INSTITUTE FOR ADVANCED COMPUTING AND SOFTWARE DEVELOPMENT, AKURDI, PUNE

"Hostel Hunt"

PG-DAC August 2024

Submitted By:

Group No: G-104

Roll No.	Name of Student
248039	Kaushal Sharma
248104	Umesh Patil

Mr. Prithviraj Shinde
Project Guide

Mr. Rohit Puranik
Centre Coordinator

ABSTRACT

The purpose of **Hostel Hunt System** is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Hostel Hunt System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

ACKNOWLEDGEMENT

I take this occasion to thank God, almighty for blessing us with his grace and taking our endeavour to a successful culmination. I extend my heartfelt thanks to our esteemed guide, **Mrs. Prithviraj Shinde** for providing me with the right guidance and advice at the crucial juncture and showing me the right way. I sincerely thank our respected Centre Coordinator, Mr. Rohit Puranik, for allowing us to use the available facilities. I would also like to thank the other faculty members at this occasion. Last but not least, I would like to thank my friends and family for the support and encouragement they have given me during our work.

Kaushal Sharma (240841220077)

Umesh Patil (240841220196)

Table of Contents

1. INTRODUCTION

1.1 Introduction of the Project Hostel Hunt System:

The "Hostel Hunt System" has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it proves it is user-friendly. Hostel Management System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus, it will help organization in better utilization of resources.

Every organization, whether big or small, has challenges to overcome and managing the information of Room, Hostel, Student, Facility, Student Registration. Every Hostel Management System has different Hostel needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and will help you ensure that your organization is equipped with the right level of information and details for your future goals. Also, for those busy executives who are always on the go, our systems come with remote access features, which will allow you to manage your workforce anytime, at all times. These systems will ultimately allow you to better manage resources.

1.2 Abstract of the Project Hostel Hunt System:

The purpose of Hostel Hunt System is to automate the existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. The required software and hardware are easily available and easy to work with.

Hostel Hunt System, as described above, can lead to error free, secure, reliable and fast management system. It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources. The organization can maintain computerized records without redundant entries. That means that one need not be distracted by information that is not relevant, while being able to reach the information.

The aim is to automate its existing manual system by the help of computerized equipment's and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically, the project describes how to manage for good performance and better services for the clients.

1.3 Overview

The Hostel Hunt System is developed in favour of the hostel management team which helps them to save the records of the students about their rooms and their things. It helps them from the manual work from which it is very difficult to find the record of the student and the information about these ones who had help the hostel years before. This solution is developed on the plight of the hostel management team through this they cannot require so efficient person to handle and manage the affairs of the students in the hostel, all you need to do is to login as administrator and you can see the information of all the students who have obtained and registered their hostel form, click verify to assertion their agility and allocate them to the available hostel. We can improve the efficiency of the hostel management, thus overcome the drawbacks of the existing management.

1.4 Objective and Goles

Hostel Hunt system is the system that manages the student data, staff data, student's admission process and create receipt for the fees paid by the student who stay in the hostel and also help in maintaining visitor 's messages.

The primary objectives of the this are:

- To provide an online platform for hostel booking and management.
- To ensure a seamless and user-friendly experience for hostel owners and tenants.
- To automate hostel-related activities, such as room booking, payments, and inquiries.
- To provide a secure and efficient payment system.
- To enable administrators to manage users, bookings, and hostel approvals efficiently.

The primary goals of this project are:

- Develop a robust and scalable HMS using Java EE, Spring Boot, and related technologies.
- Automate key hostel operations to improve efficiency and reduce manual effort.
- Enhance resident experience through online booking and streamlined communication.
- Provide secure access to sensitive data using Spring Authentication.
- Generate comprehensive reports for informed decision-making.

This system is designed in favour of the hostel management which helps them to save the records of the students about their rooms. It helps them from the manual work from which it is very difficult to final the record of the students and the mess bills of the students and the information of about those ones.

1.5 Admin Panel

1.Admin Login: - Admin can login through login form.

2. Admin Profile: -Admin can manage his own profile. Admin can also change his password

3. Rooms: - Admin can create rooms and allots seater to particular rooms and assign the fees.

4. Registration: - Admin can create student profile and allot the rooms

5. Manage the Registration: - Admin can manage the all the student Profile. Take a print out of all profiles and also delete the profile.

6. Forgot Password: - Admin can also retrieve the password if admin forgot the password

1.6 User Panel

1. User Registration - User can register through user registration form

2. User Login - User can login through login form

3. Forgot Password - user can retrieve password through forgot password link

4. User Dashboard

5. User Profile - User can manage own profile

6. Book Hostel – User can book hostel

7. Room Details - Booked Room Details

8. Change Password - User Can change own password

1.7 Functionalities provided by Hostel Management System are as follows:

- Provides the searching facilities based on various factors. Such as Hostel, Bed, Student, Student Registration
- Hostel Management System also manage the Facility details online for Student details, Student Registration details, Hostel.
- It tracks all the information of Room, Facility, Student etc
- Manage the information of Room
- Shows the information and description of the Hostel, Bed
- To increase efficiency of managing the Hostel, Room
- It deals with monitoring the information and transactions of Student.
- Manage the information of Hostel
- Editing, adding and updating of Records is improved which results in proper resource management of Hostel data.
- Manage the information of Student
- Integration of all records of Student Registration.

1.8 Modules of Hostel Management System:

- Hostel Management Module: Used for managing the Hostel details.
- Student Registration Module: Used for managing the details of Student Registration
- Facility Module: Used for managing the details of Facility
- Room Management Module: Used for managing the information and details of the Room.
- Bed Module: Used for managing the Bed details
- Student Module: Used for managing the Student information
- Login Module: Used for managing the login details
- Users Module: Used for managing the users of the system

2. SYSTEM ARCHITECTURE

2.1 The Hostel Hunt System follows a layered architecture:

- **Presentation Layer (Frontend):** Built using ReactJS, Redux, HTML, CSS, and JavaScript with Bootstrap for styling. This layer provides the user interface for interacting with the system.
- **Business Logic Layer (Backend):** Developed using Java EE, Spring Boot, and JPA. This layer handles the core logic of the application, including data validation, business rules, and interaction with the database.
- **Data Access Layer:** JPA (Java Persistence API) is used for database interaction, providing an abstraction over the underlying MySQL database.
- **Data Storage:** MySQL database stores all the application data, including resident information, room details, financial records, and staff information.

2.2 Functional Requirements Implementation

The following functional requirements are being addressed in this project:

- **Resident Management:**
  - Resident registration and profile management.
  - Room allocation and occupancy tracking.
  - Check-in/check-out management.
  - Communication tools (e.g., announcements, notifications).

- **Room/Facility Management:**
  - Room availability management.
  - Maintenance request tracking and management.
  - Facility booking.
  - Inventory management for supplies.

- **Financial Management:**
  - Fee collection and payment processing (integrated with Razorpay).
  - Expense tracking.
  - Financial reporting.

- **Security Management:**
  - User authentication and authorization using Spring Authentication.
  - Integration with access control systems (future implementation).
  - Incident reporting.
  - Visitor management.

- **Online Booking:**
  - Allow residents to book rooms online.
  - Online payment processing.
  - Booking management.

2.3 Software Interfaces:

Software configuration for back-end Services:

- Java EE (Java 17)
- Spring Boot, JPA, Razor Pay, Spring Authentication
- MySQL (8.0.39)
- STS (4)

Software configuration for front-end Services:

- ReactJS, Redux (19.0.0)
- HTML, CSS, JS
- VS Code (1.97.0)

2.4 Technology Stack and Justification

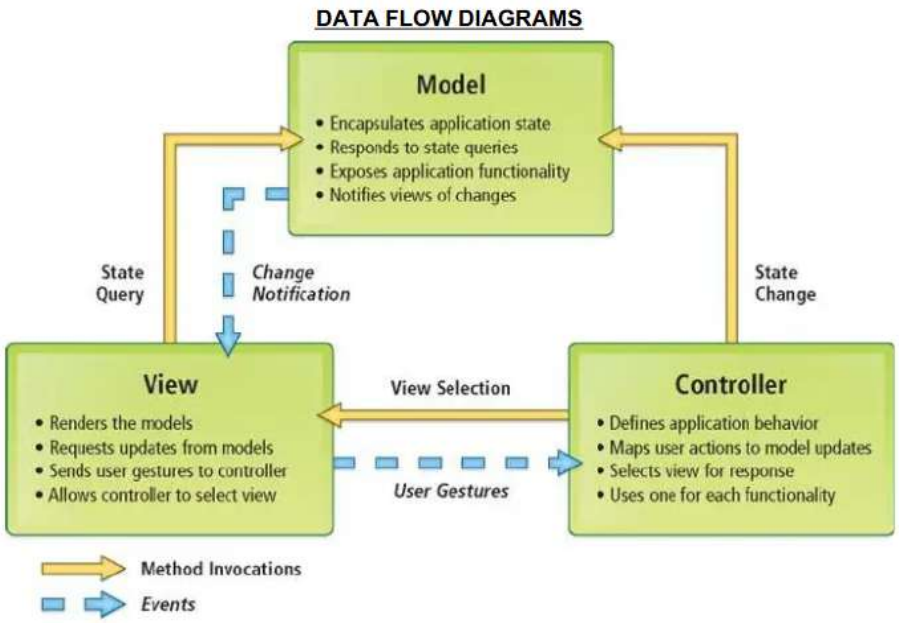
- **Java EE:** Provides a robust platform for building enterprise-level applications.
- **Spring Boot:** Simplifies the development process with auto-configuration and embedded servers.
- **JPA (Java Persistence API):** Facilitates database interaction and object-relational mapping.
- **MySQL:** A popular and reliable relational database management system.
- **ReactJS:** A powerful JavaScript library for building dynamic and interactive user interfaces.
- **Redux:** A state management library for React applications, improving performance and maintainability.
- **HTML, CSS, JS:** Core web technologies for structuring and styling the frontend.
- **Bootstrap:** A CSS framework for responsive design.
- **Razorpay:** A payment gateway integration for online transactions.
- **Spring Authentication:** Provides secure authentication and authorization mechanisms.
- **STS (Spring Tool Suite):** An IDE for developing Spring applications.

