PROJECT REPORT

ON

"HOSTEL MANAGEMENT SYSTEM"

Carried Out at

CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING BANGALORE

UNDER THE SUPERVISION OF

Mr. SHANMUGANATHAN N

SUBMITTED BY

Student Name	PRN No.
Abhinav Ablash	220350120003
Abhishek H M	220350120004
Akash Suhas Bhamare	220350120009
Nikhil Nagorao Chintawar	220350120023
Aniket Ashok Kadu	220351920012

CERTIFICATE

This is to certify that the work entitled **"Hostel Management System"** is submitted by

Abhinav Ablash (220350120003),

Abhishek H M (220350120004),

Akash Suhas Bhamare (220350120009),

Nikhil Nagorao Chintawar (220350120023),

Aniket Ashok Kadu (220351920012),

The Bonafide Students of **Diploma in Advanced Computing, CDAC, Knowledge Park, Bangalore.** The Course End Project work is carried out under my direct supervision and completed.

Mr. SHANMUGANATHAN N
C-DAC,
Bangalore - 560100, India

Candidate's Declaration

We hereby declare that the work being presented in the report entitled **"Hostel Management System"**, in partial fulfillment of the requirements for the award of PG Diploma Certificate and submitted in the department of PG-DAC of the C-DAC Bangalore, is an authentic record of our work carried out under the supervision of Mr. Shanmuganathan N, C-DAC Bangalore.

Student Name	**PRN No.**
Abhinav Ablash	220350120003
Abhishek H M	220350120004
Akash Suhas Bhamare	220350120009
Nikhil Nagorao Chintawar	220350120023
Aniket Ashok Kadu	220351920012

ACKNOWLEDGEMENT

This project **"Hostel Management System"** was truly a great learning experience for us and we are submitting this work to Advanced Computing Training School (CDAC ACTS).

We are very glad to mention **Mr. Shanmuganathan N** for his valuable guidance to work on this project. His guidance and support helped us to overcome various obstacles and intricacies during the course of project work.

We are highly grateful **to Mr. Shanmuganathan N**, Manager of ACTS training Centre, C-DAC, for his guidance and support whenever necessary during the course of our journey to acquire Post Graduate Diploma in Advanced Computing (PG-DAC) through C-DAC ACTS, Bangalore. Our heartfelt thanks go to **Ms. Uma Prasad** our Course Coordinator, PG-DAC who gave all the required support and kind coordination to provide all the necessities.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Now a days, number of people using internet is increasing exponentially. So, the use of web applications is also increasing. This is a proposal to design a web application for booking and allotment of hostel rooms for students. The purpose of the project entitled as “Hostel Management System” is to save time of users and administration. Many of them spend their time by visiting the administration directly and gather information, discuss with the people and pay the money. By this application, user can directly see room vacancy, fees and book the hostel rooms and administration can keep a track of users/student records and room booking status from his place without wasting of time.

Hostel Management System is a web application which gives information on hostel room vacancies available to the users/students, the user has to signup/sign in and register in the web application and select the room and book the room. Admin can keep track of students/users records. The user/student can select the room they prefer from the list of rooms and can pay the fees to confirm their booking status.

Chapter 1 Introduction

1.1. Introduction

Hostel Management System is a web application used primarily for booking a room in a hostel and making payment from Users/Students standpoint. From Administration standpoint, the application is used for maintaining and managing different records like student records, hostel records, payment history, complaints, booking details etc…. This system replaces paper work based management system and move towards digitalization.

1.2. Problem Statement:

In the current era of digitalization, more and more Systems and Processes are becoming automated day by day. In such an era, it is not advisable to be using a manual system to manage Hostels. Without a management system, registration of students, allotment of rooms, maintaining records becomes a tedious and time-consuming task for both students as well as the administration. Along with that, there is also an issue with the storage of records and its security. Payments can be carried out through cash which does not help in digitalization. Therefore, a new system is proposed which replaces all of these activities and move towards digitalization.

1.3. Product Scope:

The software product “Hostel Management System” will be an application that will be used for registering to a hostel and booking a room to maintaining the records in an organized manner and replace old paper work system. This project aims at automating the activities involved in booking a hostel and maintaining relevant records for smooth working of the hostels.

1.4. Aims & Objective:

The aim of the project is to come up with a solution and build a Hostel Management Web Application which addresses all the problems described in the Problem Statement section.

Chapter 2 Overall Description

2.1. Introduction

In this chapter, we will discuss about the product's existing and proposed system along with benefits and disadvantages. Users and their characteristics along with the available modules in the proposed system are described in brief. Also operating requirements both at server and client side are given along with the implementation constraints.

2.2. Product Perspective

2.2.1.1 Existing System Function

The existing system is manual based and need lot of efforts and consume enough time. In the existing system we can apply for the hostels online but the allotment processes are done manually. It may lead to errors in the allocation process as well as hostel fee calculation. The existing system does not deal with complaint registration.

- **Disadvantages:**
  - More human resources are required
  - Repetition of same procedure
  - Low Security
  - Data Redundancy
  - Difficult to handle
  - Difficult to update data
  - Record Keeping is Difficult

2.2.1.1 Proposed System

This project is aimed at developing a system for keeping records and showing information about students and hostel. This system will help the administrator to be able to manage the affairs of the hostel. This system will provide full information about a student in the hostel. It will show rooms available or not and details of a student in a particular room. This will also provide information on students who have paid fees.

This system will be developed based on Software Development Life Cycle (SDLC) with Spring Boot, React and My SQL server. Spring Boot and React is good for the development and design of web

based programs whiles My SQL is good for databases because of its security and its advanced features and properties.

2.3. Benefits

Benefits of Hostel Management System:

- This system is fully functional and flexible
- Easy to use
- Time Saving
- Lots of paper work is saved
- Data redundancy is avoided
- Available 24/7
- Data is secured
- Manual labour to maintain record is minimized
- Administrator can add Hostels and Rooms
- Administrator can see the payment status, student details, hostel and room details
- Students can see the availability of Rooms and Book the Room
- Students can pay the Hostel Fees on the portal through online mode
- Students have the choice to select the room
- Students can raise complaints/concerns
- Administrator can go through the concerns

2.4. User and Characteristics

2.4.1.1 Administrator Module

- **Admin Login:**

  Admin can login using verified username and password.

- **Student Management**

  Admin can view the student details

- **Rooms & Hostel Management**

Admin can view, add, update and delete Hostels and Hostel Details. Admin can view, add, update and delete Rooms in the Hostel.

