A PROJECT REPORT ON

Easy Stay

Online Room Rental Service

SUBMITTED IN PARTIAL

FULFILLMENT OF

DIPLOMA IN ADVANCED COMPUTING (PG-DAC)

UNDER THE GUIDANCE OF

Mr. Abhijeet Hari

PRESENTED BY

230360520085	**Yogita Bhagat**
230360520082	**Rohit Walke**
230360520076	**Suraj Bodke**
230360520007	**Akshay Kamble**
230360520046	**Nikhil Shinde**

AT

CENTER FOR DEVELOPMENT OF ADVANCED COMPUTING

C-DAC, Kochi

ACKNOWLEDGEMENT

The project “Easy Stay” was a great learning experience for us and we are submitting this work to Advanced Computing Training School (C-DAC ACTS, Pune).

We are very glad to mention the name of Mr. Abhijeet Hari for his valuable guidance to work on this project.

We are highly grateful to Ms. Sunitha Mam ,Head of STDC KOCHI Training Centre, CDAC, for her guidance and support whenever necessary during the booking of our journey to acquire PG-Diploma in Advanced Computing (PG-DAC) through CDAC KOCHI.

Our heartfelt thanks go to Mrs. Sunita (Project Coordinator, PG-DAC)who gave us all the required support and kind coordination to provide all the necessities to complete the project and throughout the booking up to the last day ofthe booking .

We would like to express our sincere gratitude towards Mr. Shailesh Pawar, our faculty for Core Java and Advanced Java, who was always there for us. Her guidance and support helped us overcome various obstacles and intricacies during our project work. Without her tremendous support, guidance, and efforts, this project would not have been possible.

TABLE OF CONTENTS

ABSTRACT

The Room management system is a software application that helps Managers efficiently manage their daily operations. The system includes various modules that enable hotel staff to perform tasks such as reservations management, room allocation, guest check-in and check-out and reporting. The system provides real-time information to the staff and management, allowing them to make informed decisions to enhance guest satisfaction and maximize revenue. This abstract outlines the main features and benefits of the hotel management system and highlights its importance in the hospitality industry. In our project, we aim to provide online hotel booking app where they can login to the portal and use it to have good stay options.

1. INTRODUCTION

We have developed Hotel Booking System on Java Spring Boot, MySQL and ReactJS which runs on the tomcat server and STS IDE, we have used Eclipse and Spring Tool Suites (STS) IDE. Java Spring Boot and ReactJS Project on Hotel Booking System is developed for automating the process of Hotel Booking System. The main feature of thisproject is to manage Hotel, Rooms, Payments, Customers, Booking and Services

This is a Major Project in Java using Spring Boot Micro Services API and ReactJS Frontend, which is good option for final year java spring boot projects with micro services, the main features and modules developed in project is Hotel module, Rooms module, Payment's module, Customers module and Booking module, which performs allthe various operations. Spring Boot and ReactJS Project on Hotel Booking System is secured web application which run inside the JVM and we have used Eclipse and STS IDE for developing this project.

For developing this Hotel Booking System in Spring boot framework and using ReactJS frontend, this isa maven-based project which you can run on STS also we have managed all dependencies through pom.xml. In Hotel Booking System ReactJS Project, all UI and frontend we have developed in ReactJS.

2. PROJECT OVERVIEW AND SUMMARY

2.1. PURPOSE

Our project, “ROOM BOOKING SERVICE” is a web-based online room booking portal which aims to provide Manager with functionalities to manage Users from Managers as well as customer base and hotel infrastructure, efficiently.

2.2. SCOPE

“ROOM BOOKING SERVICE” aims to deliver a web-based portal that manages all the customers who have signed up for the Room booking app. Customers can update their profile, browse through available hotels, check and compare services, see reviews, book rooms and also give their review. Managers can add hotel, add rooms, manage users, get feedback list, etc.

We are assuming that the organization that implements it will be using third-party platform where payments are done. Also, an API which can easily be integrated in our application can be created in future if needed. ROOM BOOKING SERVICE is a portal for both customers (for booking related activities) and Managers (for managing customer and infrastructure).

2.3. OVERVIEW

A. TECHNOLOGIES USED

i. FRONT END

- HTML
- CSS
- Bootstrap
- JavaScript
- React
- Ajax

ii. BACK END

- Spring Boot (REST API)
- Spring Data JPA
- Spring Security
- Java Mail API

iii. DATABASE MANAGEMENT SYSTEM

- MySQL

B. FEATURES PROVIDED

i. FOR CUSTOMER AND INFRASTRUCTURE

a. Register – Customers can register themselves as users.

b. Login – Successfully registered users are now eligible to Login.

c. Browse through properties – After successful login, customers can view properties according to filters applied by his own choice.

d. Booking – Customers can book the hotel of their liking for the selected dates.

e. Feedback – Customers can give feedback to show the satisfaction level.

f. Logout – After utilizing the portal, customers can Logout of the app.

ii . FOR MANAGERS

a. Login & Logout – Similar to customer, Managers can login & logout to access their account.

b. View bookings – Manager can view bookings as soon as they are made in order to prepare the services accordingly.

c. Add properties – Manager will edit infrastructure shown on the app if there is addition in property like a new room.

