A
Summer Internship Report
On
"Building Layout Guide"

(IT446 – Summer Internship - II)

Prepared by

Raghavendrasinh Jadeja(D21IT184)

Under the Supervision of

Dr. Purvi Prajapati

Submitted to

Charotar University of Science & Technology (CHARUSAT)
for the Partial Fulfillment of the Requirements for the
Degree of Bachelor of Technology (B.Tech.)
for Semester 7

Submitted at

Accredited with Grade A+ by NAAC
Accredited with Grade A by KCG

SMT. KUNDANBEN DINSHA PATEL DEPARTMENT OF INFORMATION TECHNOLOGY

Chandubhai S. Patel Institute of Technology (CSPIT)
Faculty of Technology & Engineering (FTE), CHARUSAT
At: Changa, Dist: Anand, Pin: 388421.
August, 2023

CHARUSAT
CHAROTAR UNIVERSITY OF SCIENCE AND TECHNOLOGY

Accredited with Grade A+ by NAAC
Accredited with Grade A by KCG

CERTIFICATE

This is to certify that the report entitled "**Building Layout Guide**" is a bonafied work carried out by **Raghavendra Sinh Jadeja(D21IT184)** under the guidance and supervision of **Dr. Purvi Prajapati** & **Mr. Sagar Khunt** for the subject **Summer Internship – II (IT446)** of 7th Semester of Bachelor of Technology in **Department of Information** at Chandubhai S. Patel Institute of Technology (CSPIT), Faculty of Technology & Engineering (FTE) – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate himself, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred by the examiner(s).

Under the supervision of,

Dr. Purvi Prajapati
Assistant Professor
Smt. Kundanben Dinsha Patel Department of Information Technology
CSPIT, FTE, CHARUSAT, Changa, Gujarat

Mr. Sagar Khunt
Director
ITMAKERZ

Dr. (Prof.) Parth Shah
Head of Department (IT)
CHARUSAT, Changa, Gujarat.

Chandubhai S. Patel Institute of Technology (CSPIT)

Faculty of Technology & Engineering (FTE), CHARUSAT

At: Changa, Ta. Petlad, Dist. Anand, Pin: 388421. Gujarat.

itmakerz
Solutions Private Limited

Website: www.itmakerz.com
Email: contact@itmakerz.com

Date: 15th july 2023

To Whom It May Concern

This is to certify that Raghavendrasinh Jadeja(D21IT184), a student of B.Tech in Information Technology From CSPIT(Charusat University), Changa - 388421, Gujarat has undertaken an internship at ITMakerz Solutions Pvt. Ltd, Ahmedabad from 15th May 2023 to 15th July 2023.

During the internship, he has been found to be punctual and diligent. After completion of the internship, he successfully submitted the Project Report and Code at ITMakerz Solutions Pvt. Ltd, Ahmedabad.

Project Name: Building layout guide

Project Guide: Sagar Khunt (Director)

We wish him good luck for his future endeavors.

Sincerely,

Sarfraj Mankad
Co-founder
ITMakerz Solutions

Acknowledgement

- The Success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and we would not forget to thank them.

- I am extremely thankful to Prof. Rima Patel and Mr. Sagar Khunt. for providing such nice support and guidance, although they had a busy schedule managing the college and company work. I owe my deep gratitude to myteam members, who took keen interest in project work and guided me all along, till the completion of my project work by providing all the necessary information for developing a good project. Also encouraged me till the completion of my project work.

Abstract

- ➢ The project is a web application developed using ReactJS, NodeJS, and a database backend, designed to facilitate the creation of building layout apps. The goal of the project is to provide a flexible and user-friendly platform that enables users to see interactive and detailed building layout apps that can be used for purchasing new houses..

- ➢ The application provides an intuitive and responsive user interface, powered by ReactJS, that can handle complex user interactions with ease. The use of ReactJS ensures that the application is fast, efficient, and can be accessed from any device

- ➢ The application backend is powered by NodeJS, which provides the necessary server-side functionality for processing and storing data. NodeJS also enables the application to be highly scalable, allowing it to handle large volumes of data without any performance degradation.

- ➢ To store and manage data, the project uses a database backend, which provides the ability to store data in a structured and organized manner. The use of a database backend ensures that data is stored securely and can be retrieved efficiently when required. The database is designed to be highly scalable, allowing it to handle large volumes of data and enable fast retrieval and analysis of data.

- ➢ The project also includes a range of security features to ensure that data is stored and managed securely. These features include user authentication, data encryption, and access controls, which ensure that only authorized users can access the data. The application also includes data backup and recovery features, which ensure that data is not lost in the event of a system failure or disaster.

- ➢ In summary, the project is a powerful and comprehensive web application that leverages the strengths of ReactJS, NodeJS, and a database backend to provide a flexible and user-friendly platform for building layout app creation. The use of modern web technologies ensures that the application is fast, efficient, and can be accessed from any device. The real-time collaboration feature enables multiple users to work on a layout simultaneously, making it easy to collaborate on complex projects.

Table of Contents

TABLE OF FIGURES

Chapter 1 Introduction

1.1 Problem Statement

- ➢ The problem that the building Layout guide project seeks to address is the need for a comprehensive and centralized platform that enables architects, engineers, contractors, and other stakeholders to access detailed building information quickly and easily. Currently, building details are often scattered across multiple documents and sources, making it difficult to access and maintain up-to-date information about a building. This can lead to delays, errors, and increased costs during construction and renovation projects.

- ➢ The project aims to solve these problems by providing a web application that leverages modern web technologies such as ReactJS, NodeJS, and a database backend to provide a centralized platform for building information management. The application provides features such as building information search, floor plans, blueprints, 3D models, and real-time collaboration tools, which simplify the process of accessing and managing building information. This enables stakeholders to quickly and easily access the information they need to make informed decisions, reduce errors, and improve project timelines.

- ➢ The project also includes a range of security features that ensure that data is stored and managed securely, addressing concerns around data privacy and security. This helps to ensure that sensitive building information is only accessible to authorized personnel, reducing the risk of unauthorized access and data breaches.

- ➢ In summary, the building layout guide project seeks to address the problem of inefficient and scattered building information management by providing a comprehensive and centralized platform for building information storage, access, and collaboration. The project leverages modern web technologies to provide a user-friendly and customizable interface that simplifies the process of accessing and managing building information, while also addressing concerns around data privacy and security.

1.2 Project Definition

- The Building Layout Guide is a web-based platform designed to provide a centralized repository for building information management. The application leverages modern web technologies such as ReactJS, NodeJS, and a database backend to provide a user- friendly and customizable interface for managing building information.

- The platform enables architects, engineers, contractors, and other stakeholders to access detailed building information quickly and easily. The app provides a range of features, including building information search, floor plans, blueprints, 3D models, and real-time collaboration tools, which simplify the process of accessing and managing building information. The platform also provides a customizable dashboard, allowing users to create personalized views of building data and information.

- The application includes security features that ensure that data is stored and managed securely, addressing concerns around data privacyand security. This helps to ensure that sensitive building information is only accessible to authorized personnel, reducing the risk of unauthorized access and data breaches.

1.3 Objective

➢ The objective of the Building Layout Guide project is to provide a comprehensive and user-friendly web application for people to search and find suitable apartments for rent or purchase.