2.5 Implementation Methodology:

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the

following three parts:

- **Model** - The lowest level of the pattern which is responsible for maintaining data.
- **View** - This is responsible for displaying all or a portion of the data to the user.
- **Controller** - Software Code that controls the interactions between the Model and View.



- MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.

3. DIAGRAMS

3.1Different types of UR diagrams for hostel hunt java project

1. Use Case Diagram

Purpose: Shows the interactions between users (actors) and the system functionalities (use cases).

Key Elements:

- **Actors**: User, Hostel Owner, Admin
- **Use Cases**:
  - Login/Signup
  - Search Hostels
  - Book Hostel
  - Add/Edit Hostel Listings
  - Manage Users (Admin)
  - View Reports (Admin).

2. Class Diagram

Purpose: Models the static structure of the system, showing classes, their attributes, methods, and relationships.

Key Classes:

1. User
2. Hostel
3. Booking
4. Service Classes (e.g., HostelService, AuthenticationService)
5. DAO Classes (e.g., HostelDAO, UserDAO)

Example Relationships:

- User -> Booking (1-to-Many)
- User -> Review (1-to-Many)
- Hostel -> Review (1-to-Many).

3. Sequence Diagram

Purpose: Displays the order of interactions in a specific scenario.

Example Scenario: User Books a Hostel

1. User sends a booking request via the frontend.
2. Controller validates the request.
3. Service processes the booking logic.
4. DAO interacts with the database to store the booking.
5. User receives confirmation.

4. Activity Diagram

Purpose: Models the workflow for specific functionalities.

Example Scenario: Search Hostels

1. User enters location and filters.
2. System validates inputs.
3. DAO retrieves matching hostels from the database.
4. Results are displayed.

5. Component Diagram

Purpose: Describes the architecture and components of the system.

Components**:**

- Frontend: HTML/CSS & ReactJs.
- Backend: SpringBoot, Java
- Database: MySQL
- External APIs: Payment Gateway, Google Maps

6. **Deployment Diagram**

Purpose: Models the physical deployment of the system's components.

Nodes**:**

1. Client Machine (Browser/JavaFX Desktop App)
2. Application Server
3. Database Server

7. State Diagram

Purpose: Models the state changes of a particular object over time.

Example Object**:** Booking

1. **States**: Pending → Confirmed → Checked-In → Checked-Out → Completed/Cancelled
2. **Transitions**: Triggered by user or admin actions.

8. ER (Entity-Relationship) Diagram

Purpose: Models the database structure.

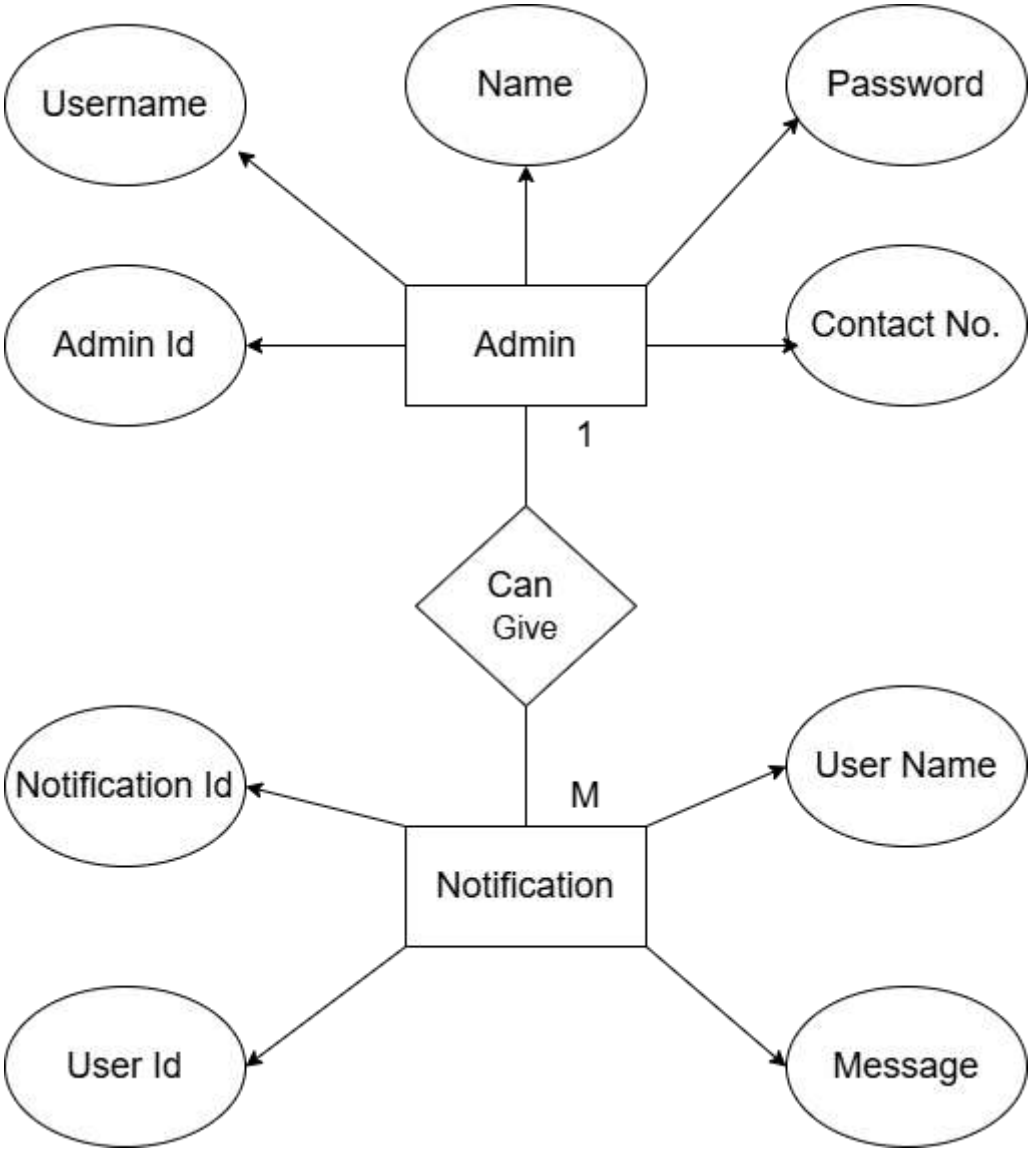
Entities:

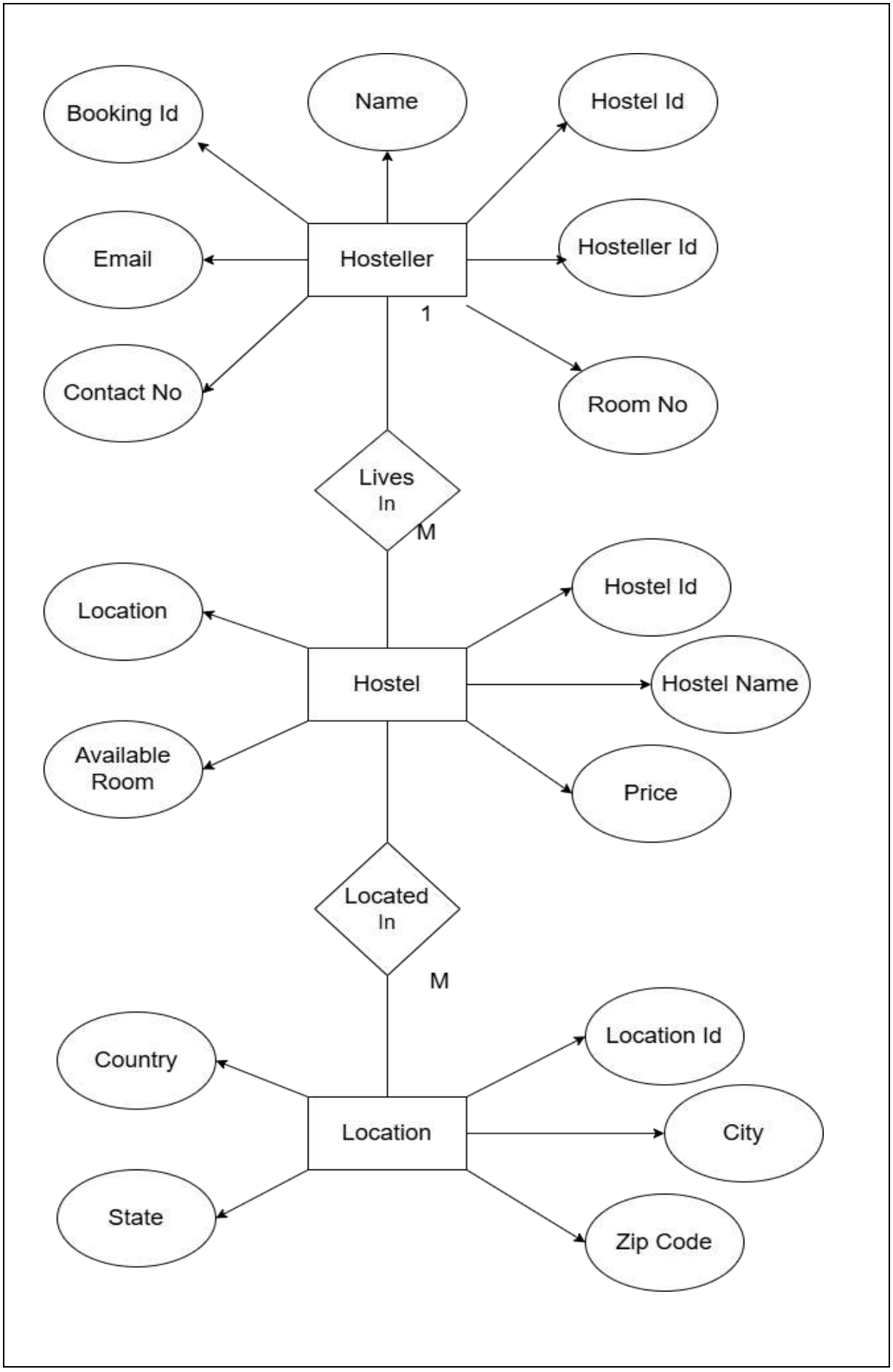
1. Users
2. Hostels
3. Bookings
4. Reviews