2.4. FEASIBILITY STUDY

Feasibility is the determination of whether a project is worth undertaking or not. Before recommending the new system, it is important to investigate if it is feasible to develop it.

Before developing and implementing a system, we must make sure that the system is feasible in the following ways:

A. TECHNICAL FEASIBILITY

In this type of feasibility study, the system analyst must check whether it is possible or not to develop the requested system with the available manpower, software, hardware, etc.
This project makes use of cross-platform software and solutions like Java, and hence can run on any operating system. JavaScript, used in front-end, is swift and versatile framework when it comes to delivering the requested page. Also, as JavaScript is popular, it is easy to learn it and utilizing it as front-end technology. The combination of Spring Boot, Spring Data JPA and Spring Security for backend make for a fast, easy to set-up and reliable system to interact with the database, as they are secure and transactional in nature. Since the sensitive data of customers and Managers need to be stored in a robust and secure database, MySQL database management system was chosen as it is an industry standard.

B. OPERATIONAL FEASIBILITY

In this type of feasibility study, the operation of the system is considered. An analysis is performed on whether it is feasible for the user department to use the application. Thus, the proposed system is said to be operationally feasible only if clients can understand the system clearly and correctly, and can use it with ease.

In the design of this project, we always kept user experience in mind. We tried to have a good user interface with consistent theme and alluring design to keep the users interested and engaged. In our project, the use of universally known icons and instructions that are easy to understand makes sure that the user will not need any special technical know-how to use the application. We made sure that the information available throughout the application is arranged in a logically coherent and consistent manner, guaranteeing that the users will have a smooth and effortless experience and even enjoy using the application.

C. ECONOMIC FEASIBILITY

In this type of feasibility study, the benefits of the system to the organization are considered by taking into consideration the cost-benefit analysis. All the software and technologies used in our project free, open-source, and widely available, with each of the technologies having an extensive community support. This makes “ROOM BOOKING SERVICE” an economically feasible solution to the organizations that wish to implement it.

3. REQUIREMENTS FULFILLED

3.1. FUNCTIONAL REQUIREMENTS

Following are the functional requirements fulfilled by our project:

- Customers can register themselves as users so that they can access the web site from anywhere to book the rooms as per requirements.
- Further, they are asked to fill in their personal details and email address.
- Manager registers infrastructure.
- Customers can login and book the properties.
- Managers can view bookings and reviews.
- Managers can also edit the properties id required .

3.2. NON-FUNCTIONAL REQUIREMENTS

Following are the non-functional requirements fulfilled by our project:

- Since the application uses lightweight and established software components that are also cross-platform, it is remarkably performant and has good support for every operating system.
- The use of JavaScript and JQuery for front end and Spring Boot, Spring Data JPA and Spring Security for back end delivers quick response times to Managers and customer and infrastructure.
- Card-style UI and well-known icons and symbols used throughout the application provides a consistent theme and user-friendly interface that anyone can grasp easily, even without a technical background.

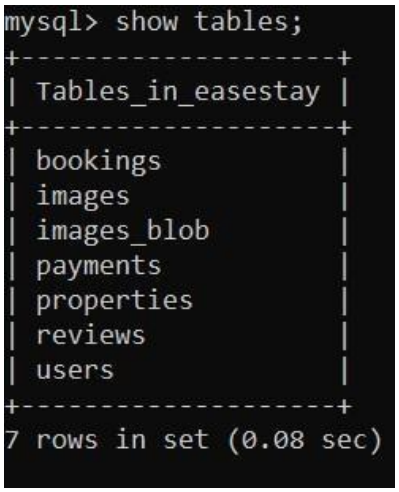
4. PROJECT DESIGN

4.1. DATA MODEL

The following tables depict the database design used for “ROOM BOOKING SERVICE” application:

A. Tables for Registration and Login

a. Database tables



b. Users

```
mysql> desc users;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| user_type     | varchar(31)  | NO   |     | NULL    |                |
| id            | bigint       | NO   | PRI | NULL    | auto_increment |
| date_of_birth | date         | YES  |     | NULL    |                |
| email         | varchar(255) | YES  |     | NULL    |                |
| name          | varchar(255) | YES  |     | NULL    |                |
| password      | varchar(255) | YES  |     | NULL    |                |
| phone_number  | varchar(255) | YES  |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)
```

a. Properties Table

```
mysql> desc properties;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| id              | bigint       | NO   | PRI | NULL    | auto_increment |
| city            | varchar(255) | YES  |     | NULL    |                |
| country         | varchar(255) | YES  |     | NULL    |                |
| state           | varchar(255) | YES  |     | NULL    |                |
| street          | varchar(255) | YES  |     | NULL    |                |
| zip_code        | varchar(255) | YES  |     | NULL    |                |
| capacity        | int          | NO   |     | NULL    |                |
| description     | varchar(255) | YES  |     | NULL    |                |
| features        | varchar(255) | YES  |     | NULL    |                |
| name            | varchar(255) | YES  |     | NULL    |                |
| price_per_night | double       | YES  |     | NULL    |                |
| type            | int          | YES  |     | NULL    |                |
| owner_id        | bigint       | YES  | MUL | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
13 rows in set (0.01 sec)

mysql> desc reviews;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | bigint       | NO   | PRI | NULL    | auto_increment |
| comment     | varchar(255) | YES  |     | NULL    |                |
| rating      | int          | YES  |     | NULL    |                |
| booking_id  | bigint       | YES  | MUL | NULL    |                |
| customer_id | bigint       | YES  | MUL | NULL    |                |
| property_id | bigint       | YES  | MUL | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)
```

B. Tables after Student Login

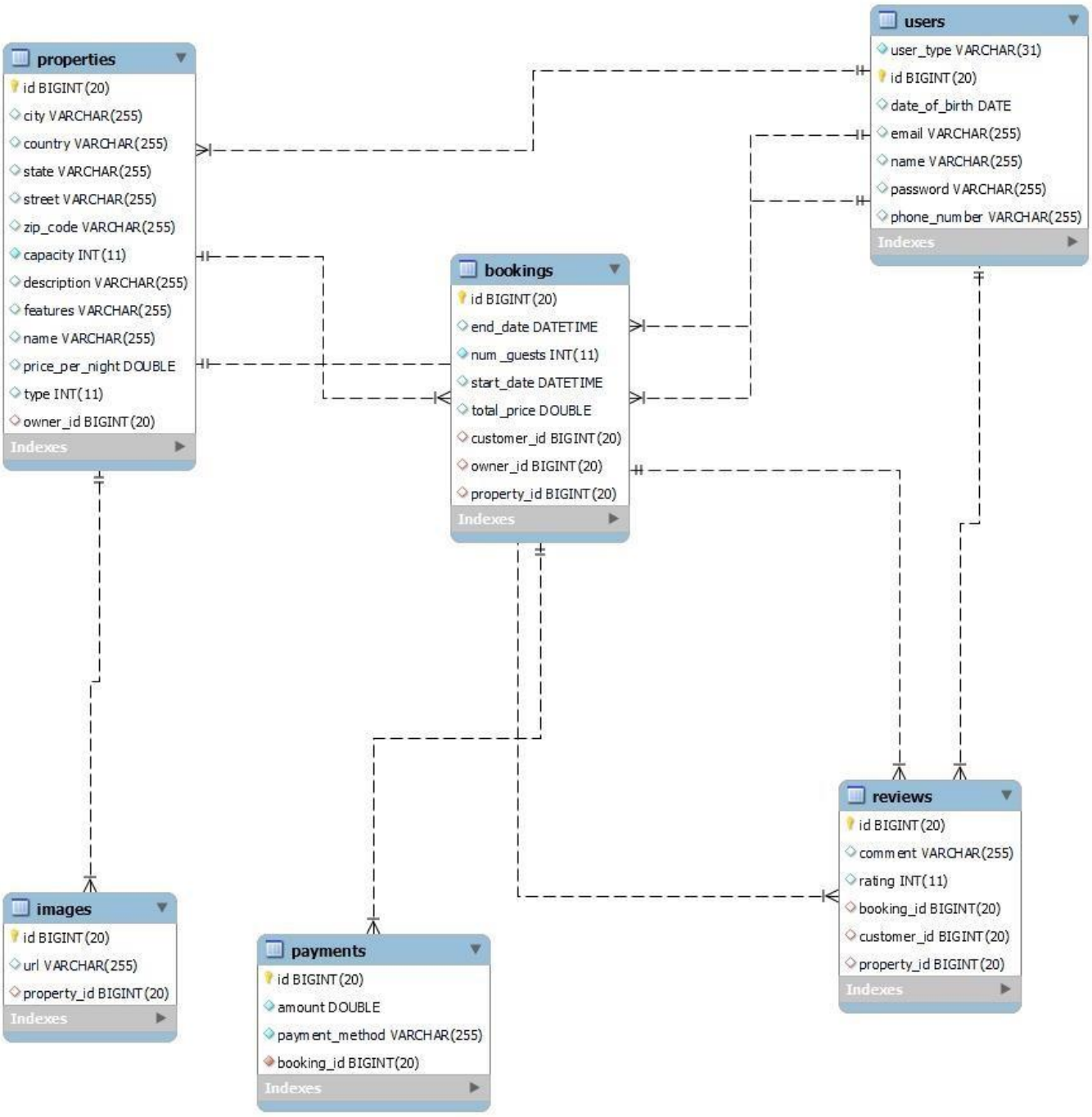
a. Booking and images

```
mysql> desc bookings
    -> ;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| id          | bigint      | NO   | PRI | NULL    | auto_increment |
| end_date    | datetime(6) | YES  |     | NULL    |                |
| num_guests  | int         | NO   |     | NULL    |                |
| start_date  | datetime(6) | YES  |     | NULL    |                |
| total_price | double      | YES  |     | NULL    |                |
| customer_id | bigint      | YES  | MUL | NULL    |                |
| owner_id    | bigint      | YES  | MUL | NULL    |                |
| property_id | bigint      | YES  | MUL | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
8 rows in set (0.02 sec)

mysql> desc images;
+-------------+--------------+------+-----+---------+----------------+
| Field       | Type         | Null | Key | Default | Extra          |
+-------------+--------------+------+-----+---------+----------------+
| id          | bigint       | NO   | PRI | NULL    | auto_increment |
| url         | varchar(255) | YES  |     | NULL    |                |
| property_id | bigint       | YES  | MUL | NULL    |                |
| image_data  | longblob     | YES  |     | NULL    |                |
+-------------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

a. Image blob and payments Table