➢ The app leverages modern web technologies such as ReactJS, NodeJS, and a database backend to provide a flexible and customizable platform for apartment search. The specific objectives of the project include:

- **Simplify the process of apartment search:** The app should simplify the process of searching for apartments by providing intuitive search tools, filters, and sorting options.

- **Provide a customizable platform:** The app should provide a customizable platform that can be tailored to meet the specific needs of users, such as filtering by location, price, size, and other parameters.

- **Improve accuracy and reduce errors:** The app should improve the accuracy of apartment search results by automating certain tasks, reducing the potential for human error.

- **Increase efficiency:** The app should increase the efficiency of apartment search, reducing the time and resources required for the task.

- **Enable real-time communication**: The app should enable real-time communication between users and property managers, allowing users to ask questions, schedule visits, and make inquiries about apartments.

- **Ensure data privacy and security**: The app should include security features that ensure that user data is stored and managed securely, addressing concerns around data privacy and security.

➢ In summary, the objective of the Building Layout Guide project is to provide a comprehensive and user-friendly platform for apartment search that simplifies the process, improves accuracy, increases efficiency, and enables real-time communication between users and property managers, while also addressing concerns around data privacy and security.

Chapter 2 Project Planning

Here is a project plan for the development ofthe Building Layout Guide:
1)Requirements Gathering and Analysis

- Define the project scope and requirements for the app
- Conduct market research to understand user needs and preferences
- Identify key features and functionality required for the app

2) Design and Prototyping

- Develop wireframes and prototypes to visualize the app's layout and user interface
- Define the database schema and data models required for the app
- Finalize the design and user interface of the app

3) Front-end Development

- Set up the ReactJS environment and install required libraries and dependencies
- Implement the UI design using ReactJS components, HTML, CSS, and JavaScript
- Integrate APIs and services required for the app

4) Back-end Development

- Set up the NodeJS environment and install required libraries and dependencies
- Develop the server-side APIs and services required for the app
- Implement the database integration and ensure data integrity and security

5) Testing and Quality Assurance

- Develop test cases and conduct functional testing of the app
- Conduct usability testing with real users to identify and resolve any issues
- Implement necessary improvements based on feedback and testing results

6) Deployment and Maintenance

- Deploy the app to a production environment and ensure it is accessible to users
- Implement monitoring and maintenance tools to ensure the app remains stable and secure
- Provide ongoing support and maintenance to fix anybugs or issues that arise

The timeline and duration of each phase will depend on the project scope, team size, and other factors. It is important to have a project manager or scrum master oversee the project and ensure that it is completed on time and within budget.

2.1 Gantt Chart

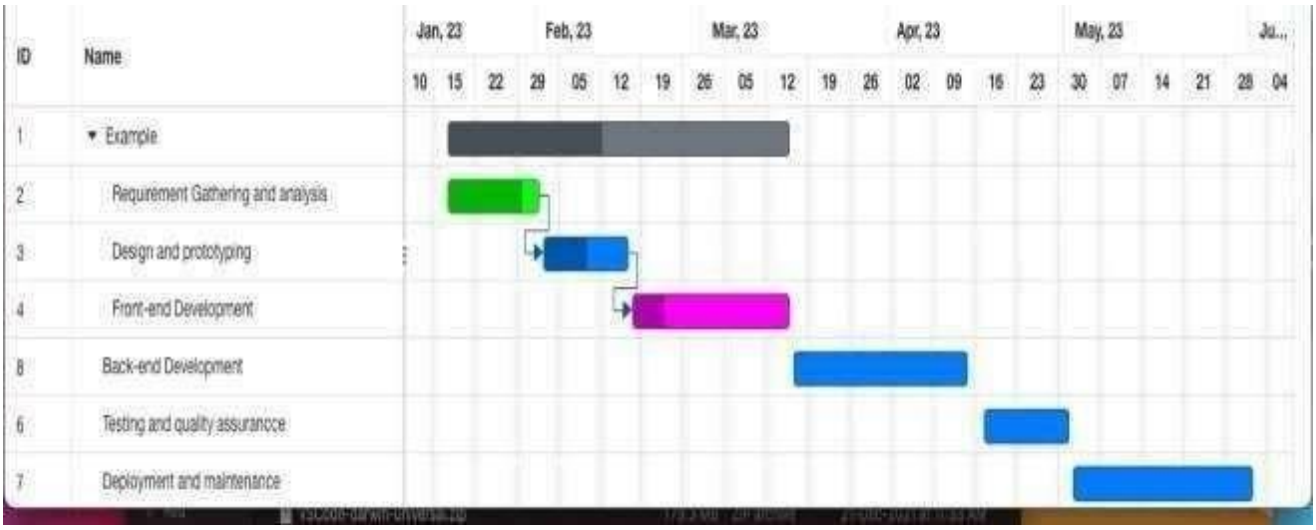


FIGURE 1 GANTT CHART

Chapter 3 Technologies and Tools Used:

3.1 Technologies

The Building Layout Guide can be developed using a wide range of technologies depending on the specific requirements of the project.

Here are some of the technologies that can be used to build the app:

1) **Front-end development:**

- **ReactJS:** A popular JavaScript library used for buildinguser interfaces.
- **HTML/CSS:** The markup and styling languages used for structuring andpresenting the user interface.
- **JavaScript:** The programming language used to add interactivity and dynamic behavior to the user interface.

FIGURE 2 FRONT END TECHNOLOGIES

2) **Back-end development:**

- **NodeJS:** A server-side JavaScript runtime environment that allows forbuilding scalable and high-performance web applications.
- **ExpressJS:** A web application framework for NodeJS that simplifies the development of server-side APIs and services.

- **MongoDB:** A NoSQL database that can be used to store and manage the app's data.

3) **Other technologies:**

- **RESTful APIs**: A standard protocol for building web services that can be used to exchange data between the front-end and back-end.
- **Git:** A version control system that can be used to manage code changes and collaborate with other developers. Heroku: A cloud platform that can be used to deploy and host the app.
- **Amazon Web Services (AWS)**: A cloud computing platform that can be used to store and manage data, and provide other services such as authentication and authorization.

Using these technologies and tools, developers can build a robust and scalable Building Layout Guide that meets the requirements of the project and provides a user-friendly and efficient platform for apartment search.