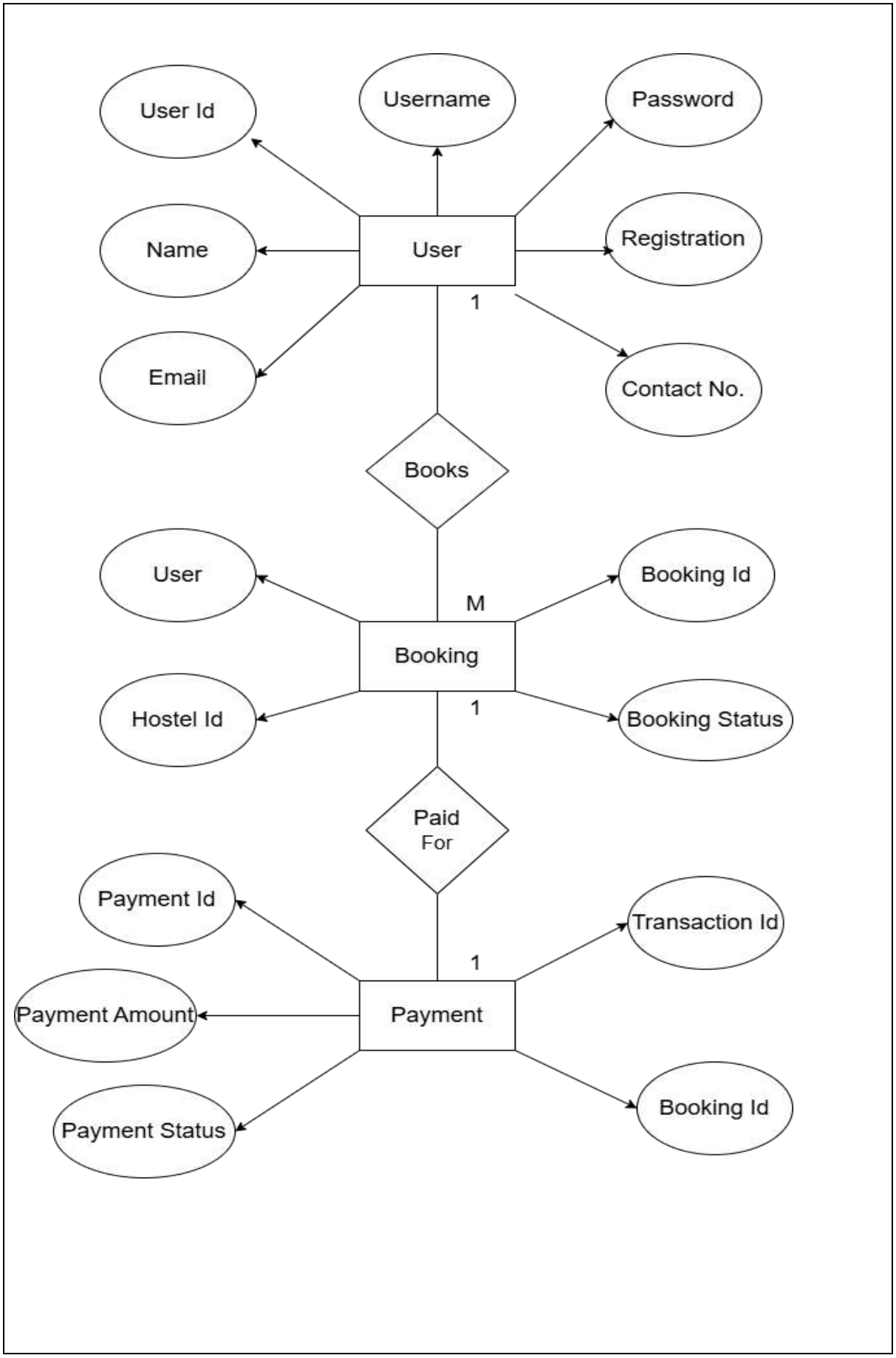
Relationships:

- Users can create multiple bookings.
- Users can leave multiple reviews for hostels.
- Hostels can have multiple reviews and bookings.

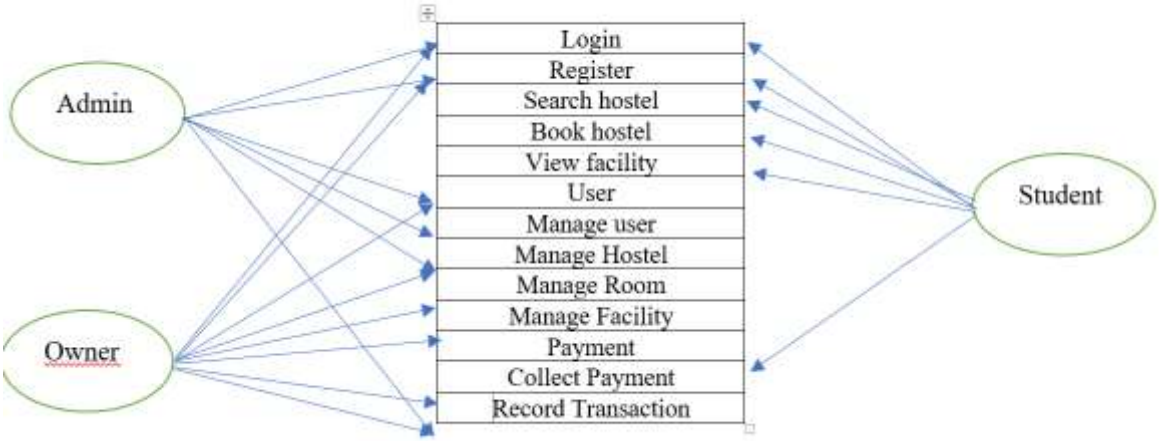
3.1 ER Diagram




Booking Id
Name
Hostel Id
Email
Hosteller
Hosteller Id
1
Contact No
Room No
Lives In
M
Location
Hostel Id
Hostel
Hostel Name
Available Room
Price
Located In
M
Country
Location Id
Location
City
State
Zip Code

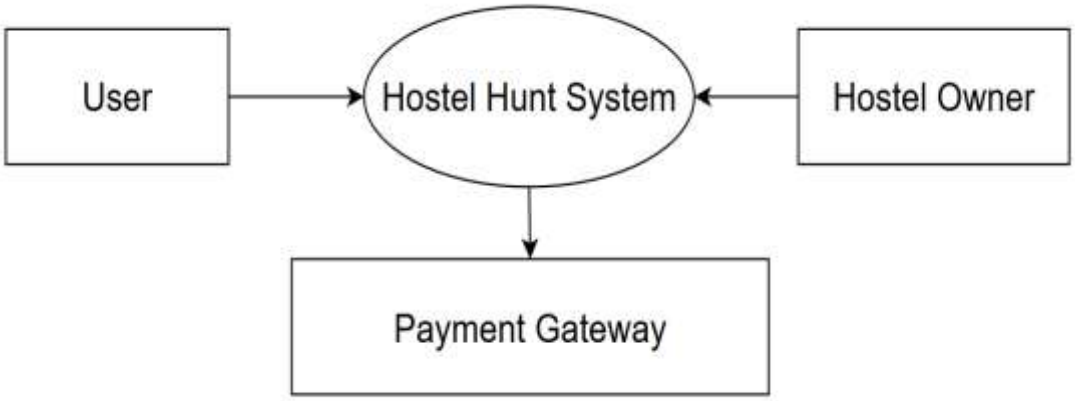

User Id
Username
Password
Name
User
Registration
1
Email
Contact No.
Books
User
Booking Id
M
Booking
Hostel Id
1
Booking Status
Paid
For
Payment Id
Transaction Id
1
Payment Amount
Payment
Payment Status
Booking Id