- **Payment and Allotment Management**

  Admin can view the payment history, view allotment status of the student, check the vacant rooms.

- **Concern Management**

  Admin can view the concerns raised by the student and details of the student.

2.4.1.1 Student Module

- **Student Login:**

  Student can create an account, register and become a member. After login he can view, add, update personal information.

- **Room Selection Module**

  Student can see the availability of rooms and choose the room of choice.

- **Payment Module**

  Student can pay the fees for selected room through online mode.

- **Allotment Module**

  Student can see the details of allotted hostel and room after successful fee payment.

- **Concern Module**

  Student can raise concerns which will be visible to admin.

2.5. Operating Environment

Server Side:

Processor: Intel® Xeon® processor 3500 series

HDD: Minimum 500GB Disk Space

RAM: Minimum 4GB OS: Windows 10, Linux 6 Database: MySQL

Client Side (minimum requirement):

Processor: Intel Dual Core

HDD: Minimum 80GB Disk Space

RAM: Minimum 2GB

OS: Windows 7, Linux

2.6. Design and Implementation Constraints

- The application will use HTML, JavaScript, Bootstrap and CSS, ReactJS as main web technologies.
- HTTP Protocols are used as communication protocols
- JSON Web Token is used for authentication and authorization providing security
- Several types of validations make this web application a secured one and SQL Injections can also be prevented.
- As the Hostel Management System is a web-based application, internet connection must be established.
- The Hostel Management System will be used on PCs and will function via internet or intranet in any web browser.

Chapter 3 Requirements Specification

3.1. External Interface Requirements

3.1.1. User Interfaces

- All the users will see the same page when they enter in this website. This page asks the users a username and a password.
- After being authenticated by correct username and password, user will be redirect to their corresponding profile where they can do various activities.
- The user interface is simple and consistent, using terminology commonly understood by intended users of the system. The system will have simple interface, consistence with standard interface, to eliminate need for user trainingof infrequent users.

3.1.2. Hardware Interface

- No extra hardware interfaces are needed.
- The system will use the standard hardware and data communication resources.
- This includes, but not limited to, general network connection at the server/hosting site, network server and network management tools.

3.2. Non-Functional Requirements

3.2.1. Application Interfaces

OS: Windows/Linux/MacOS

Web Browser: The system is a web-based application; clients need a modern web browser such as Mozilla Firebox, Internet Explorer, Opera, and Chrome. The computer must have an Internet connection in order to be able to access the system.

3.2.2. Communications Interfaces

- This system uses communication resources which includes but not limited to, HTTP protocol for communication with the web browser and web server and TCP/IP network protocol with HTTP protocol.

- This application will communicate with the database that holds all the hostel booking information. Users can contact with server side through HTTP protocol by means of a function that is called HTTP Service. This function allows the application to use the data retrieved by server to fulfil the request fired by the user.
- Operating environment for the Hostel management system is as listed below.
    - Client/Server system
    - Operating system: Windows/Linux/MacOS.
    - Database: MySQL database
    - Platform: HTML/Java/JavaScript