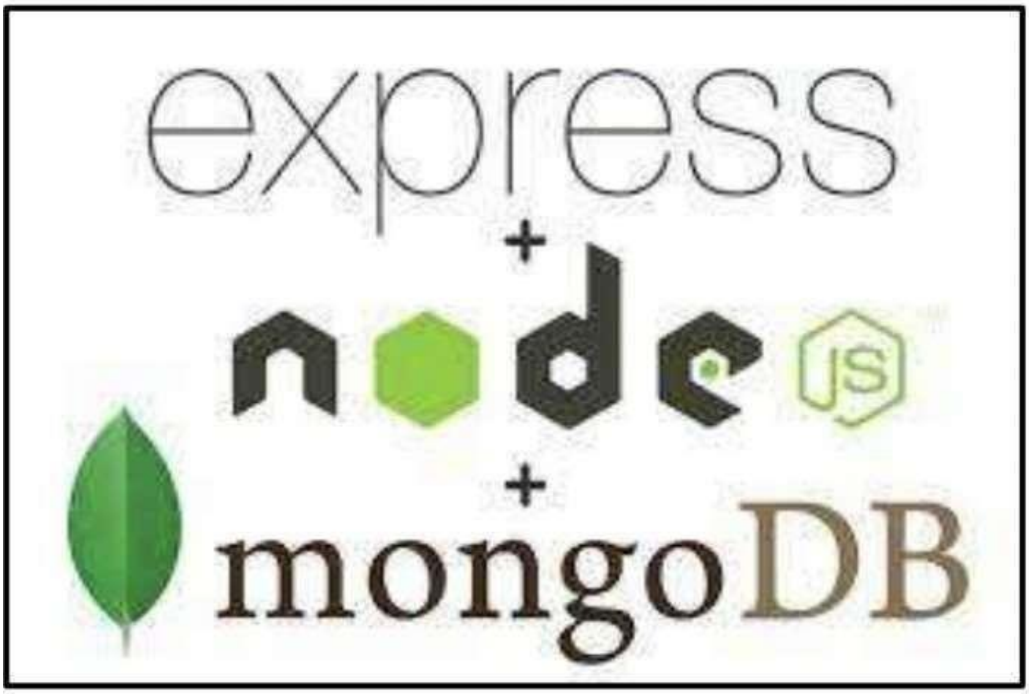


FIGURE 3 NODE AND MONGODB

3.2 Tools

Here are some of the tools that can be used to build the Search Apartments App:

1)Integrated Development Environment (IDE):

- **Visual Studio Code:** A popular open-source code editor that supports multiple programming languages and provides features such as syntax highlighting, code completion, and debugging.

2) Front-end development:

- **ReactJS:** A JavaScript library used for building user interfaces.
- **Bootstrap:** A popular front-end framework for building responsive and mobile-first web applications.
- **Material-UI:** A React-based UI library that provides pre-designed components and themes for building modern and attractive user interfaces.

3) Back-end development:

- **NodeJS:** A server-side JavaScript runtime environment.

FIGURE 4 TOOLS

Chapter 4 Requirements

4.1 Functional Requirements

- ➢ **User Registration and Authentication**: The app should allow users to create a new account or login with an existing account to access the app's features.

- ➢ **Search Functionality:** Users should be able to search for apartments based on various criteria, such as location, price, number of bedrooms, and amenities.

- ➢ **Apartment Listings:** The app should display a list of available apartments that match the user's search criteria, along with details such as photos, descriptions, and contact information.

- ➢ **Favorites:** Users should be able to save apartments to a "favorites" list for future reference.

- ➢ **Reviews and Ratings**: The app should allow users to leave reviews and ratings for apartments they have visited.

- ➢ **Notifications:** The app should send notifications to users about new listings or updates to their saved apartments.

4.2 Non Functional Requirements

- ➢ **Performance:** The app should be fast and responsive, with quick loading times and minimal lag.
- ➢ **Security:** The app should be secure, with measures in place to protect user data and prevent unauthorized access.
- ➢ **Usability:** The app should be user-friendly, with a clear and intuitive interface that is easy to navigate.
- ➢ **Reliability:** The app should be reliable and available at all times, with minimal downtime or errors.
- ➢ **Scalability:** The app should be able to handle a large number of users and listings, with the ability to scale up as needed.
- ➢ **Accessibility**: The app should be accessible to users with disabilities, with features such as screen readers and keyboard navigation.

4.3 Hardware Requirements

Here, mentioned below specifications are needed for smooth execution of the project.