3.2 Use Case Diagram

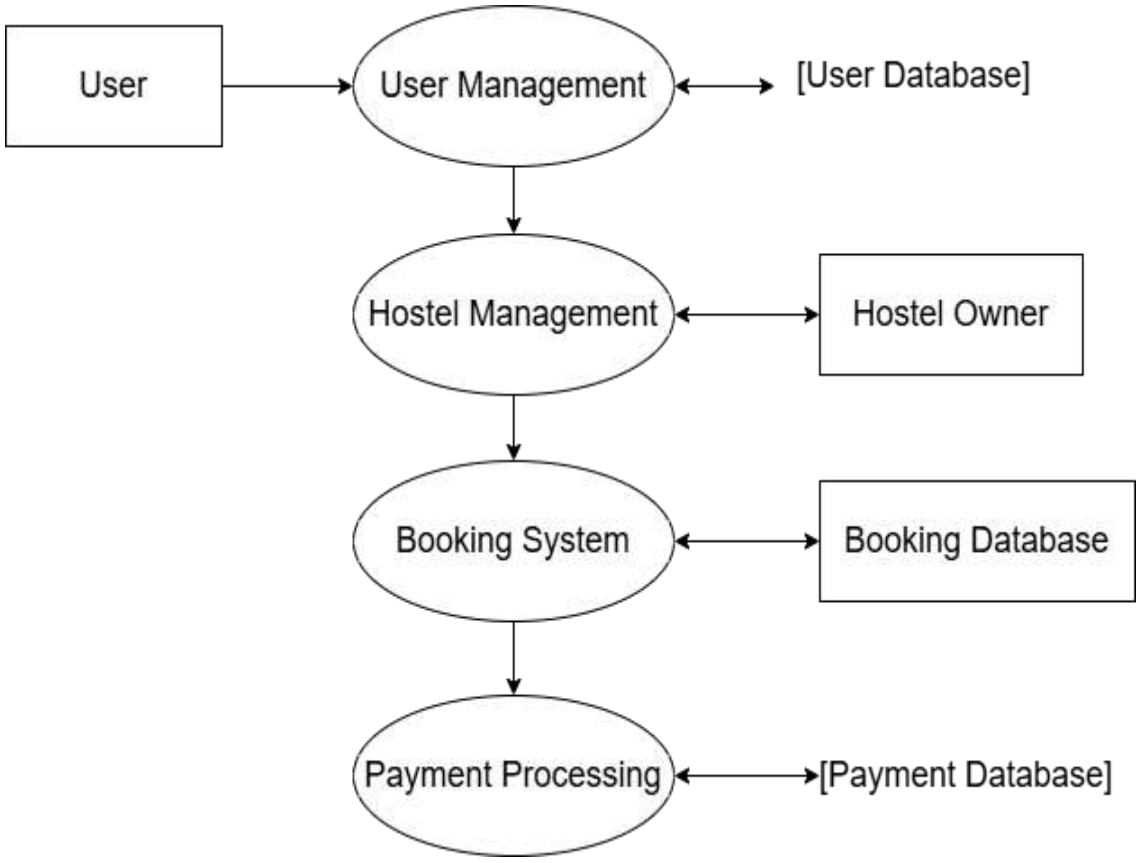


3.3 Data Flow Diagram

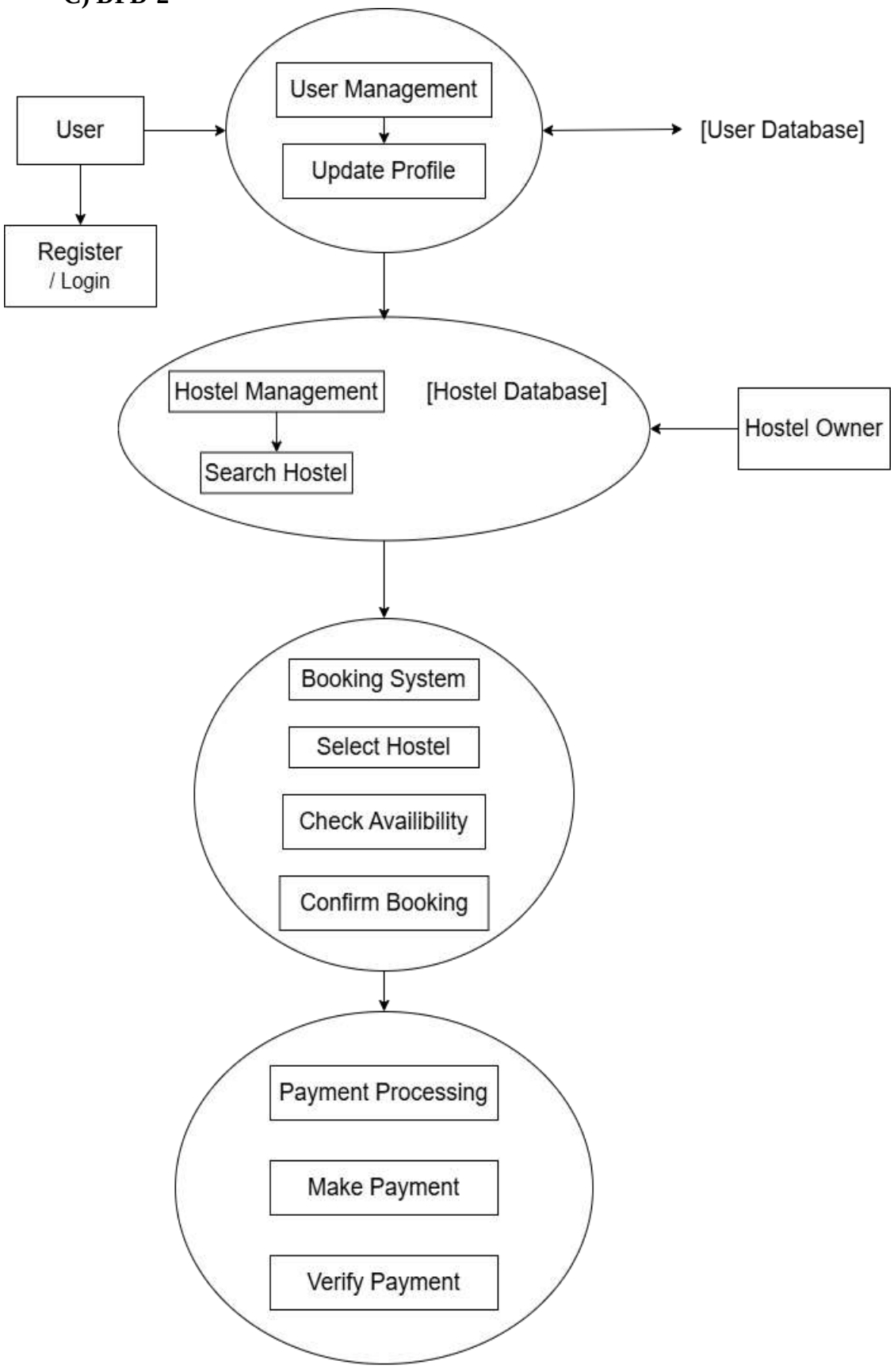
A) DFD-0



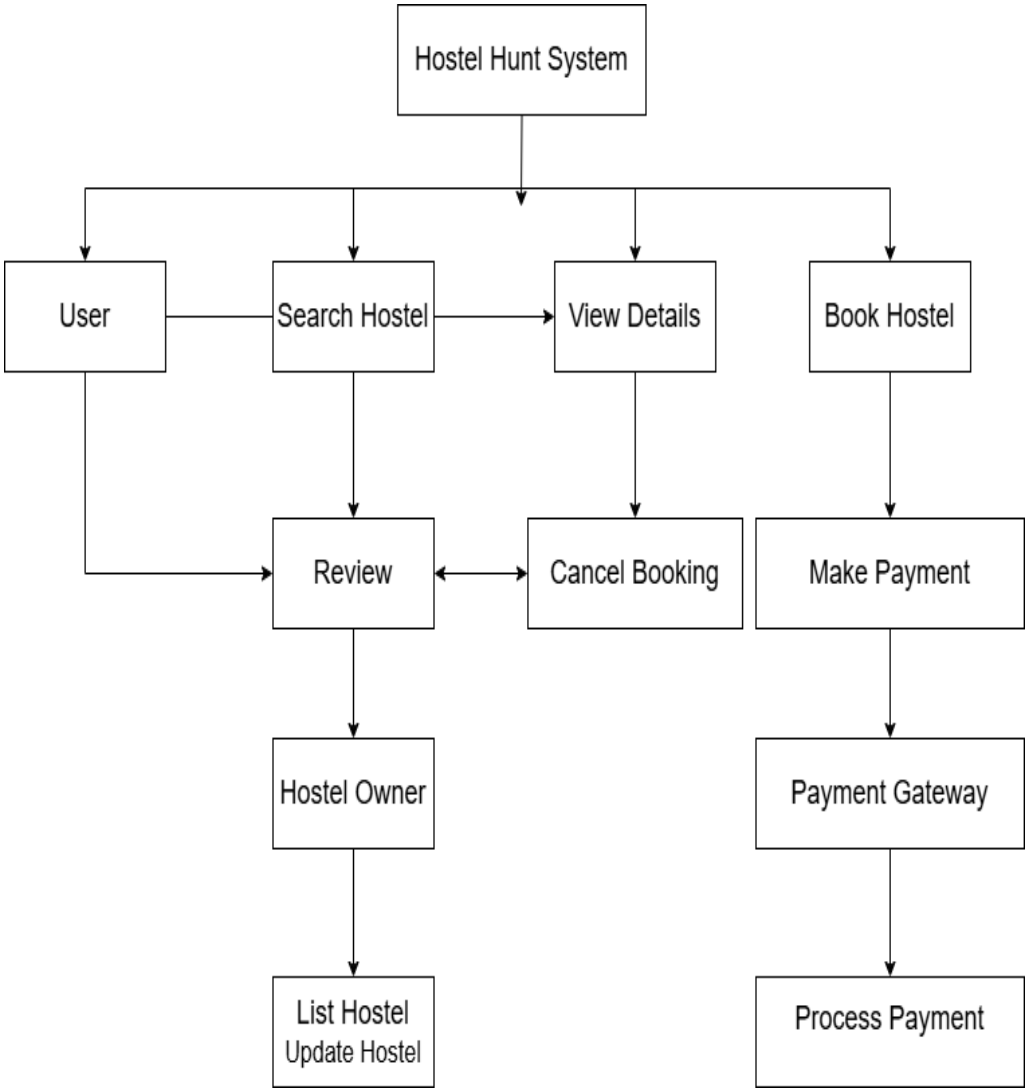
B) DFD-1


User
User Management
[User Database]
Hostel Management
Hostel Owner
Booking System
Booking Database
Payment Processing
[Payment Database]