Chapter 4 System Design

4.1. Activity Diagram

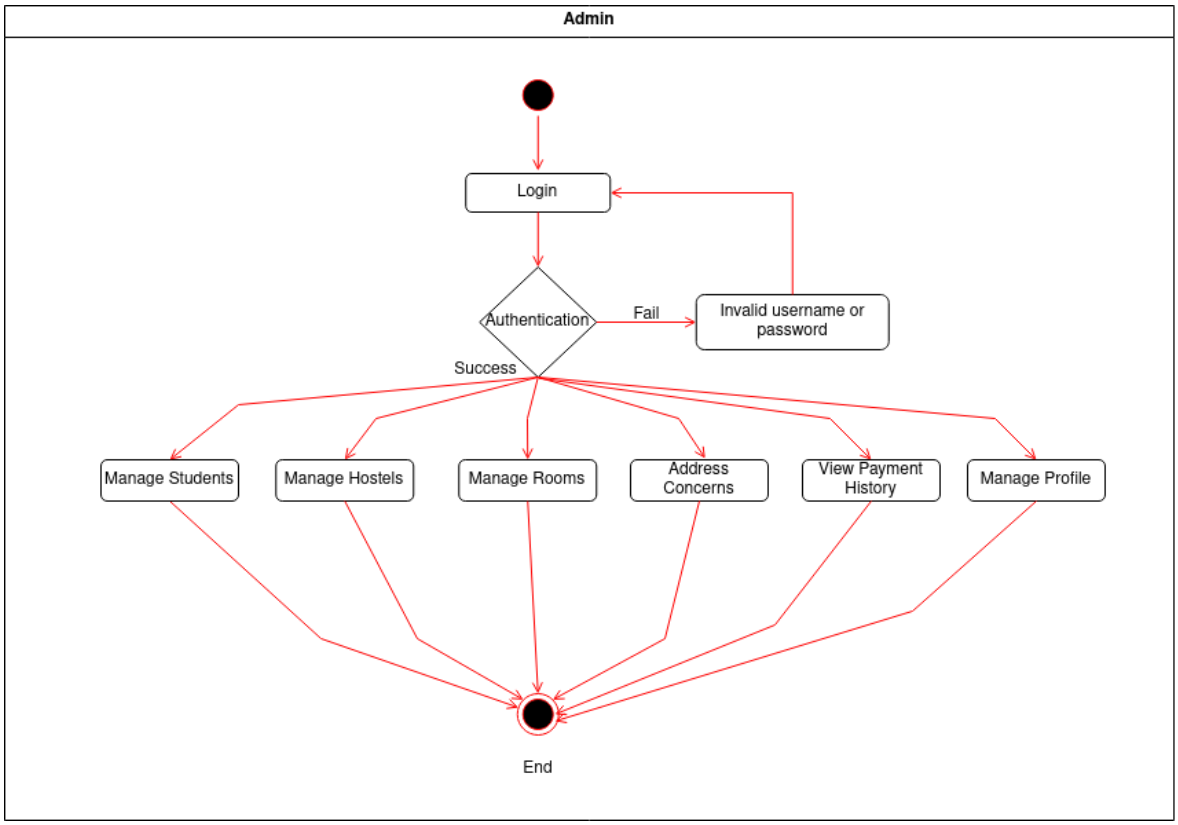

Figure 1 Admin Activity Diagram

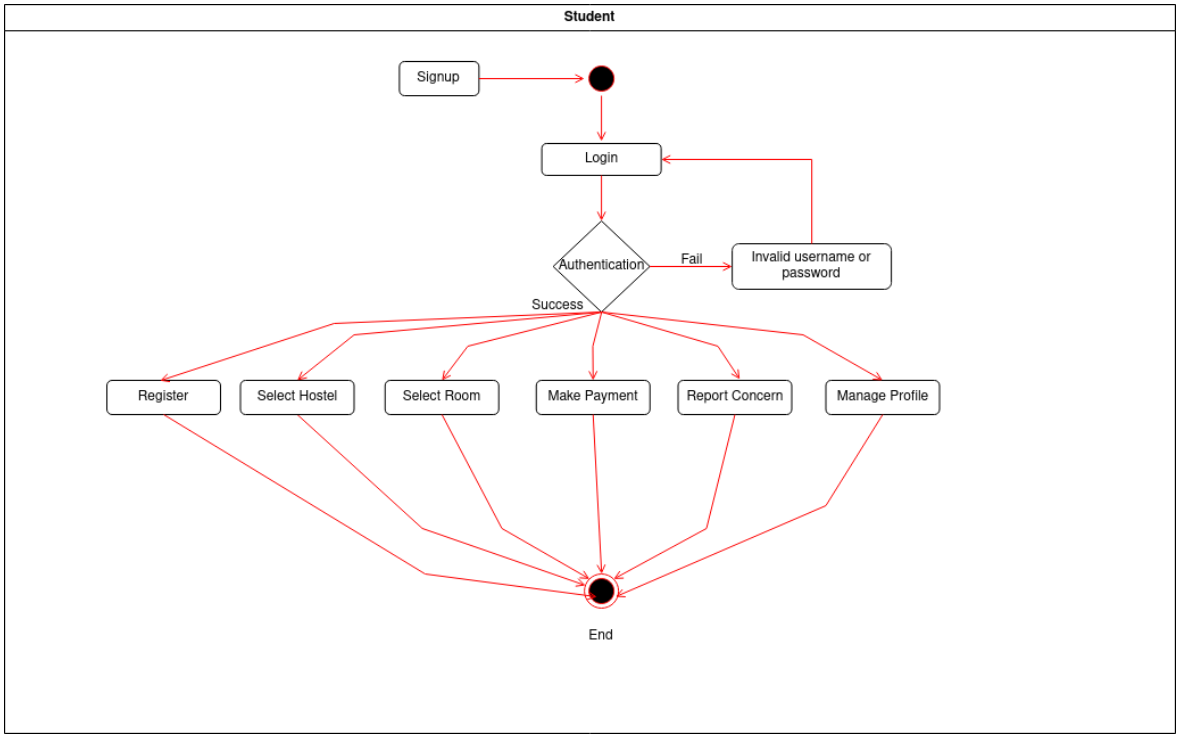

Figure 2 Student Activity Diagram

4.2. Use Case Diagram

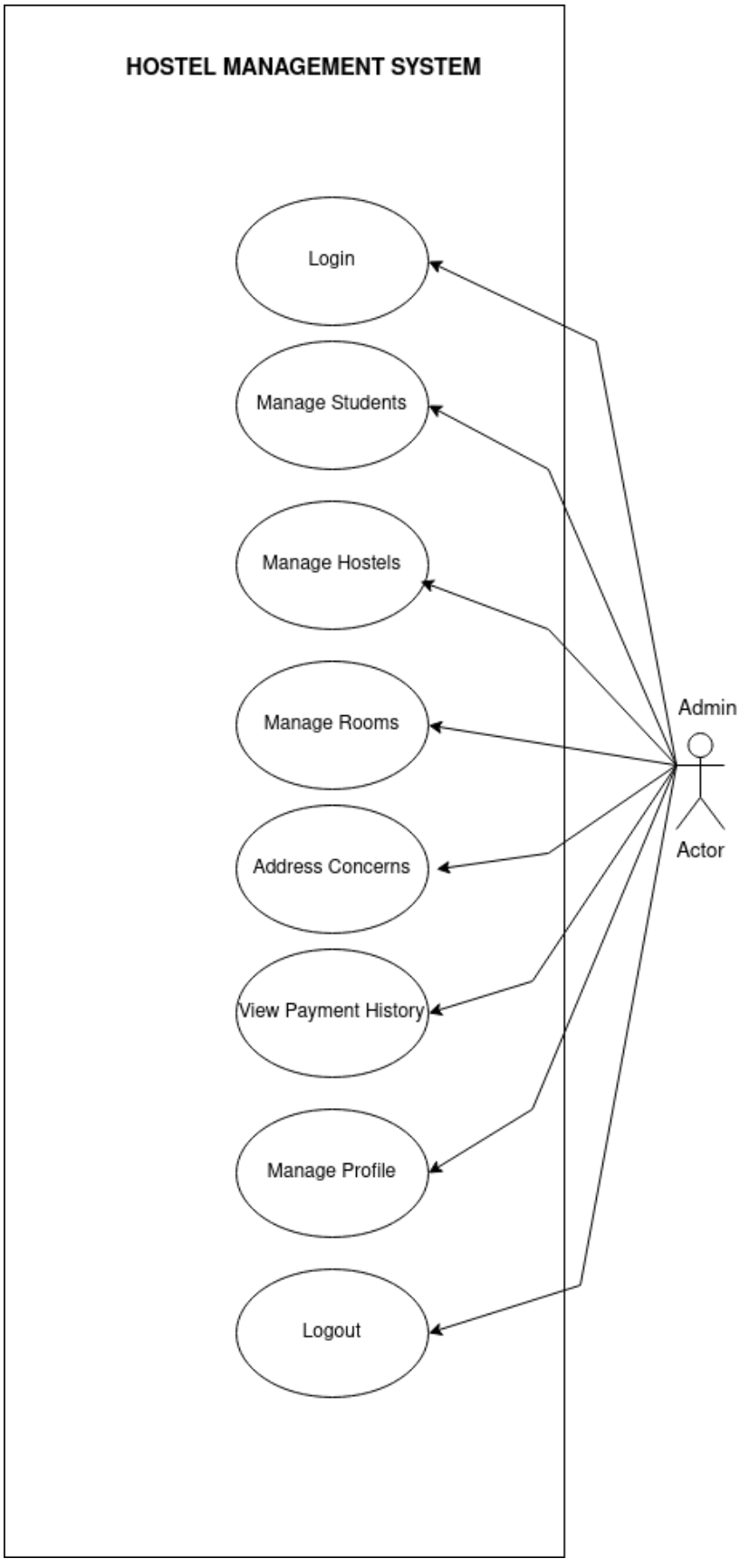

Figure 3 Admin Use Case Diagram

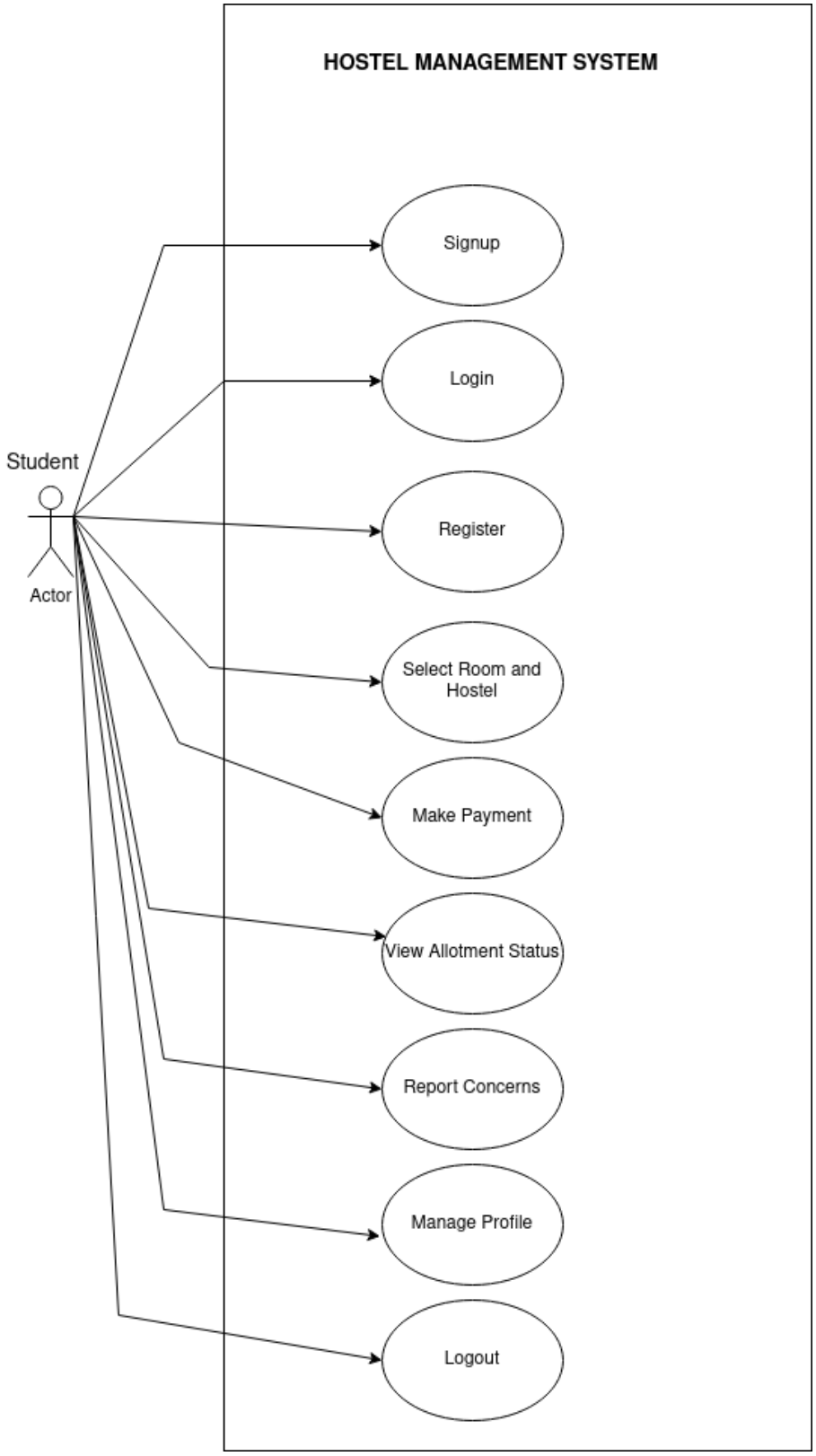

Figure 4 Student Use Case Diagram

4.3. ER Diagram

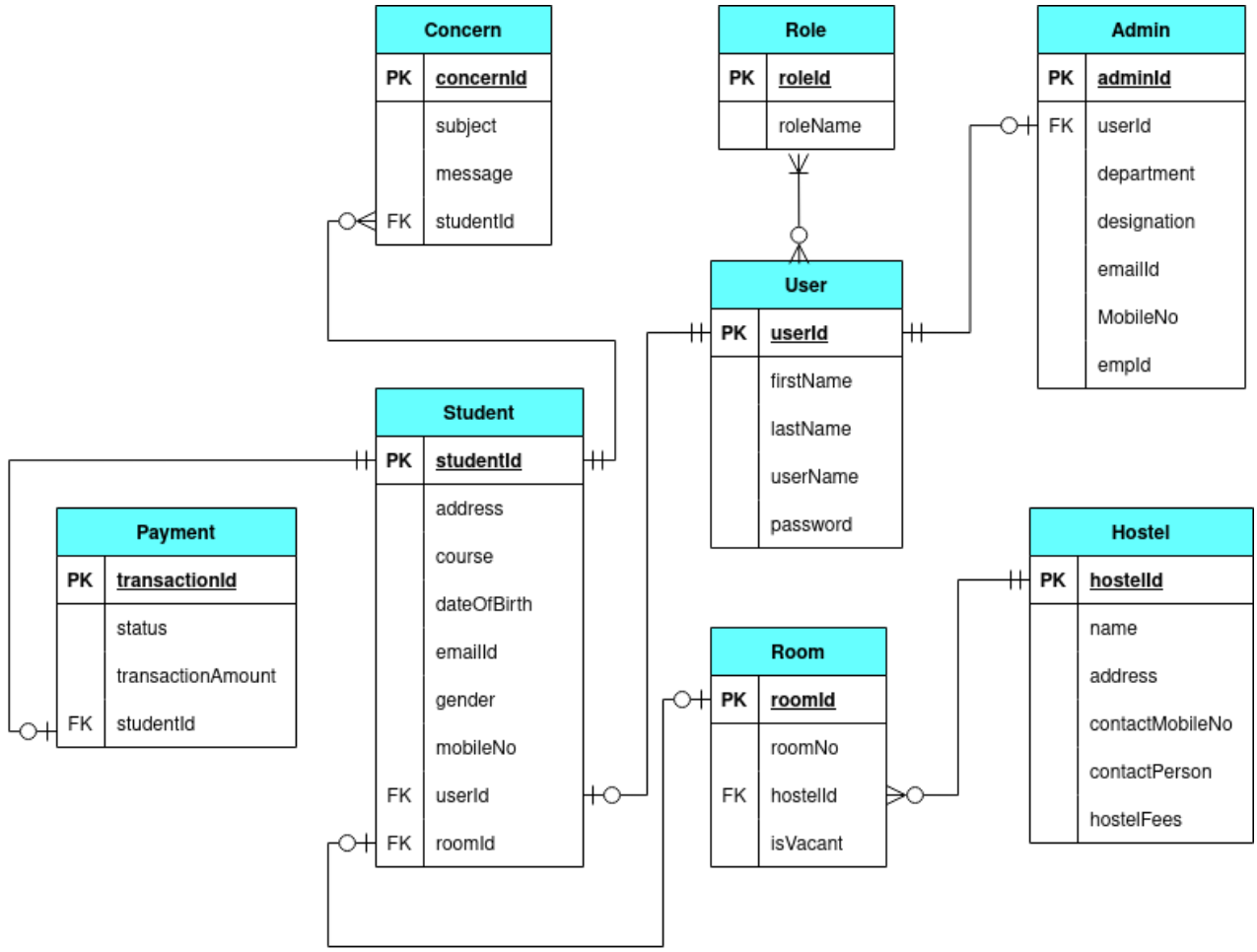

Figure 5 ER Diagram

Chapter 5 Database Design & GUI

5.1. Database Design

5.1.1. Admin Table

Table 1 Admin

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
department	varchar(255)	YES		NULL	
designation	varchar(255)	YES		NULL	
email_id	varchar(255)	YES		NULL	
emp_id	varchar(255)	YES		NULL	
mobile_no	bigint	YES		NULL	
user_id	bigint	YES	MUL	NULL	

5.1.2. Concerns Table

Table 2 Concerns

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
message	varchar(255)	YES		NULL	
subject	varchar(255)	YES		NULL	
student_id	bigint	YES	MUL	NULL	

