

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



An-Najah National University
Faculty of Engineering and Information Technology
Computer Engineering Department

Graduation Project 1
Travel Nest



Ahmad Shaer & Arwa Maroof

Supervisor: **Dr. Sofyan Samara**

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

24-January-2024

Acknowledgment

We would first want to express our sincere appreciation to the Almighty. Without His blessings that led us every step of the way our route, leading to the successful completion of this project.

We are really grateful to our parents and other loved ones. We have been continuously inspired and uplifted by your constant encouragement along our journey. We have gained strength and stability from your kind and supportive remarks, which have created a solid foundation.

And to our supervisor, Dr. Sofyan Samara, we express our gratitude for your guidance and your significant suggestions. The precise insights, insightful remarks, provoking debates, and useful feedback you have provided have greatly improved our project's capability.

We would especially want to thank An-Najah National University's Computer Engineering Department for their continuous support.

Disclaimer

The team is comprised of two computer engineering students, Ahmad Shaer and Arwa Maroof, currently pursuing their education at An-Najah University. Collaborated on the completion of this paper. The authors acknowledge full responsibility for any errors identified within. The ideas, suggestions, findings, and opinions expressed in this paper solely belong to the authors and do not reflect the viewpoint of An-Najah National University.

Table of Contents

Acknowledgment	2
Disclaimer	3
Abstract	7
1 Introduction.....	8
2 Constraints, Standards and Earlier course work	9
3 literature Review	10
4 Methodology	11
4.1 Technologies:	11
4.1.1 Database:.....	11
4.1.2 Server side:.....	12
4.1.3 UI design:	13
4.1.4 Application Architecture:.....	14
4.1.5 Agile Methodology:	16
4.1.6 Testing:	17
4.1.7 Version Control:.....	19
4.2 Features:	21
4.2.1 Login / Signup Pages:	21
4.2.2 Main Page /Home page	23
4.2.3 Profile Page:	24
4.2.4 Hotel Page:.....	25
4.2.5 Booking page:	26
4.2.6 Favorites Page:.....	29
4.2.7 Search Page:.....	30
4.2.8 User Callender:	34
4.2.9 User bookings:	36
4.2.10 Notification system:	37
4.2.11 Login:.....	38
4.2.12 Dashboard:	39
4.2.13 Hotels Page:	40

4.2.14	Users page:	41
4.2.15	Add Hotel Page:	41
4.2.16	Bookings Page:	42
4.2.17	Rooms Page:	42
5	Conclusion/ Recommendations and Future Works.....	44
6	References.....	45

Table of Figures:

Figure 1:	Application Architecure	14
Figure 2:	MVC Model	15
Figure 3:	Agile Methodology.....	17
Figure 4:	Testing	18
Figure 5:	Version Control_1	20
Figure 6:	Version Control_2	20
Figure 7:	Login Page.....	21
Figure 8:	Sign Up page	22
Figure 9:	Home Page	23
Figure 10:	Profile Page	24
Figure 11:	Hotel Page	25
Figure 12:	Booking Page	27
Figure 13:	Booking Page (specifying the dates)	28
Figure 14:	Favorites Page	29
Figure 15:	Search Page 1	30
Figure 16:	Search Page Filter.....	31
Figure 17:	Search Page Filter (Price).....	32
Figure 18:	Search Page Filter (Rating)	33
Figure 19:	User Calendar	34
Figure 20:	User Calander (Extra Info).....	35
Figure 21:	User Bookings	36
Figure 22:	Notification System.....	37
Figure 23:	web Login page	38
Figure 24:	web Login Page 2	38

Figure 25:Admin Dashboard.....	39
Figure 26:Hotel Admin Dashboard.....	39
Figure 27:Hotels Page.....	40
Figure 28: Users Page	41
Figure 29:Add Hotel Page	41
Figure 30:Bookings.....	42
Figure 31:Rooms Page 1	42
Figure 32:Rooms Page 2.....	43
Figure 33:Rooms Page 3.....	43

Abstract

During the Pandemic that started in the end of 2019 and continued for about three years most of people started relaying on online services to buy their grocery or to do business or to work. So, many more and more companies started to creating an online application for their services and started promoting their application through social media like Instagram, Facebook, Twitter etc. and most of the local business followed that pattern to keep up to the change.

Our application is an online booking mobile application. That gives users the chance to visit the beautiful city of Palestine and its many landmarks and connect them with the closest hotel in that area in which they don't have to spend much time on traveling and they can search for the closest hotel for their destination and also choose the range of prices they would like to spend and in addition to a recommendation system to give them a personalized experience.

The hotel manager can edit the information and post images and add offers and manage the reservation that he gets. The application provides a feedback report to give the hotel manager an easy way to keep up with their customers ratings.

The idea of Online booking has been done before but it has never been implemented in Palestine as far as we know and a booking application that contains almost all of the hotels available in Palestine in one site and a Recommendation system based on number of factors such as rating and pricing and location visited.

1 Introduction

The "Travel Nest" project is a network-based application that was created using Node.js and Flutter. The name of the project is appropriate given that it offers the ability to book residences, hotels, flats, and other types of places to stay.

This system was designed to take the role of the tedious manual tasks involved in locating the right hotel, exploring the area, and finally reserving rooms. The utilization of database-driven applications that successfully reduce the issues that arise from manual processes, such as erroneous recordkeeping that delays corporate operations.

In addition to being able to locate a facility precisely, these features would enable customers to look up the rooms of any hotel, apartment, house, etc. and determine in advance whether or not the facility they are browsing meets their needs. They also enable customers to view prices so they won't be uneasy when they discover that the rooms are worth significantly more than what they are able to afford, and they can pay the owners at the time of checkout.

Our application also shows the user his scheduled booking using a Callender that display the details of the booking and the dates and also offers the user a way to keep track of all his old bookings and current ones using a separate page.

2 Constraints, Standards and Earlier course work

One of the difficulties we encountered was learning a new language. We are aware that this will be difficult at first, but with practice, it will get easier. Since we had no prior experience making mobile apps, learning was necessary to finish this project, and putting what we learned into practice was even more important. It was challenging, particularly because there wasn't enough time. The absence of courses and occasionally not having the necessary skills to study a certain subject, particularly how to work with commonly used tools like Android Studio and certain languages like Flutter, as well as the usage of Node.js for backend and MongoDB.

Then we revisited the courses we studied in the previous years, such as database, web development, and critical thinking. Our strategy for developing in this The project started with gathering the most data and necessities needed to finish it. Next, we constructed an initial database, which would allow for further growth or modification as the design progressed according to our desired layout.

First, we began the administration's part of the process, known as the design phase. It was only an initial design. Next, from the owner's perspective. We check to see if the administrator's component has to be included after every addition. We began with the user side when it was finished. Furthermore, we aim to determine whether the admin's part needs to be added, modified, or deleted. Keeping an eye on the smallest aspects to ensure a seamless and user-friendly program, the administrator manages the owners, users, and all of the features it offers.

3 literature Review

With the development of information technology, online travel agency has become an important information and communication source in the hospitality industry.

The previous studies assessing the online hotel decision making behaviors, however, mainly focused on behavioral intentions as well as identifying factors that directly influence the transactional behaviors, which rely on a static approach rather than employing the holistic viewpoints. Thus, this research adopted the choice-set model as a theoretical lens to explore online hotel booking behaviors by using multiple methods (i.e., observation and survey methods). The findings of this research shed light on the dynamic patterns of online hotel decision making process and identify important factors (i.e., internal and external information sources) across sequential stages of the choice-set model. Therefore, this research provides useful implications to online hospitality marketers.[2]

In recent years, the travel and tourism industry has begun to develop internet applications more extensively. many travel service suppliers, such as hotels and airline companies, already use the internet as a direct channel for customer reservations/Bookings and have improved their business growth. It becomes more and more difficult for traditional travel agents to compete with web sites that offer online booking possibilities. One of the main reasons for the increasing number of people booking online is that online experience has much greater potential in visualizing travel destinations, thus travel agents using online booking are likely to grow and expand their business. Tourists still used professional services and advice offered by travel agencies. [7]

A country's performance depends on its enterprises, which certainly includes its hospitality and tourism industry. The hotel industry is among the most critical departments of the hospitality and tourism industry. Globally, there is a rising demand for quality hotel services. [5]

The manual approach to reserving hotel services poses several challenges, including limited information for clients about hotels in their area, potential booking of overly expensive or unsuitable accommodations, and delays in retrieving guest information. This can negatively impact a hotel's business. [4]

The main objective of this project is to develop a web application that addresses these issues, allowing clients to access hotel information and make bookings online using their mobile phones, tablets, or computers, thereby improving the efficiency and convenience of the reservation process.

4 Methodology

4.1 Technologies:

To create mobile apps, a variety of tools and programming languages are needed for mobile application development. This chapter will include a list of the tools, programming languages, and technologies that were used to build our project.

4.1.1 Database:

We choose to use a NoSQL database, MongoDB, for storing and managing our application data because of its many benefits that meet our needs. MongoDB is a popular document-oriented database that stores information in BSON (Binary JSON), a robust format like JSON. Because it enables us to store data with different forms, this structure is very useful for our application because it is perfect for handling a variety of information.

The dynamic and changing data requirements of our application are well suited for MongoDB because of its prowess in managing substantial amounts of unstructured or semi-structured data. The essential flexibility of the data model in our application is reflected in its schema-less architecture, which makes it simple to react to changes in data structure. The main benefit of MongoDB is its intuitive handling of relationships and nested structures. This is important because in our application, entities may have intricate interactions, and we need to be able to express these relationships effectively. MongoDB is a document-based database. MongoDB's support for horizontal scalability is essential for accommodating the expected growth in data size with an increasing number of users. It employs sharding, enabling seamless distribution of data across multiple servers, ensuring high availability and performance even as the dataset expands.

In terms of analytics and reporting, MongoDB offers powerful aggregation framework and indexing capabilities. These features empower us to perform complex queries, aggregations, and analyses efficiently, supporting our application's requirement to provide the hotel owner with insightful business analytics.

In summary, MongoDB, with its flexible schema design, scalability, and powerful querying capabilities, emerged as the optimal choice for our application. As our user base grows and data complexity increases, MongoDB provides a robust foundation to meet the evolving demands of our application.

4.1.2 Server side:

We decided to use the robust and open-source JavaScript runtime environment Node.js for our application's server-side development. Node.js has an event-driven, non-blocking design that makes it ideal for developing scalable network applications. In this case, the server-side functionality was handled by Node.js, which allowed for easy HTTP request connection with our MongoDB database. Given its effectiveness in managing asynchronous processes and real-time applications, Node.js is a great fit for our application. Node.js satisfies these needs precisely because our server-side processes entail interactions with a NoSQL database like MongoDB, which frequently depends on asynchronous operations.

Node.js's capacity to manage several concurrent connections with minimal overhead is one of its main features. Ensuring maximum performance and responsiveness is vital for our program, especially as our user base grows.

The dynamic community and package manager npm (Node Package Manager), which gives users access to a vast array of libraries and modules, are other advantages of Node.js. We can effectively manage our codebase and use different function modules for data representation thanks to the broad library support, which guarantees flexibility and extensibility in our server-side implementation. We can also use JavaScript on the client and server sides with Node.js, which helps to maintain consistency throughout our software. This single language strategy improves code maintainability, lowers the overhead of context change, and streamlines development.

In conclusion, Node.js seems to be a great option for our server-side requirements, including scalability, effectiveness in managing asynchronous processes, and a robust library ecosystem. Its full stack compatibility with JavaScript complements the design of our program and makes our server-side code more efficient and manageable overall.

4.1.3 UI design:

In order to create a captivating and intuitive hotel reservation application, I investigated the domain of modern mobile application development norms. I choose to develop cross-platform using Flutter, an open-source, elegant user interface software development kit (UI SDK) created by Google, to guarantee a smooth experience across several devices.

Flutter makes it possible to create a single codebase that works with both iOS and Android devices, resulting in a uniform and effective development environment. Dart is the programming language used in Flutter, and it's renowned for being dependable and sturdy. Using Flutter's vast library of customizable UI components, the reservation hotel app's user interface was thoroughly created. This toolkit of parts makes it easier to design expressive and adaptable user interfaces, giving users a clear and simple experience.

