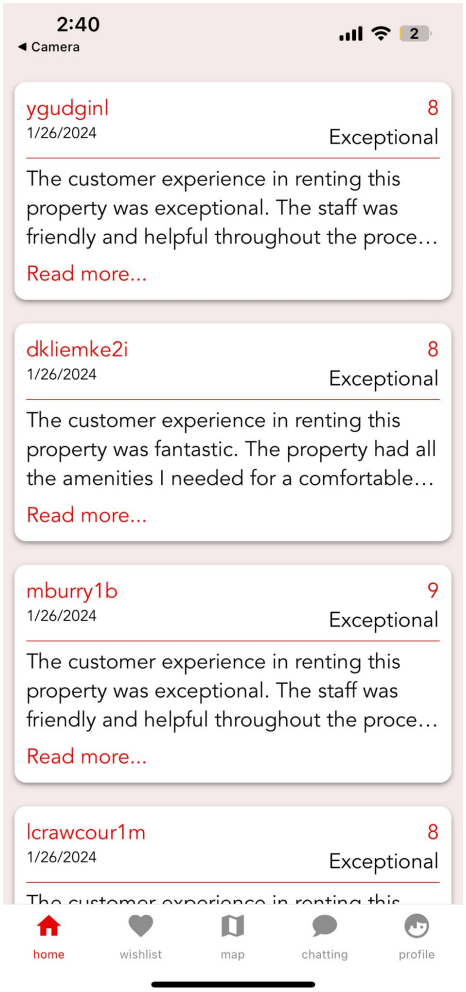


Figure 23: Reviews Screen

- **owner profile Screen**: this page is to display the details about the owner of the property

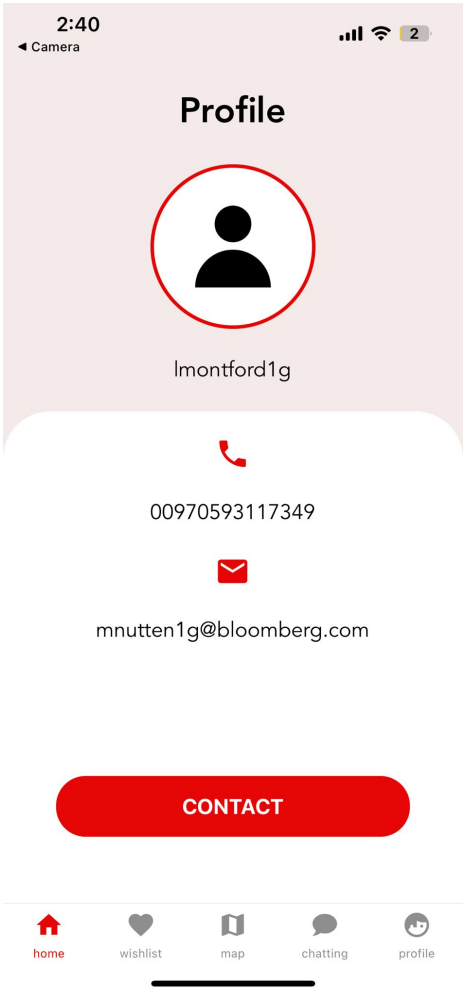


Figure 24: owner profile Screen

- **Wishlist Screen**: this screen is to view your own likes that was added to the wishlist

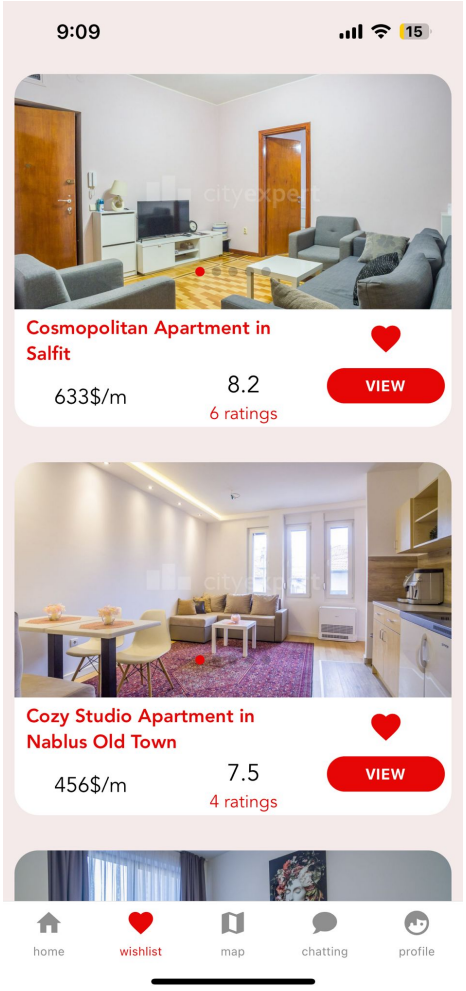


Figure 25: wishlist Screen

- **Pick Map Screen**: this screen is to choose the city and view the properties available on maps