5.1.3. Hostels Table

Table 3 Hostels

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
address	varchar(255)	YES		NULL	
contact_mobile_no	bigint	YES		NULL	
contact_person	varchar(255)	YES		NULL	
hostel_fees	int	YES		NULL	
name	varchar(255)	NO		NULL	

5.1.4. Payments Table

Table 4 Payments

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
transaction_date	date	YES		NULL	
transaction_id	varchar(255)	YES		NULL	
transaction_status	varchar(255)	YES		NULL	
student_id	bigint	YES	MUL	NULL	

5.1.5. Roles Table

Table 5 Roles

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
name	varchar(255)	NO		NULL	

5.1.6. Rooms Table

Table 6 Rooms

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
is_vacant	tinyint	NO		NULL	
room_no	int	YES	UNI	NULL	
hostel_id	bigint	YES	MUL	NULL	

5.1.7. Students Table

Table 7 Students

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
address	varchar(255)	YES		NULL	
course	varchar(255)	YES		NULL	
date_of_birth	date	YES		NULL	
email_id	varchar(255)	YES		NULL	
gender	varchar(255)	YES		NULL	
mobile_no	bigint	YES	UNI	NULL	
room_id	bigint	YES	MUL	NULL	
user_id	bigint	YES	MUL	NULL	

5.1.8. Users Table

Table 8 Users

Field	Type	Null	Key	Default	Extra
id	bigint	NO	PRI	NULL	auto_increment
first_name	varchar(255)	NO		NULL	
last_name	varchar(255)	NO		NULL	
password	varchar(255)	NO		NULL	
user_name	varchar(255)	NO		NULL	

5.1.9. Users Roles Table

Table 9 User Roles

Field	Type	Null	Key	Default	Extra
user_id	bigint	NO	MUL	NULL	
role_id	bigint	NO	MUL	NULL	

5.2. Login and Signup Section

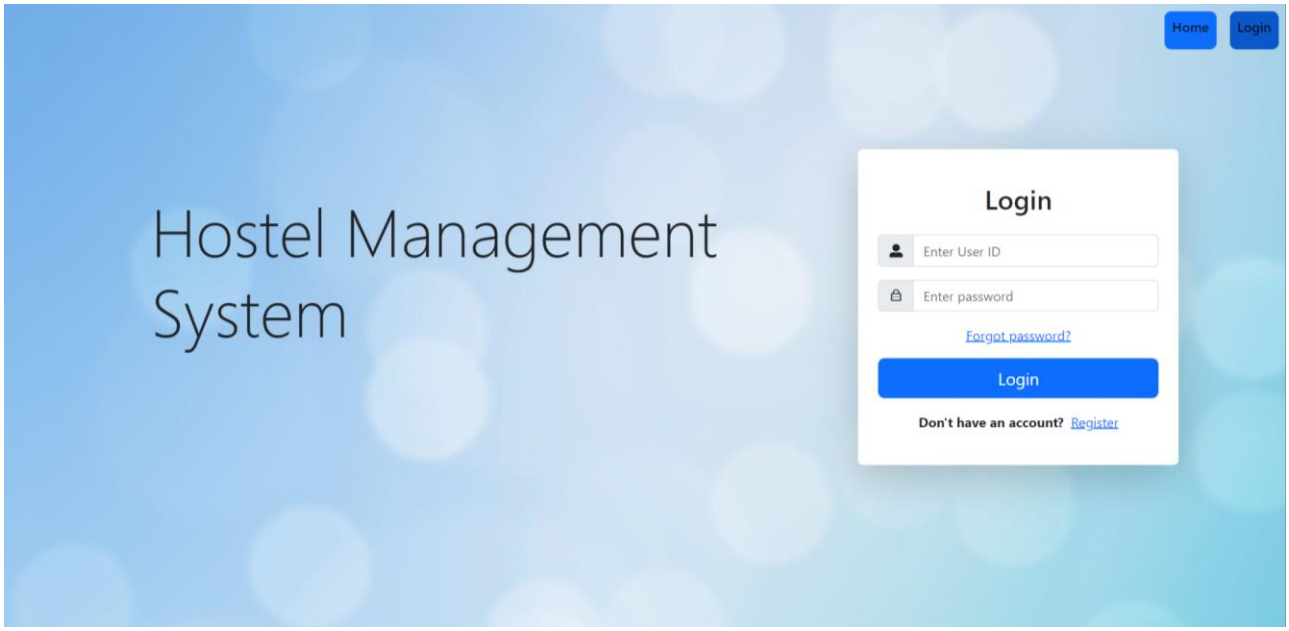

Figure 6 Login Page

Student and Admin can login using the login page shown above using username and password. The information is sent to server and it checks for match in database. If no match is found, then an appropriate error message is shown to the user. If data matches in databases then he/she will be redirected to dashboard page.