Flutter's remarkable quality is its ability to handle important platform changes like scrolling, navigation, fonts, and icons with ease. This guarantees a unified and native-like experience across a variety of devices, which enhances user experience. Moreover, the efficiency of Flutter is heightened by Dart's native compilers, which compile Flutter code into native machine code. This compilation process results in a performant and optimized application, enhancing the overall responsiveness and speed of the reservation hotel app.

In conclusion, I was able to create a cross-platform hotel reservation application with an appealing, intuitive UI by utilizing Flutter and Dart. Dart's compilation speed and Flutter's configurable UI components made for a smoother development process and a consistent, enjoyable user experience across Android, iOS, and other platforms.

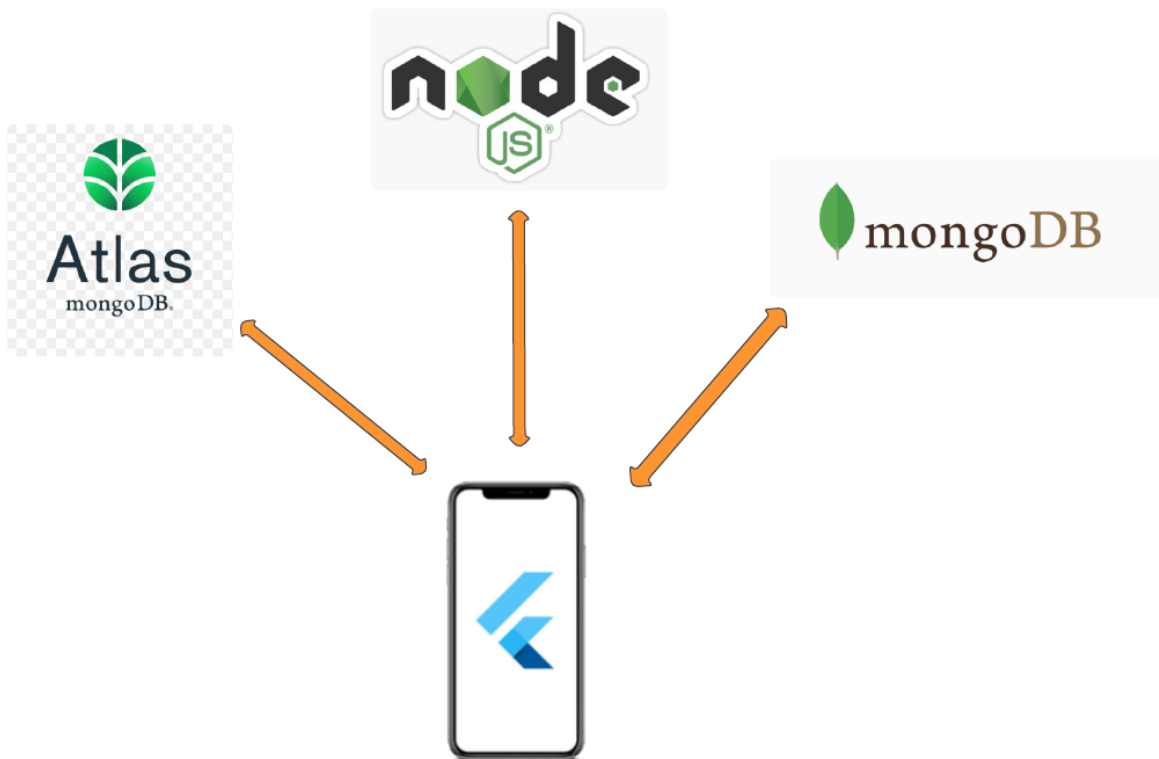


Figure 1:Application Architecture

4.1.4 Application Architecture:

In our project, we adopted the Model-View-Controller (MVC) architectural pattern, a robust paradigm that allowed us to organize and streamline our system efficiently:

Model: The Model, which forms the foundation of our MVC design, contains the essential information and business logic for our system. The main data store, MongoDB, houses all of the important data. In addition to handling requests from the user interface (View) and backend activities (Controller), the Model is in charge of maintaining data integrity. It serves as the official data source and is essential to preserving information consistency across the system. The Model is a representation of the underlying framework that guides and supports system functioning as well as user interactions.

View: In our MVC paradigm, the graphical user interface (GUI) acts as the View. The View offers an interactive platform with several capabilities that are tailored to meet the demands of users. Users are able to do things like book reservations, keep track of them, and use the rating and search functions. The user-facing part of the system is called The View, and it provides a simple and easy-to-use interface for easy system interaction.

Controller: Designed to be the core of our architecture, the Controller is responsible for managing backend functions for the website and the application. It is created with Node.js. This essential element coordinates the information and action flow between the Model and the View by acting as a go-between. The Controller makes it possible for the Model and the View to communicate with one other in a way that makes the system work as a whole. It is crucial in handling user requests, controlling data flow, and preserving the general consistency of system operations.

In conclusion, our project's MVC architecture creates an effective and transparent division of responsibilities. The Controller coordinates the interactions between the Model and the View, the View maintains the user interface, and the Model manages the data. This methodical approach improves our system's overall resilience, scalability, and maintainability.

MVC Architecture Pattern

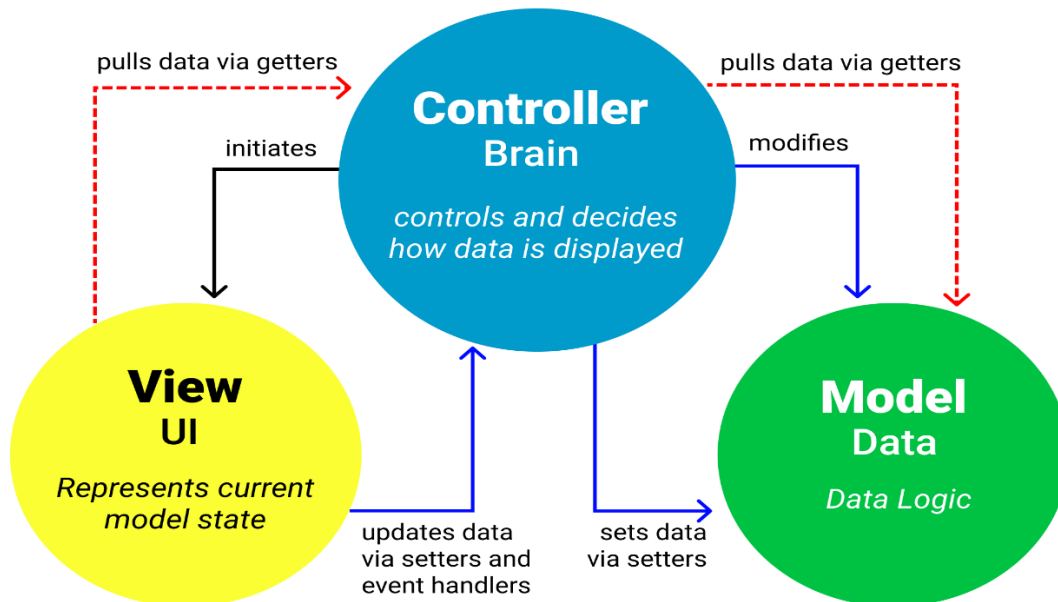


Figure 2: MVC Model

4.1.5 Agile Methodology:

We used the Agile model technique to execute our software system, strategically segmenting the process into five separate stages:

1. Requirement Analysis (Stage 1):

held lengthy brainstorming meetings to determine and list the core features that were intended for the program.

2. Planning (Stage 2):

Created a thorough schedule for completing the tasks, giving each activity a specified deadline.

conducted routine evaluations and conversations with the project supervisor once each assignment was finished.

3. Designing the Requirements (Stage 3):

sorted project work into priority categories, starting with the most important ones.

Regularly convened to assess advancements, deliberate on forthcoming tasks, and tackle obstacles, guaranteeing prompt resolution.

User stories were created for each project component, with the purpose, target audience, and technical details of each intended feature planned.

4. Development (Stage 4):

began coding each page's front end and back end respectively.

5. Testing (Stage 5):

Postman testing was used to verify the back-end's functioning and make sure everything was correct.

Ensured that each page functions properly by establishing and testing the link between the front end and back end.

We were able to create and improve our software system iteratively thanks to the Agile model approach, which also ensured alignment with changing priorities and needs.



Figure 3: Agile Methodology

4.1.6 Testing:

With its many features, Postman is a very feature-rich and easy-to-use application that makes testing backend programs much easier. Because of its user-friendly interface, developers may create and execute a variety of HTTP requests, including GET, POST, PUT, and DELETE, with ease, facilitating extensive testing of backend features. Request setup is where Postman shines; it makes it simple to create parameters, headers, and body content, simulating real-world circumstances for reliable testing.

One of Postman's main advantages is its organization, which enables users to organize requests into collections and offers an organized method for managing and running test cases. Developers may move between several testing environments with ease since this arrangement includes environment variables. With the addition of automation processes, Postman goes beyond manual testing and makes it possible to write JavaScript test scripts for intricate testing scenarios and repetitive activities. Because of this automation, testing is far more efficient and produces repeated, reliable findings.

Postman also offers a thorough view of API answers, including headers, content, and status codes, through its deep response inspection feature. By allowing team members to share collections and requests, the technology makes cooperation easier. Testing is easily included into the software development lifecycle because to its connection with monitoring tools, continuous integration platforms, and version control systems. By enabling developers to record API endpoints, request forms, and anticipated answers, Postman's documentation features further increase its usefulness and make it an invaluable tool for both development and testing teams. In conclusion, Postman proves to be a priceless tool for developers, providing a comprehensive solution for backend testing through its collaboration features, automation capabilities, and user-friendly design.

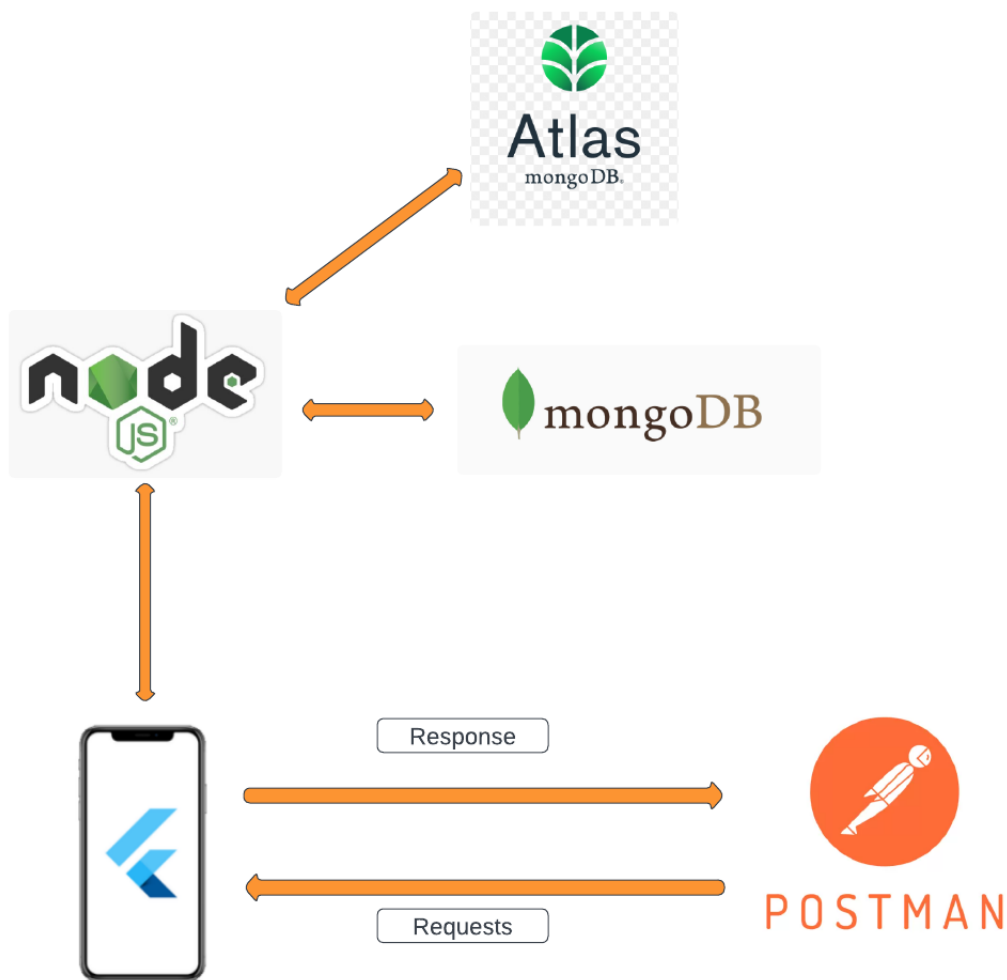


Figure 4: Testing

4.1.7 Version Control:

When it comes to version control and cooperation, GitHub is a vital tool that greatly improves collaborative software development processes. As a distributed version control system, GitHub helps teams easily monitor and manage changes to their codebase. The incorporation of GitHub into our development process has been crucial in promoting effective collaboration and maintaining the integrity of version control.