- ➢ Windows 10
- ➢ Intel i7-7500u
- ➢ 8 GB RAM
- ➢ 256 GB SSD

CHAPTER 5 IMPLEMENTATIONS

1) Login

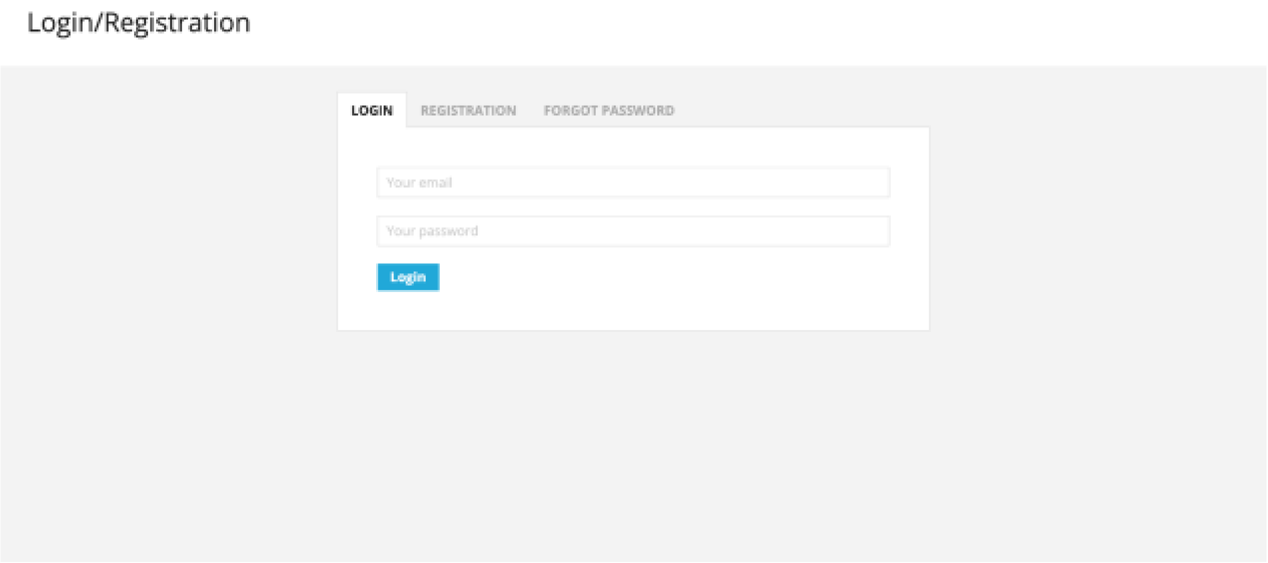


FIGURE 5 LOGIN

2) Registration

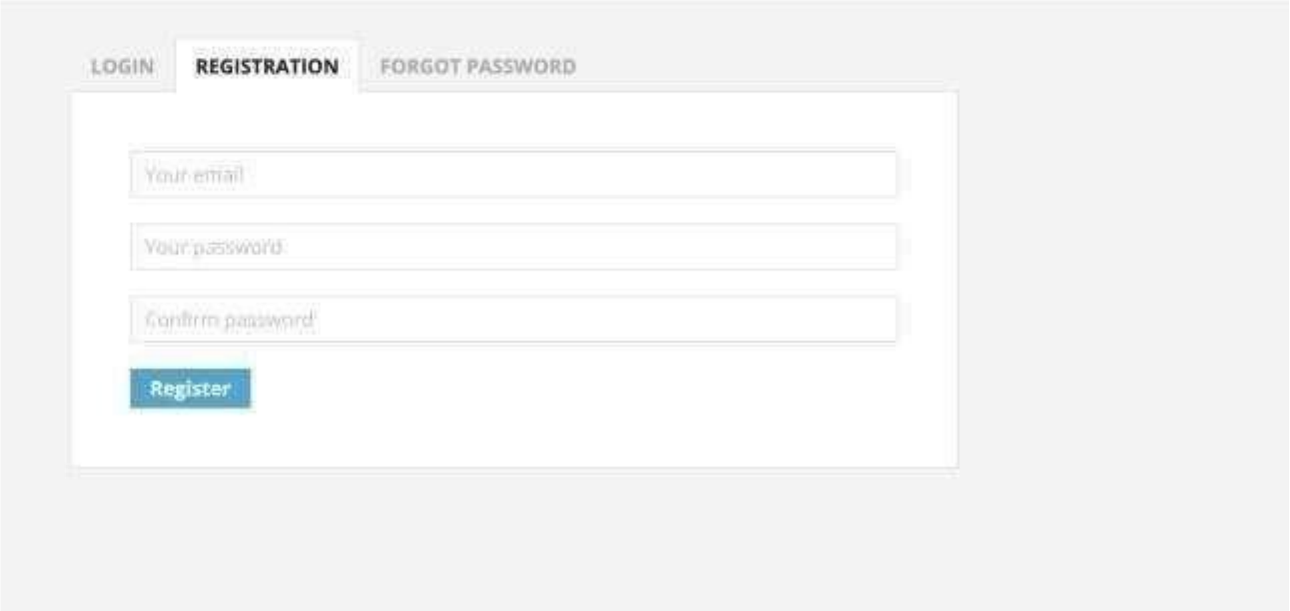


FIGURE 6 REGISTRATION

3) Forgot Password

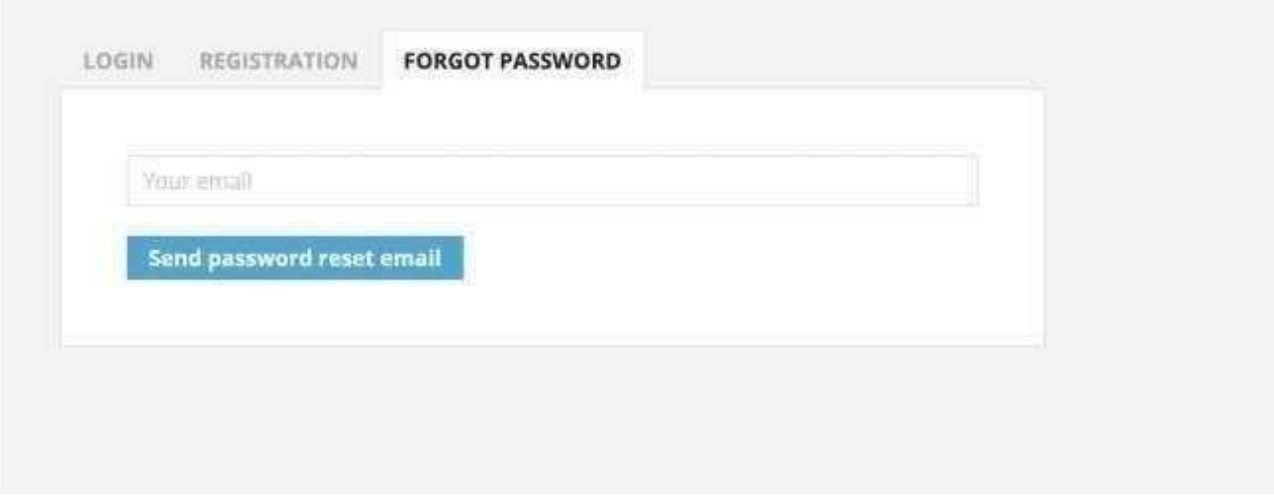
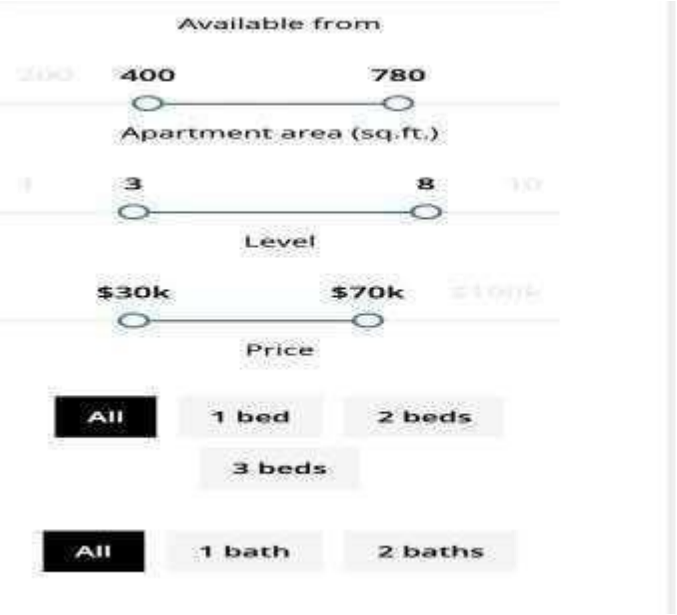


FIGURE 7 FORGOT PASSWORD

4) Filter


Available from

200 400 780

Apartment area (sq.ft.)

1 3 8 10

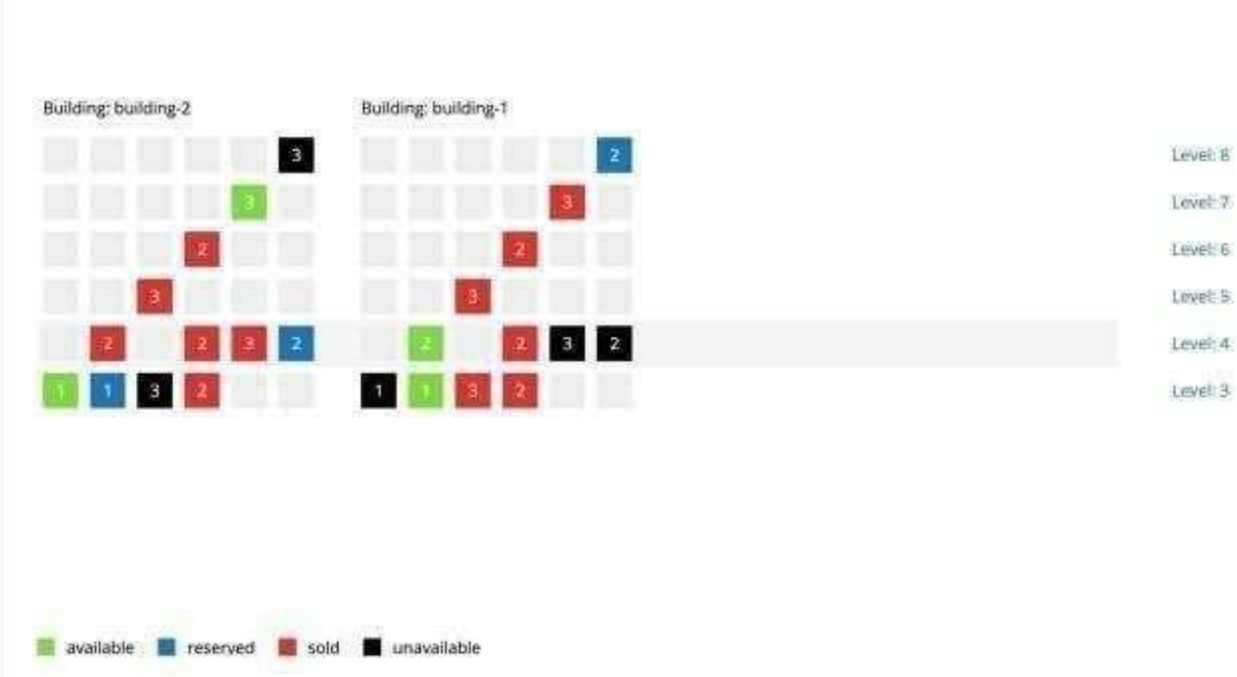
Level

$30k $70k $100k

Price

FIGURE 8 FILTER

5) Scheme



Building: building-2						Building: building-1						
					⬛ 3						🟦 2	Level: 8
				🟩 3						🟥 3		Level: 7
			🟥 2						🟥 2			Level: 6
		🟥 3						🟥 3				Level: 5
	🟥 2		🟥 2	🟥 3	🟦 2		🟩 2		🟥 2	⬛ 3	⬛ 2	Level: 4
🟩 1	🟦 1	⬛ 3	🟥 2			⬛ 1	🟩 1	🟥 3	🟥 2			Level: 3