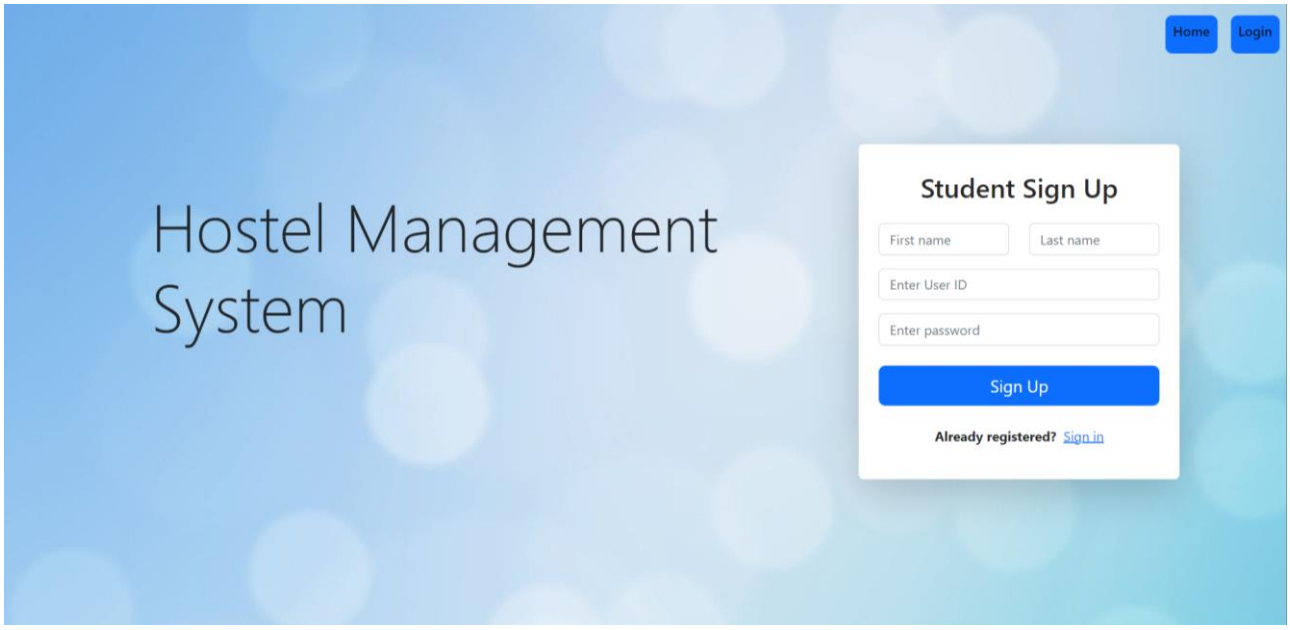

Figure 7 Student Sign Up page

Students can register using First Name, Last Name, username and password. The data will be sent to server where it will be checked if username is available or not and will send suitable response. Once the signup is successful, success message will be displayed and user will be redirected to Login page.

5.3. Admin Account

Admin can update his/her details in profile section. Admin can see the student details from Students section.

Admin can view, edit, add and delete Hostels and Rooms from Hostels tab.

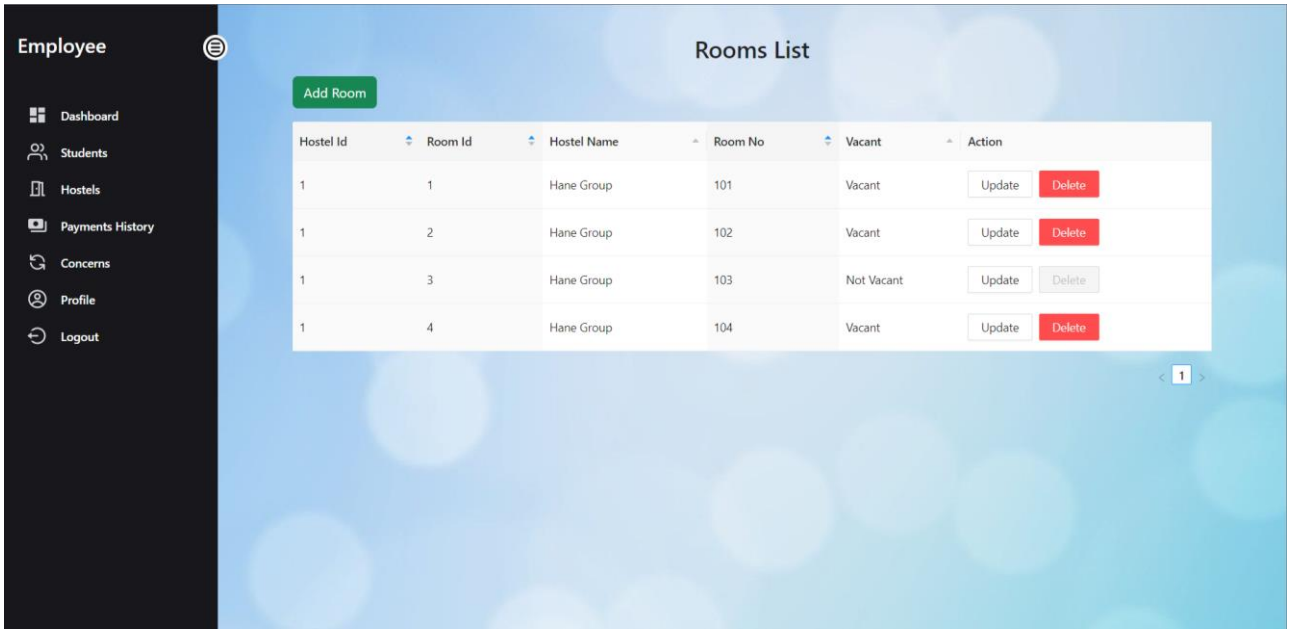

Figure 8 Rooms List (Admin)

Admin can see the payments history from the Payments History tab.