Serving as a single repository for our codebase is one of GitHub's main contributions to our collaborative efforts. With the ability to push code to GitHub, any team member may create a centralized, well-organized place for the project. This protects against potential data loss and facilitates cooperation in addition to acting as a safe backup. The version control features of GitHub are essential for supporting an organized and methodical approach to group coding. Developers may work on features or fixes in isolation (branching) and smoothly incorporate their changes back into the main codebase (merging) with the help of branching and merging techniques. By doing this, a clean and functioning codebase is maintained and it is ensured that several team members may contribute concurrently without interfering with each other's work.

One essential part of our collaboration approach is the pushing and pulling of code from GitHub. The revised code is available to the whole team when a team member pushes their modifications to GitHub after finishing a job or adding a new feature. On the other hand, team members pull the updates from GitHub when they require the most recent version of the code, guaranteeing that everyone is working with the most recent and well-organized codebase.

With capabilities like pull requests and code reviews, GitHub offers strong project management, problem tracking, and collaboration tools as well. Before integrating code into the main branch, pull requests provide team members the opportunity to suggest improvements, have a discussion about revisions, and check the quality of the work. This cooperative review procedure reduces possible problems and improves the quality of the code.

To sum up, GitHub is essential to our methods for collaboration and version control. Our development processes are now far more organized and efficient because to its centralized code storage, version control, and collaboration features. GitHub has become an essential component of our workflow because to its smooth pushing and pulling of code procedures, which enable our team to collaborate on projects, monitor changes methodically, and produce code that meets high quality standards.

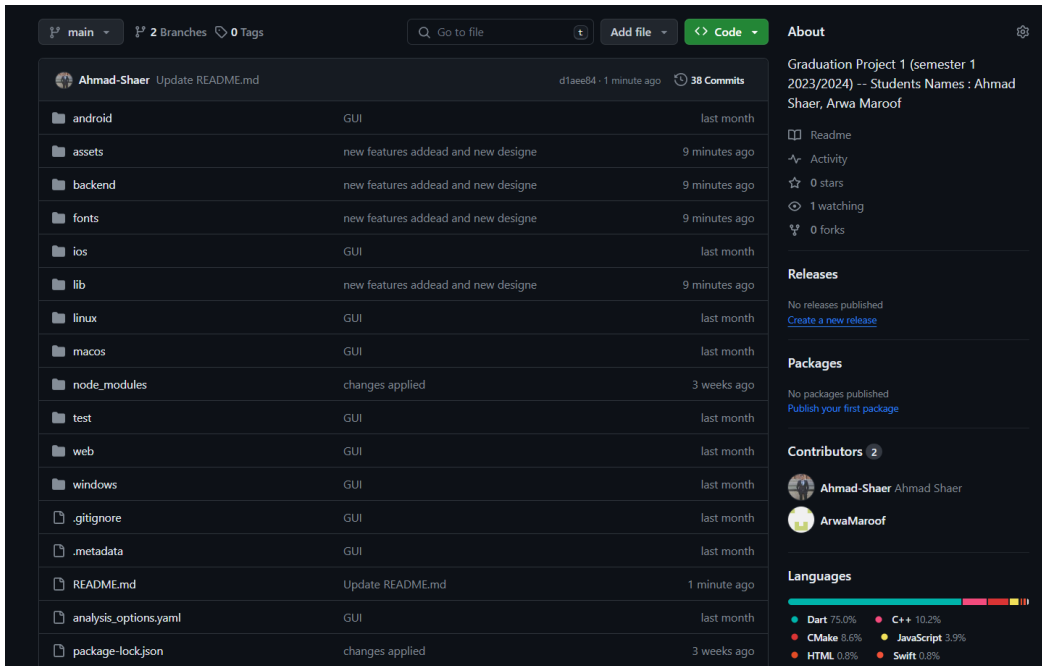


Figure 5:Version Control_1

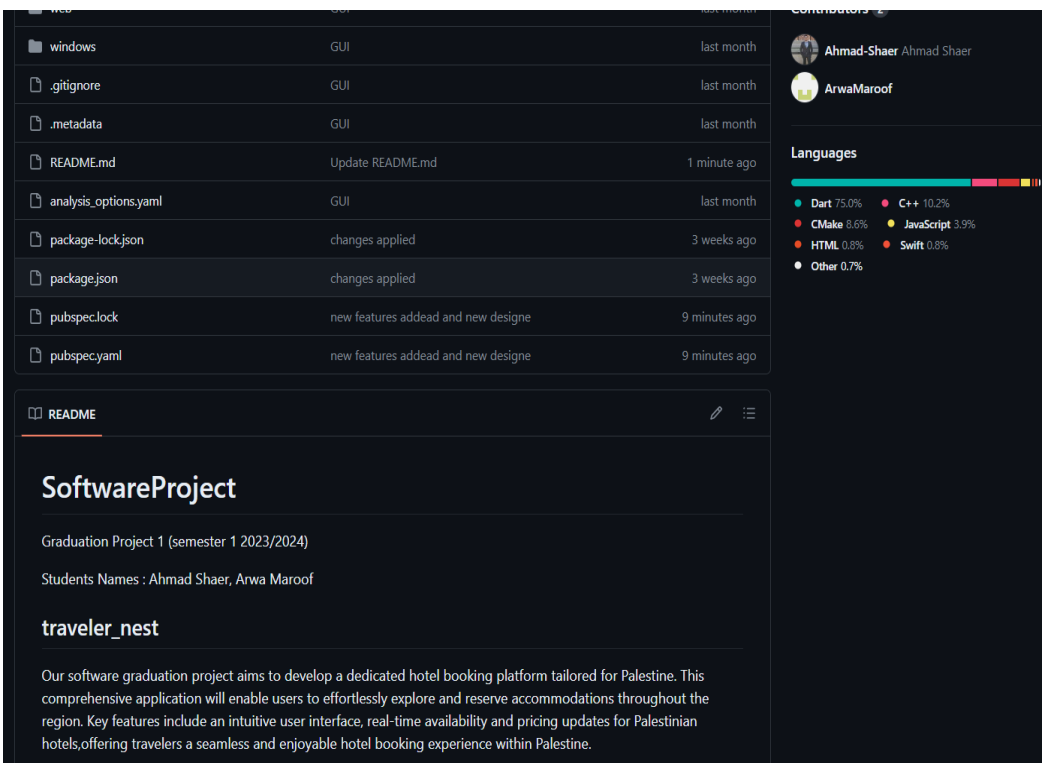


Figure 6:Version Control_2

4.2 Features:

4.2.1 Login / Signup Pages:

To access the application, users are required to possess an account and verify the accuracy of their password; otherwise, entry will be denied. For individuals without an existing account, registration on the designated page is mandatory. During registration, users must input and confirm their name, email, password, phone number, etc. On both the login and registration pages, completion of all fields is essential; failure to do so will prompt a warning message.

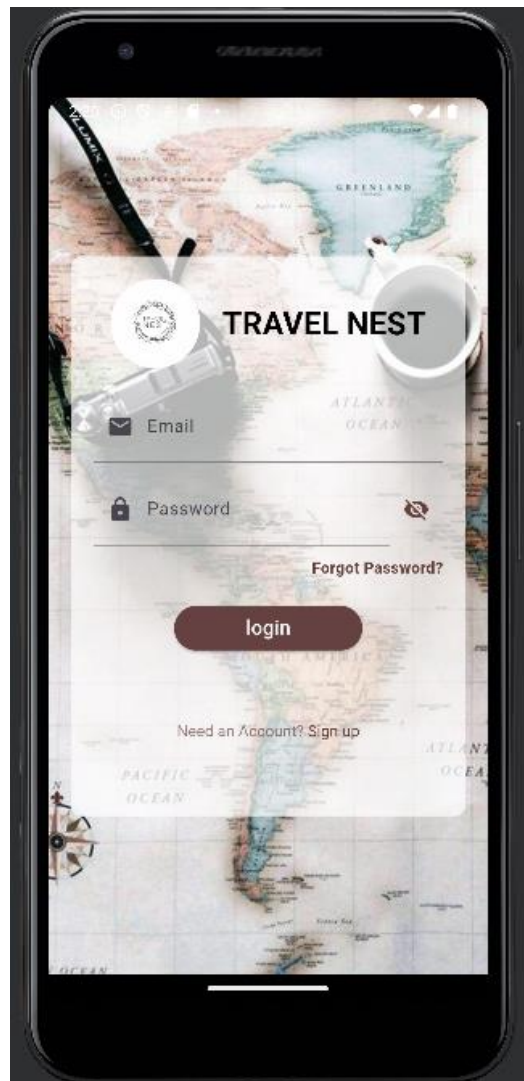


Figure 7:Login Page

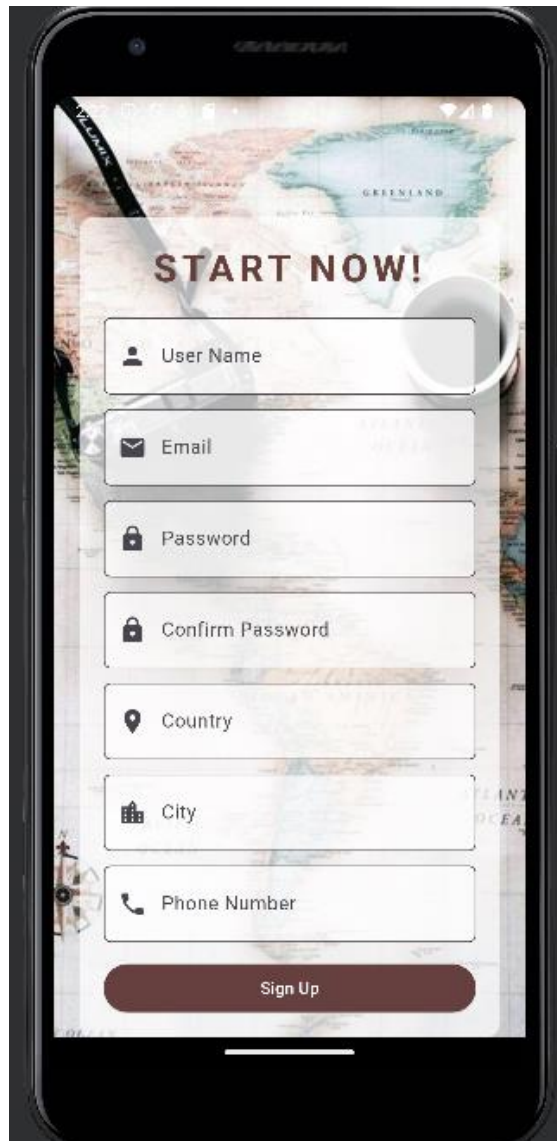


Figure 8: Sign Up page

4.2.2 Main Page /Home page

Regarding the homepage, it is crucial to emphasize its well-designed nature, incorporating various accessibility features to ensure user-friendliness. Let's delve into the specific functionalities of each feature.

The homepage encompasses a primary menu comprising Home, Calendar, Reservations, Favorite Reservations, and Profile. Additionally, it features a prominent search box and notifications for enhanced user interaction.

Notably, the home page showcases a compilation of the most frequently booked hotels. It also provides a list of Palestinian cities, allowing users to seamlessly book accommodations in their desired locations. The emphasis is on creating an intuitive and accessible interface for an enhanced user experience.