🟩 available 🟦 reserved 🟥 sold ⬛ unavailable

FIGURE 9 SCHEME

6) Building View

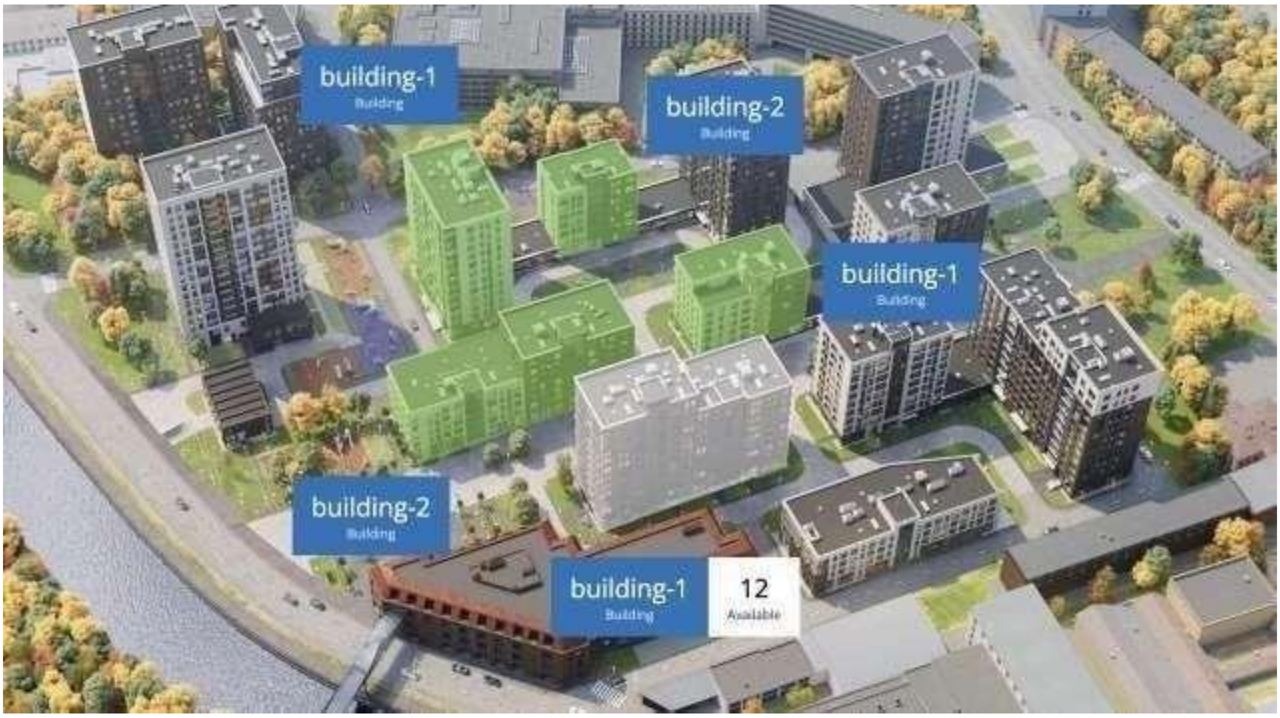
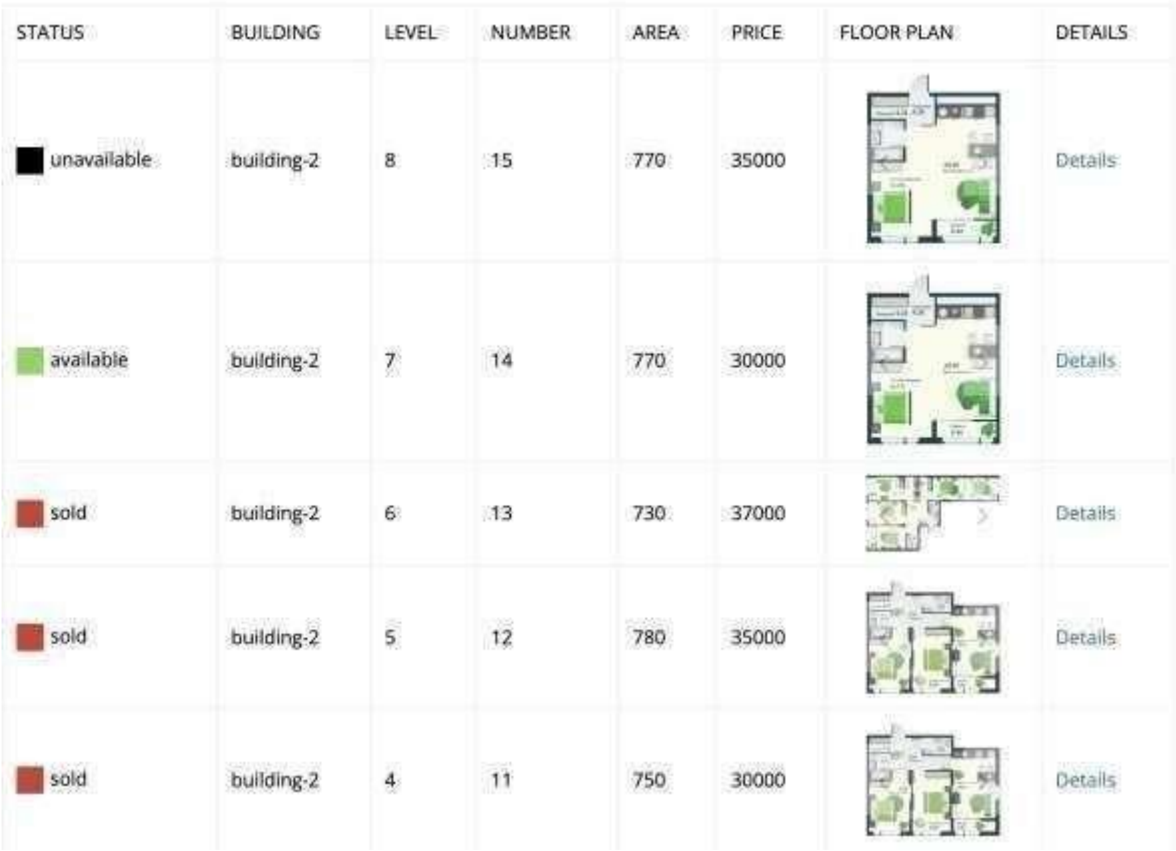