```
mysql> desc images_blob;
+------------+--------------+------+-----+---------+----------------+
| Field      | Type         | Null | Key | Default | Extra          |
+------------+--------------+------+-----+---------+----------------+
| id         | bigint       | NO   | PRI | NULL    | auto_increment |
| image_data | longblob     | YES  |     | NULL    |                |
| image_name | varchar(255) | YES  |     | NULL    |                |
+------------+--------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> desc payments;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| id             | bigint       | NO   | PRI | NULL    | auto_increment |
| amount         | double       | NO   |     | NULL    |                |
| payment_method | varchar(255) | NO   |     | NULL    |                |
| booking_id     | bigint       | NO   | MUL | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
4 rows in set (0.01 sec)
```

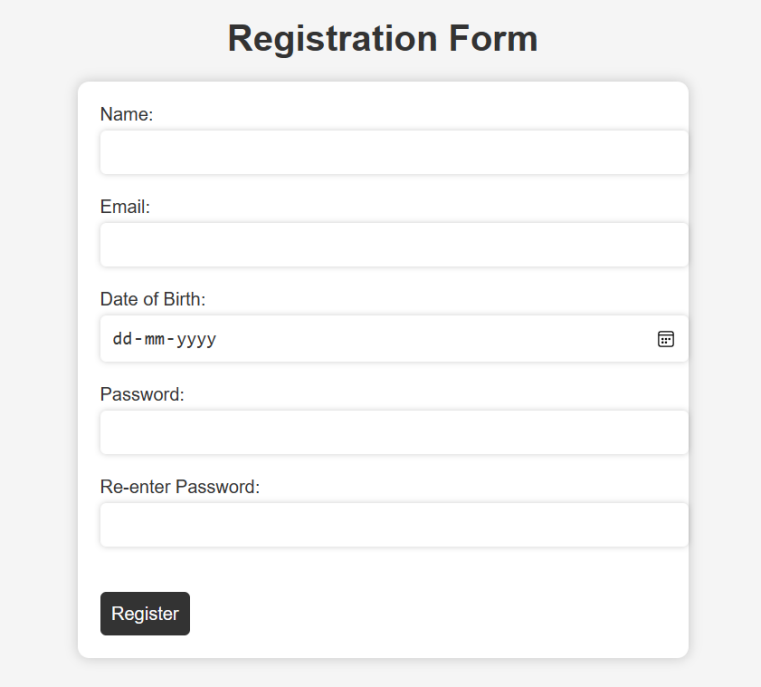
ER-DIAGRAM


properties
id BIGINT(20)
city VARCHAR(255)
country VARCHAR(255)
state VARCHAR(255)
street VARCHAR(255)
zip_code VARCHAR(255)
capacity INT(11)
description VARCHAR(255)
features VARCHAR(255)
name VARCHAR(255)
price_per_night DOUBLE
type INT(11)
owner_id BIGINT(20)
Indexes
users
user_type VARCHAR(31)
id BIGINT(20)
date_of_birth DATE
email VARCHAR(255)