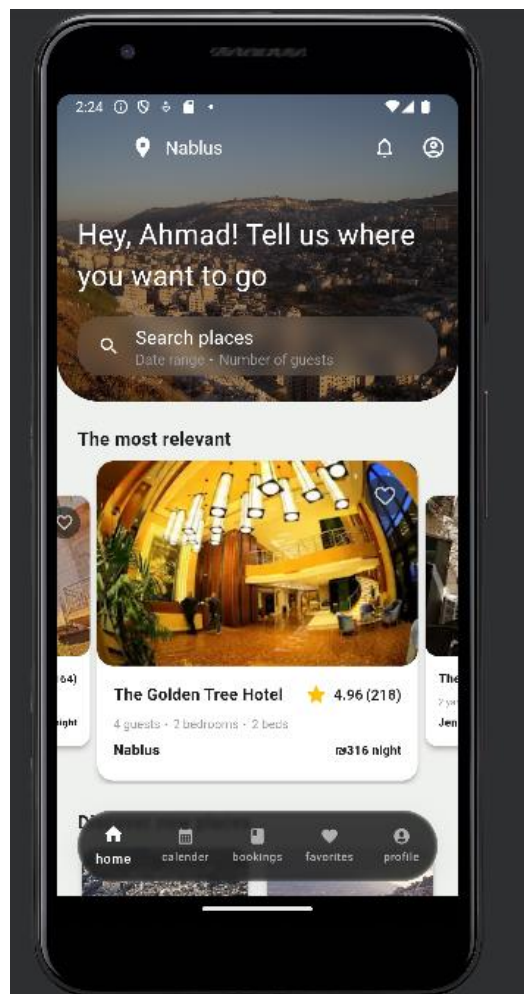


Figure 9: Home Page

4.2.3 Profile Page:

The user's personal page serves as a private hub housing information directly related to the user. Within this space, users have the capability to exert control by modifying their personal details, including their name, email, and personal number. Additionally, users possess the flexibility to personalize their personal page by changing their cover photo. This empowers users to curate and manage their individual information, fostering a sense of ownership and customization within the platform.

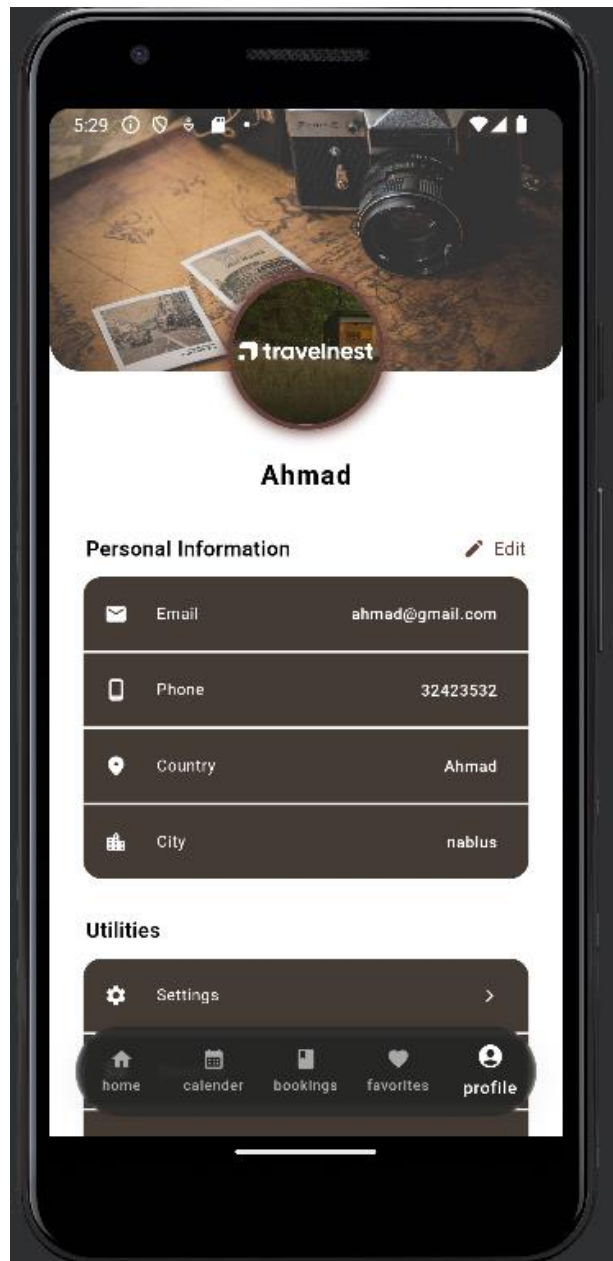


Figure 10: Profile Page

4.2.4 Hotel Page:

The hotel page is meticulously crafted to showcase the unique features and characteristics of each property, drawing in users and inspiring them to book a memorable stay. This page precisely highlights the special aspects of the hotel's location, offering captivating images that encapsulate the essence of the surroundings. Additionally, it provides detailed insights into the distinctive features of the hotel rooms, offering a comprehensive overview of the accommodations, complete with a 5-star rating system for user guidance.

Transparent information about current room rates and availability is prominently featured on the hotel page, empowering users to make well-informed decisions. Should a user find a particular hotel appealing, they are presented with the option to proceed with the booking or add the hotel to their favorites list. This ensures a seamless and personalized experience, catering to individual preferences for a delightful hotel selection process.

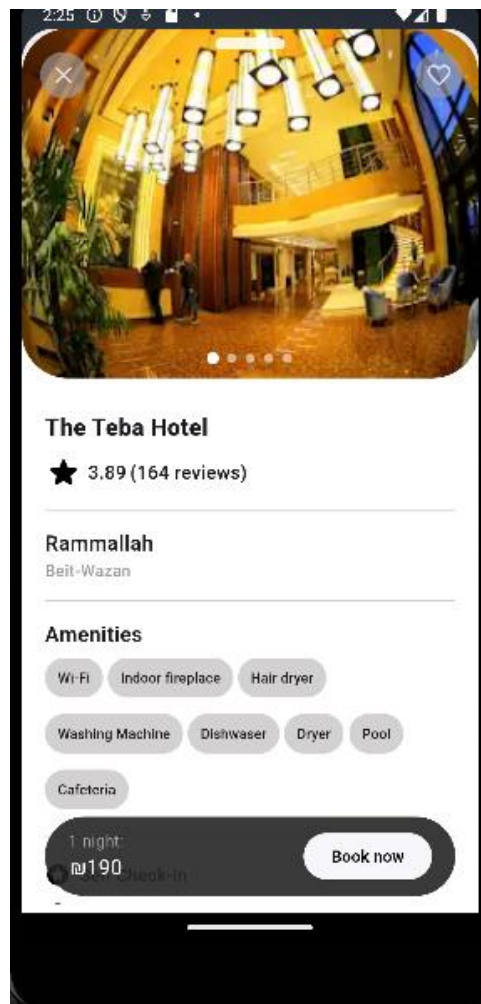


Figure 11: Hotel Page

4.2.5 Booking page:

Upon selecting the desired hotel, the user proceeds to make a reservation, where they can tailor their stay according to their preferences. The user is prompted to pick the dates that align with their schedule from the calendar. Additionally, they have the flexibility to choose from the available rooms within the chosen hotel, considering prices and specifications that best suit their needs.

This intuitive reservation process ensures that users have a seamless experience, allowing them to customize their stay by selecting specific dates and choosing from a variety of available rooms based on their preferences, pricing considerations, and desired specifications. This user-centric approach enhances the overall booking experience, offering flexibility and personalization.

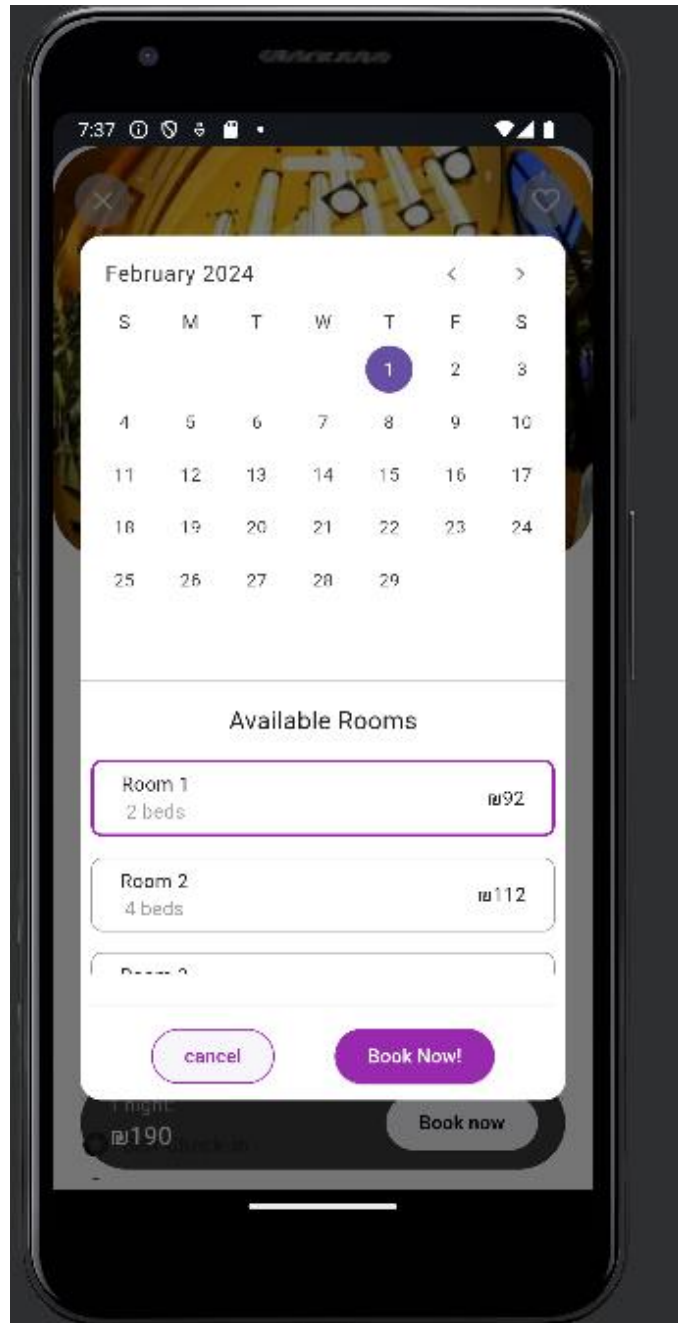


Figure 12:Booking Page

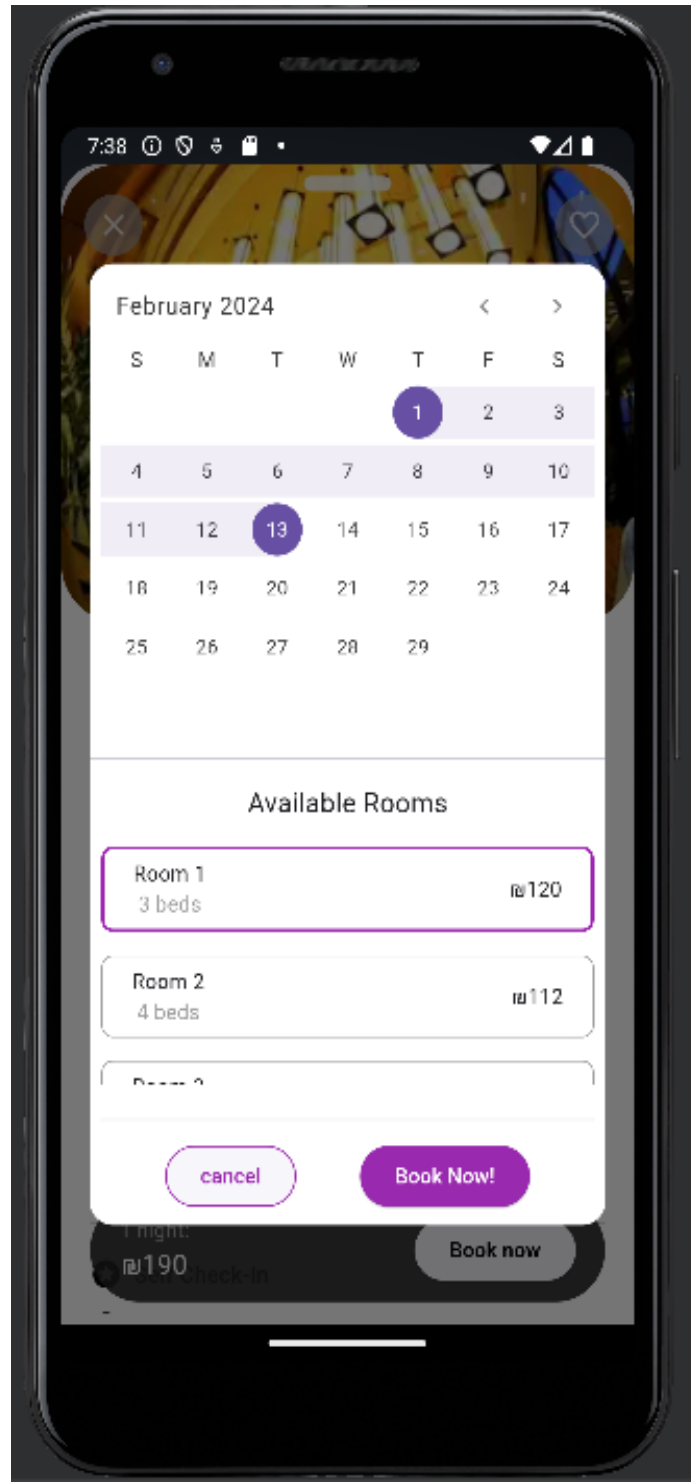


Figure 13: Booking Page (specifying the dates)

4.2.6 Favorites Page:

As mentioned earlier, the homepage provides a convenient display of the user's favorite hotels, showcasing those that have been saved on the Favorites page. This feature enables users to quickly access and review the hotels they have expressed interest in.