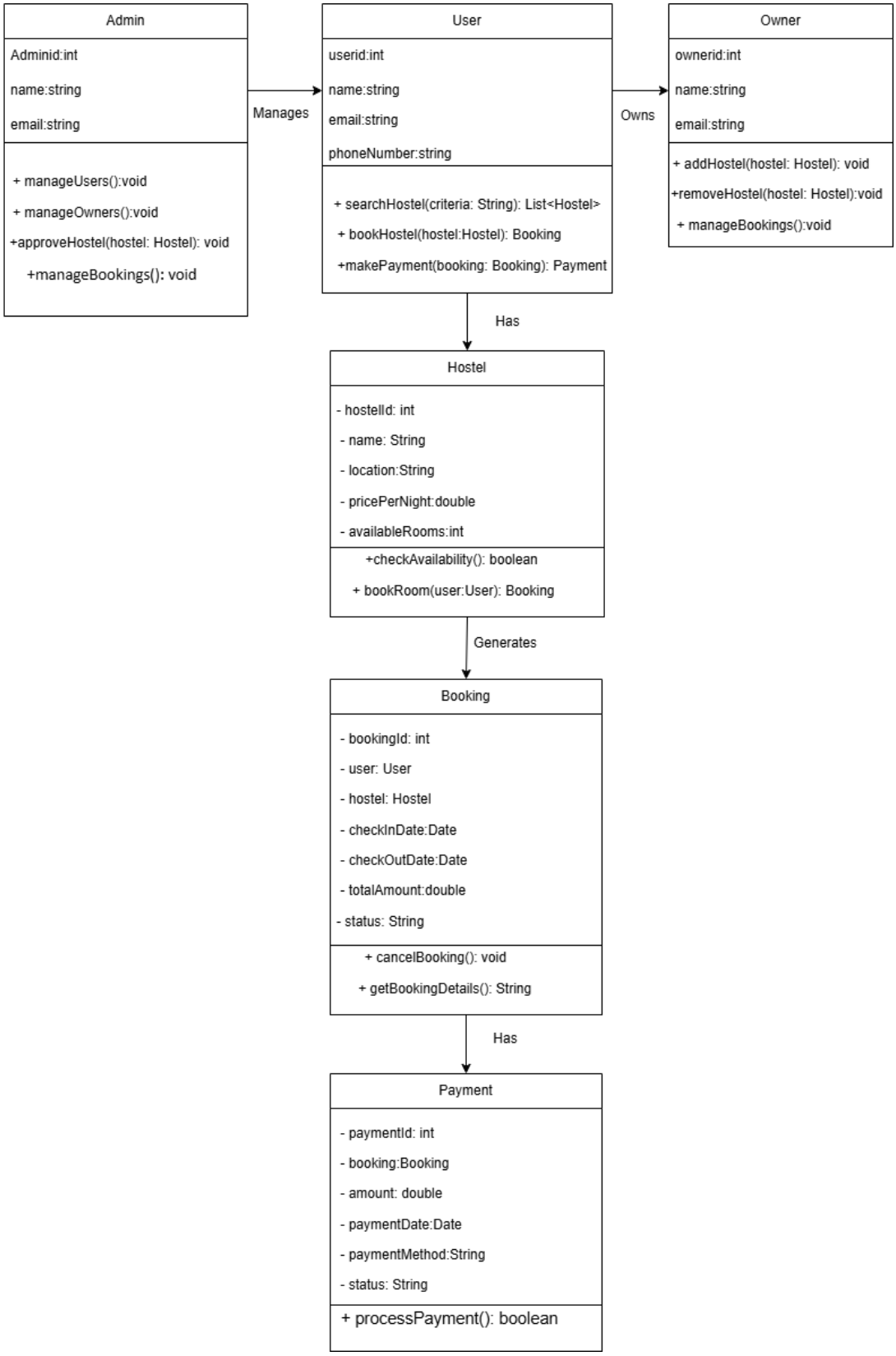
C) DFD-2


User Management
User
Update Profile
[User Database]
Register
/ Login
Hostel Management
[Hostel Database]
Hostel Owner
Search Hostel
Booking System
Select Hostel
Check Availibility
Confirm Booking
Payment Processing
Make Payment
Verify Payment

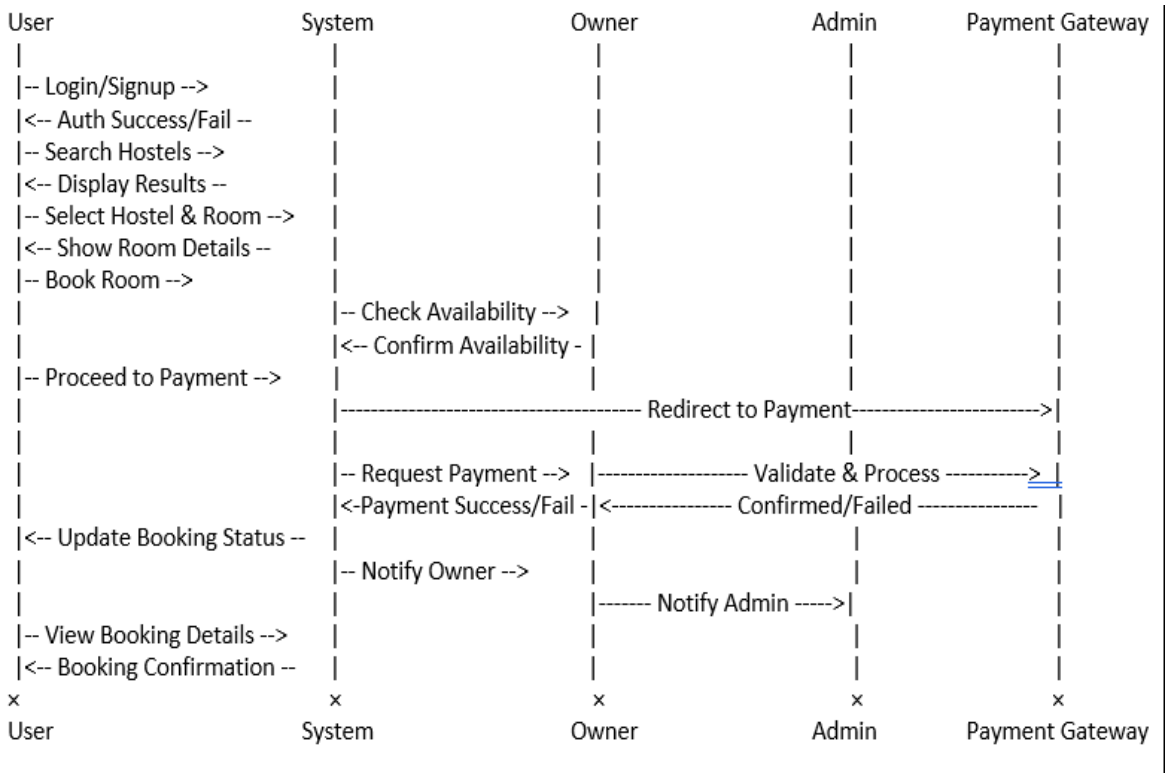
3.4 Activity Diagram



3.5 Class Diagram



3.6 Sequence Diagram



4. DATABASE

4.1 Tables

```
mysql> show tables;
+-----------------------+
| Tables_in_hostel_hunt |
+-----------------------+
| admin                 |
| hostel                |
| hostel_owner          |
| student               |
+-----------------------+
4 rows in set (0.07 sec)
```

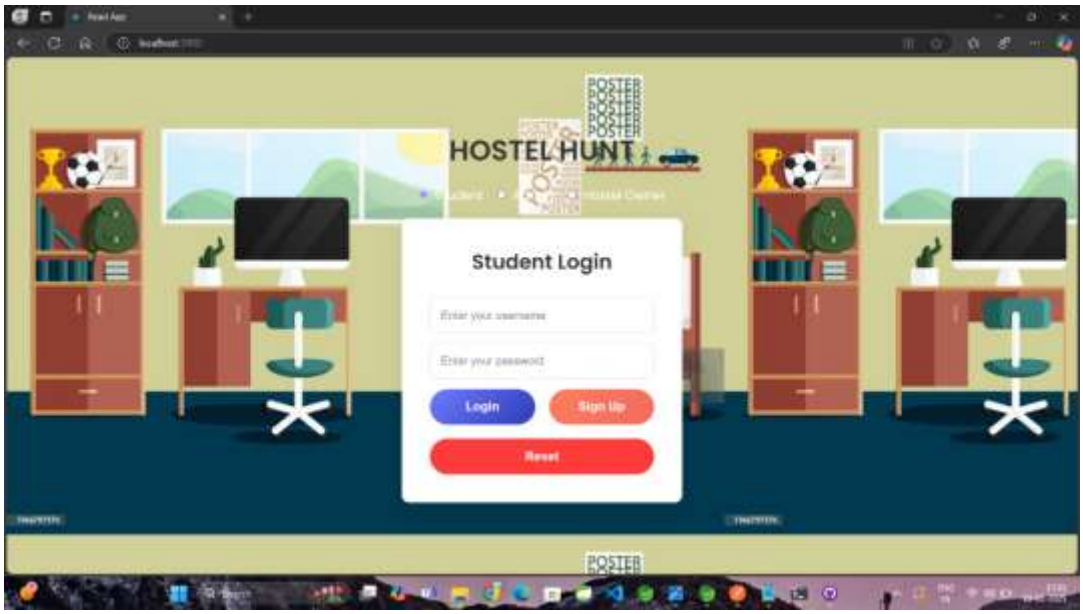
```
mysql> desc admin;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | bigint       | NO   | PRI | NULL    | auto_increment |
| password | varchar(255) | YES  |     | NULL    |                |
| username | varchar(255) | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
3 rows in set (0.03 sec)
```

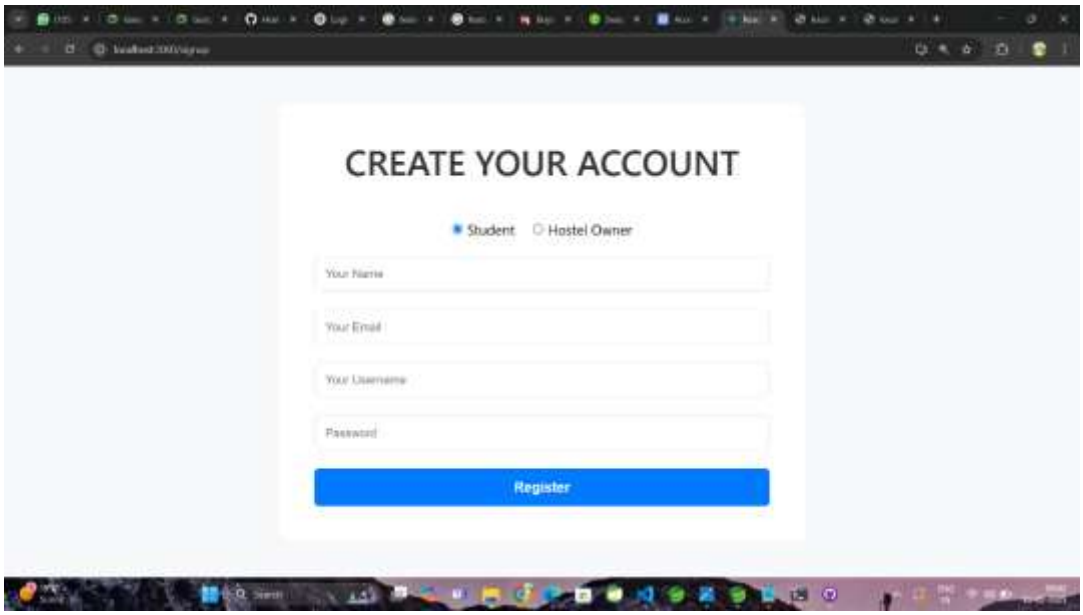
```
mysql> desc hostel;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | bigint       | NO   | PRI | NULL    | auto_increment |
| capacity | int          | NO   |     | NULL    |                |
| deposit  | double       | NO   |     | NULL    |                |
| location | varchar(255) | YES  |     | NULL    |                |
| name     | varchar(255) | YES  |     | NULL    |                |
| rent     | double       | NO   |     | NULL    |                |
| owner_id | bigint       | NO   | MUL | NULL    |                |
| imageurl | varchar(255) | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
8 rows in set (0.00 sec)
```

