SAVITRIBAI PHULE PUNE UNIVERSITY

A PROJECT REPORT ON

Wanderlust – Vacation Rental Platform

SUBMITTED TOWARDS THE

FINAL FULLFILLMENTS OF THE REQUIREMENTS OF

BACHELOR COMPUTER APLLICATION (SCIENCE)

BY

RUTUJA GANJE

UNDER THE GUIDANCE OF

PROG :- PRITI JADHAV

SWARAJ COLLEGE OF ARTS COMMERCE AND SCIENCE

PUNE – 411043

2025-26

INDEX

INTRODUCTION

The Airbnb Clone (Wanderlust) project is a full-stack web application developed to simulate the basic functionality of a property listing platform. The system allows users to perform various operations such as viewing property listings, adding new listings, editing existing listings, deleting listings, and posting reviews. Additionally, the system includes user authentication to ensure secure access.

This project is developed using technologies such as Node.js, Express.js, MongoDB, EJS (Embedded JavaScript), and Bootstrap. Unlike React-based applications, this system uses EJS as a templating engine to dynamically render data on web pages. Bootstrap is used to design a responsive and userfriendly interface.

In this architecture, the frontend and backend are tightly integrated. The server renders dynamic HTML pages using EJS templates, and data is fetched from MongoDB using backend logic written in Node.js and Express.js. This makes the system simple and efficient for understanding server-side rendering.

The main objective of this project is to understand how web applications work using server-side technologies. It demonstrates CRUD operations, routing, middleware usage, authentication, and database management.

An important feature of this project is that it is deployed using Render, making it accessible online. Deployment ensures that

the application works in a real-world environment and can be accessed by users over the internet.

The project focuses only on essential features and avoids advanced functionalities like booking systems or payment gateways. This keeps the system simple and suitable for academic learning.

Overall, this project provides hands-on experience in building, testing, and deploying a full-stack web application using Node.js and EJS-based architecture.

Subpoints:

- Full-stack web application
- Uses Node.js, Express.js, MongoDB
- EJS used for dynamic page rendering
- Bootstrap used for UI design
- CRUD operations implemented
- User authentication and reviews included
- Deployed on Render (live project

Need of Computerization

Computerisation plays a very important role in modern systems as it helps in improving efficiency, accuracy, and speed of operations. In traditional manual systems, data is maintained on paper or through simple methods, which can lead to problems such as data loss, duplication, and difficulty in accessing information.

In this project, computerisation is used to manage property listings in a digital format. Users can easily add, update, and delete listings using the system. This reduces manual effort and ensures that all operations are performed quickly and accurately.

Another important advantage of computerisation is that it allows easy data retrieval. Users can view listings instantly without searching through large amounts of data manually. This improves overall system performance and user experience.

Computerisation also enhances data security. With the help of authentication, only authorized users can access and modify data. This helps in maintaining data integrity and prevents unauthorized access.

In addition, computerised systems provide a better interface for users. In this project, Bootstrap is used to design a responsive and user-friendly interface which makes interaction easy and efficient.

Overall, computerisation helps in making the system faster, more reliable, and user-friendly.

Subpoints:

- Reduces manual work and human errors
- Improves speed and efficiency
- Provides secure data storage
- Enables quick data retrieval
- Enhances user experience
- Supports authentication and security

Fact Finding Techniques

Fact finding techniques are used to collect information about the system requirements and to understand how the system should be designed and developed. These techniques help the developer to gather accurate data, analyze user needs, and build a system that fulfills all required functionalities.

In this project, different fact finding techniques were used to understand how property listing platforms work and what features should be included. By studying existing systems and analyzing user requirements, a clear idea of the system design was obtained.

Fact finding plays a very important role in software development because it ensures that the system is built according to real-world requirements. Without proper fact finding, the system may not meet user expectations.

The techniques used in this project helped in identifying the need for features such as listing management, user authentication, and review system. It also helped in deciding which features should be included and which should be excluded to keep the system simple.

Subpoints:

1. Observation

Observation involves studying existing systems and understanding how users interact with them. In this project, property listing platforms were observed to understand how listings are displayed and managed.

- Helps in understanding real system behavior

- Identifies user interaction patterns
- Provides practical knowledge

2. Internet Research

Internet research involves collecting information from online sources such as tutorials, documentation, and articles.

- Helps in learning modern technologies
- Provides implementation ideas
- Improves technical knowledge

3. Requirement Analysis

Requirement analysis is the process of identifying what features the system should have.

- Identifies system needs
- Defines project scope
- Helps in planning development

4. Comparative Study

Comparative study involves comparing existing systems with the proposed system.

- Identifies strengths and weaknesses
- Helps in improving system design
- Removes unnecessary features

Study of Existing System

The study of the existing system is an important phase in system development. It helps in understanding how current systems work, what features they provide, and what limitations they have. By analyzing existing systems, developers can identify areas of improvement and design a better system.

In the context of this project, existing property listing platforms were studied to understand their functionality and working process. These systems provide various features such as property listings, booking systems, payment integration, and user reviews.

However, these systems are often complex and include many advanced features which are not required for academic projects. Therefore, studying the existing system helps in selecting only the necessary features and avoiding unnecessary complexity.

4.1 Existing System

The existing system refers to real-world platforms that provide property listing services. These platforms allow users to browse properties, view details, and perform various actions such as booking and reviewing.

These systems are designed for commercial use and include multiple modules such as user management, booking system, payment gateway, and review system. They are built to handle large amounts of data and multiple users simultaneously.

Although these systems are powerful, they are complex in nature and require advanced knowledge to understand and maintain.

Subpoints:

- Provides property listing functionality
- Includes booking and payment systems
- Supports user authentication
- Allows user reviews and ratings
- Designed for large-scale usage

4.2 Drawbacks of Existing System

Even though existing systems are feature-rich, they have certain drawbacks, especially from a learning perspective.

Subpoints:

- Complex interface which is difficult for beginners
- Includes many unnecessary features
- Hard to understand complete system flow
- Not suitable for academic learning
- Requires more resources and maintenance

4.3 Features of Existing System

Existing systems provide a wide range of functionalities that make them powerful and useful for real-world applications.

Subpoints:

- Property listing and search
- Booking functionality
- Payment integration
- User authentication and authorization
- Reviews and ratings system

Study of Proposed System

The proposed system is designed to overcome the limitations of the existing system by focusing only on the essential functionalities required for property listing management. Unlike real-world platforms, this system is simplified and developed mainly for academic and learning purposes.

The system allows users to perform CRUD operations such as creating, viewing, updating, and deleting property listings. It also includes user authentication and review functionality, making it a complete basic full-stack application.

The proposed system is designed using Node.js, Express.js, MongoDB, EJS, and Bootstrap. These technologies help in building a dynamic and interactive system where data is processed efficiently and displayed properly to the user.

The system avoids unnecessary complexity by not including advanced features such as booking systems or payment gateways. This makes it easier to understand and implement while still covering all important concepts of web development.

5.1 Need for Proposed System

The need for the proposed system arises from the complexity of existing systems. Many platforms include features that are not necessary for academic learning. Therefore, a simplified system is required.