Furthermore, the homepage incorporates a powerful search functionality enhanced by a filter option. This empowers users to refine their search criteria, ensuring a tailored and efficient exploration of available accommodations. The combination of showcasing favorites and offering robust search and filter capabilities contributes to a user-friendly interface, facilitating a seamless and personalized experience for finding the perfect hotel.



Figure 14: Favorites Page

4.2.7 Search Page:

Certainly, the application enhances user convenience by offering comprehensive search functionalities. Users can conduct a general search or utilize the filter option for a more refined exploration. The search capabilities extend to locating hotels based on various criteria such as the city name, price range, or rating.

Whether users are seeking a specific city, aiming for accommodations within a particular price bracket, or prioritizing hotels with certain ratings, the filter empowers them to tailor their search precisely. This flexibility ensures that users can efficiently find hotels that match their preferences and requirements, contributing to a more streamlined and personalized experience within the application.



Figure 15: Search Page 1

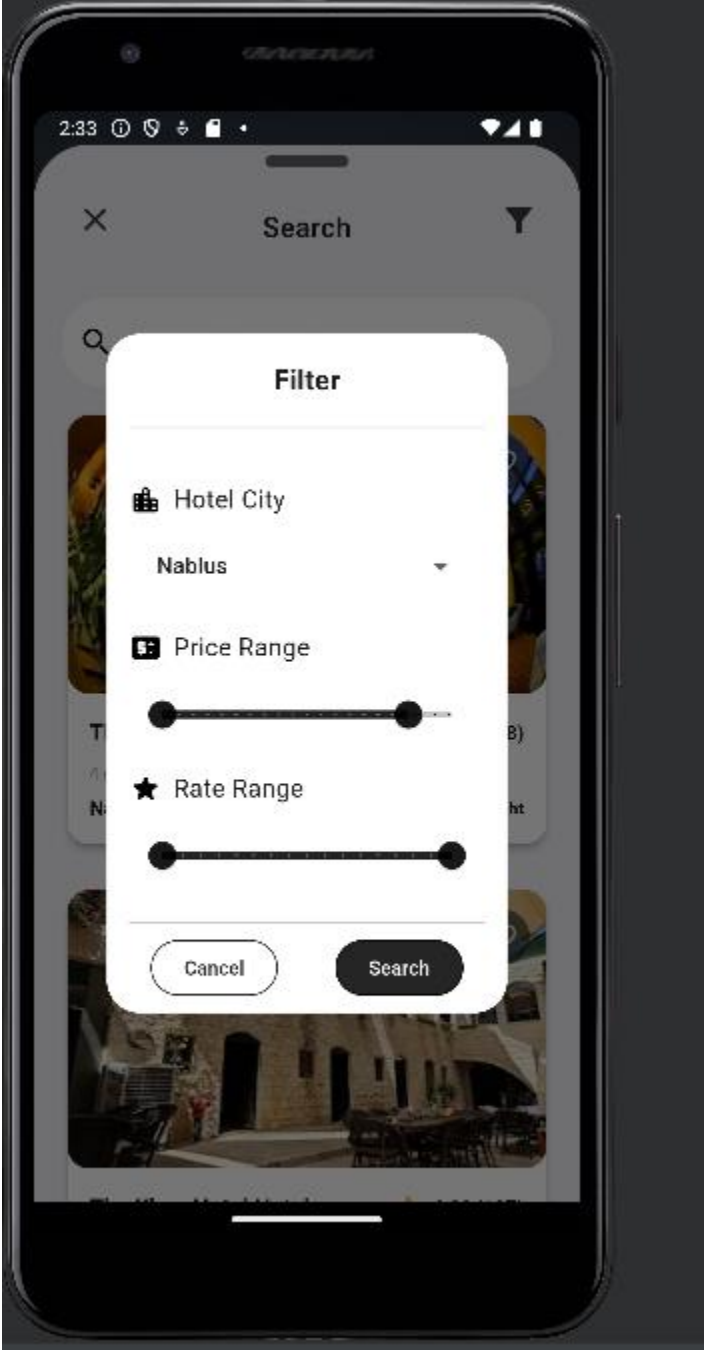


Figure 16: Search Page Filter

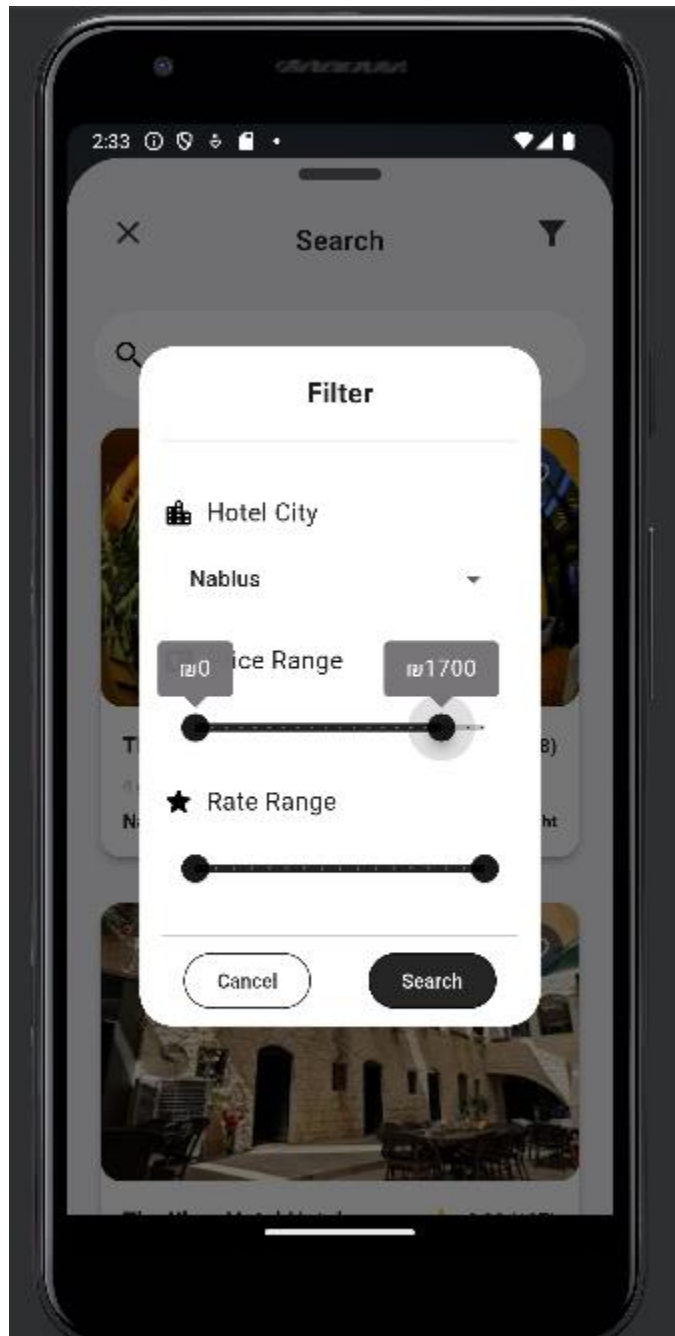


Figure 17: Search Page Filter (Price)

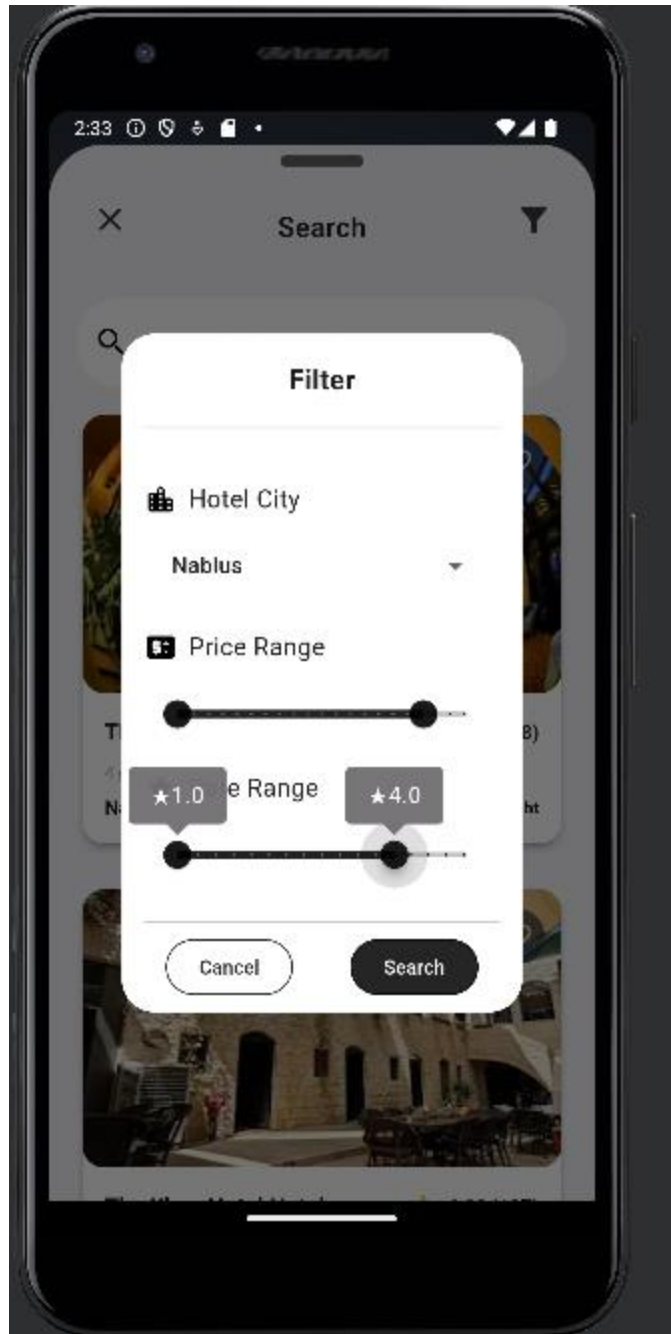


Figure 18: Search Page Filter (Rating)

4.2.8 User Callender:

The application further enhances user organization and convenience by providing a dedicated calendar feature. This calendar serves as a personal repository, storing the dates of the user's hotel reservations. This proactive reminder system ensures that users stay informed about the upcoming dates of their hotel bookings, facilitating a seamless and organized experience.

With the calendar functionality, users can easily keep track of their travel plans, making it more straightforward to manage and coordinate their schedule. This user-friendly feature adds an extra layer of convenience, contributing to a positive and efficient user experience within the application.