Figure 26: Pick Map Screen

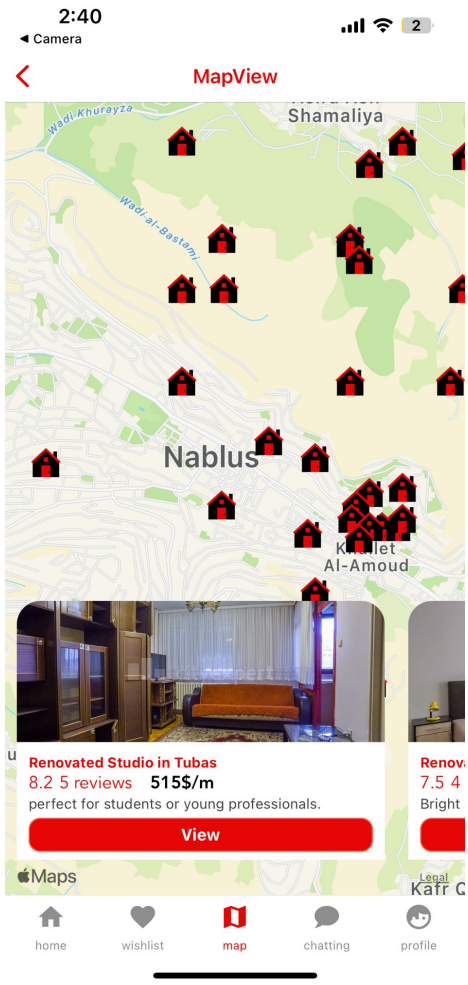


Figure 27: Map Screen

- **Chat Screen**: view previous chats and chat screen

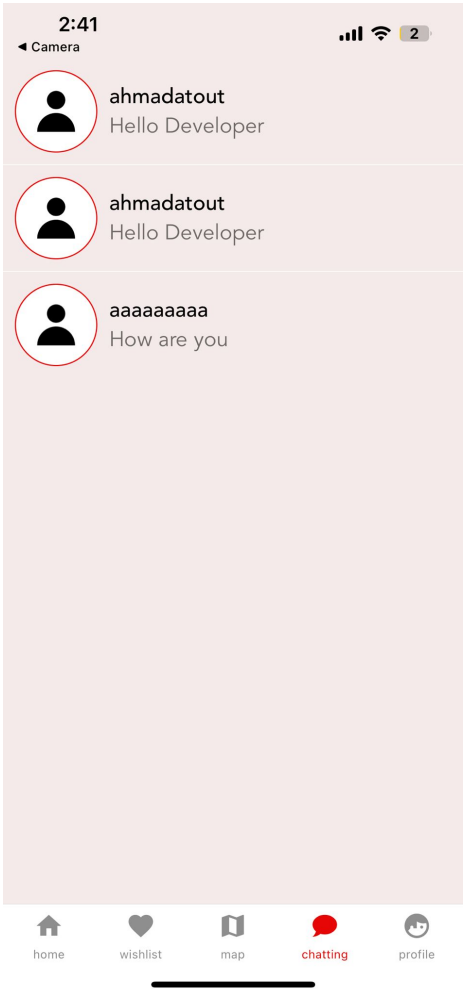


Figure 28: Chat Screen

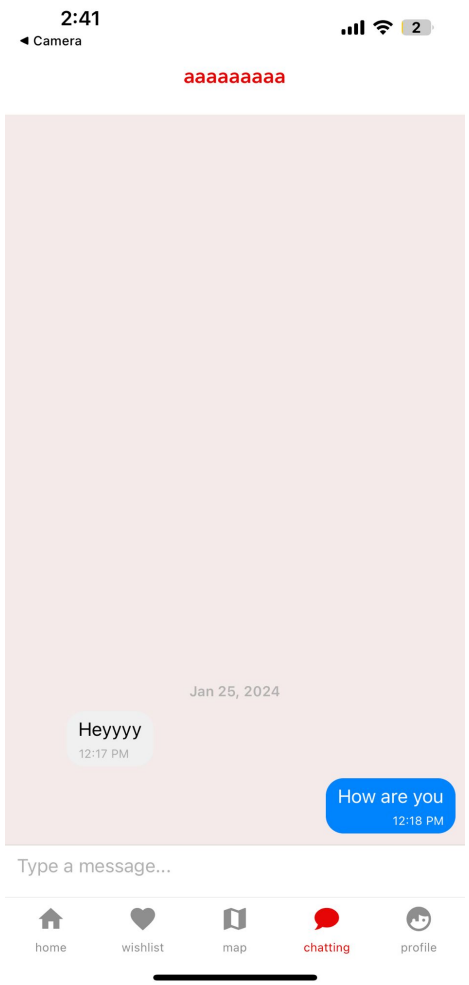


Figure 29: Chat Screen

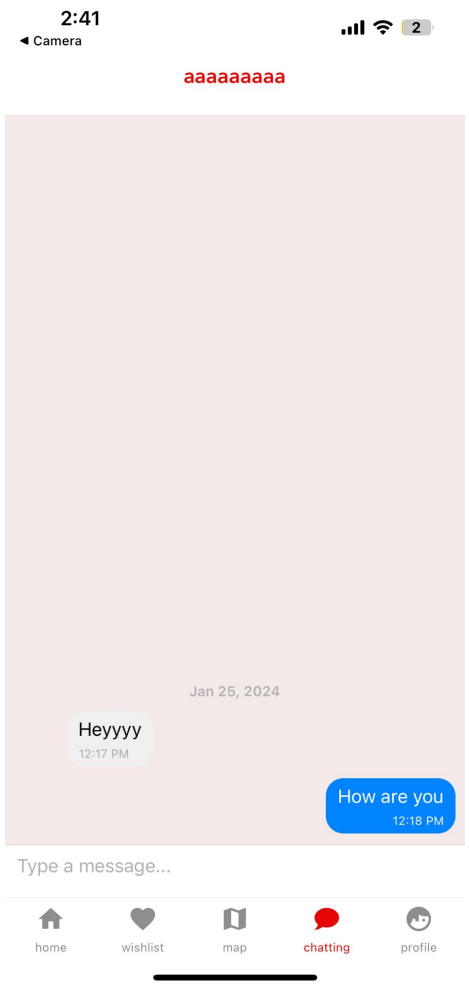


Figure 30: Chat Screen

- **Profile Screen**: this screen is to view your own profile with multiple options you can perform

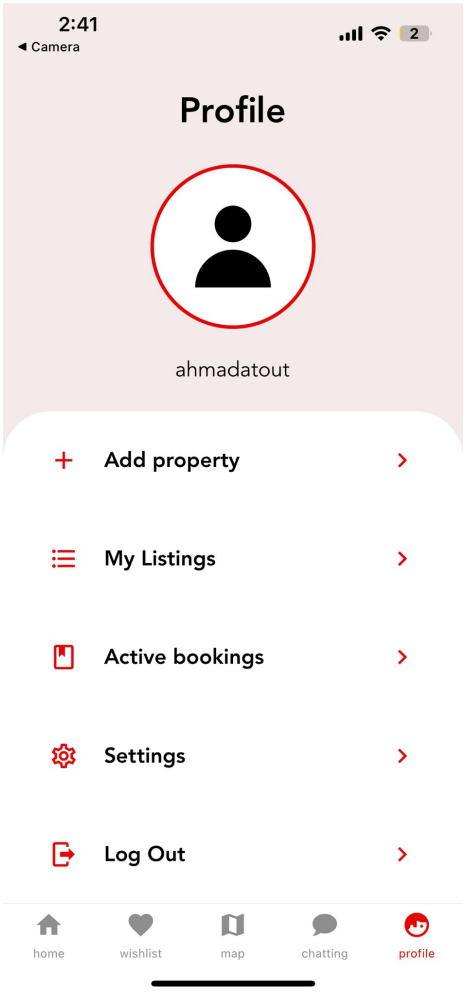


Figure 31: Profile Screen

- **Add property Screens**: those screens are to let you make a new post of your property