Subpoints:

- Simplifies complex real-world systems
- Focuses on essential features only
- Easy to understand and implement

-

Suitable for academic learning

5.2 Feasibility Study

Feasibility study determines whether the system is practical and possible to implement.

A. Technical Feasibility

The system is technically feasible because it uses widely accepted technologies such as Node.js, Express.js, and MongoDB. These technologies are easy to learn and implement.

Subpoints:

- Uses modern web technologies
- Easy to develop and maintain
- Good community support

B. Economic Feasibility

The system is cost-effective as it uses open-source tools and does not require any paid software.

Subpoints:

- No licensing cost
- Uses free tools
- Can run on personal laptop

C. Operational Feasibility

The system is easy to operate and user-friendly.

Subpoints:

- Simple interface
- Easy navigation
- No training required

5.3 Drawbacks of Proposed System

Even though the system is useful, it has some limitations.

Subpoints:

- No booking functionality
- No admin panel
- Limited advanced features

5.4 Features of Proposed System

The system provides essential functionalities required for property listing management.

Subpoints:

- User Registration and Login (Authentication)
- Add new property listing
- View all listings
- Edit existing listings
- Delete listings
- Add reviews and ratings

5.5 Objectives of Proposed System

The main objectives of the system are to provide learning and practical implementation of web development concepts.

Subpoints:

- To develop a full-stack web application
- To understand CRUD operations
- To integrate frontend, backend, and database
- To provide user-friendly interface
- To implement authentication and review system

5.6 System Specifications

System specifications define the hardware and software requirements.

- Hardware Requirements:
- Processor: Intel i3 (HP Laptop)
- RAM: Minimum 4GB
  Storage: 500GB

•

Software Requirements:

- Operating System: Windows 11
- IDE: Visual Studio Code
- Technologies: Node.js, Express.js, MongoDB, EJS, Bootstrap

Technology Stack

The Wanderlust project is developed using modern web technologies that help in building a scalable and user-friendly application. The technology stack is divided into frontend, backend, database, and other supporting tools.

Frontend Technologies

The frontend of the application is responsible for designing the user interface and improving user experience.

- HTML (Hypertext Markup Language):

Used to create the basic structure of web pages.

- CSS (Cascading Style Sheets):
  Used to style the web pages and improve visual appearance.
- Bootstrap:
  A CSS framework used to create responsive and mobile-friendly designs quickly.
- EJS (Embedded JavaScript):
  A templating engine used to generate dynamic HTML pages by embedding JavaScript code.
- EJS-Mate:

  Used for layout management and reusability of templates like header and footer.

Backend Technologies

The backend handles server-side logic and processes user requests.

- Node.js:

A JavaScript runtime environment used to run server-side code efficiently.

- Express.js:

A web framework for Node.js used to build APIs and handle routing.

- Express Router:

Helps in organizing routes into separate files for better structure.

- Middleware:

Functions that run between request and response, used for authentication, validation, and error handling.

Database

The database is used to store and manage application data.

- MongoDB:

A NoSQL database used to store data in flexible JSON-like format.

- MongoDB Atlas:

A cloud-based database service that allows remote access and easy deployment.

Mongoose:

An object data modelling (ODM) library used to interact with MongoDB easily.

Authentication & Authorization

Security is an important part of the application.

- Authentication:

Used to verify user identity (Login/Signup system).

- Authorization:

Used to control access, ensuring only authorized users can edit or delete data.

- Prebuilt Methods:

Used for simplifying authentication and authorization implementation.

Validation

Joi:

Used for validating user input data to prevent errors and improve security.

- **Map Integration**

Leaflet :

Used to display location of listings on the map.

- **Architecture**

MVC (Model-View-Controller):

Used to organize code into three parts:

Model → Data handling

View → User interface

Controller → Business logic

- **Version Control** Git &

GitHub:

Used for version control and storing project code.

System Design

System Design is an essential step in the development of a software application. It is the blueprint of the system which defines the system architecture, components, and their interactions. The goal of the system design phase is to plan how the system will work, ensuring it meets all requirements defined earlier in the project.

In this project, we focused on designing a web application that manages property listings. The system design provides a clear structure of how the components will interact with each other, and how the flow of data will happen. Proper system design ensures the smooth functioning of the application by allowing efficient data handling, secure access, and seamless user experience.

We will explain the system design through several diagrams that represent different components of the system, including their relationships and interactions.

7.1 Entity Relationship Diagram (ERD)

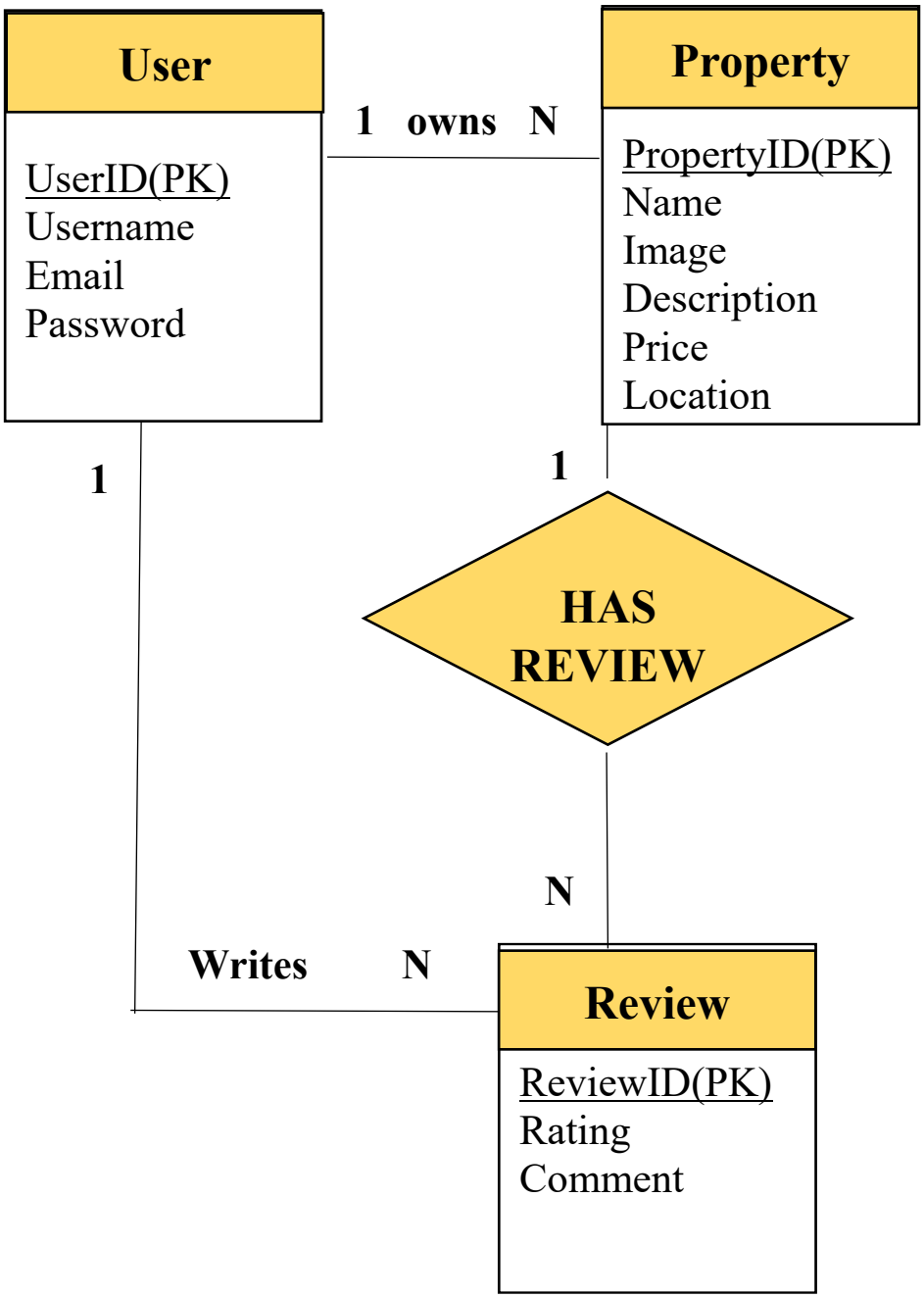
An Entity-Relationship Diagram (ERD) is used to visually represent the data model. It shows the relationships between various entities in the system. In the context of this project, the entities include Users, Properties, Reviews, and Bookings.