Figure 19:User Calendar

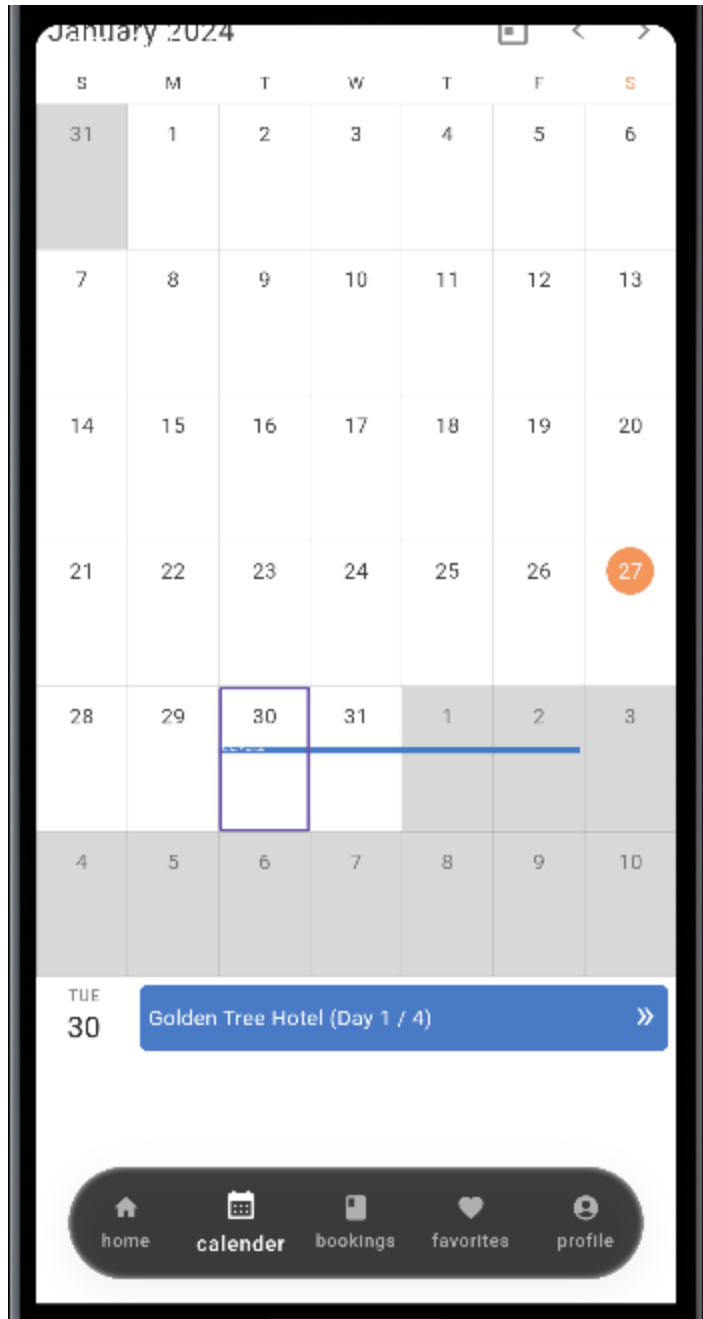


Figure 20: User Calander (Extra Info)

4.2.9 User bookings:

As mentioned earlier, the reservations page acts as a centralized hub within the application, consolidating all the confirmed bookings made by the user. This page serves as a comprehensive record of the user's hotel reservations, providing easy access to important details such as reservation dates, hotel names, and any additional relevant information.

By offering a dedicated space for confirmed reservations, the application ensures that users can efficiently review and manage their upcoming stays. This feature adds an extra layer of organization and accessibility, contributing to a user-centric design that prioritizes clarity and ease of use.

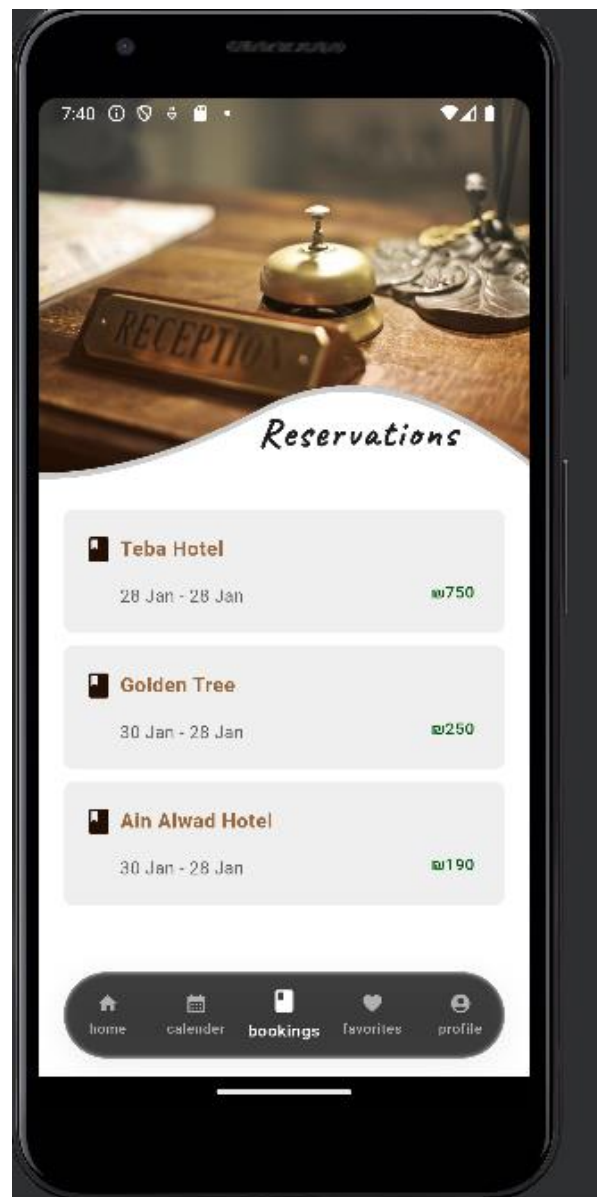


Figure 21: User Bookings

4.2.10 Notification system:

The notifications page, a pivotal feature within the app, plays a crucial role in keeping users informed and ensuring a safe and transparent experience. Notifications are sent to the user whenever their personal data changes, providing a quick update of any amendments made.

Moreover, notifications act as a confirmation mechanism for the hotel booking process. Users receive notifications confirming the success or failure of their hotel bookings, providing them with reassurance and ensuring their information is verified and processed securely. This proactive communication not only keeps users well informed, but also instills confidence in the app's reliability and commitment to user satisfaction.

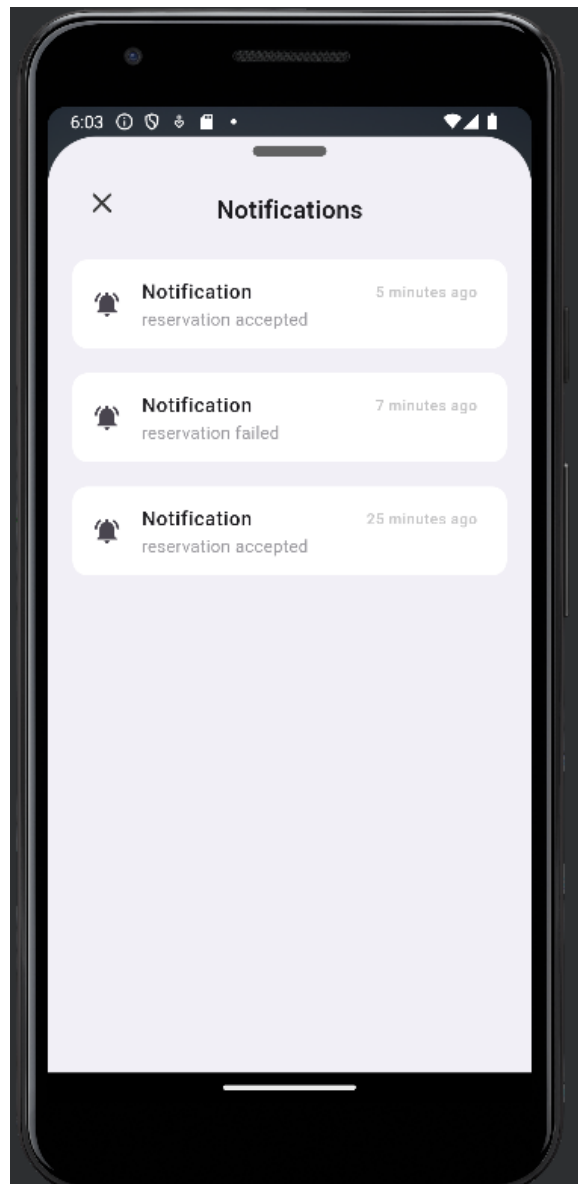


Figure 22: Notification System

4.2.11 Login:

This page for hotel admin and application admin.