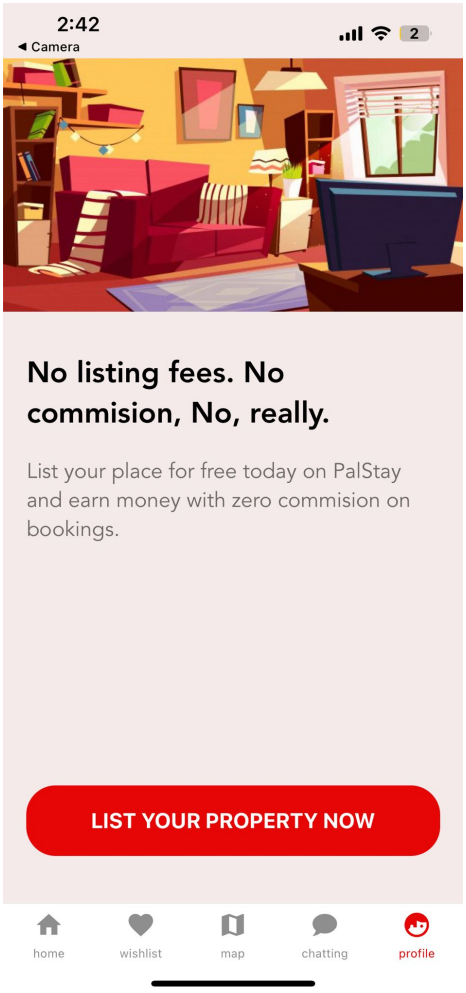


Figure 32: Add property Screens

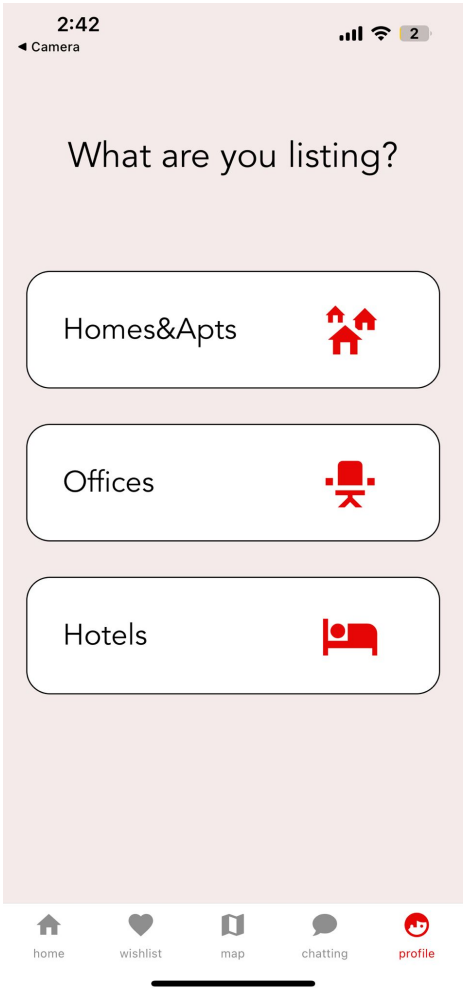


Figure 33: Add property Screens

Figure 34: Add property Screens

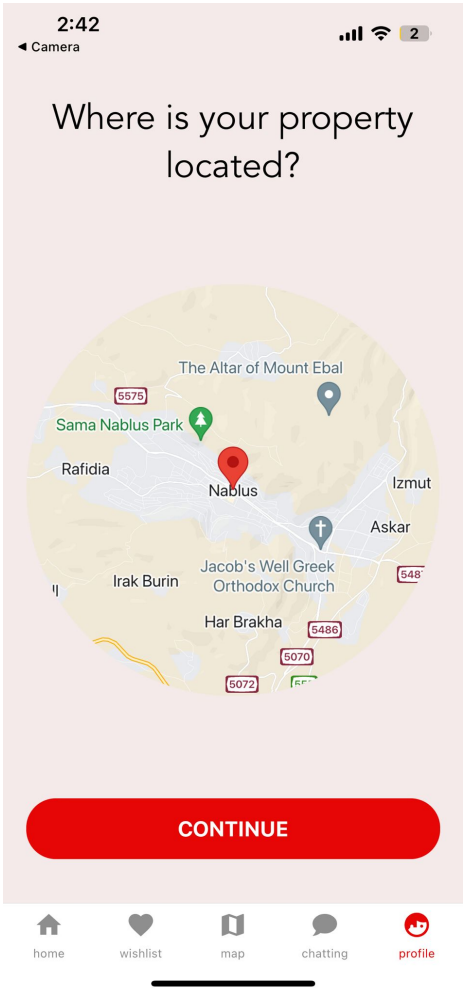


Figure 35: Add property Screens

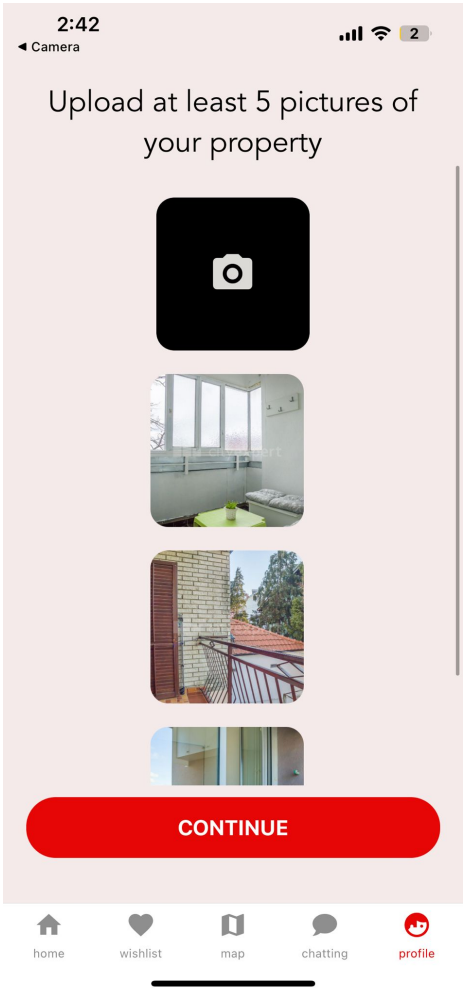


Figure 36: Add property Screens

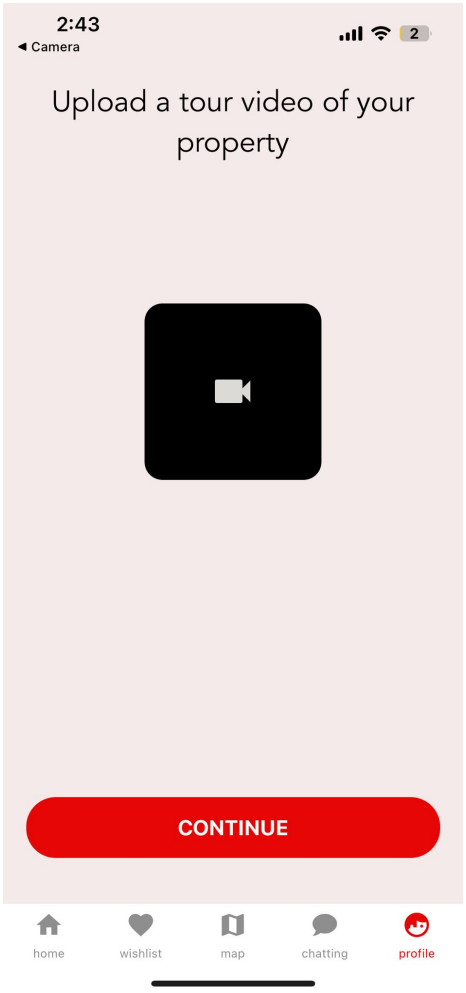


Figure 37: Add property Screens

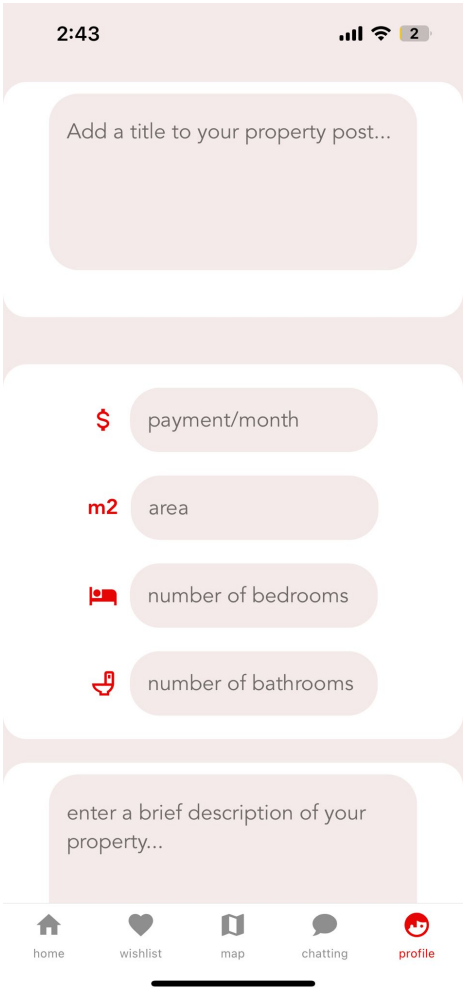


Figure 38: Add property Screens

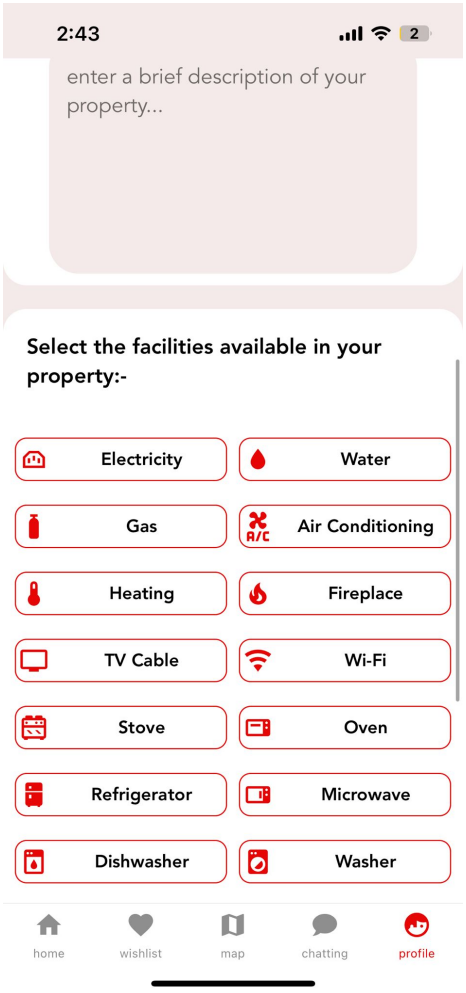


Figure 39: Add property Screens

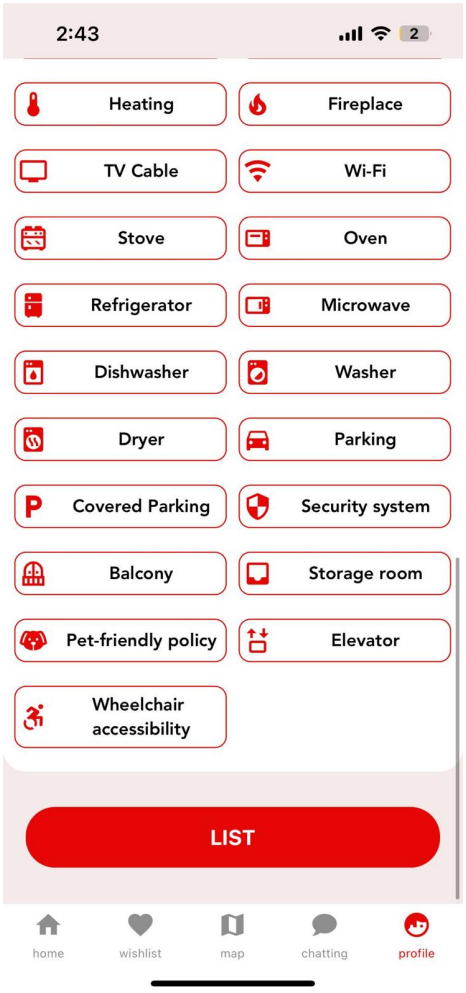


Figure 40: Add property Screens

- **My Active bookings Screen**: this screen is to display your active bookings own, the active booking has 3 status, first status in (upcoming) which means that your booking duration has not started yet, the second status is (active) which means your booking is running now, the third is (finished) which means that your stay duration has ended and then new button appears for you to give a review