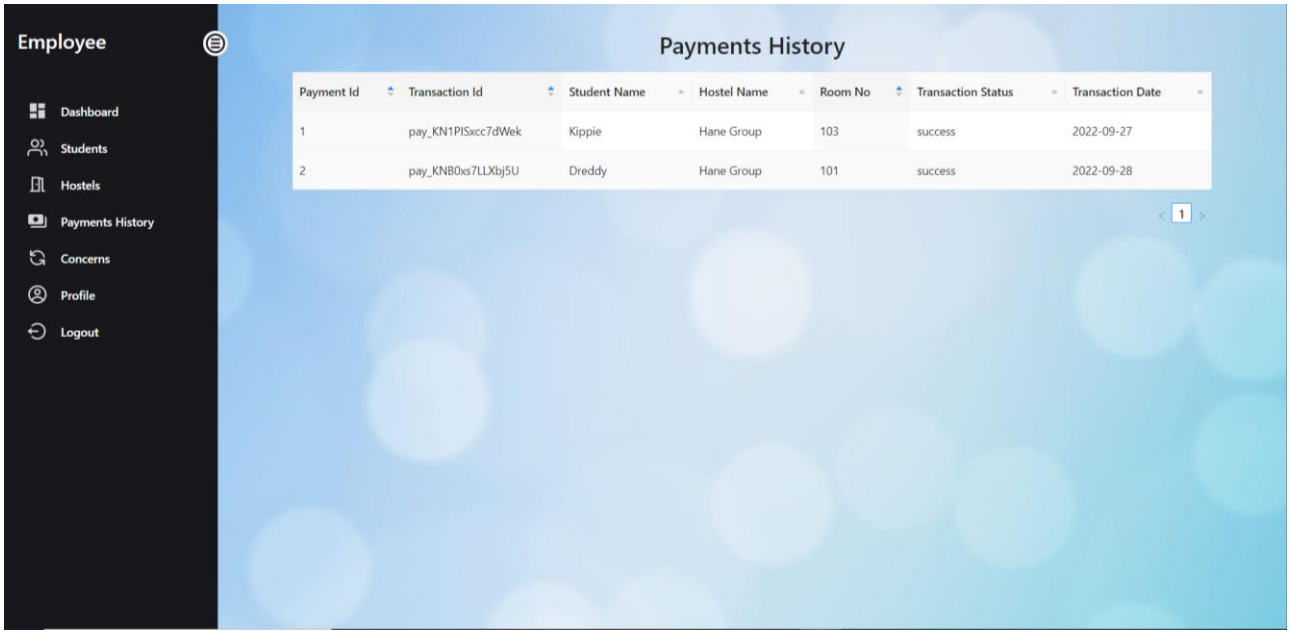

Figure 9 Payment History Page

Also Admin can see the concerns raised by the students along with their details from concerns tab.

5.4. Student Account

After Login student will be redirected to Student Dashboard page.

From Register section, he can add his details.

From Room Status tab he can book the room and if he has already booked the room then allotment details like hostel name, address, details of contact person will be shown along with room id and payment status.

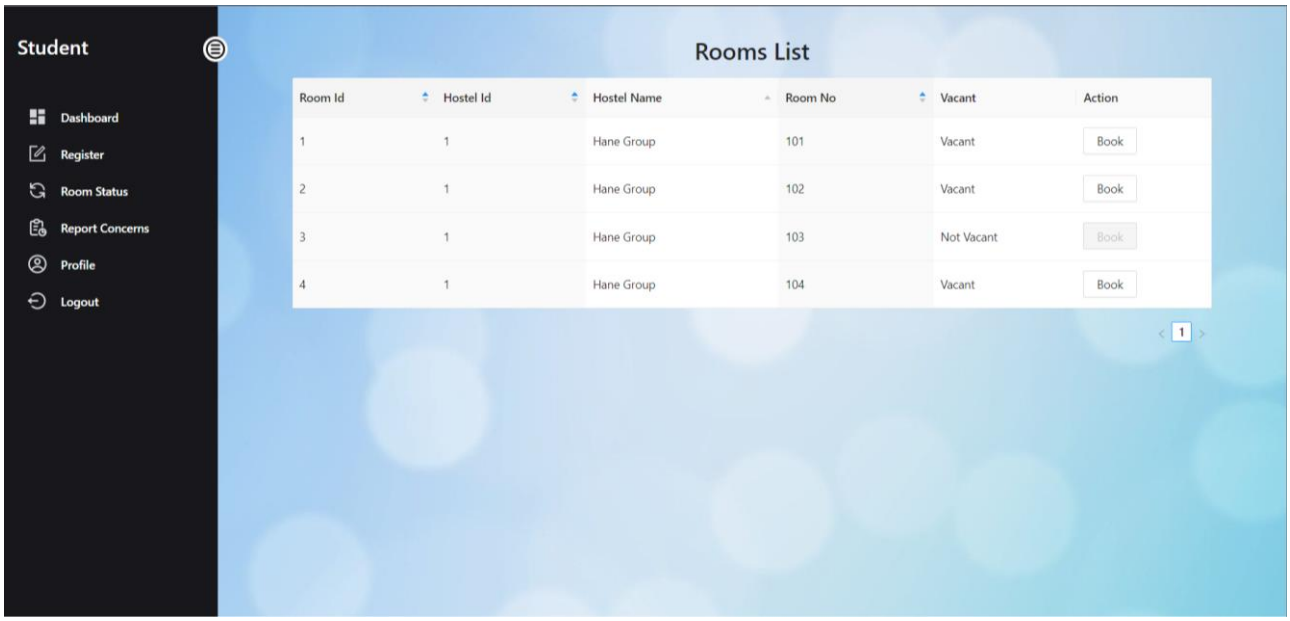

Figure 10 Room Selection Page

Student will be redirected to Payment API after he clicks on book button.