name VARCHAR(255)
password VARCHAR(255)
phone_number VARCHAR(255)
Indexes
bookings
id BIGINT(20)
end_date DATETIME
num_guests INT(11)
start_date DATETIME
total_price DOUBLE
customer_id BIGINT(20)
owner_id BIGINT(20)
property_id BIGINT(20)
Indexes
images
id BIGINT(20)
url VARCHAR(255)
property_id BIGINT(20)
Indexes
payments
id BIGINT(20)
amount DOUBLE
payment_method VARCHAR(255)
booking_id BIGINT(20)
Indexes
reviews
id BIGINT(20)
comment VARCHAR(255)
rating INT(11)
booking_id BIGINT(20)
customer_id BIGINT(20)
property_id BIGINT(20)
Indexes

PROJECT SCREENSHOTS

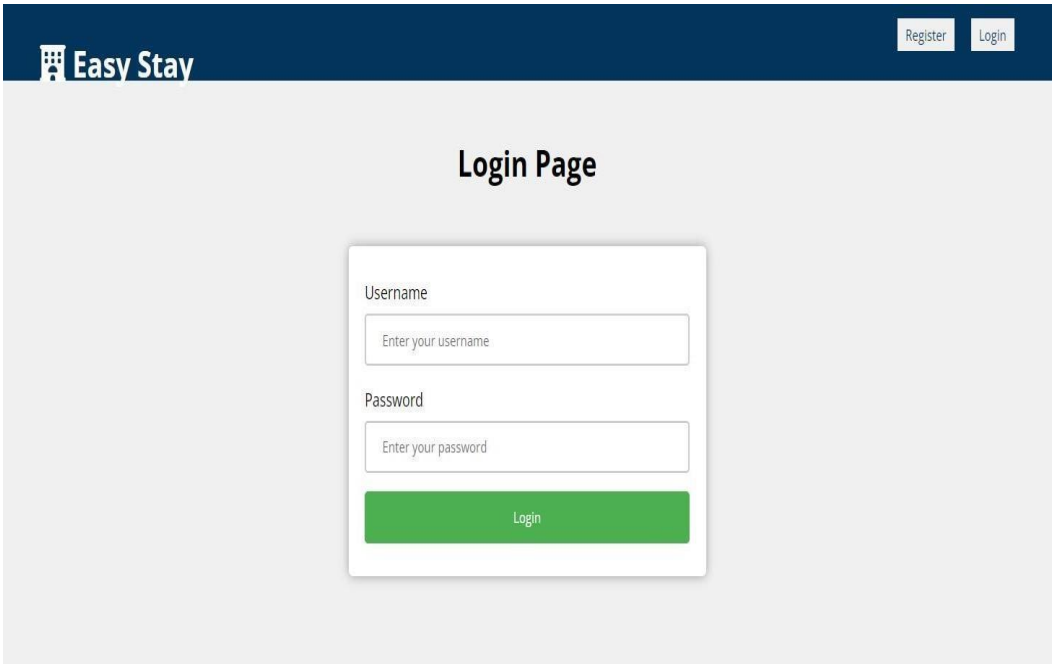
1.2. STUDENT

Register



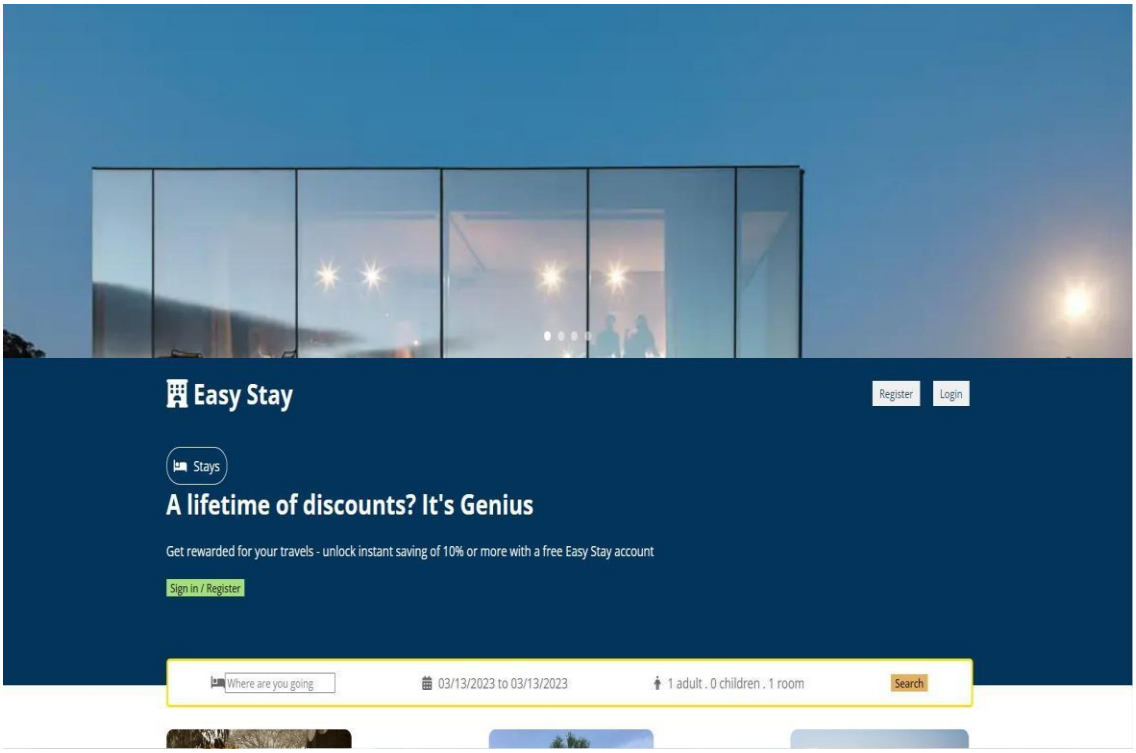
- Customer must give his/her personal details and make a password for registration
- Password must have one Capital Letter and one Special Symbol

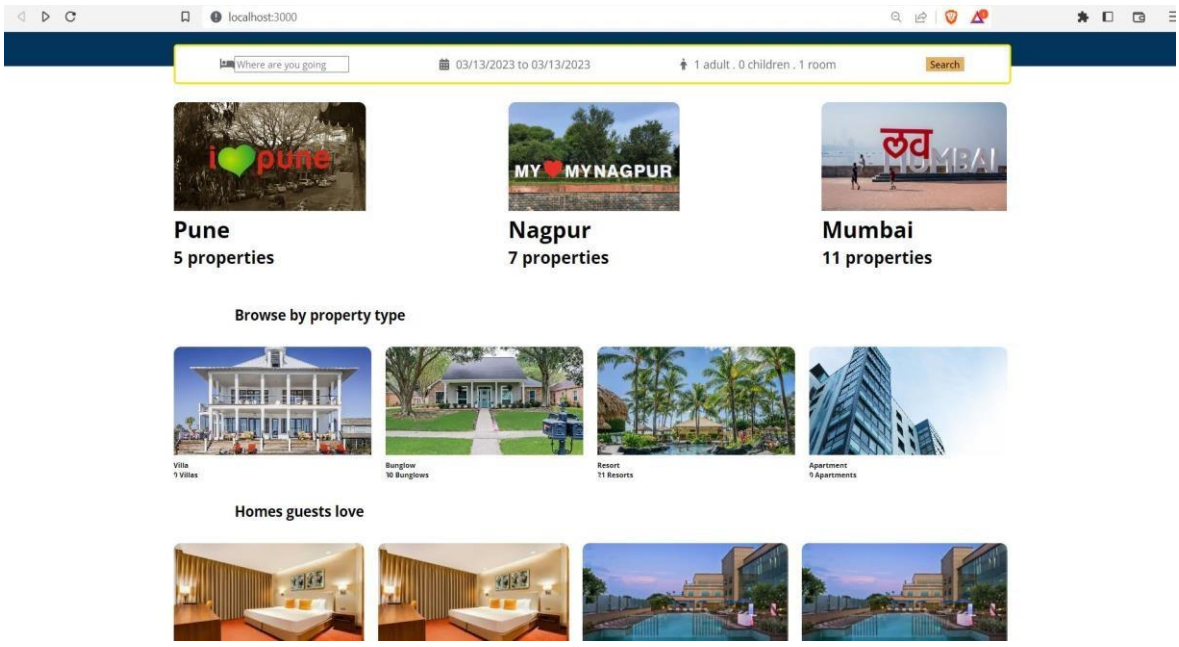
Login



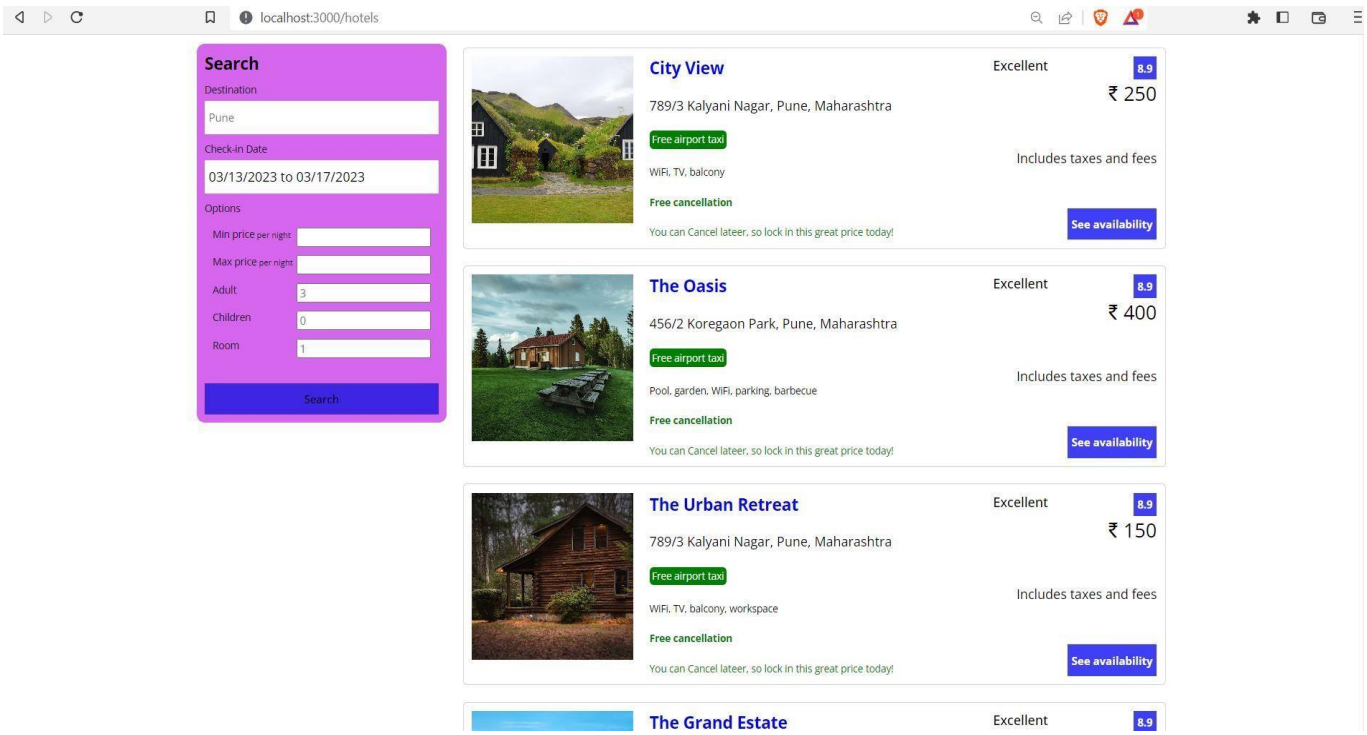
- On Click on login link, the above page will be displayed
- Customer must enter correct email with password given at registration time
- If customer enters some wrong data it will give wrong data message in alert box

Customer Dashboard



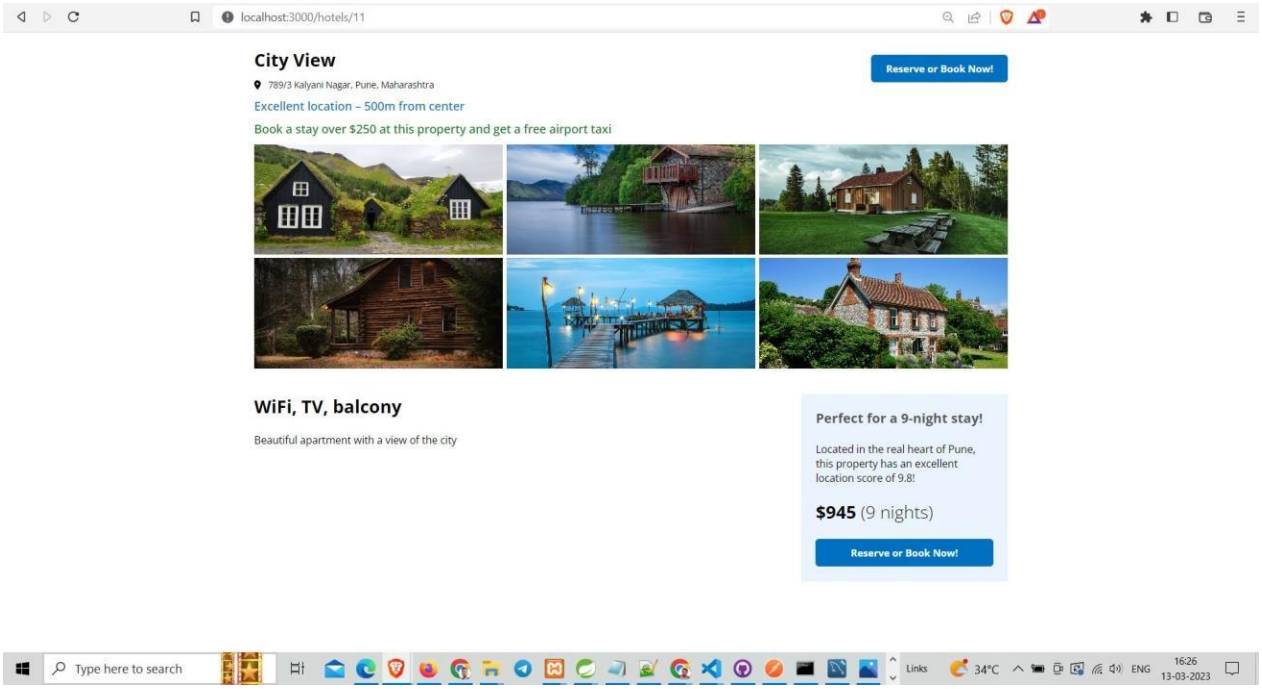


Custom Search



- After Searching a in a particular city ,this page will displayed on screen
- The result takes input as city name
-