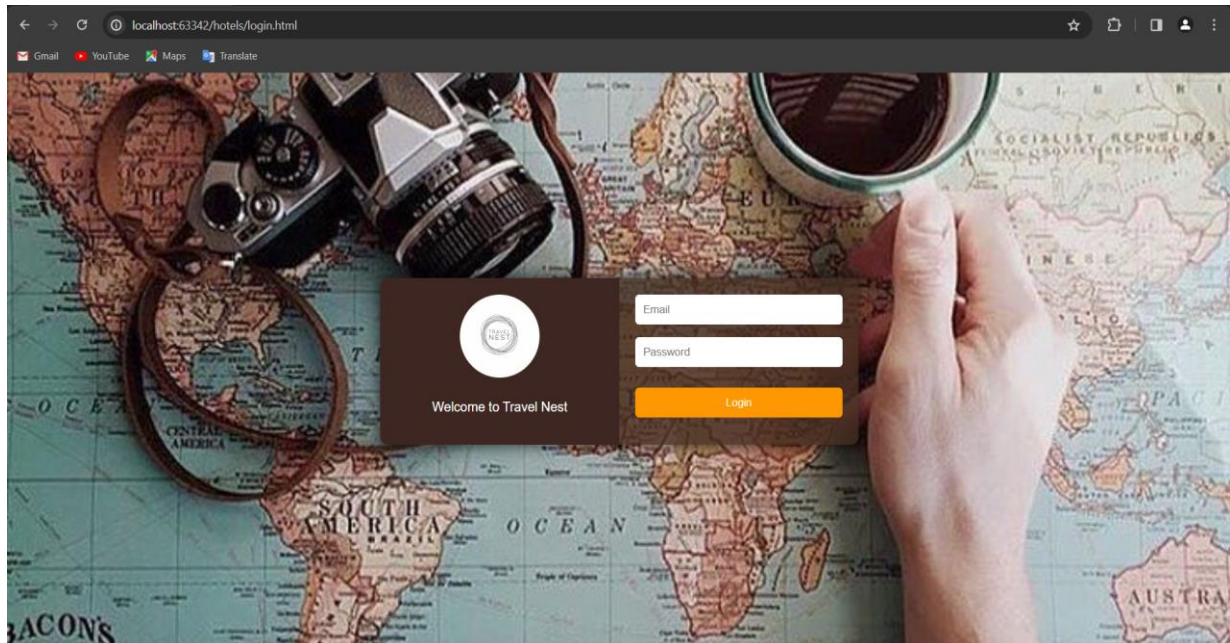


Figure 23:web Login page

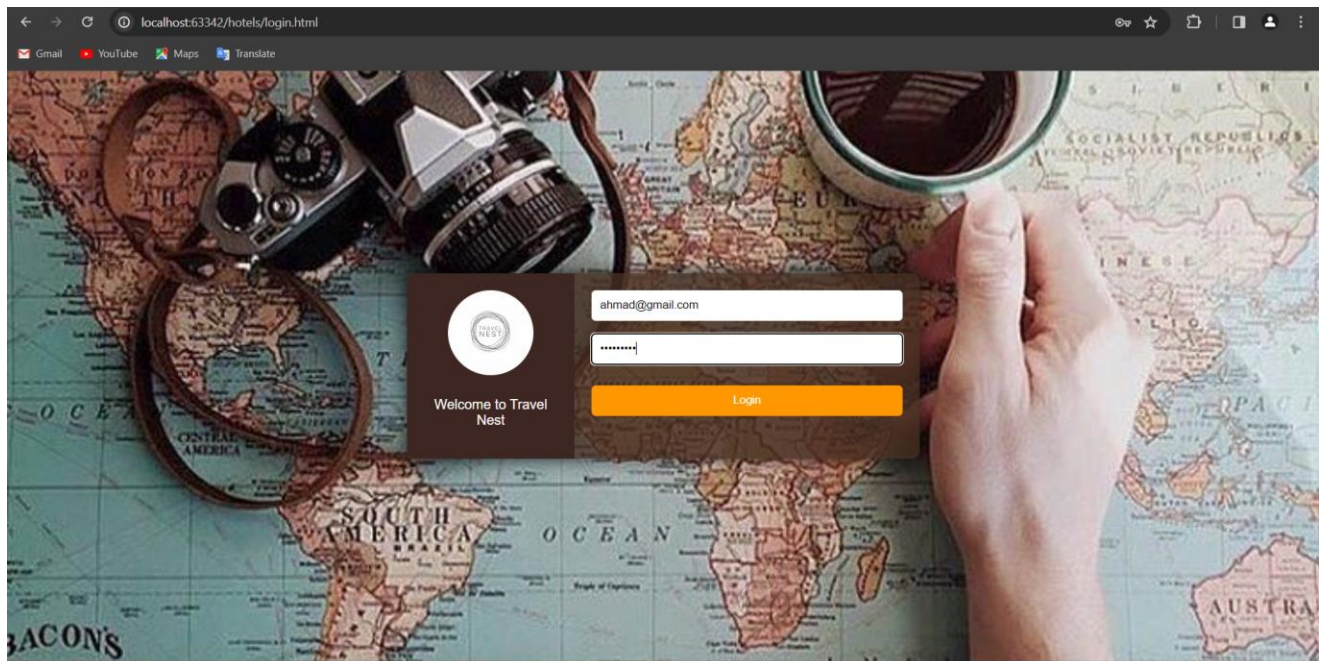


Figure 24:web Login Page 2

4.2.12 Dashboard:

This page for application admin

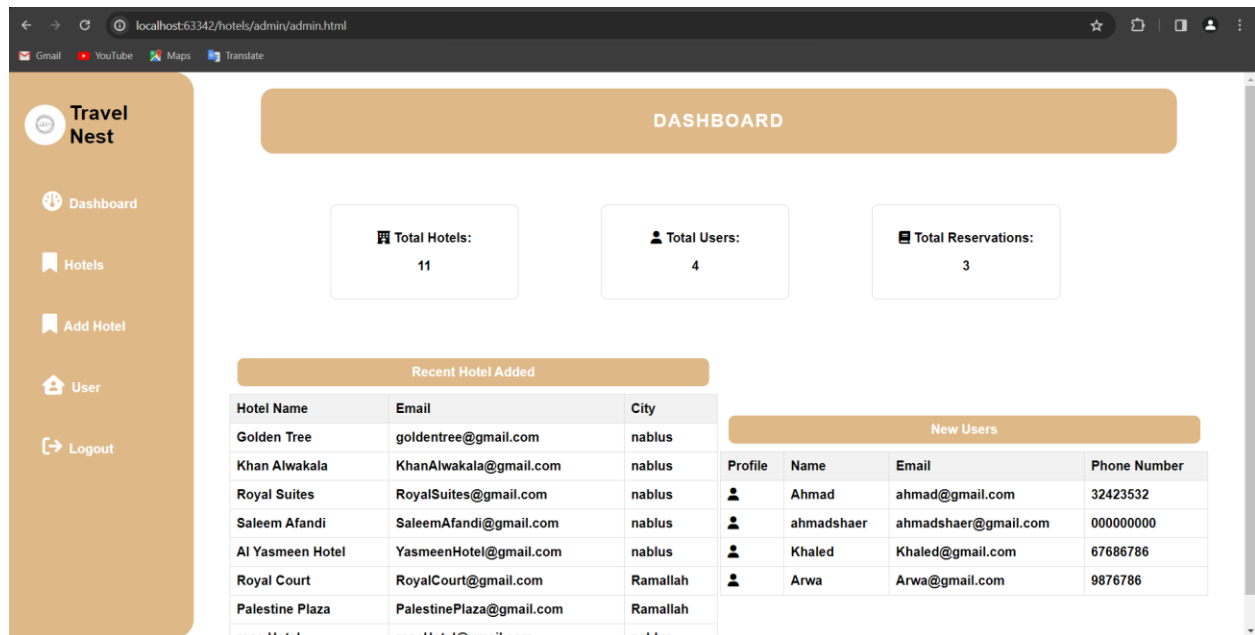


Figure 25: Admin Dashboard

In which it shows him the total hotels registered in the application and the number of users who also signed up for to use it and the total reservations and two tables showing the recent hotels added and new users added

Now for the hotel owner:

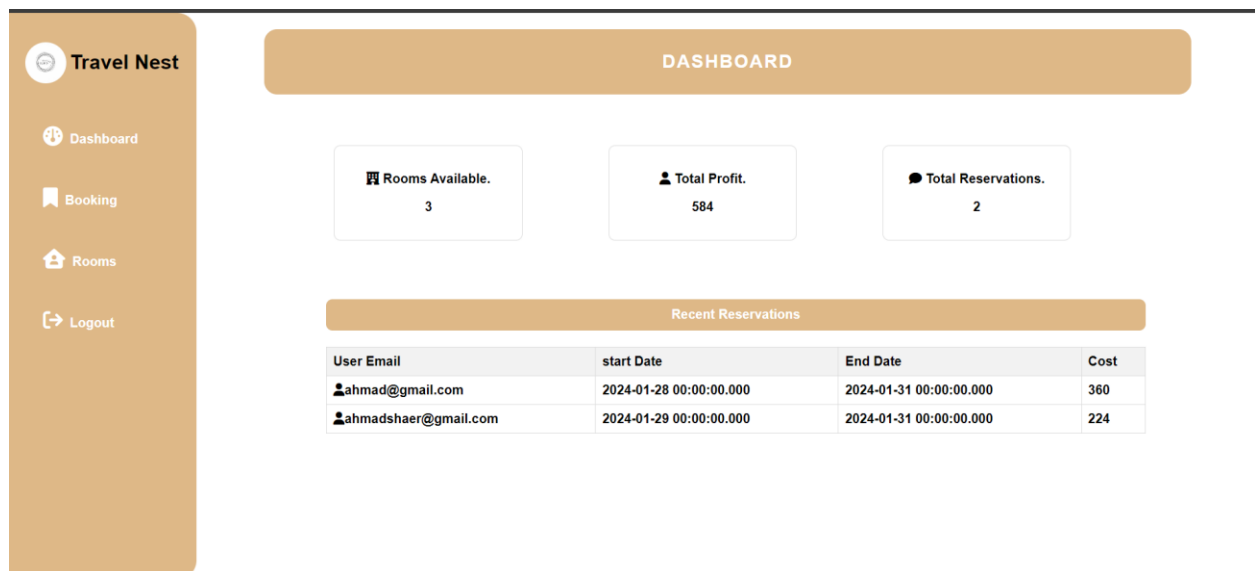


Figure 26: Hotel Admin Dashboard

And this page is dynamically filled with data depending on the hotel email and it shows the hotel the number of rooms available and the total profit and the total number of reservations and shows a table of the resent reservations with the start date and end date and user email and the cost

4.2.13 Hotels Page:

This page is only for the admin page in which he can see all the hotels registered to the application and some information about each hotel

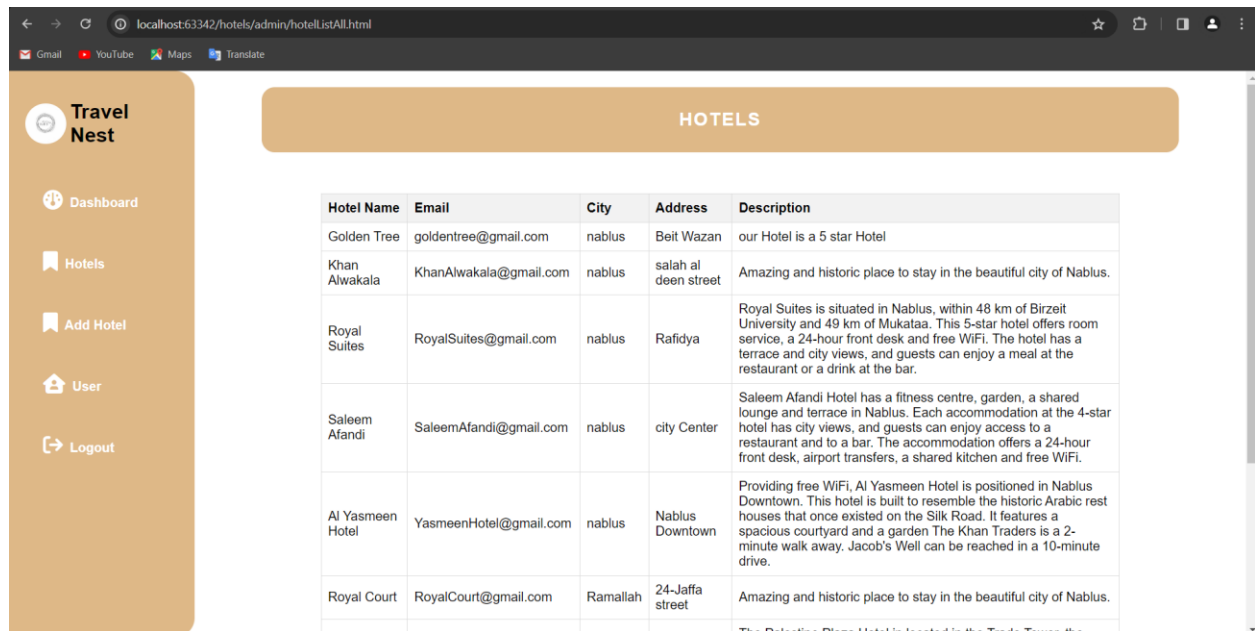
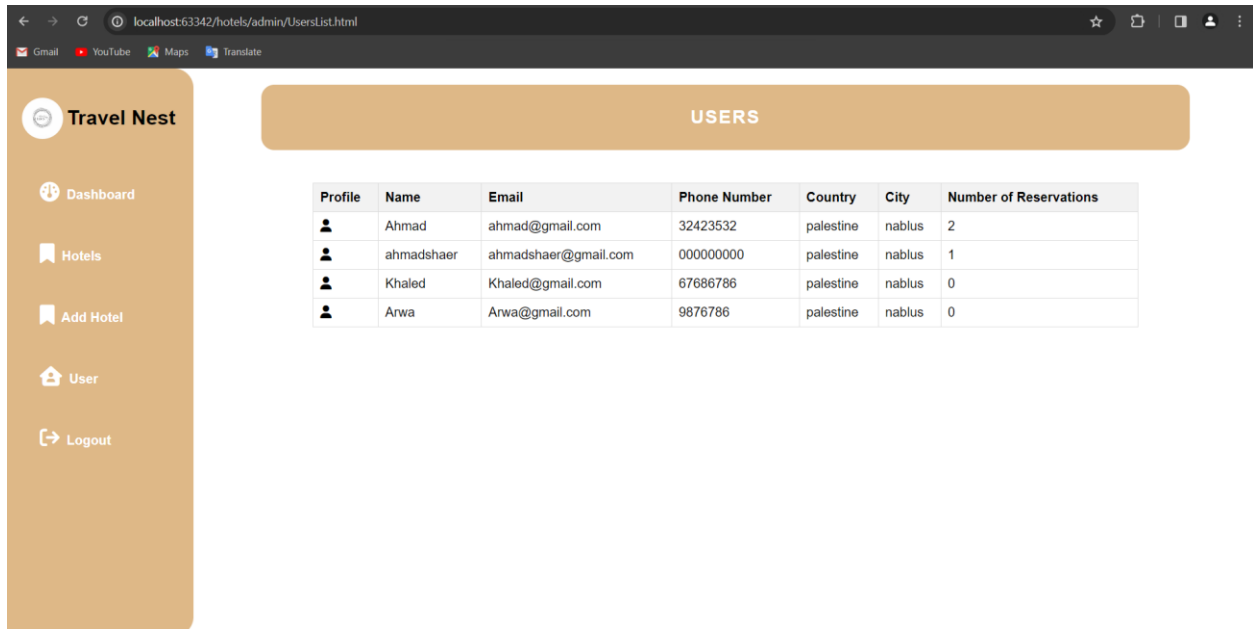