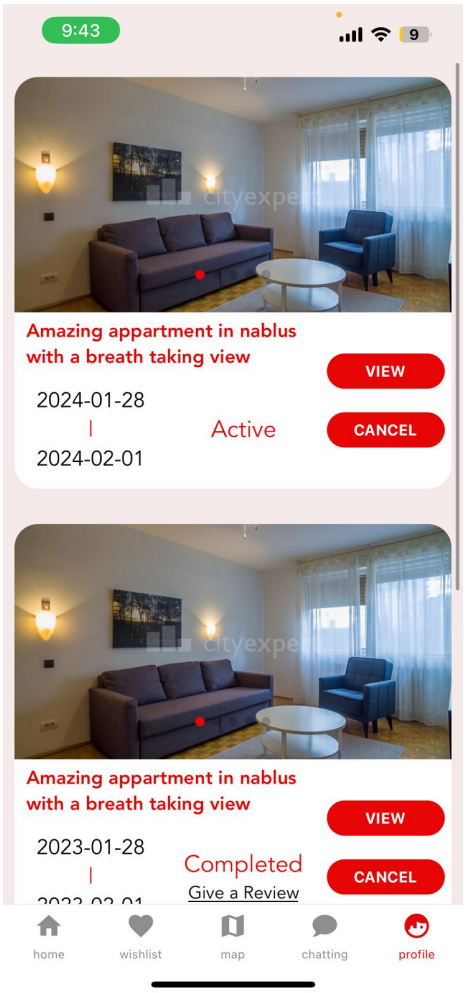


Figure 41: My Active bookings Screen

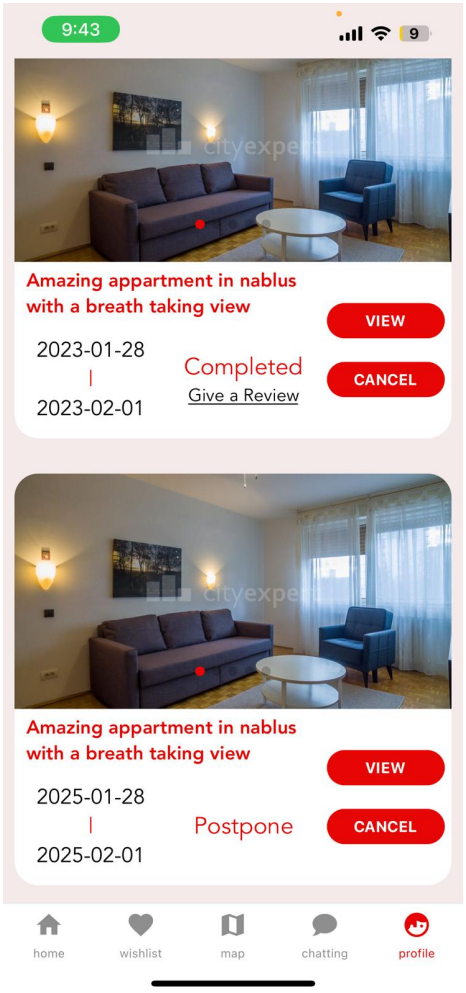


Figure 42: My Active bookings Screen

- **give reviews Screen**: this screen is to give a review after you finished your stay in the property

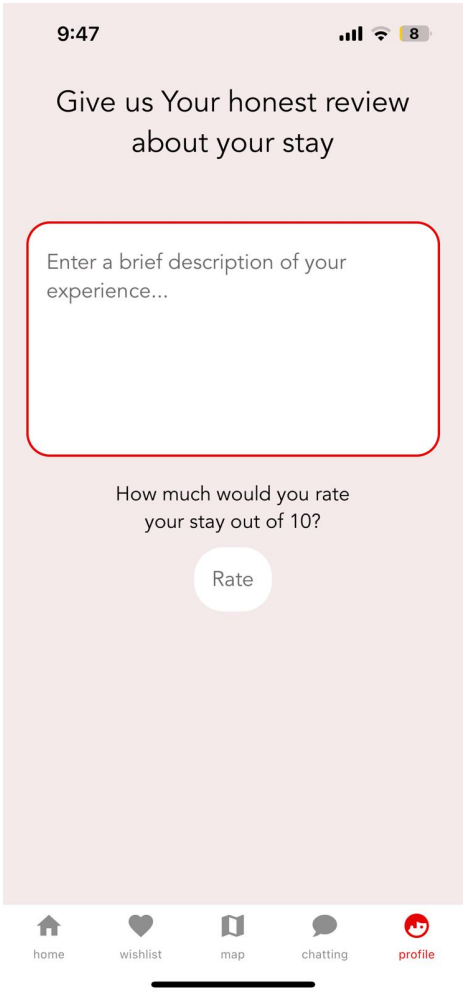


Figure 43: give reviews Screen

- **My Listings Screen**: this screen is to display the user's own listings

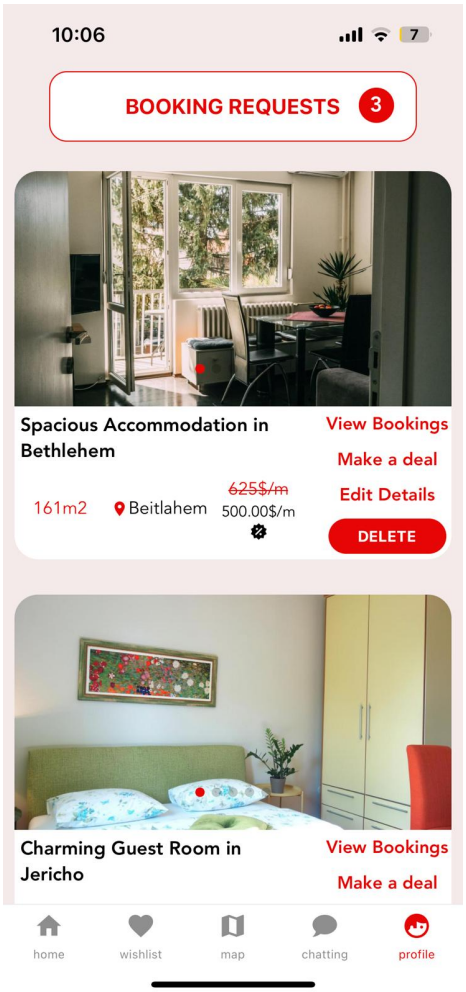


Figure 44: My Listings Screen

- **Booking Requests Screen**: this screen is to display the requests for booking your properties.

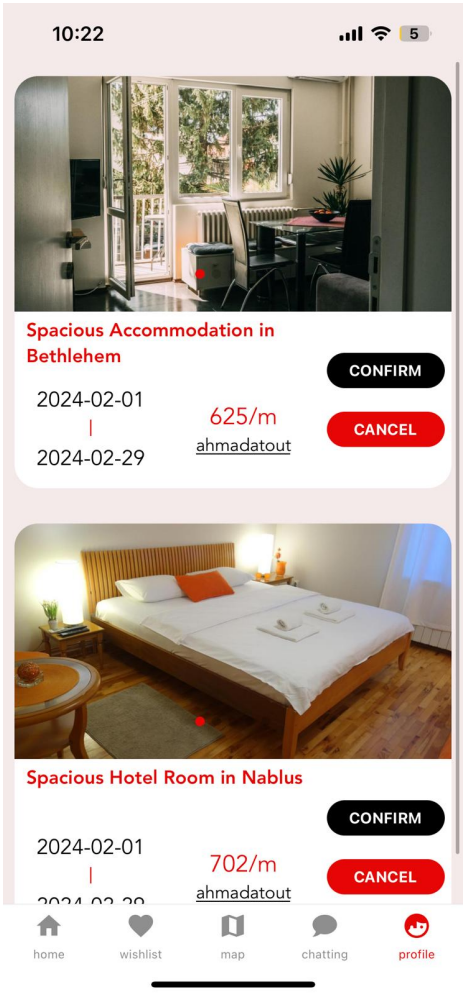


Figure 45: Booking Requests Screen

- **Make Offer Screen**:this screen is to let the user make an offer of the selected property

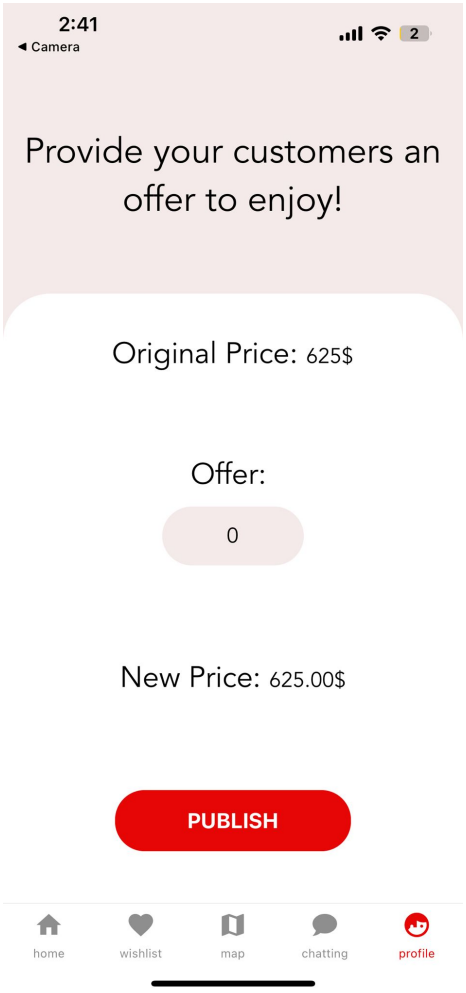


Figure 46: Make Offer Screen

- **Property active bookings screen**: this screen is to display the property's active bookings by different users