Each entity is represented by a box and is linked with other entities through relationships. For example, a User can have multiple Reviews, and a Property can have many Bookings.

Subpoints:

- User (One-to-Many) Reviews
- User (One-to-Many) Bookings
- Property (One-to-Many) Bookings
- Property (One-to-Many) Reviews

ERD


User
UserID(PK)
Username
Email
Password
1 owns N
Property
PropertyID(PK)
Name
Image
Description
Price
Location
1
1
HAS REVIEW
N
Writes N
Review
ReviewID(PK)
Rating
Comment

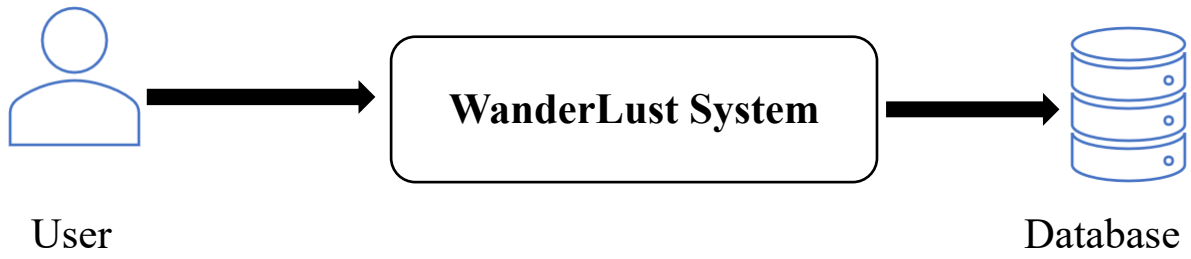
7.2 Data Flow Diagram (DFD)

A Data Flow Diagram (DFD) illustrates how data moves through the system. It shows the flow of information between the user, system, and database.

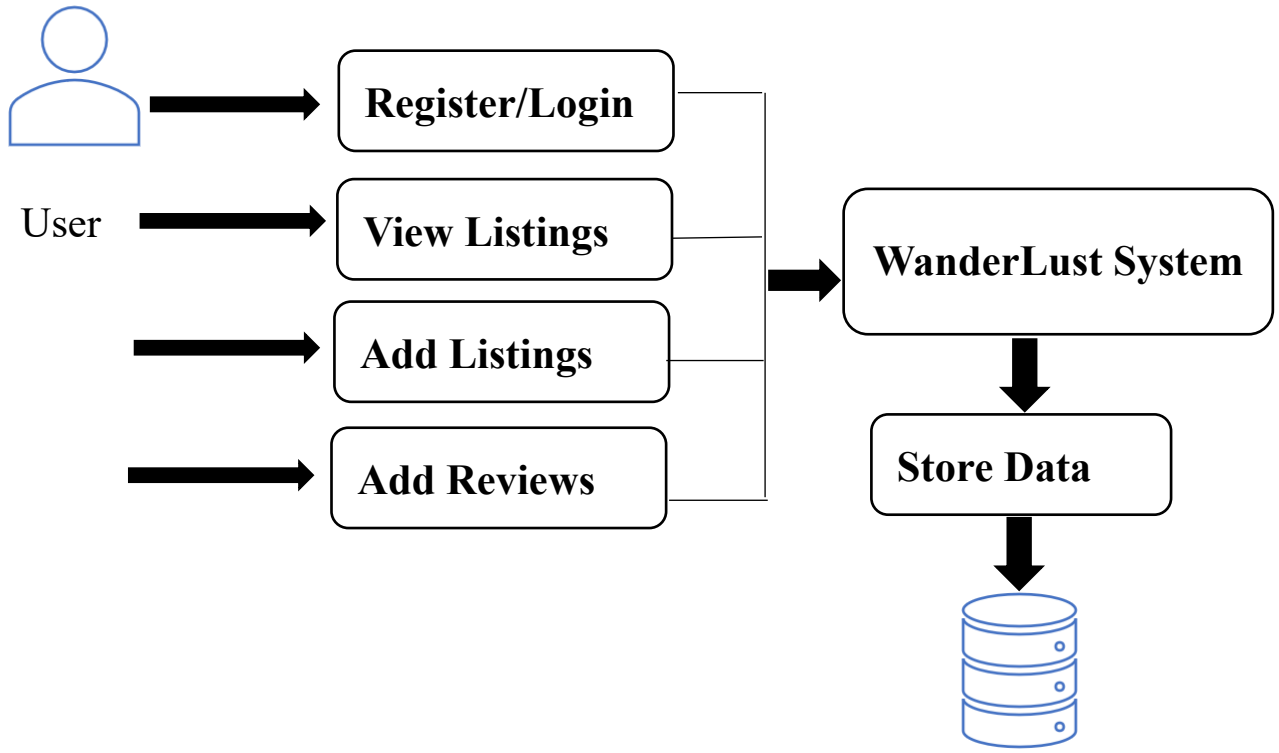
For example, when a user adds a new property listing, the data flow starts from the user interface and is passed through the application logic before being stored in the database.