FIGURE 10 BUILDING VIEW

7) Availability



STATUS	BUILDING	LEVEL	NUMBER	AREA	PRICE	FLOOR PLAN	DETAILS
⬛ unavailable	building-2	8	15	770	35000		Details
🟩 available	building-2	7	14	770	30000		Details
🟥 sold	building-2	6	13	730	37000		Details
🟥 sold	building-2	5	12	780	35000		Details
🟥 sold	building-2	4	11	750	30000		Details

FIGURE 11 AVAILABILITY

8) Community

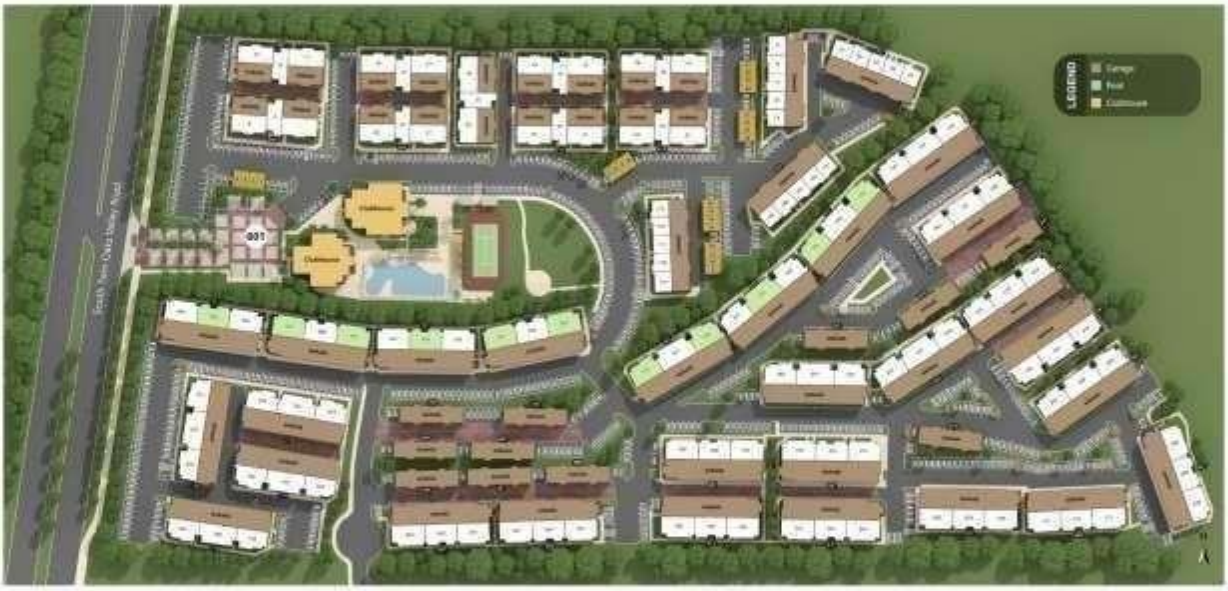


FIGURE 12 COMMUNITY

5.2 Cocomo Model

The COCOMO (Constructive Cost Model) is a model used to estimate the effort, cost, and time required to develop software. The model uses a set of factors that influence the development effort, such as the size of the project, the complexity of the software, and the experience of the development team. To apply the COCOMO model for a Building Layout Guide, we would need to consider the following factors:

- ➢ **Size of the project:** This refers to the number of lines of code required to build the application. The more complex and feature-rich the application, the more lines of code it will require.

- ➢ **Complexity of the software:** The complexity of the software refers to the difficulty of designing and implementing the application. For a Building Layout Guide, this could be influenced by factors such as the number of search filters, the ability to save searches, and the integration of third-party APIs for map and location data.

- ➢ **Required reliability:** This refers to the level of reliability needed in the application. For a Building Layout Guide users would expect reliable and accurate search results, which would require more extensive testing and quality assurance efforts.

- ➢ **Required development experience:** This refers to the level of experience required by the development team to build the application. Developing a search apartment app would require expertise in mobile app development, user experience design, and possibly data analysis.

Using these factors, we can estimate the effort required to build a building layout guide using the COCOMO model. However, it's worth noting that this model is not always accurate and may require adjustments based on the specific requirements and constraints of the project.

here's a demonstration of a COCOMO model for a Building Layout Guide:

- ➢ **Size of the project:** To estimate the size of the project, we can use lines of code (LOC) as a measure. Let's say that the Building Layout Guide has 10000 lines of code.

- ➢ **Complexity of the software:** For Building layout guide, we can assume a medium level of complexity. The app will require features such as search filters, location-based search results, and a user profile system. We can estimate the complexity factor as 1.15.

- ➢ **Required reliability**: We can assume that the building layout guide will require a high level of reliability, as users will expect accurate and up-to-date search results. We can estimate the reliability factor as 1.20.

- ➢ **Required development experience:** To App Building Layout guide , we will need a team with expertise in mobile app development, user experience design, and possibly data analysis. We can estimate the experience factor as 1.10.

Using these factors, we can calculate the effort required to develop the search apartment app as follows:
Effort = 2.94 * (LOC^1.15) * 1.20 * 1.10
Effort = 2.94 * (10000^1.15) * 1.20 * 1.10
Effort = 34.9 person-months

5.3 Theory

1) React Js

ReactJS is an open-source JavaScript library used for building user interfaces (UIs) for web and mobile applications. It was developed and is maintained by Facebook and a community of developers. ReactJS allows developers to build complex UIs with reusable and modular components.

One of the key features of ReactJS is its use of a virtual DOM (Document Object Model). The virtual DOM is a lightweight representation of the actual DOM, which allows React to efficiently update and render components when there are changes in the data or state of the application. This helps to improve performance and minimize the number of updates needed to the actual DOM, resulting in a smoother user experience.

ReactJS also uses a declarative programming model, which allows developers to describe how the UI should look based on the current state of the application, rather than having to manually manipulate the DOM. This makes it easier to write and maintain code, as well as reducing the likelihood of errors.

Another benefit of ReactJS is its large ecosystem of tools and libraries. This includes popular state management tools like Redux and MobX, as well as numerous other packages and plugins that make it easier to build complex applications.

Overall, ReactJS is a powerful and flexible library for building high-performance user interfaces, and it has become a popular choice for web and mobile application development.

2) Node Js

Node.js is an open-source, cross-platform JavaScript runtime environment that enables developers to build scalable and efficient network applications. Node.js is built on top of the Google Chrome V8 JavaScript engine, which provides fast execution of JavaScript code.

One of the key features of Node.js is its event-driven, non-blocking I/O model, which allows for efficient handling of multiple simultaneous connections. This makes Node.js well-suited for building real-time applications that require fast and frequent data exchanges, such as chat applications, online gaming platforms, or stock market tickers.