Room Information Page



- After Searching the result will be displayed and after click on any room link we will be directed to this page
- The page displays what all room provides as a facility

Property Form

Country:

State:

City:

Street:

Zip Code:

Capacity:

Description:

Zip Code:

Capacity:

Description:

Features:

Name:

Price per night:

Type:

Select a type

Regiester

TESTING

One of the main purposes of testing is to validate and verify that the system works as intended. No program or system design is perfect. However, if we implement the system without proper testing, then it may cause problems and lead to a bad user experience.

Testing and checking outcomes of each test gives us the best chance to detect and correct errors before the system is implemented in a production environment.

In the booking of our project, we tried to manually test each component. In all cases, we obtained the desired results as demonstrated below.

A. CUSTOMER FEATURES TEST

#	Description	Outcome	Result
1.	Register as customer	Register a new Customer	Passed
2.	Login as Customer	Successful login.	Passed
3.	Browse hotels	Can view different hotels available for booking.	Passed
4.	Filter according to cities	If you apply the filter then other city hotels are no longer shown.	Passed
5.	Selecting date	If the dates for booking are not available then the booking is not made otherwise successful booking is done.	Passed
6.	Review	Customer giving review according to satisfaction.	Passed
7.	Logout	The session was cleared.	Passed

B. MANAGER FEATURES TEST

#	Description	Outcome	Result
1.	Login as Manager	Logged in as Manager.	Passed
2.	View bookings	Can see the booked rooms	Passed
3.	Edit the properties	Rooms are added to update properties	Passed
4.	Get Feedback List	Feedbacks given to property by customers.	Passed
5.	Logout	Successfully logged out.	Passed

CONCLUSION

"ROOM BOOKING SERVICE" is an online room booking app, was developed by our project team to provide a platform for booking the hotel room and to simplify the Manager's work. We tried using the latest technologies that are cross-platformand robust. Each and every software we used was open-source in nature, which keeps the cost of production at a minimum.

We were also meticulous about the user experience aspect of our application so that navigating our website is an easy and seamless experience.

In conclusion, "ROOM BOOKING SERVICE" as a portal would definitely be beneficial for the Hotels who have joined the app to manage customers in Online Mode efficiently. We are confident that the numerous features and visually appealing look of the portal will definitely make this portal effective for all the customers and Managers. This portal can be scaled on higher side to add more features to it.

FUTURE SCOPE

Using whatever we have learnt over the duration of this course, we tried to make our project as user-friendly and gave it as many features as possible in the limited time allotted for the project work. That said, there are certainly more features that can be added to our application. Some of those are mentioned below:

1. Manager functionalities can be improved for larger scale.
2. More number of booking filters can be added in the app.
3. If a user forgets password, OTP can be sent to his/her registered mobile number and registered Email Id to reset password.
4. To improve the app on very large scale, lectures and labs can be conducted in the portal only.
5. CAPTCHA can be added to login page.
6. Overall portal can be built on a higher scale so that all functionalities that can rival professional sites that are there.

REFERENCES

Following is the list of websites we referred during the booking of our project:

1. https://getbootstrap.com/docs/5.1/getting-started/introduction/
2. https://www.baeldung.com/
3. https://www.w3schools.com/
4. https://docs.spring.io/spring-data/jpa/docs/current/reference
5. https://javaee.github.io/javaee-spec/javadocs/
6. https://javadoc.io/doc/org.springframework.data/spring-data-jpa/latest/index.html