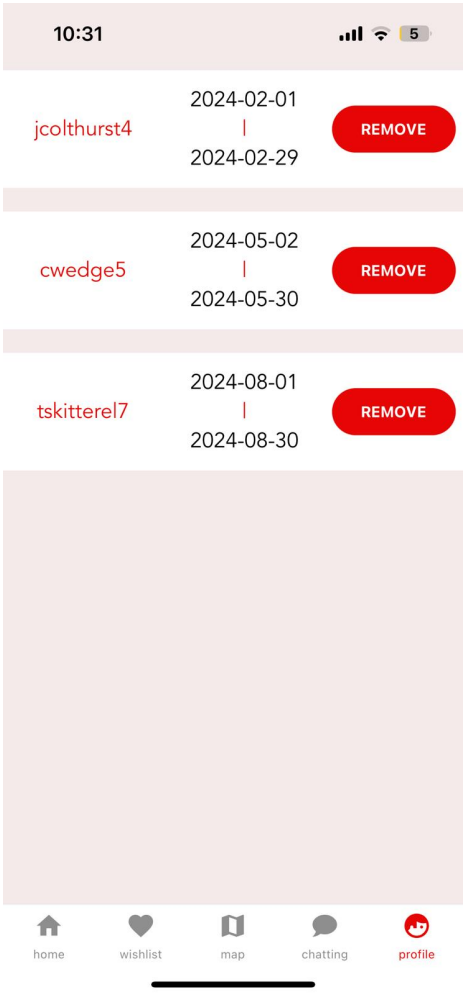


Figure 47: Property active bookings screen

- **Edit Property details screen**: this screen is to edit your listed property details