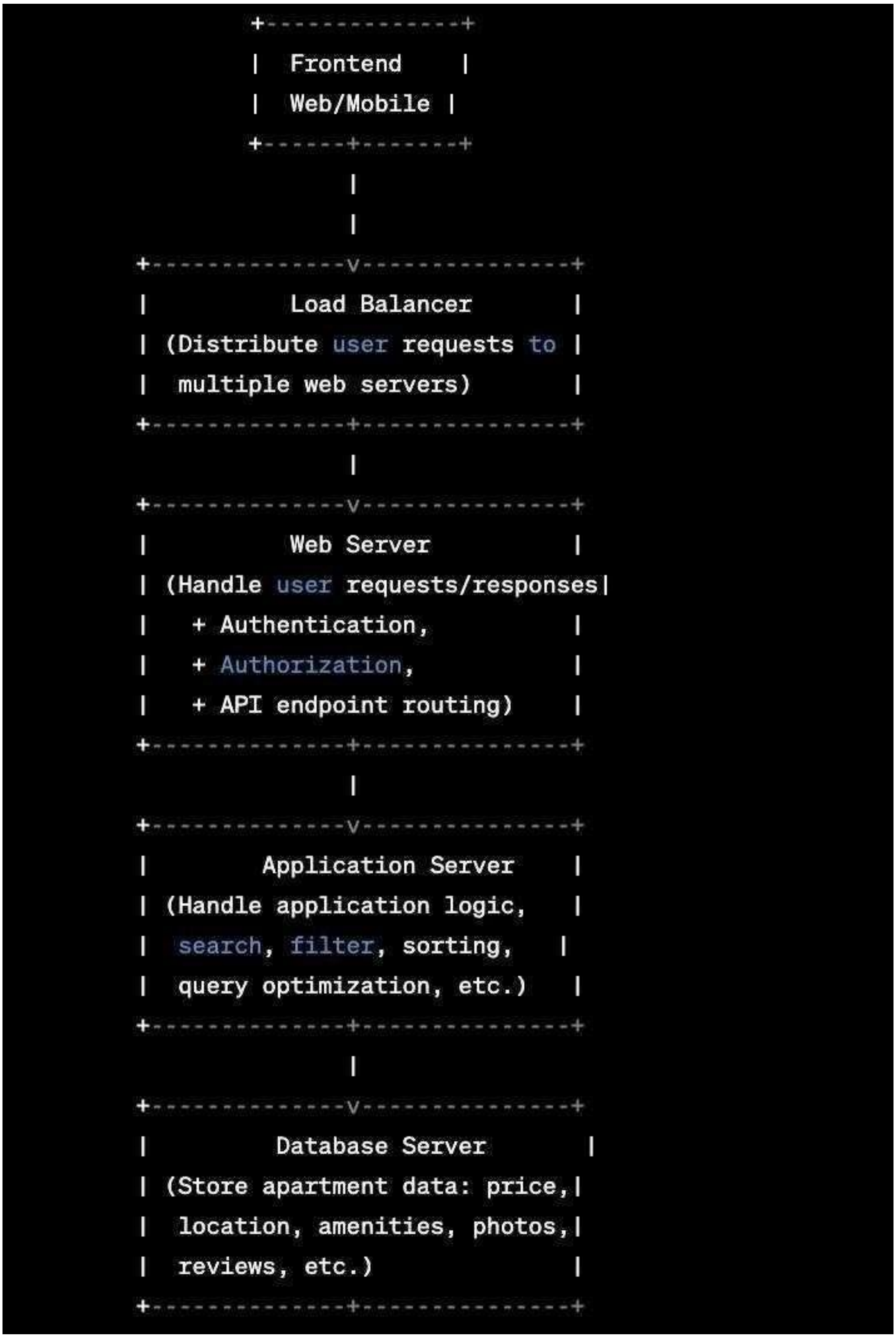
Node.js also has a large and active ecosystem of libraries and tools, which makes it easy for developers to build complex applications. The Node Package Manager (NPM) is a popular package manager that comes bundled with Node.js, and it provides access to over a million packages that can be easily installed and integrated into applications.

Node.js is widely used in the industry, especially for building server-side applications such as web servers, APIs, and microservices. It is also used for front-end development in combination with popular front-end frameworks such as React, Angular, and Vue.

Overall, Node.js is a versatile and powerful platform for building high-performance network applications that require scalability, real-time communication, and efficient handling of I/O operations.

CHAPTER 6 UML DIAGRAMS

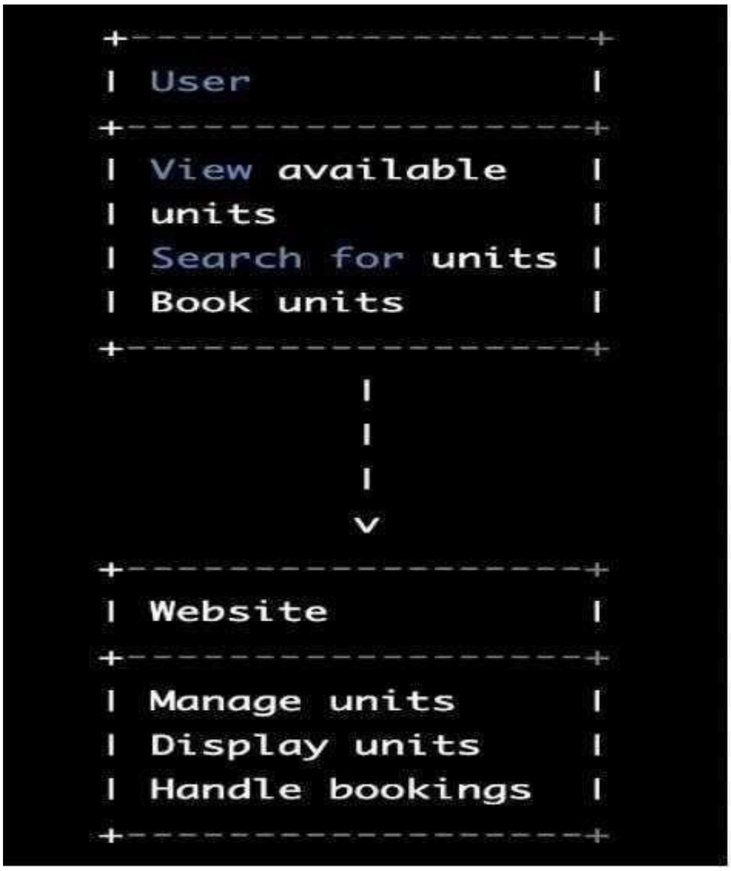
6.1 System design



```
          +-------------+
          |  Frontend    |
          |  Web/Mobile |
          +------+------+
                 |
                 |
+-------------v-------------+
|          Load Balancer      |
| (Distribute user requests to |
|  multiple web servers)      |
+-------------+-------------+
                 |
+-------------v-------------+
|          Web Server         |
| (Handle user requests/responses|
|    + Authentication,        |
|    + Authorization,         |
|    + API endpoint routing)   |
+-------------+-------------+
                 |
+-------------v-------------+
|        Application Server    |
| (Handle application logic,   |
|  search, filter, sorting,   |
|  query optimization, etc.)   |
+-------------+-------------+
                 |
+-------------v-------------+
|         Database Server       |
| (Store apartment data: price,|
|  location, amenities, photos,|
|  reviews, etc.)              |
+-------------+-------------+
```