Subpoints:

Level 0 DFD (Overview of the entire system flow)



Level 1 DFD (Detailed flow for individual modules like listing, booking, and review)

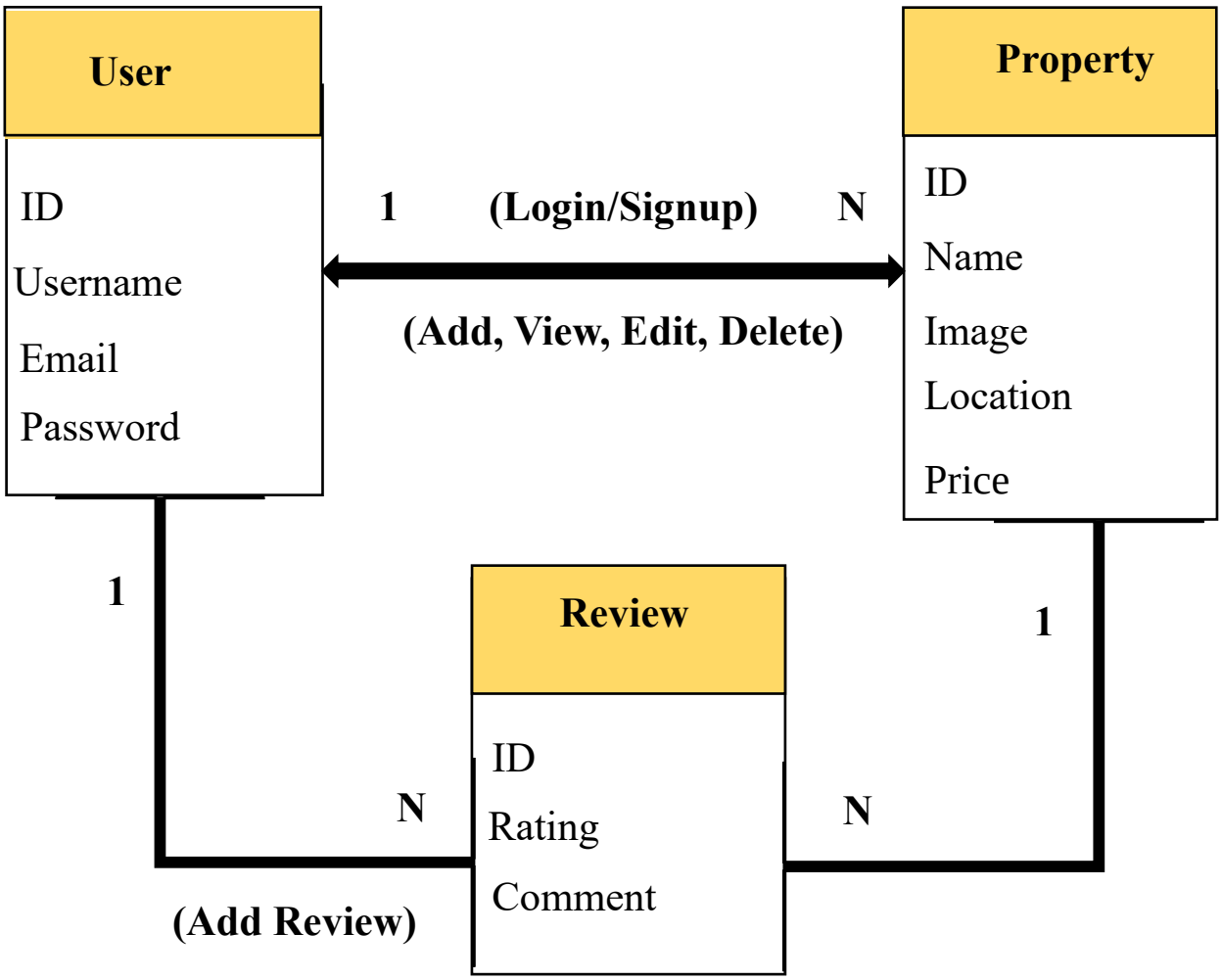


7.3 Class Diagram

A class diagram shows the structure of the system by representing the system's classes and their relationships. In this case, the classes include User, Property, Review, etc. Each class has attributes and methods that describe its state and behaviour.

Subpoints:

- User Class: Attributes: name, email, password; Methods: login, register
- Property Class: Attributes: name, description, price; Methods: add, edit, delete
- Review Class: Attributes: rating, comment; Methods: add, edit

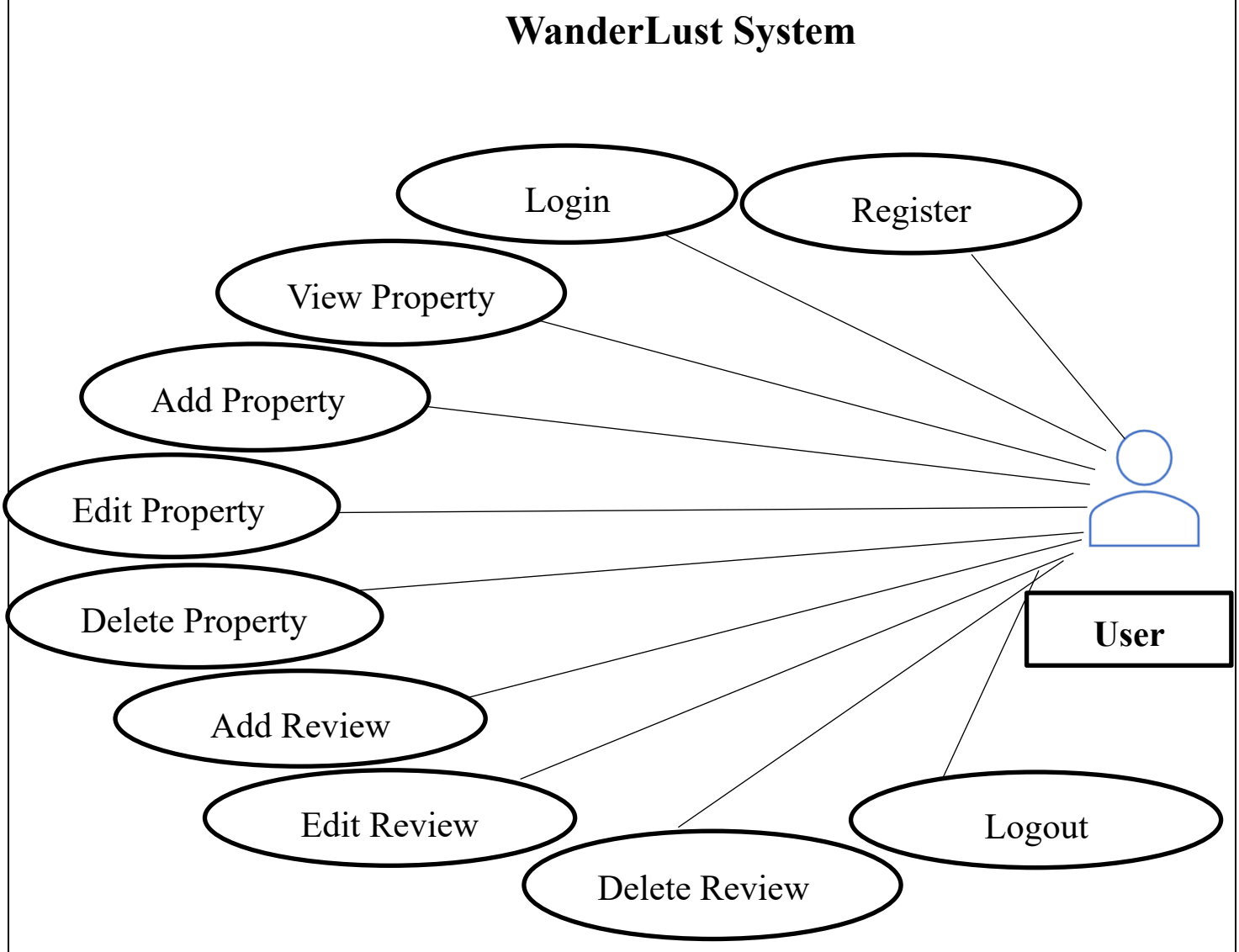


7.4 Use Case Diagram

A Use Case Diagram represents the system's functionalities from a user's perspective. It defines what the users can do with the system. For example, a User can Add a Property, View Listings, Edit a Property, and Write a Review.