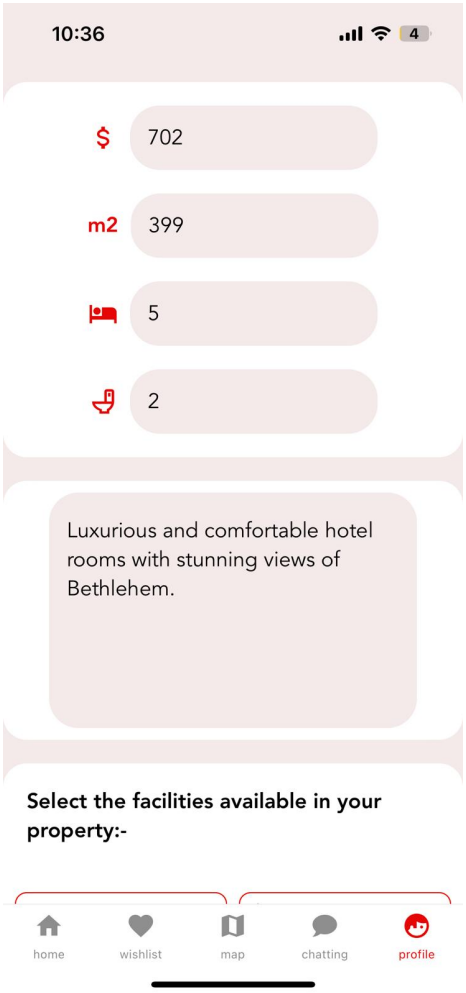


Figure 48: Edit Property details screen

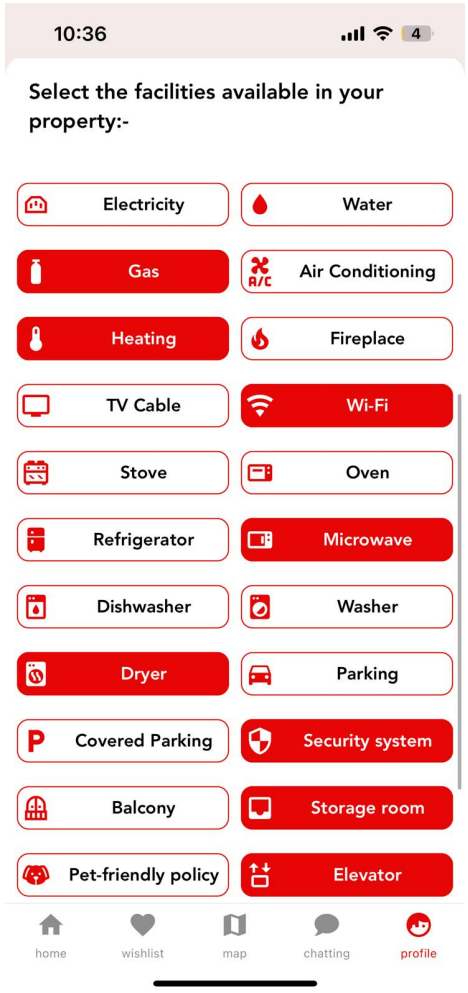


Figure 49: Edit Property details screen

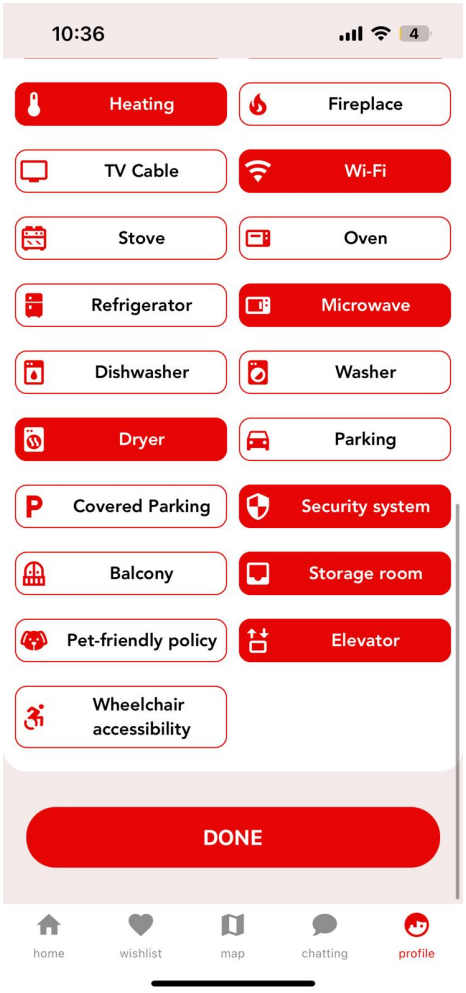


Figure 50: Edit Property details screen

4.3.5 Web Application

- **Login Admin Screen**: this screen is for the admin to login

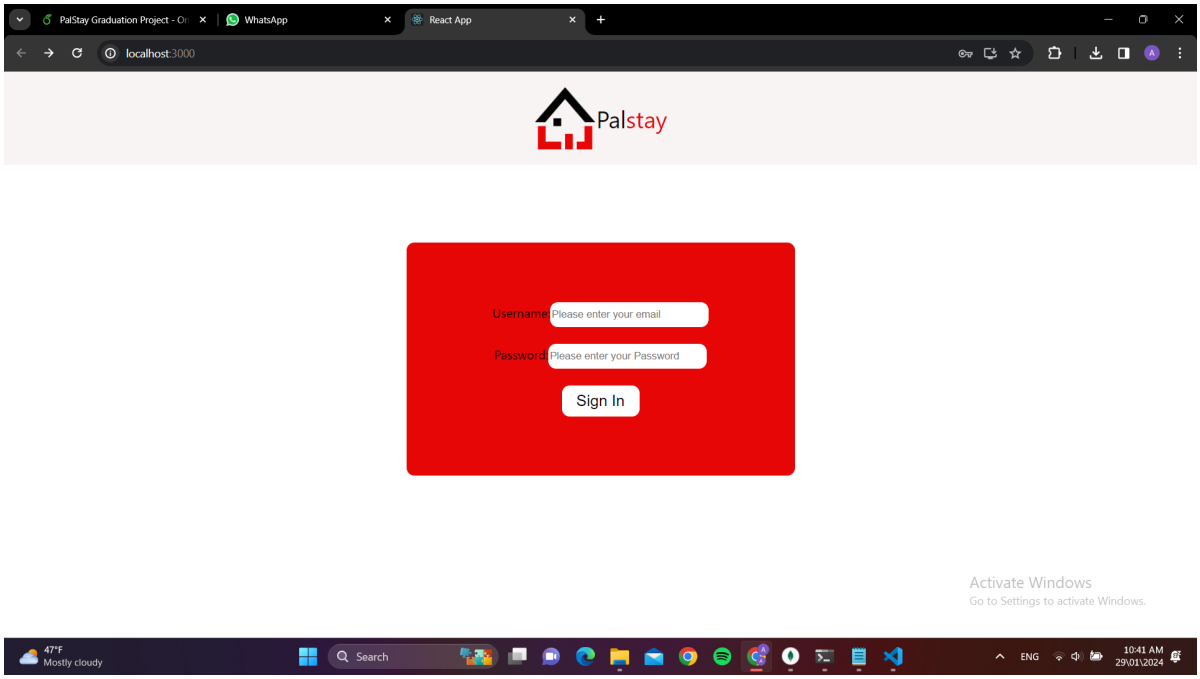


Figure 51: Admin Screen

- **Admin Panel Screen**: this screen is to display the admin Panel

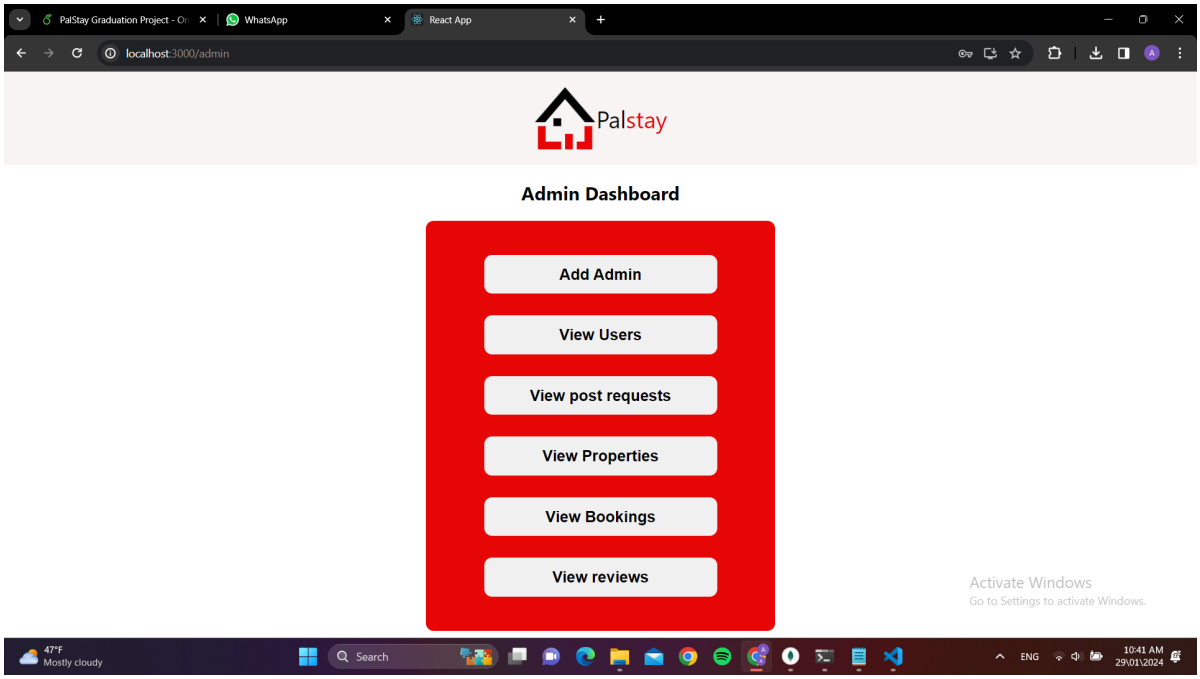


Figure 52: Admin Panel Screen

- **Add new Admin Screen**:this screen is to add anew Admin

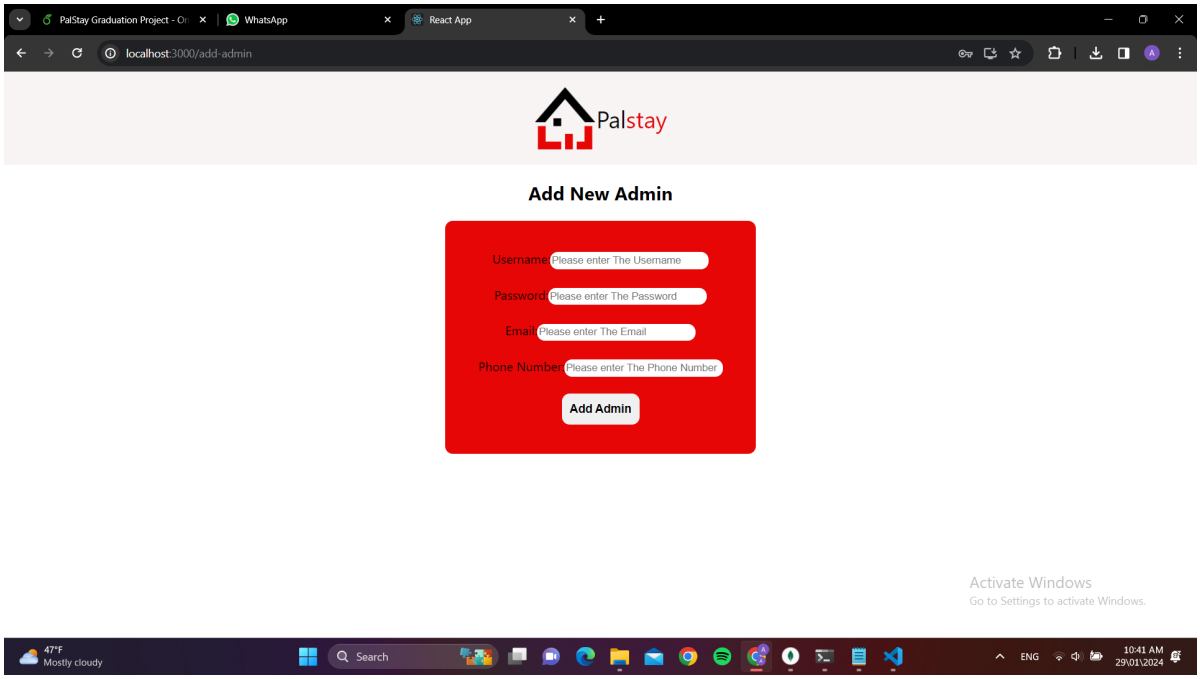


Figure 53: Add new Admin Screen

- **Admin View Users Screen**: this screen is for the admin to view all the users of the application and make filters and search for certain users

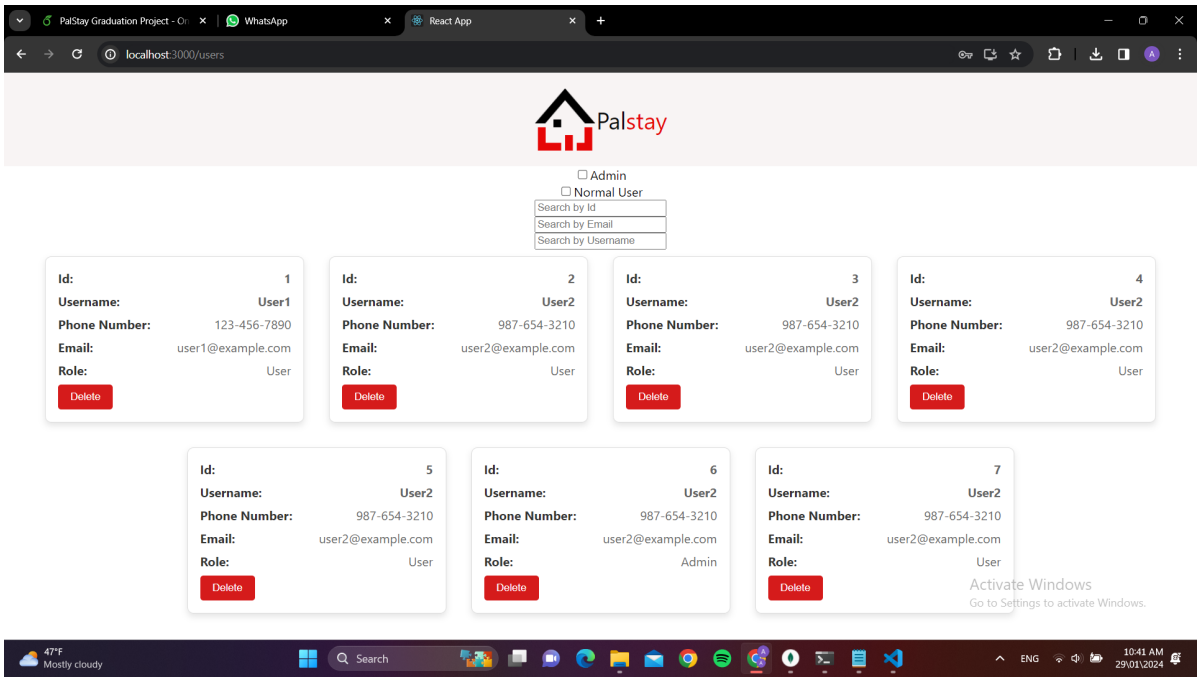


Figure 54: Admin View Users Screen

- **Admin View post requests Screen**: =this screen is for the admin to view post requests and accept or deny them