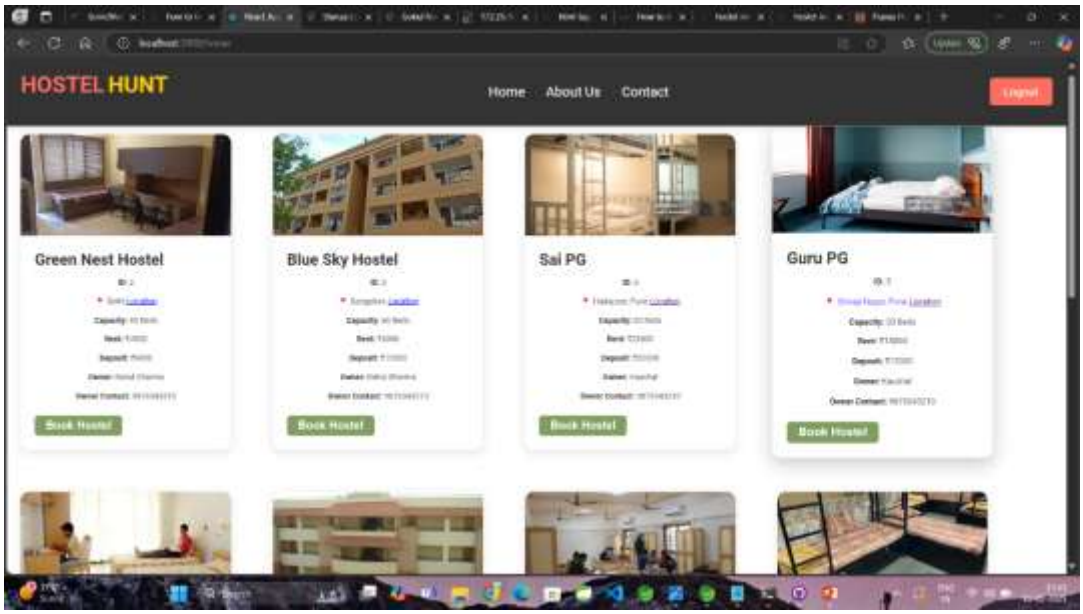
```
mysql> desc hostel_owner;
+--------------+--------------+------+-----+---------+----------------+
| Field        | Type         | Null | Key | Default | Extra          |
+--------------+--------------+------+-----+---------+----------------+
| id           | bigint       | NO   | PRI | NULL    | auto_increment |
| contact_info | varchar(255) | YES  |     | NULL    |                |
| name         | varchar(255) | YES  |     | NULL    |                |
| password     | varchar(255) | YES  |     | NULL    |                |
| username     | varchar(255) | YES  |     | NULL    |                |
+--------------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

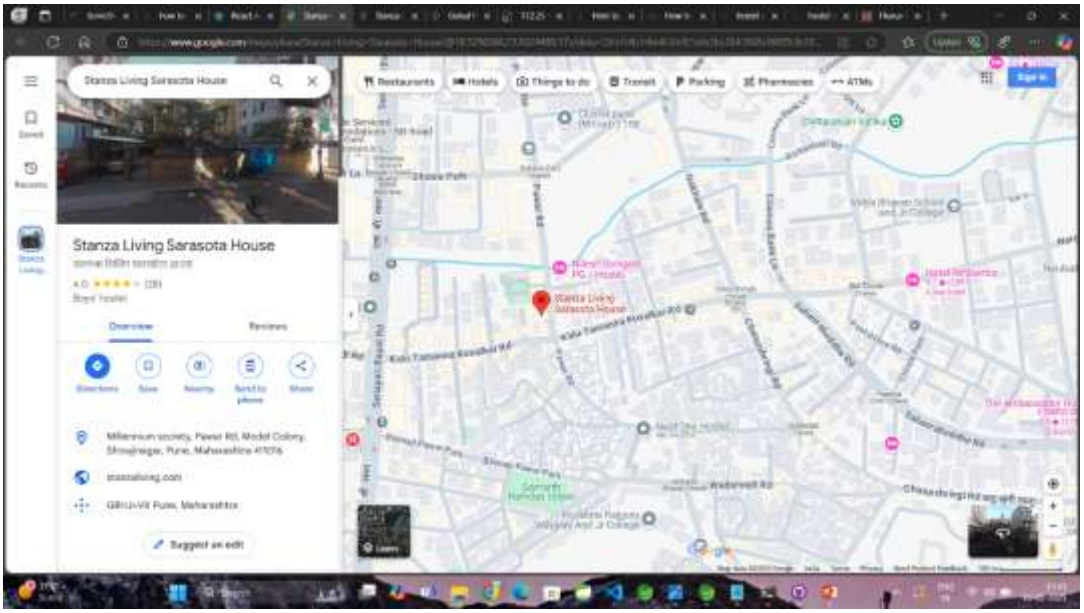
```
mysql> desc student;
+----------+--------------+------+-----+---------+----------------+
| Field    | Type         | Null | Key | Default | Extra          |
+----------+--------------+------+-----+---------+----------------+
| id       | bigint       | NO   | PRI | NULL    | auto_increment |
| email    | varchar(255) | YES  |     | NULL    |                |
| name     | varchar(255) | YES  |     | NULL    |                |
| password | varchar(255) | YES  |     | NULL    |                |
| username | varchar(255) | YES  |     | NULL    |                |
+----------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

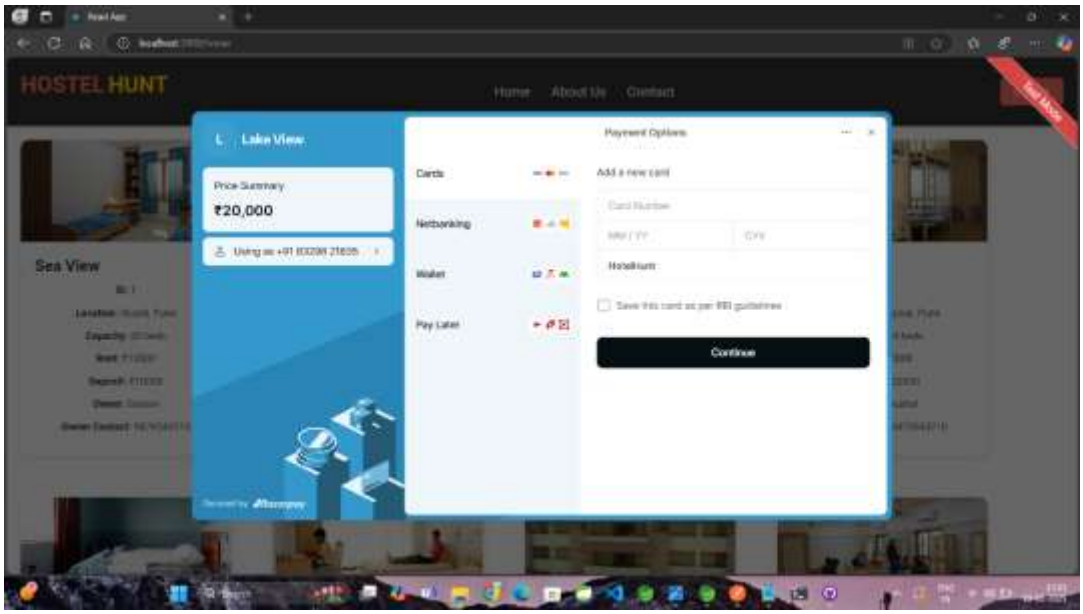
5.SNAPSHOT (SCREENSHOT)