Subpoints:

- Actors: User
- Use Cases: Add Property, View Property, Edit Property, Write Review
- System Boundary: Limits the actions of users to within the app's functionality

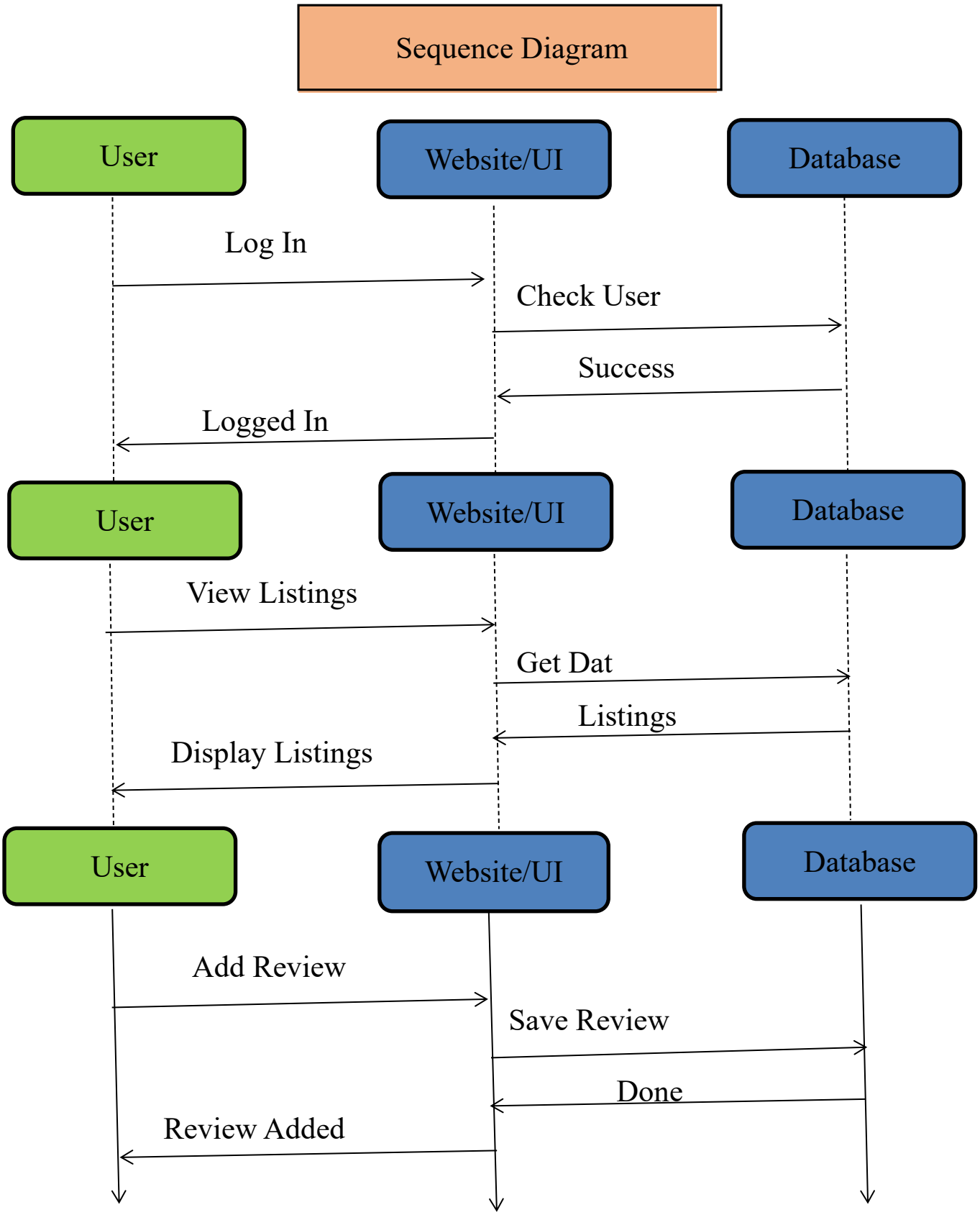


7.5 Sequence Diagram

A Sequence Diagram illustrates how objects interact in a sequential manner. It shows the order of operations between objects for a specific process. For example, for the User Registration process, the sequence involves the user entering details, the system validating the details, and then the data being saved in the database.

Subpoints:

- Sequence of User Registration
- Sequence of Property Listing
- Sequence of Review


Sequence Diagram
User
Website/UI
Database
Log In
Check User
Success
Logged In
User
Website/UI
Database
View Listings
Get Dat
Listings
Display Listings
User
Website/UI
Database
Add Review
Save Review
Done
Review Added

Database Design

Database Design refers to the structure of storing data in an organized manner. It helps to manage user records, property listings, and reviews efficiently. In the WanderLust system, MongoDB database is used for storing all information securely.

Subpoints:

1.User Collection

Field Name	Description
User ID	Unique ID of user
Username	User name
Email	Email address
Password	Encrypted password

2.Property Collection

Field Name	Description
Property ID	Unique property ID
Title	Property title
Description	Property details
Image	Property image
Price	Rent Price
Location	Property Location

Owner ID	User reference ID

3.Review Collection

Field Name	Description
Review ID	Unique review ID
Rating	Rating given by user
Comment	Review message
User ID	Review writer ID
Property ID	Property reference ID

Relationships:

- One User can own many Properties.
- One Property can have many Reviews.
- One User can write many Reviews.

Benefits of Database Design:

- Easy data storage and retrieval
- Better security
- Organized records
- Fast system performance
- Easy maintenance

Testing

Testing is the process of checking whether the system is working properly or not. It helps to identify errors and ensure that all features of the WanderLust system function correctly.

Subpoints:

9.1 Objective of Testing

- To verify system performance
- To detect bugs and errors
- To ensure user-friendly functionality
- To check security and reliability

9.2 Types of Testing Used

1. Unit Testing:

Testing individual modules like Login, Register, Add Property.

2. Integration Testing:

Testing connection between frontend, backend, and database.

3. System Testing:

Testing the complete WanderLust website as one system.

4. User Acceptance Testing:

Checking whether the system meets user requirements.

9.3 Test Cases

Test Case	Expected Result
User Registration	User account created successfully
Login	User Logged in successfully
Add Property	Property added to listings
Edit Property	Property Updated
Delete Property	Property removed

Add Review	Review submitted successfully

9.4 Result of Testing

All modules were tested successfully. The system works properly with accurate output.

9.5 Benefits of Testing

- Improves quality
- Removes errors
- Increases reliability
- Better user experience
- Smooth performance

Limitation And Future Modification

Every system has some limitations in the present version. These limitations can be improved in future updates. The WanderLust system can be enhanced by adding more advanced features later.