Figure 27:Hotels Page

4.2.14 Users page:

This page also is only shown to the admin of the application



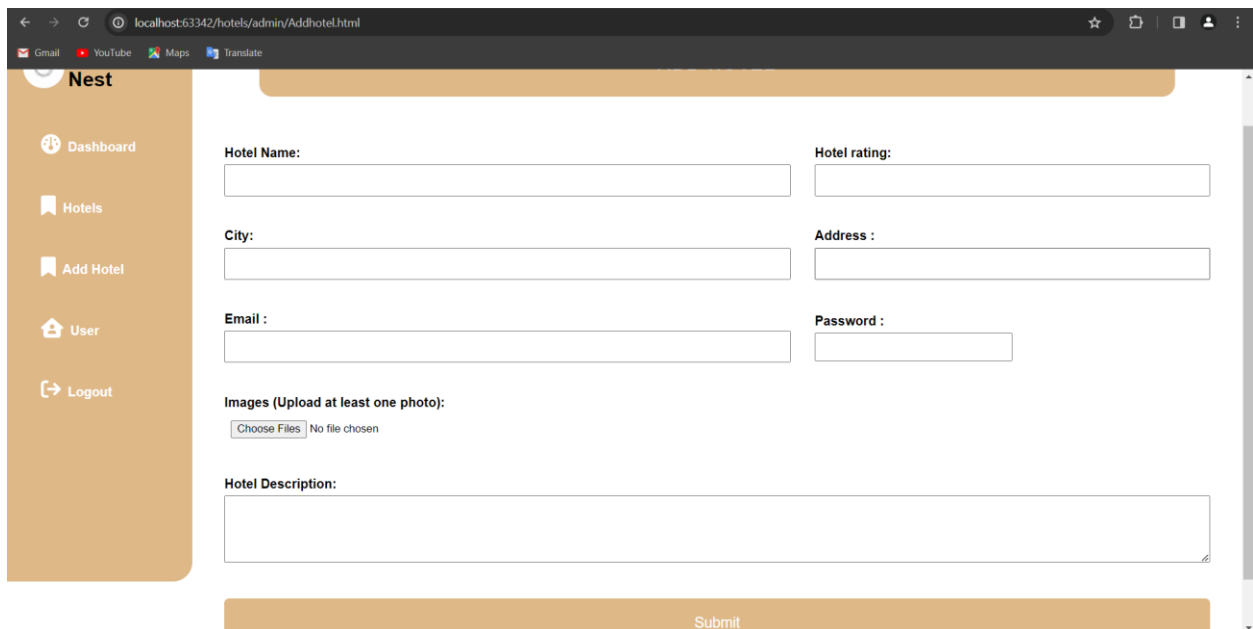
The screenshot shows the 'Users' page in the Travel Nest application. The page features a sidebar with navigation options: Dashboard, Hotels, Add Hotel, User, and Logout. The main content area displays a table of users with the following data:

Profile	Name	Email	Phone Number	Country	City	Number of Reservations
	Ahmad	ahmad@gmail.com	32423532	palestine	nablus	2
	ahmadshaer	ahmadshaer@gmail.com	000000000	palestine	nablus	1
	Khaled	Khaled@gmail.com	67686786	palestine	nablus	0
	Arwa	Arwa@gmail.com	9876786	palestine	nablus	0

Figure 28: Users Page

4.2.15 Add Hotel Page:

This is an admin page in which he can adds hotels to the data base



The screenshot shows the 'Add Hotel' page in the Travel Nest application. The page features a sidebar with navigation options: Dashboard, Hotels, Add Hotel, User, and Logout. The main content area displays a form with the following fields:

- Hotel Name:
- Hotel rating:
- City:
- Address:
- Email:
- Password:
- Images (Upload at least one photo): No file chosen
- Hotel Description:

A Submit button is located at the bottom of the form.

Figure 29: Add Hotel Page

4.2.16 Bookings Page:

This page is shown for the hotel admin and its dynamically filled for each hotel

The screenshot shows the 'Bookings' page in the Travel Nest admin interface. On the left is a sidebar with navigation options: Dashboard, Booking, Rooms, and Logout. The main content area has a header 'BOOKINGS' and a sub-header 'Reservations'. Below this is a table with the following data:

User Email	start Date	End Date	Cost
ahmad@gmail.com	2024-01-28 00:00:00.000	2024-01-31 00:00:00.000	360
ahmadshaer@gmail.com	2024-01-29 00:00:00.000	2024-01-31 00:00:00.000	224

Figure 30:Bookings

4.2.17 Rooms Page:

This page shows the rooms for the hotel and the hotel can add rooms also

The screenshot shows the 'Rooms' page in the Travel Nest admin interface. On the left is a sidebar with navigation options: Dashboard, Booking, Rooms, and Logout. The main content area has a header 'ROOMS' and a sub-header 'Add room'. Below this are two input fields: 'Number Of Beds:' with a value of 4, and 'Price:' with a value of 150. A 'Submit' button is located below these fields. Below the form is a table with the following data:

Room Number	Number of Beds	Price
1	3	100
2	2	60

Figure 31:Rooms Page 1

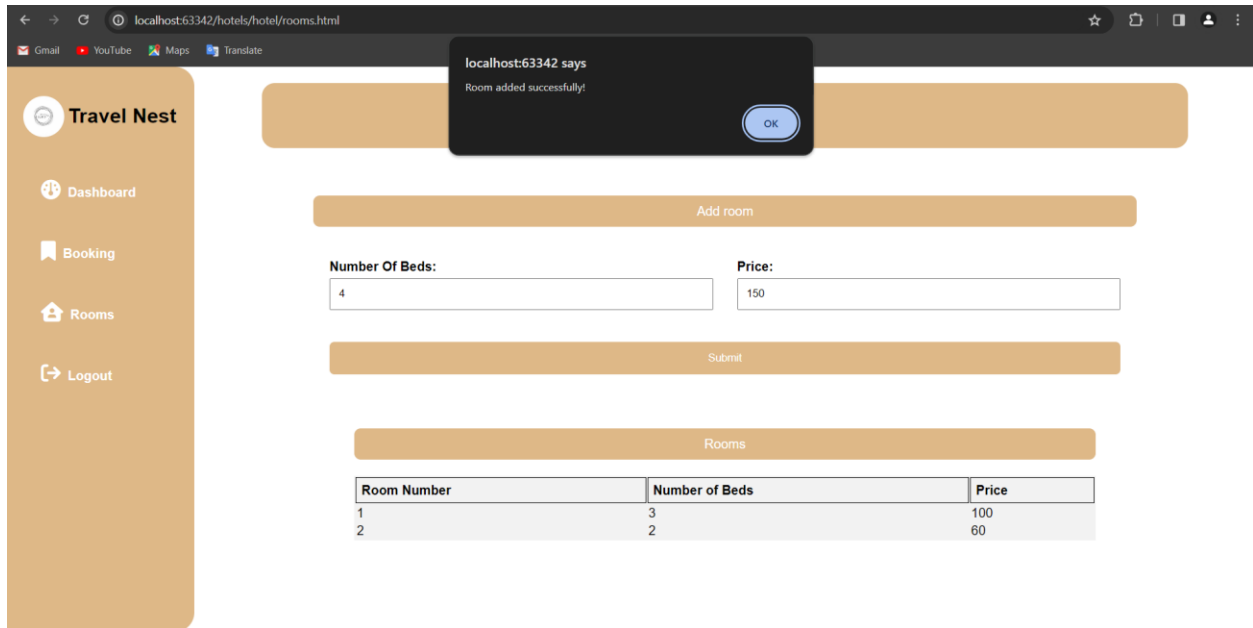


Figure 32:Rooms Page 2

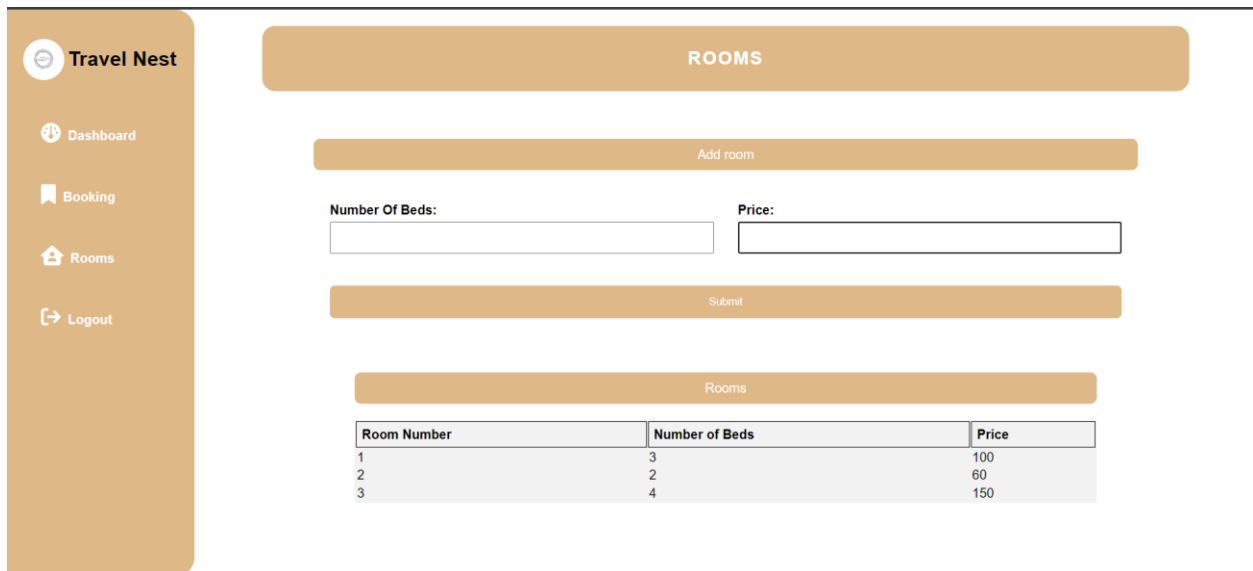


Figure 33:Rooms Page 3

5 Conclusion/ Recommendations and Future Works

In conclusion, by offering both tourists and hotel owners an application they can interact with, we have successfully developed a cross-platform mobile application that will assist people in finding precisely what they are looking for in a hotel, be it a particular view, location, room type, or feature like a swimming pool.

Each tourist has an account with a suggestion system, that depends on rating and areas he visited and expenses. Travelers may quickly search, browse hotels, add them to their favorites, and make a reservation.

We provide hotel owners with a complete user interface, which includes a page that shows all of the hotel's data. Additionally, it has a thorough analysis page of the reservations and earnings as well as a complete page with analytical data on their booking sales. giving hotel owners the most comprehensive understanding of their financial performance

And for the future work we could integrate our application with google maps to show the traveler where exactly the hotel resides.

6 References

1. Mayr, Thomas, and Andreas H. Zins. "Acceptance of online vs. traditional travel agencies." *Anatolia* 20.1 (2009): 164-177.
2. Park, Sangwon, Yizhen Yin, and Byung-Gak Son. "Understanding of online hotel booking process: A multiple method approach." *Journal of Vacation Marketing* 25.3 (2019): 334-348.
3. Van De Logt, Jetske. "The Impact of Online Travel Agencies: Is it still relevant for hotels to have their own booking system?." (2017).
4. Moindjie, Housnata, Khalid Camara, and Ghaith Mansoor Taqi. *Online Booking System*. Diss. Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Board Bazar, Gazipur-1704, Bangladesh, 2022.
5. Alhamad, Ibrahim Abdullah, and Harman Preet Singh. "Predicting Key Factors Impacting Online Hotel Ratings Using Data Mining Approach: A Case Study of Makkah of Saudi Arabia." *International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies* 12.2 (2021): 1-12.
6. Mohamad, E., et al. "Online Booking Systems for Managing Queues at The Road Transport Department." (2019).
7. Languye, Justin. "The challenge facing online booking among local travel agencies in Arusha." (2020).