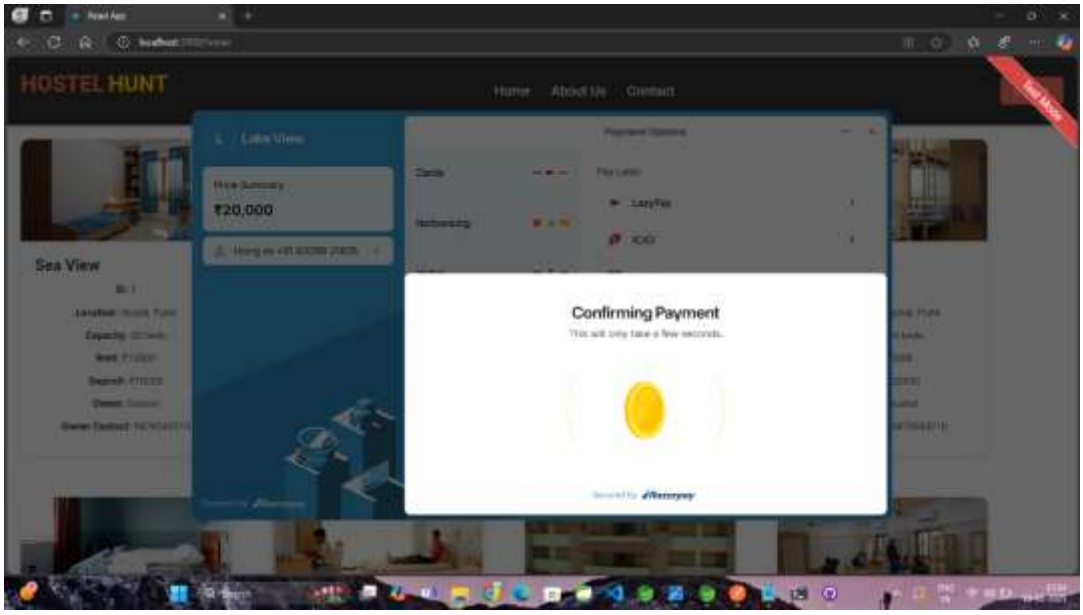


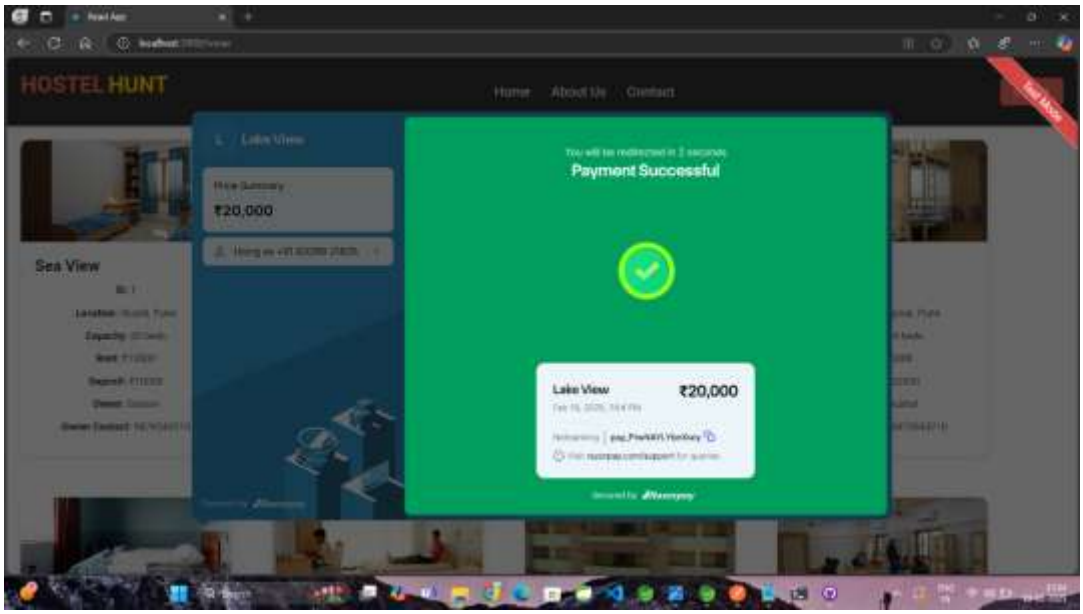


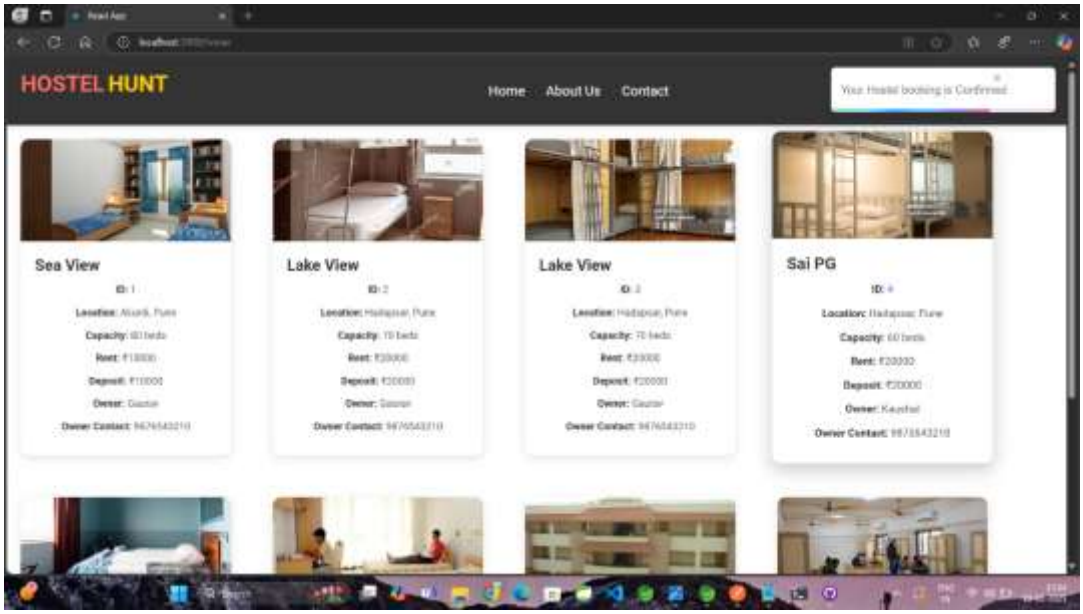

HOSTEL HUNT
Home
About Us
Contact
Logout
Green Nest Hostel
Blue Sky Hostel
Sai PG
Guru PG
Book Hostel
Book Hostel
Book Hostel
Book Hostel

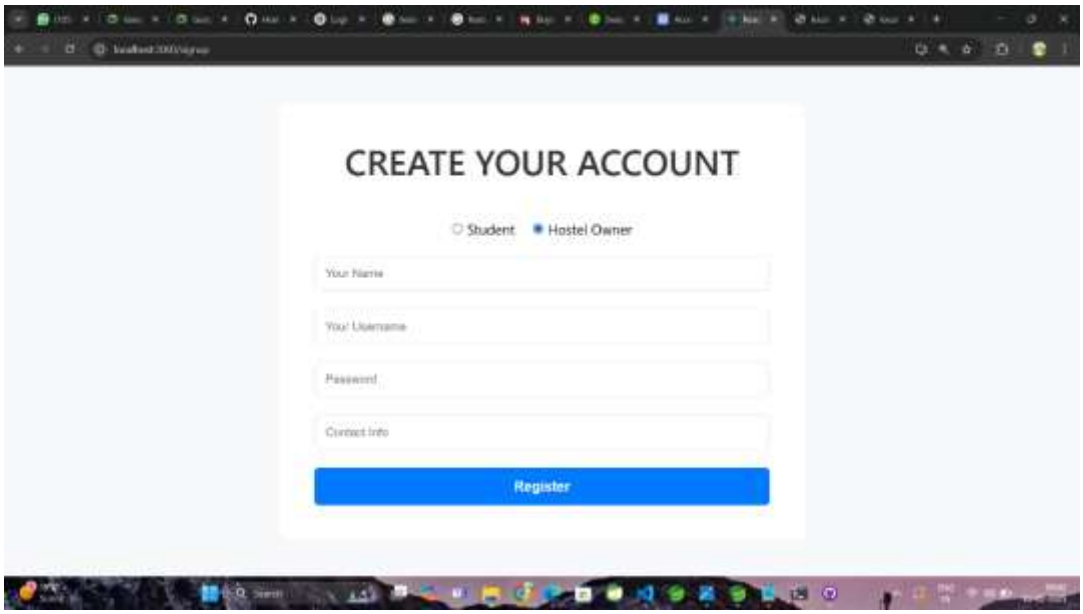

Stanza Living Sarasota House
Stanza Living Sarasota House
Overview
Reviews
Restaurants
Hotels
Things to do
Transit
Parking
Pharmacies
ATMs
Sign in

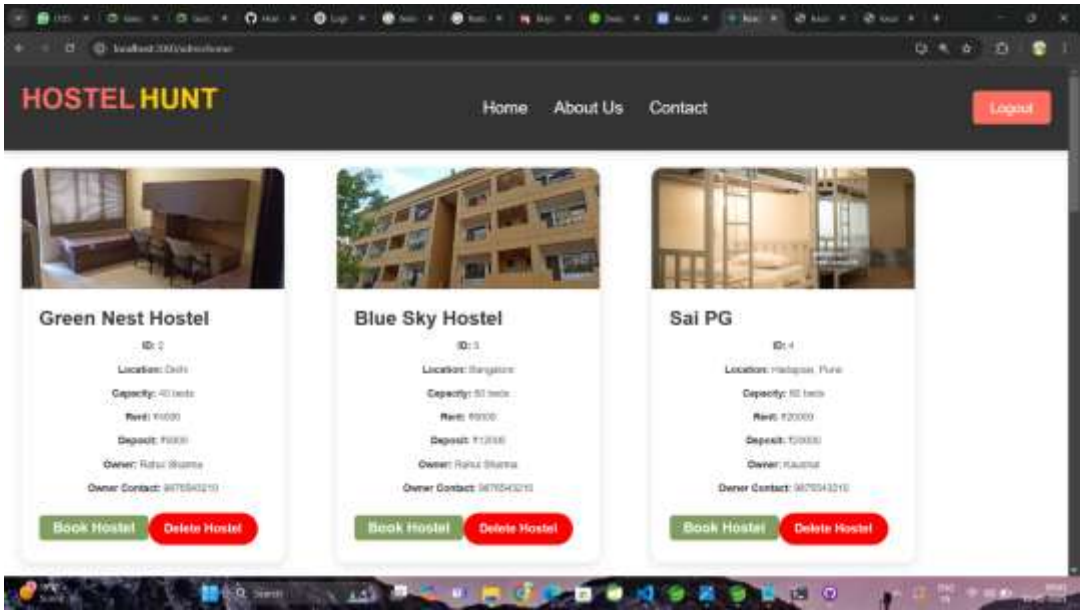

HOSTEL HUNT
Home
About Us
Contact
Lake View
Price Summary
₹20,000
Sea View
Payment Options
Cards
Netbanking
Wallet
Pay Later
Add a new card
Continue