Subpoints:

10.1 Limitations of Current System

- Limited payment options
- Booking system is not implemented currently
- Search and filter functions are UI level
- No live chat support between user and owner
- Basic search and filter options

10.2 Future Modifications

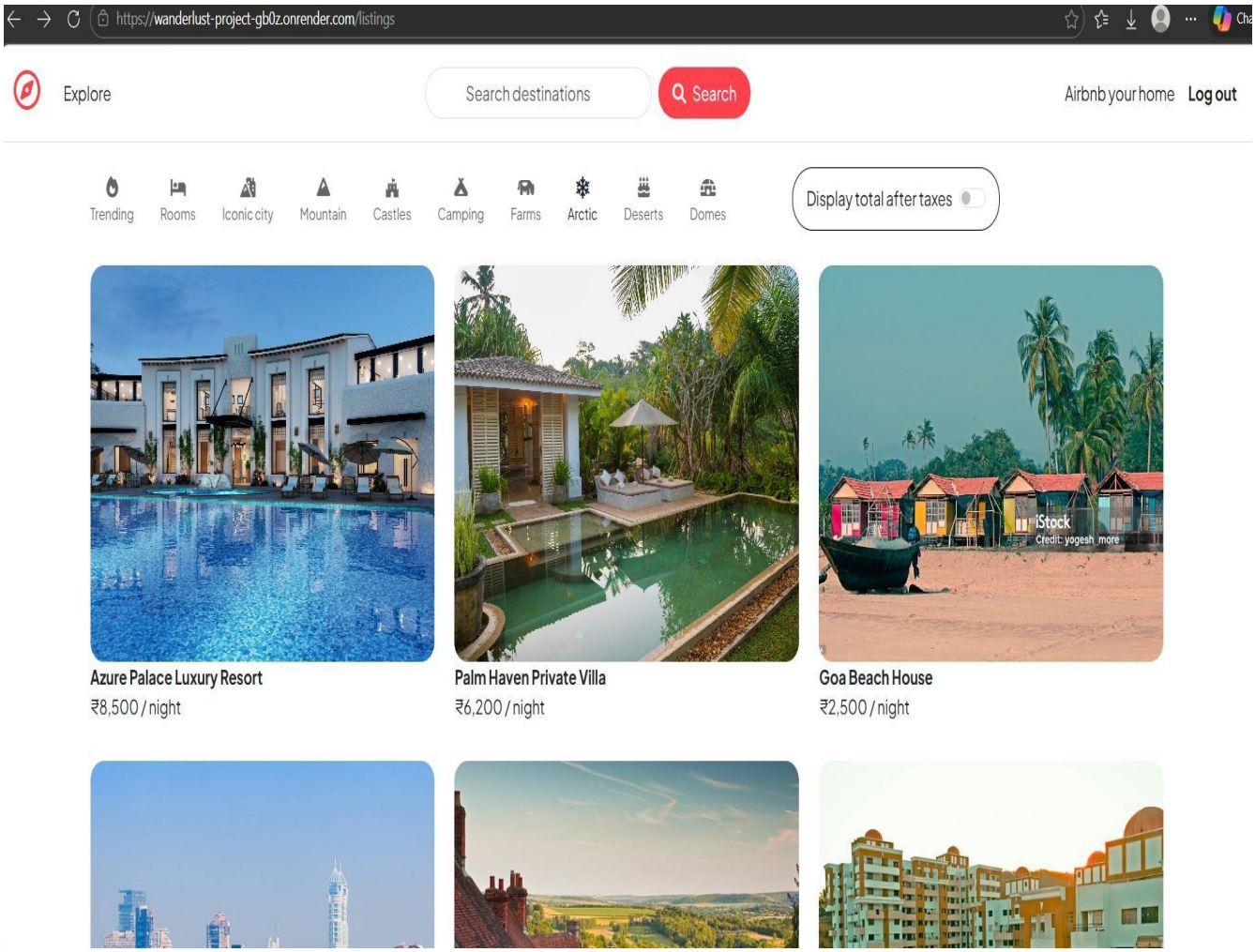
- Online payment gateway integration
- Live chat system
- Advanced search filters
- Mobile application development
- Better security and OTP verification
- Add Complete booking system
- Create admin panel

10.3 Benefits of Future Modifications

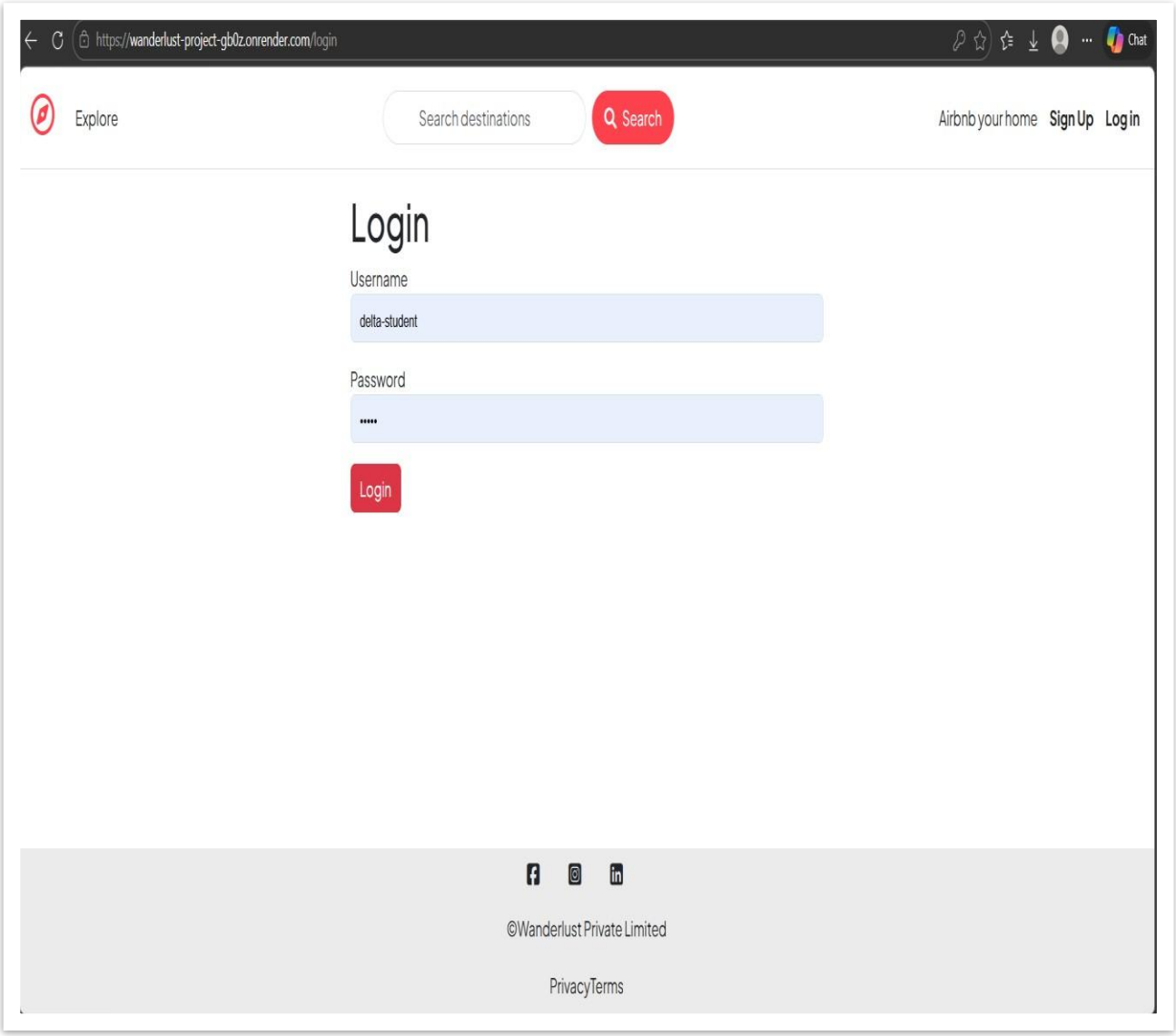
- Better user experience
- Easy booking process
- More secure transactions
- Easy communication between users
- Increased system efficiency