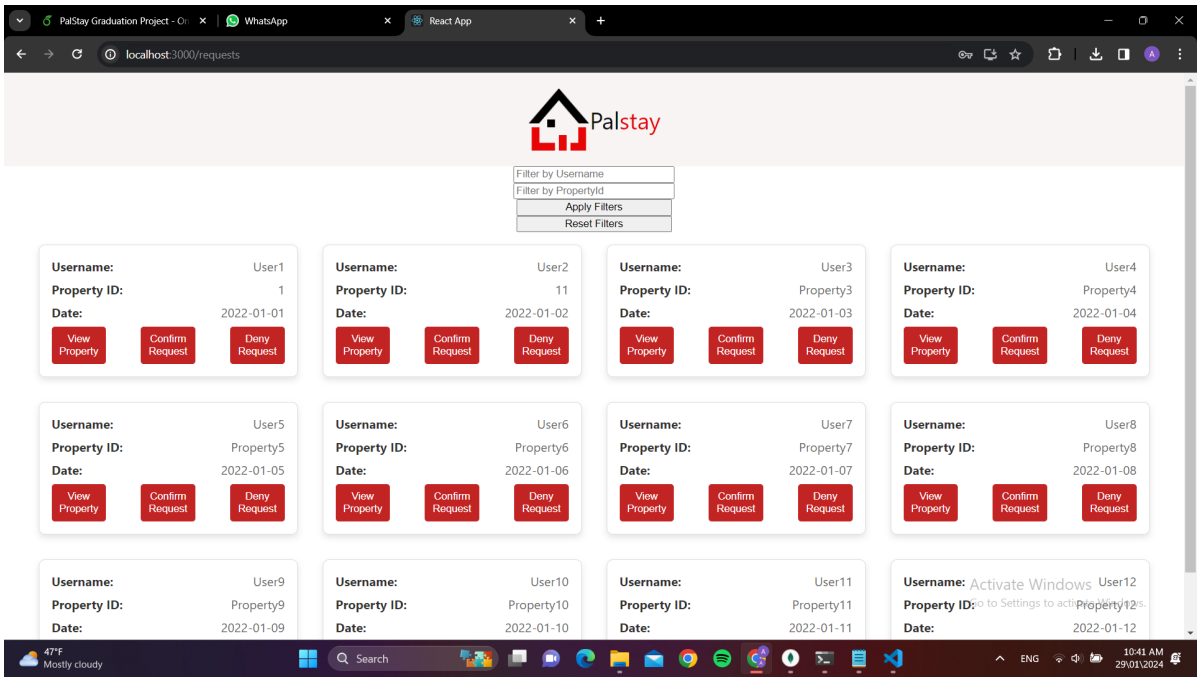


Figure 55: Admin View post requests Screen

- **Admin View Properties**:this screen is for the admin to view and search between the properties listed on the application

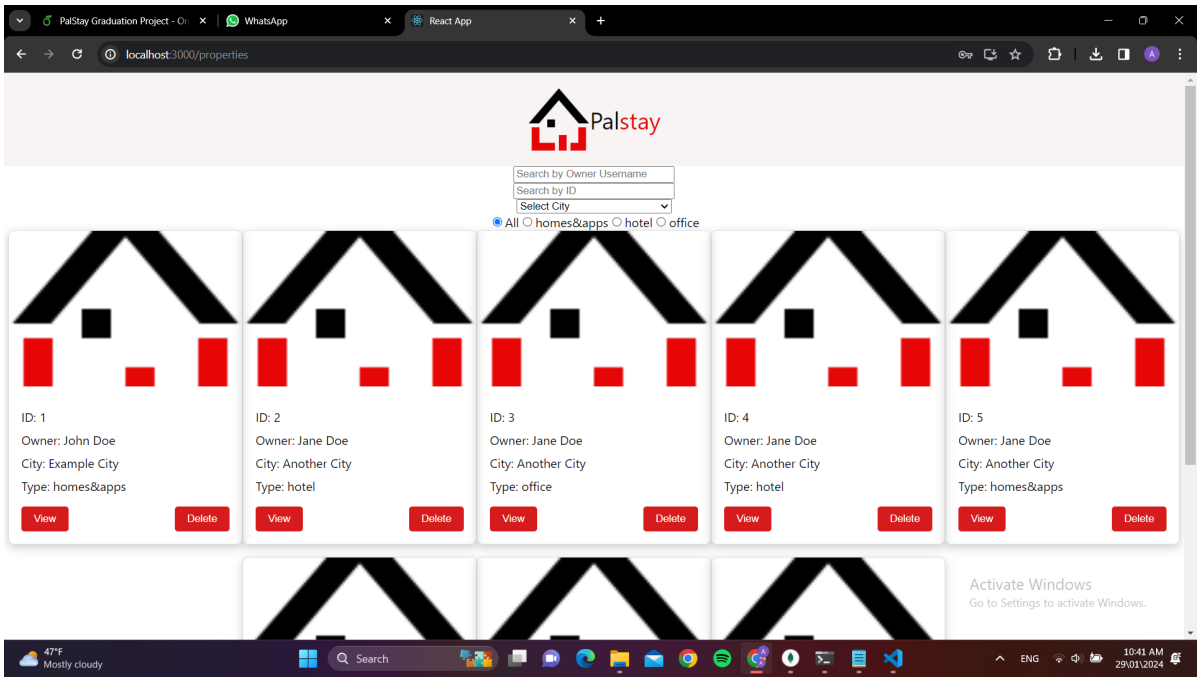


Figure 56: Admin View Properties

- **Admin view bookings Screen**: this screen is for the admin to view all the bookings

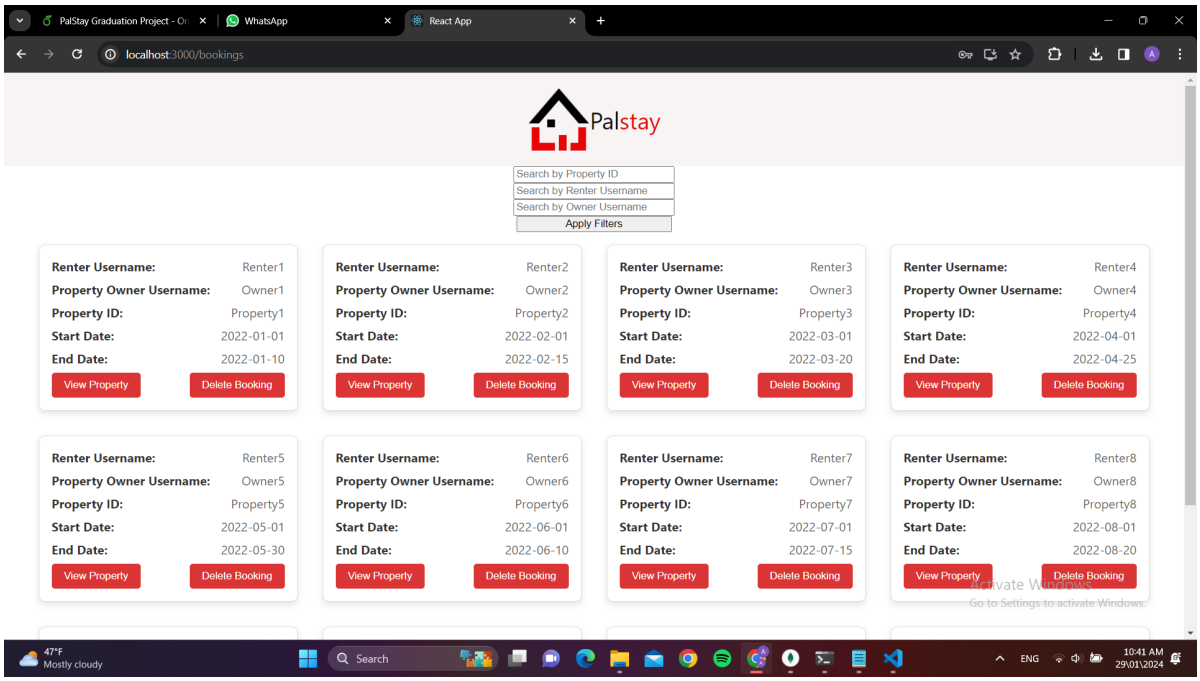


Figure 57: Admin view bookings

- **Admin View Reviews Screen**: this screen is for the admin to View all the reviews

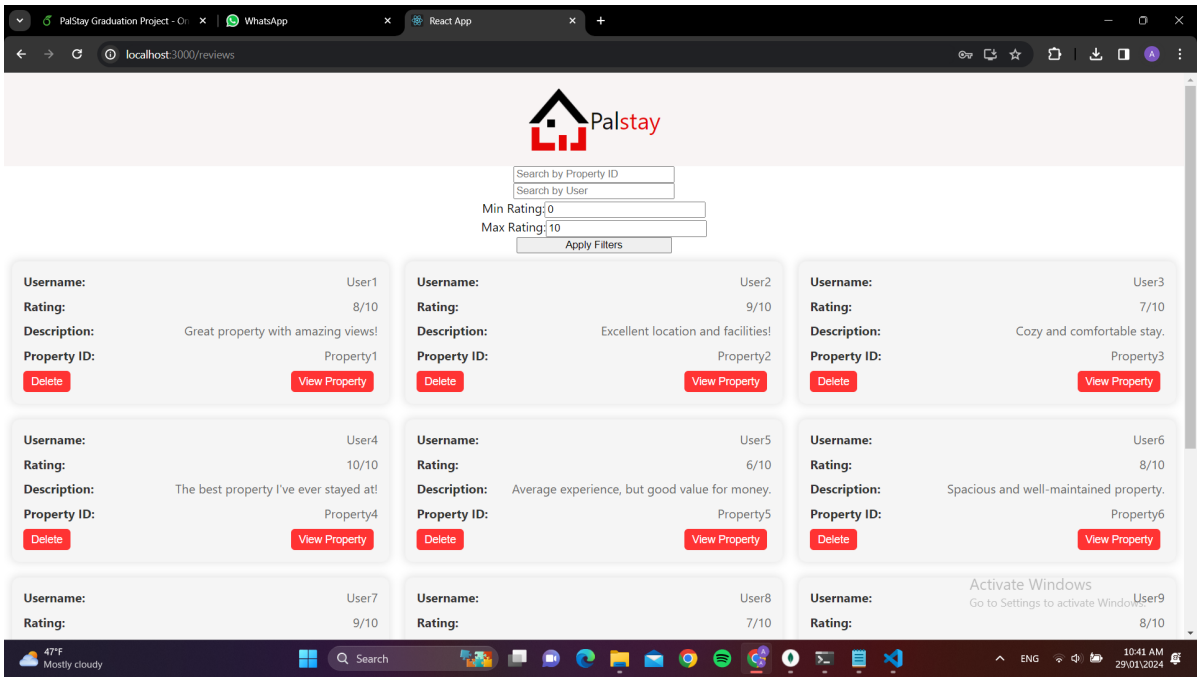


Figure 58: Admin View Reviews Screen

- **Admin View Property Details Screen**: this screen is for the admin to View Property Deatils