FIGURE 13 SYSTEM DESIGN

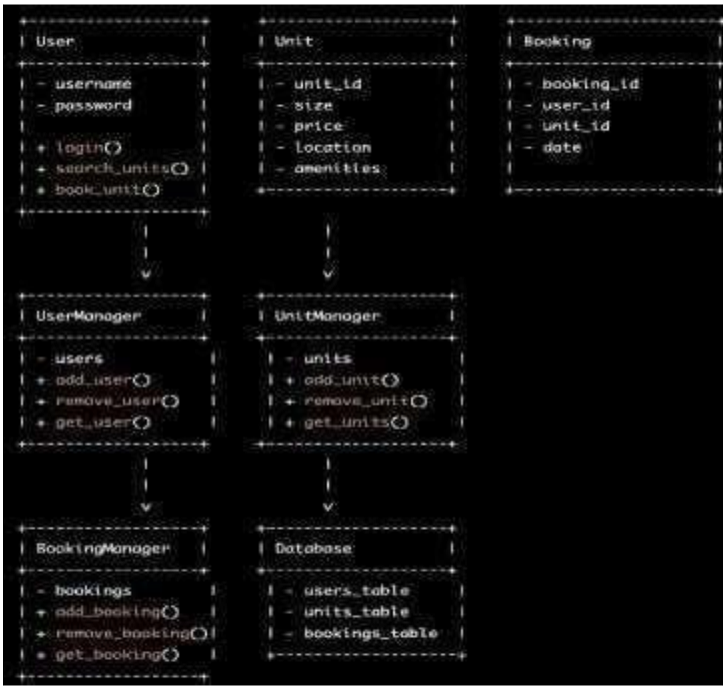
6.2 Use Case Diagram



```
+------------------+
| User             |
+------------------+
| View available   |
| units            |
| Search for units |
| Book units       |
+------------------+
          |
          |
          |
          v
+------------------+
| Website          |
+------------------+
| Manage units     |
| Display units    |
| Handle bookings  |
+------------------+
```

FIGURE 14 USE CASE DIAGRAM

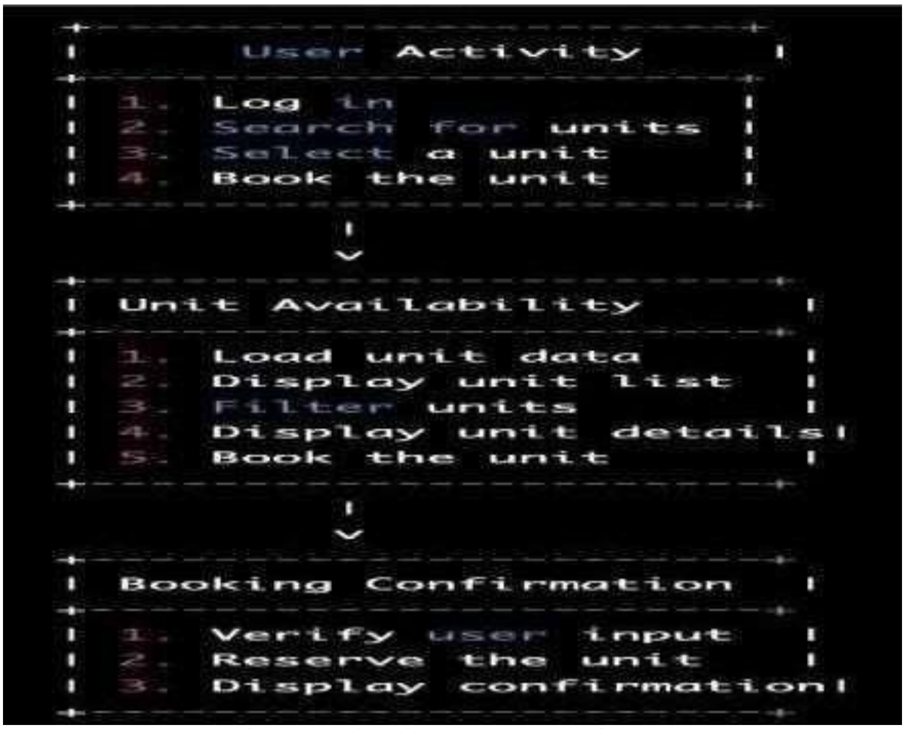
6.3 Class Diagram



```
+-------------------+   +-------------------+   +-------------------
| User              |   | Unit              |   | Booking           
+-------------------+   +-------------------+   +-------------------
| - username        |   | - unit_id         |   | - booking_id      
| - password        |   | - size            |   | - user_id         
|                   |   | - price           |   | - unit_id         
| + login()         |   | - location        |   | - date            
| + search_units()  |   | - amenities       |   |                   
| + book_unit()     |   +-------------------+   +-------------------
+-------------------+
          |                       |
          |                       |
          v                       v
+-------------------+   +-------------------+
| UserManager       |   | UnitManager       |
+-------------------+   +-------------------+
| - users            |   | - units            |
| + add_user()       |   | + add_unit()       |
| + remove_user()    |   | + remove_unit()    |
| + get_user()       |   | + get_units()      |
+-------------------+   +-------------------+
          |                       |
          |                       |
          v                       v
+-------------------+   +-------------------+
| BookingManager    |   | Database          |
+-------------------+   +-------------------+
| - bookings         |   | - users_table      |
| + add_booking()    |   | - units_table      |
| + remove_booking()|   | - bookings_table   |
| + get_booking()    |   +-------------------+
+-------------------+
```

FIGURE 15 CLASS DIAGRAM

6.4 Activity Diagram



```
+---------------------------+
|        User Activity      |
+---------------------------+
| 1. Log in                 |
| 2. Search for units       |
| 3. Select a unit          |
| 4. Book the unit          |
+---------------------------+
              |
              v
+-----------------------------+
| Unit Availability           |
+-----------------------------+
| 1. Load unit data           |
| 2. Display unit list        |
| 3. Filter units             |
| 4. Display unit details|
| 5. Book the unit            |
+-----------------------------+
              |
              v
+-----------------------------+
| Booking Confirmation        |
+-----------------------------+
| 1. Verify user input        |
| 2. Reserve the unit         |
| 3. Display confirmation|
+-----------------------------+
```

FIGURE 16 ACTIVITY DIAGRAM

CHAPTER 7 CONCLUSION

In conclusion, a website that provides information about available units in a building can be a useful tool for both potential buyers and real estate agents. The website should be designed with a user-friendly interface that allows users to search for units based on various criteria, such as size, price.

To build such a website, a frontend developer would be responsible for creating the user interface and integrating it with the backend systems. The backend would consist of a database to store information about the units and a server to handle user requests and perform data processing.

CHAPTER 8 REFERENCES

- ➢ Martin, J., & Brown, S. (2019). Building Web Applications with HTML5, CSS3, and Javascript: An Introduction to HTML5, CSS3, and Javascript. CRC Press.
- ➢ Van den Ende, J., & Mooij, M. (2020). Web Application Architecture: Principles, Protocols, and Practices. CRC Press.
- ➢ Freeman, E., & Robson, E. (2014). Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages. O'Reilly Media, Inc.
- ➢ Booth, A. (2015). Web Design: A Complete Introduction. John Wiley & Sons.
- ➢ W3Schools. (2022). HTML Tutorial. Retrieved from https://www.w3schools.com/html/
- ➢ MDN Web Docs. (2022). CSS: Cascading Style Sheets. Retrieved from https://developer.mozilla.org/en-US/docs/Web/CSS
- ➢ Node.js. (2022). Retrieved from https://nodejs.org/
- ➢ MongoDB. (2022). Retrieved from https://www.mongodb.com/
- ➢ Apache Tomcat. (2022). Retrieved from https://tomcat.apache.org/
- ➢ MySQL. (2022). Retrieved from https://www.mysql.com/