INPUT AND OUTPUT SCREENS

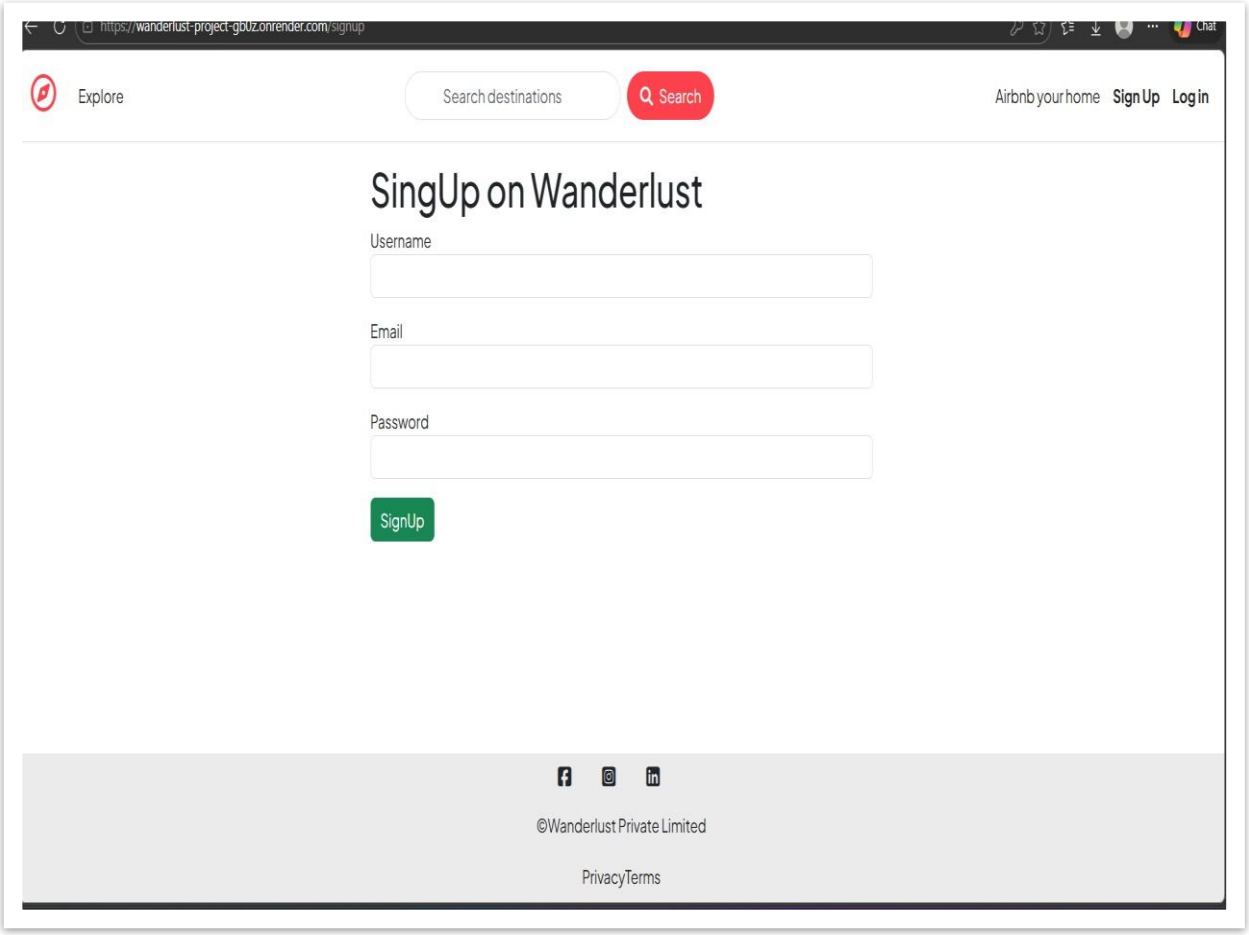
1.HOME PAGE



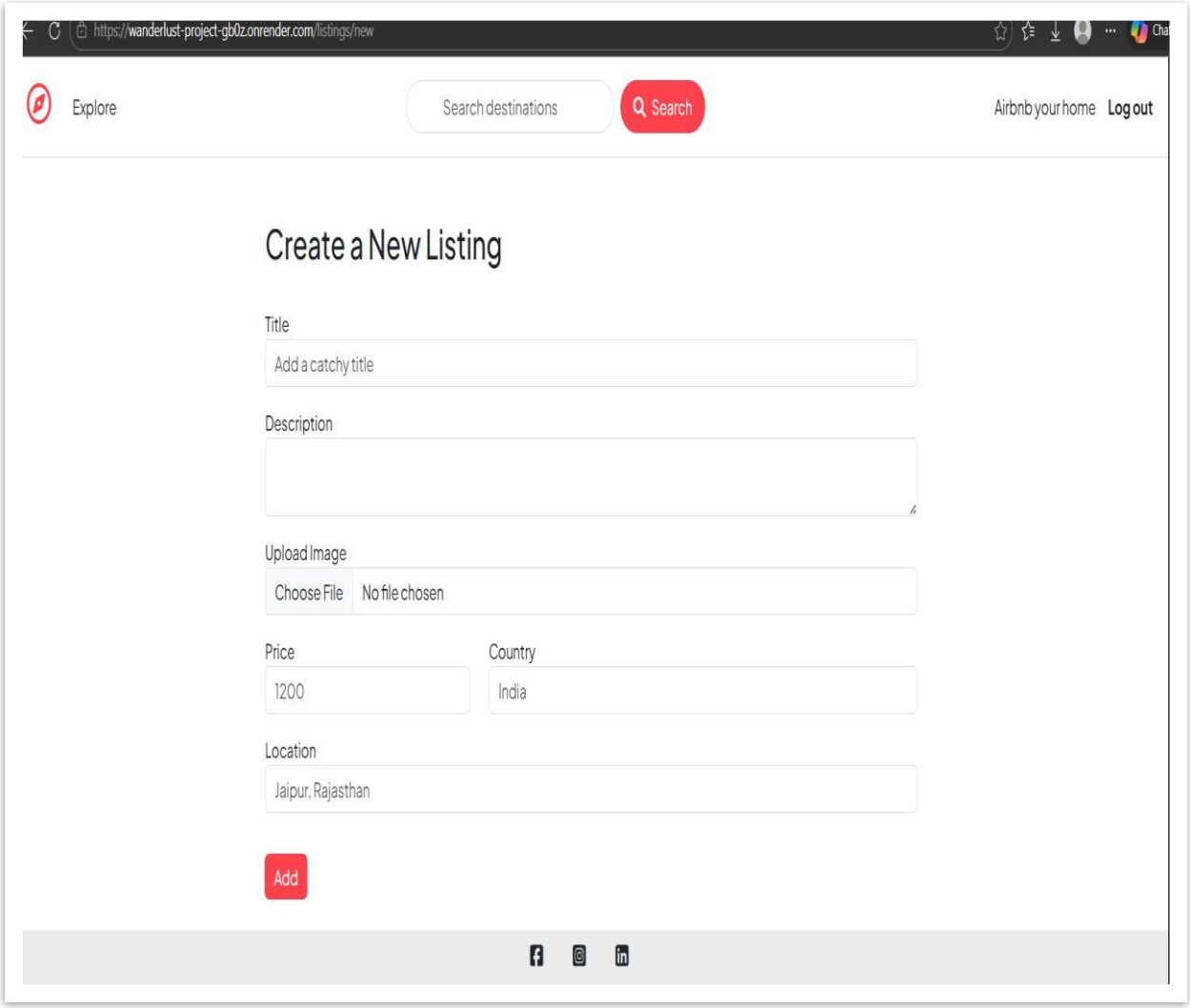
2.LOGIN PAGE


https://wanderlust-project-gb0z.onrender.com/login
Chat
Explore
Search destinations
Search
Airbnb your home
Sign Up
Log in
Login
Username
delta-student
Password
Login
©Wanderlust Private Limited
PrivacyTerms

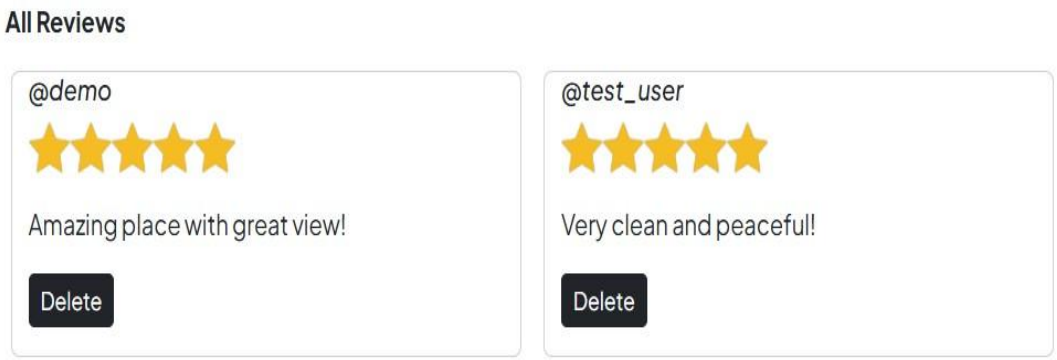
3.SIGNUP PAGE



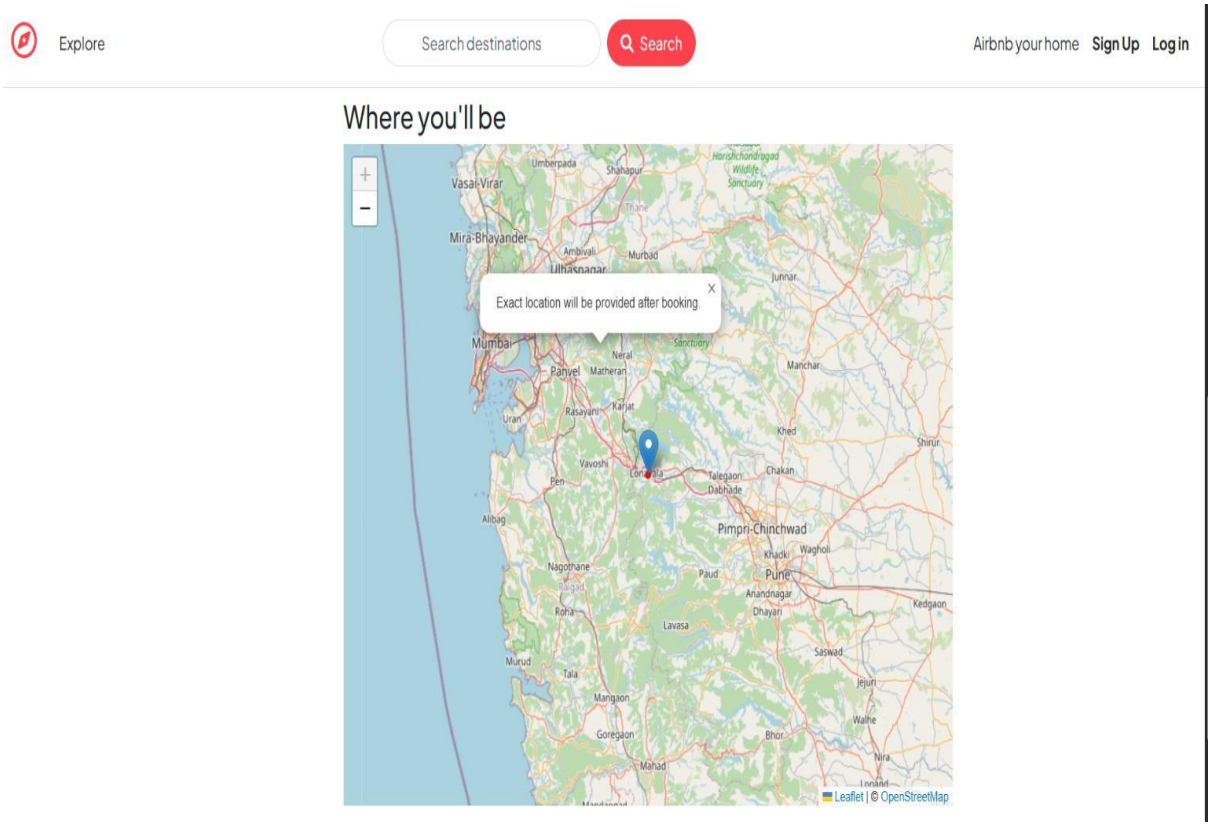
4.ADD NEW PROPERTY



5.REVIEW SECTION



6.MAP SECTION



Conclusion

The WanderLust system is a web-based property listing application developed to manage property details efficiently. It provides features like user login, property management, and review system. The project helped in learning web development concepts such as CRUD operations, database connectivity, and authentication. Overall, the system is simple, useful, and user-friendly.

Bibliography

1. Node.js Official Website – https://nodejs.org
2. Express.js Documentation – https://expressjs.com
3. MongoDB Official Website – https://www.mongodb.com
4. Bootstrap Official Website – https://getbootstrap.com
5. MDN Web Docs – https://developer.mozilla.org
6. Render Deployment Platform – https://render.com
7. NPM Official Documentation – https://docs.npmjs.com
8. Tutorials and Online Learning Resources