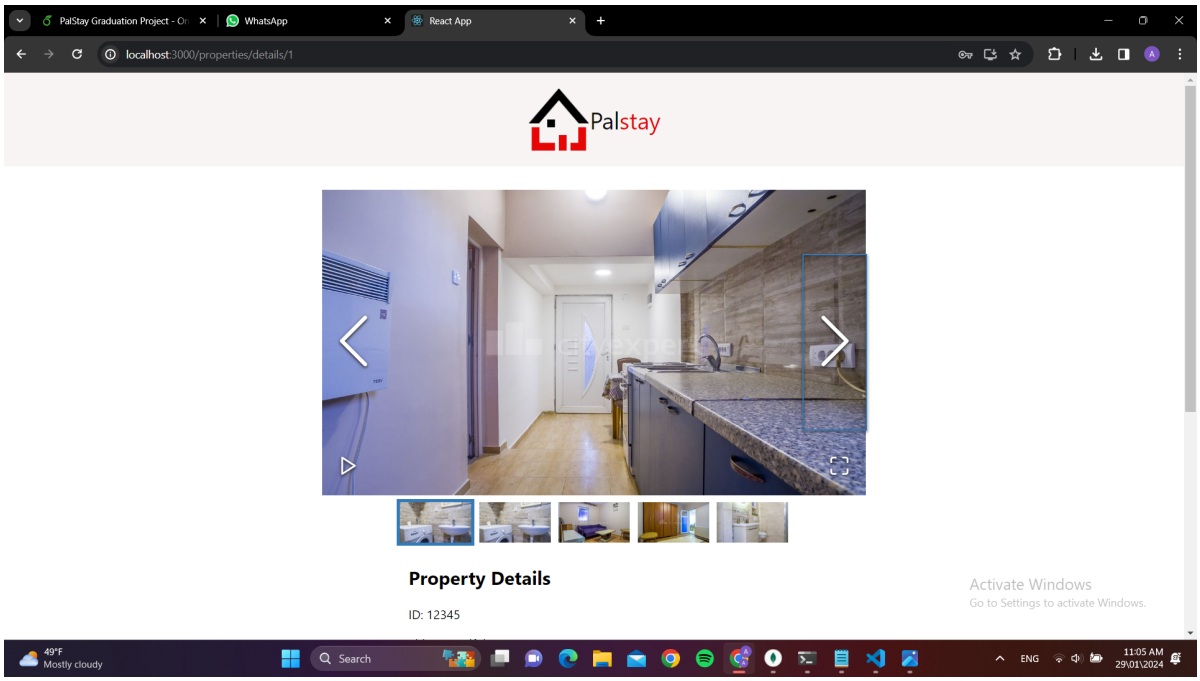


Figure 59: Admin View Property Details Screen

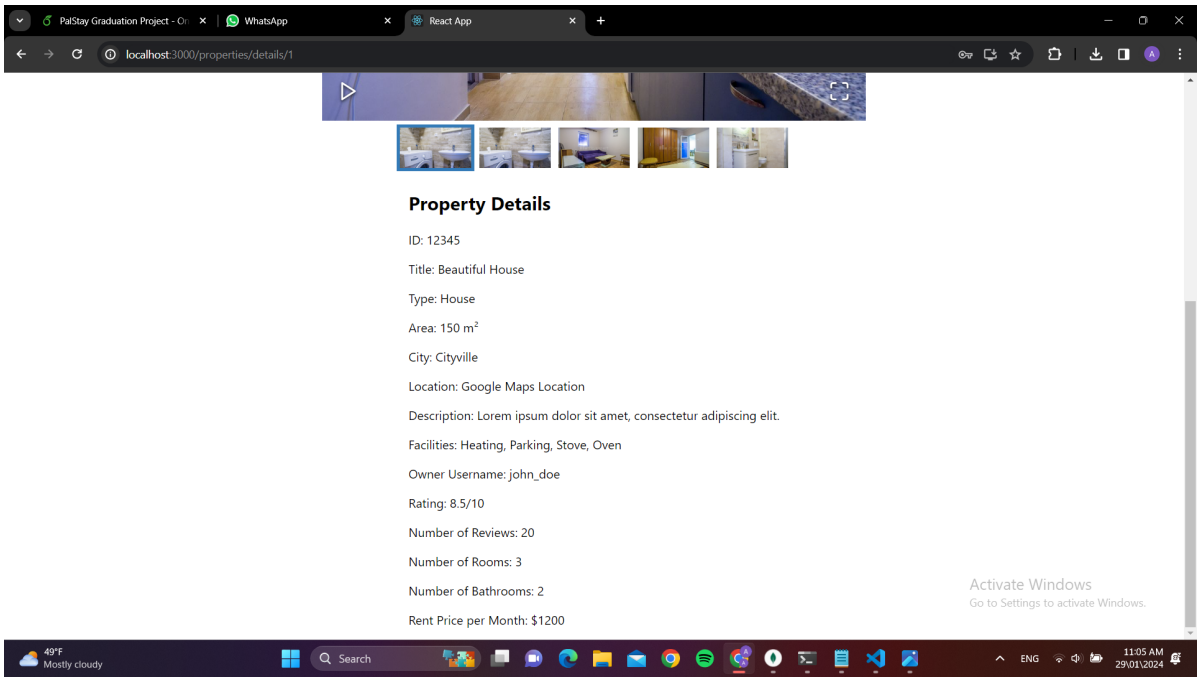


Figure 60: Admin View Property Details Screen

5 Results and Analysis

The final result is a user-friendly application with many features built with many tools and programming languages.

To help users find a place to stay easily and list their properties to make it easy for people to find them easily including so many features to make the application desired to use other that other applications available

6 Discussion

6.1 Testing

The mobile application was reviewed and tested constantly with the supervisor Dr. Emad Natsheh, as well as it was given to a small group of family and friends and gave their experience and helpful advice through out the development process.

6.2 Learning Curve

At first building the application was very hard as there was three new languages needed to be learnt in addition to another architectural style which separates client side and serve-side. So, the first few months were very unproductive from the application development. But, with a great set of documents, plenty of online tutorials, and a community of supportive people, the learning mission was not as challenging or steep as it initially seemed. The following months after that were very productive as the application was built frontend and backend separately. After that the connection process took some time to get used to, but the following months were much easier as we got the hang of the work.

7 Conclusions and Recommendation

7.1 Things we learned

· how to build mobile application using react native and node js, building end points, using recommendation system with machine learning

· How to set up expo to render the application without using emulator .

· How to solve errors and installing packages in the right way.

· how to handle designing and implementing big software projects in a limited amount of time and using technologies I have not used before

7.2 Future Work

· Implement every feature available on the application to the website so that we have a complete website with all the features.

· Include all the vilages in the westbank instead of only the major cities and include the Gaza Strip.

· include a chatbot that could listen to the user prefrences and recommend the best fits.

· list properties for selling not only for renting.

8 References

References

[1] what is collaborative filtering. https://realpython.com/build-recommendation-engine-collaborative-filtering/#:~:text=beautiful%20%2B%20Pythonic%20code.-,What%20Is%20Collaborative%20Filtering%3F,similar%20to%20a%20particular%20user., 2017. Accessed: date-of-access.

[2] Node.js ecommerce: Benefits, architecture, and best practices. https://cloudinary.com/guides/e-commerce-platform/node-js-ecommerce-benefits-architecture-and-best-practices: :text=js

[3] React native vs flutter, cross-platforms mobile application frameworks. https://www.theseus.fi/handle/10024/146232, 2020. Accessed: date-of-access.

[4] e-commerce reccommendation engine. https://medium.com/web-mining-is688-spring-2021/e-commerce-recommendation-engine-with-collaborative-filtering-cb19cd542c18, 2021. Accessed: date-of-access.

[5] How can you use mongodb for e-commerce? https://fabric.inc/blog/commerce/mongodb-ecommerce, 2021.

[6] Mepalestine. https://palestine.io/en, 2024. Accessed: date-of-access.

[7] Fernando Doglio. Pro rest api development with node. js. *Apress*, 2015.

[8] Subodh Kumar. Impact of technology on various aspects of human life. *Research gate*, 15:123–456, 2021.

[9] Ha Van. "building a universal application with react and react native. 2020.