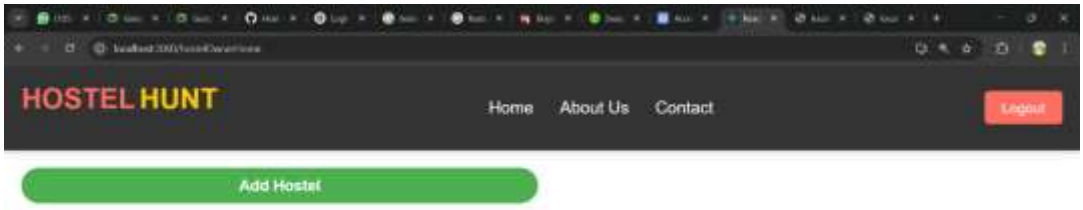

HOSTEL HUNT
Home
About Us
Contact
₹20,000
Sea View
Confirming Payment

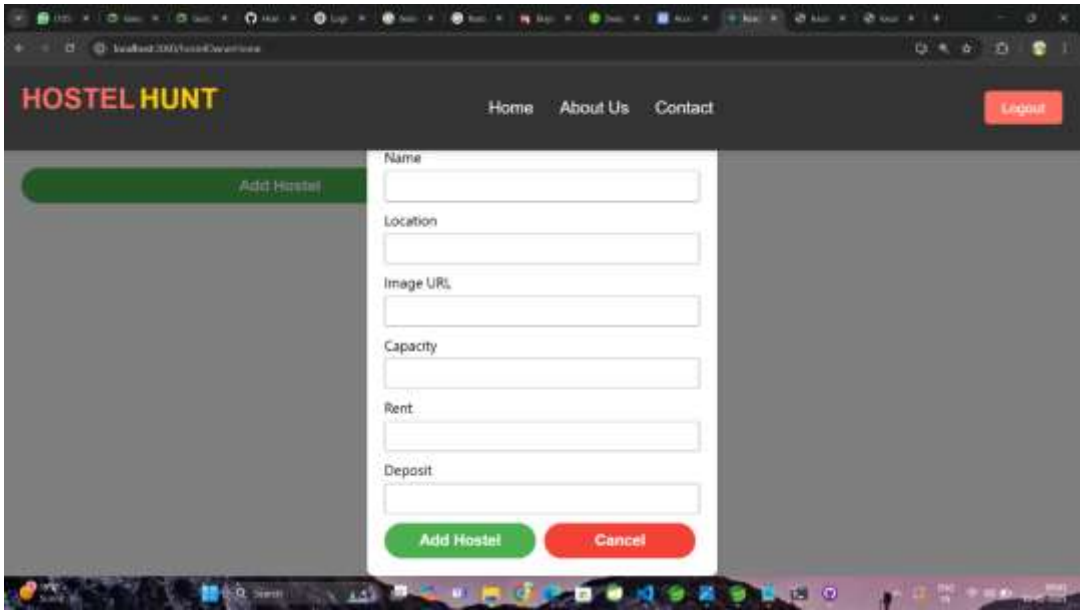

HOSTEL HUNT
Home
About Us
Contact
Payment Successful
₹20,000
Lake View
₹20,000
Sea View

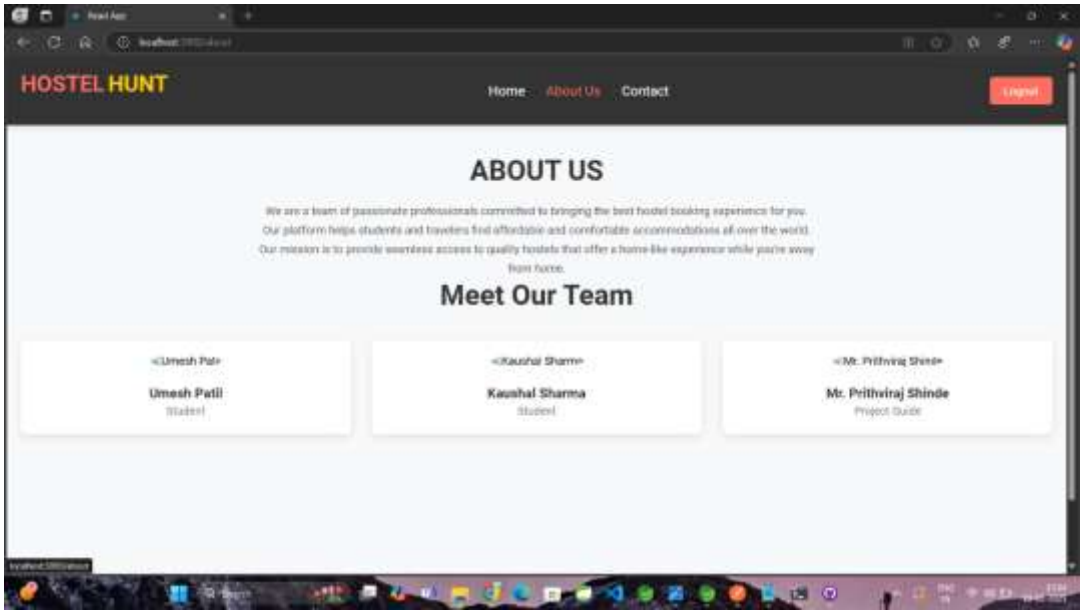

HOSTEL HUNT
Home
About Us
Contact
Sea View
Lake View
Lake View
Sai PG

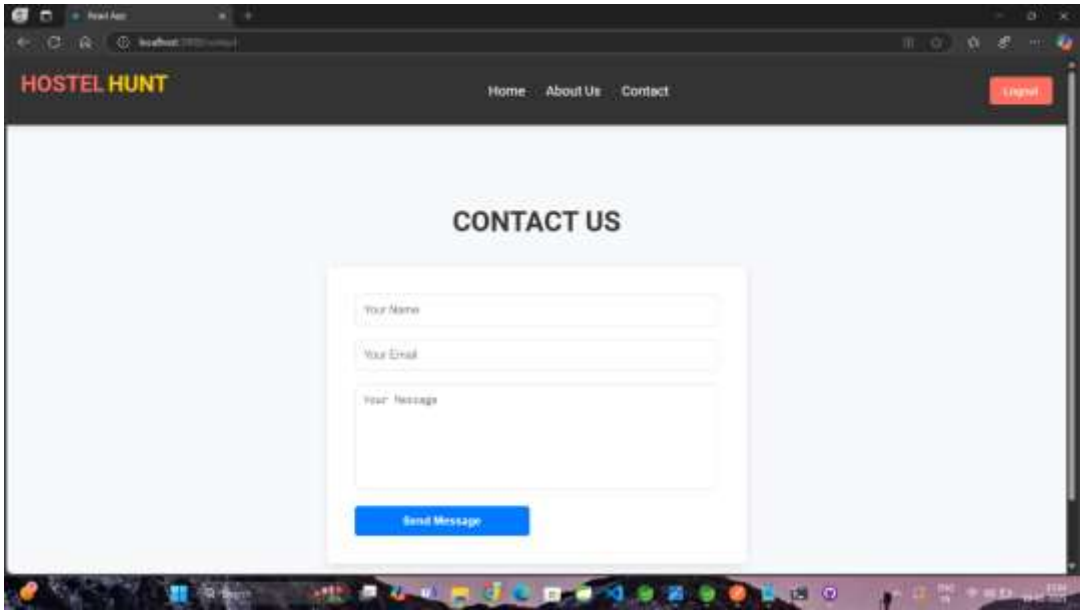

CREATE YOUR ACCOUNT
Student
Hostel Owner
Your Name
Your Username
Password
Contact Info
Register


HOSTEL HUNT
Home
About Us
Contact
Logout
Green Nest Hostel
Blue Sky Hostel
Sai PG
Book Hostel
Delete Hostel
Book Hostel
Delete Hostel
Book Hostel
Delete Hostel


HOSTEL HUNT
Home
About Us
Contact
Logout
Add Hostel


HOSTEL HUNT
Home
About Us
Contact
Logout
Add Hostel
Name
Location
Image URL
Capacity
Rent
Deposit
Add Hostel
Cancel


HOSTEL HUNT
Home
About Us
Contact
Logout
ABOUT US
We are a team of passionate professionals committed to bringing the best hostel booking experience for you.
Our platform helps students and travelers find affordable and comfortable accommodations all over the world.
Our mission is to provide seamless access to quality hostels that offer a home-like experience while you're away
from home.
Meet Our Team
<Umesh Pati>
Umesh Patil
Student
<Kaushal Sharma>
Kaushal Sharma
Student
<Mr. Prithviraj Shinde>
Mr. Prithviraj Shinde
Project Guide


HOSTEL HUNT
Home
About Us
Contact
Logout
CONTACT US
Your Name
Your Email
Your Message
Send Message

6. CHALLENGES AND SOLUTIONS

- **Integration with Razorpay:** Integrating the payment gateway required careful handling of API calls and security considerations. Solution: Thoroughly studied Razorpay documentation and implemented secure API integration.
- **Data Security:** Protecting sensitive data required robust authentication and authorization mechanisms. Solution: Implemented Spring Authentication and followed security best practices.
- **Frontend-Backend Communication:** Efficient communication between the React frontend and the Java backend was crucial. Solution: Used RESTful APIs for seamless data exchange.

7. FUTURE ENHANCEMENTS

- Integration with access control systems.
- Implementation of payroll management.
- Advanced reporting and analytics.
- Mobile app development.
- Cloud deployment.

8. CONCLUSION

The Hostel Hunt System is a comprehensive and efficient platform that simplifies hostel booking and management for users, owners, and administrators. By automating the booking process and integrating secure payment systems, it enhances user experience and operational efficiency. The system ensures transparency, security, and ease of access for both owners and tenants. The scalability of the system allows it to support multiple hostels and users while maintaining high performance and reliability.

Future enhancements such as AI-powered recommendations, chatbots for customer support, and mobile app integration will further improve the system's functionality and usability. By implementing these advancements, the Hostel Management System can provide a more personalized and interactive experience for users, increasing overall satisfaction and engagement.

Moreover, the continuous evolution of technology presents opportunities to integrate blockchain for enhanced security and trust, IoT-enabled smart room management, and predictive analytics to better understand user preferences. These advancements can further refine the hostel booking experience and create a fully digitized ecosystem that benefits all stakeholders involved.

9. REFERENCE

1. React.js Documentation - Official React documentation for frontend development. Available at: https://reactjs.org
2. Spring Boot Documentation - Official documentation for backend development. Available at: https://spring.io/projects/spring-boot
3. MySQL Documentation - Database management and SQL queries reference. Available at: https://dev.mysql.com/doc/
4. Google Maps API - Documentation for location services integration. Available at: https://developers.google.com/maps/documentation
5. Payment Gateway API - Online transaction processing references. Available from the respective payment providers (e.g., PayPal, Stripe, Razorpay).
6. Hostel Management System Research Papers - Various academic papers on hostel management automation and booking systems from IEEE Xplore and Google Scholar.
7. Sharma, A., & Gupta, P. (2022). "Automated Hostel Booking Systems Using Web Technologies." International Journal of Computer Science and Technology.
8. Patel, R., & Kumar, S. (2021). "Enhancing Hostel Management through Online Systems." Journal of Information Technology & Research.
9. Yadav, N., & Singh, B. (2020). "A Study on Web-Based Hostel Management Systems: Trends and Challenges." International Conference on Emerging Technologies in Computing.
10. Das, P., & Bose, R. (2019). "Smart Hostel Management System Using Cloud Computing." IEEE Xplore.