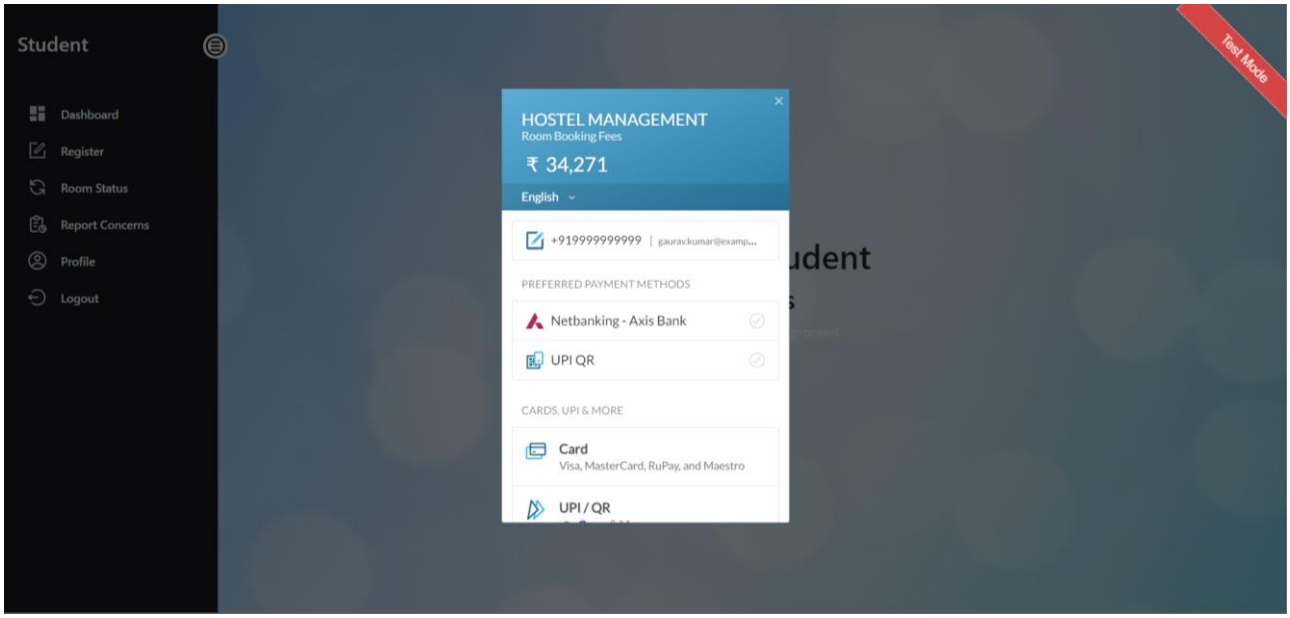

Figure 11 Payment Page

After successful payment, message will be displayed and the room will be allotted to the student. He can see the status of allocation using same room status tab. Allotment status will be displayed to the student.

He can report concerns which will be sent to server and admin can view the concerns raised by the students.

Student can view his details in Profile page. He can also update the details using the update button provided.

Chapter 6 Implementation

6.1. General

The created application is based entirely on free source applications and with the intention to cut cost. All of the functionalities need to be carried out in an orderly manner. The technologies used for the project is CSS, Bootstrap, Ant Design, ReactJS, Java and MySQL.

6.2. Implementation of Functions

Customer Interface and admin interface are the main interfaces created in this project. With the customer and admin interface, we used ReactJS. The jsx was used to create the structure of page while the bootstrap and Ant Design is used in styling of pages. Javascript is dynamic language, so we can make use of asynchronous API calls. These API calls were used for transferring data from front-end to back-end and vice versa.

```
import axios from "axios";

class AuthService {

  signin(userlogin) {
    console.log(userlogin);
    return axios.post('http://localhost:7777/signin', userlogin);
  }

  signup(studentdetail) {
    return axios.post('http://localhost:7777/signup', studentdetail);
  }

  getCurrentUser() {
    return JSON.parse(localStorage.getItem('user'));;
  }

}

export default new AuthService();
```

6.3. Rest API

RestFull API is used in backend. All the data that is passed between front-end and back-end is in JSON format and uses Http Protocol. The REST API is secured with Spring security and JWT. The response with appropriate status code is sent wrapped in the Response Entity to the frontend.

```java
@GetMapping("/rooms/display")
public ResponseEntity<List<RoomDto>> displayRooms(@RequestParam long hostelId) {
    System.out.println(hostelId);
    return ResponseEntity.ok().body(roomServiceImpl.displayRooms(hostelId));
}

@PutMapping("/room/update")
public ResponseEntity<?> updateRoom(@RequestBody RoomDto roomDto) {
    roomServiceImpl.updateRoom(roomDto);
    return ResponseEntity.ok().body(new ApiResponse(true, "Room updated successfully"));
}

@DeleteMapping("/room/delete")
public ResponseEntity<?> deleteRoom(@RequestBody RoomDto roomDto) {
    roomServiceImpl.deleteRoom(roomDto);
    return ResponseEntity.ok().body(new ApiResponse(true, "Room deleted successfully"));
}
```

6.4. Basic React Code

We are using ReactJS as our view layer as it is one of the most popular programming frameworks of UI. The requests are sent and received using 'axios' API of React. The JWT tokens in front end are managed with the help of session storage. The tokens are stored in session storage of browser and are used in further requests until they are expired. At the time of logout, the tokens are removed from the session storage.

```javascript
import { useNavigate } from 'react-router-dom';
import { useEffect } from 'react';

export default function Logout() {

    let navigate = useNavigate()
    useEffect(() => {
        localStorage.clear();
        sessionStorage.clear();
        navigate('/');
        alert("Logged-out Successfully")
        window.location.reload()
    }, []);
    return (
        <></>
    );
}
```

Basic Functional Component

6.5. Login Module

Admin and student login are provided using same module. Student and admin can login using their registered username and password. Spring security will validate the username and password and after

validation jwt token and user role are sent to the user. If the validation fails then the appropriate message will be sent to the user.

6.6. Student Services

Student with connectivity to internet can access the application. Student can register and create account to access the services. He can select hostel and room based on his choice and can book the room using online payment. After the room booking, booking status along with hostel details will be displayed. He can report concerns which can be directly seen by the admin.

6.7. Admin Services

Admin can change his details using update profile tab. He can see the list of students and their details.

Admin can add Hostels along with hostel details. Admin can add rooms in the hostel. Also hostel can be deleted, updated. Same goes for the rooms, room details can be updated and can be deleted. Admin can check the payments history. Admin can go through the concerns raised by the students.

Chapter 7 Conclusion

Education is a major global industry which is growing at a high rate like any other industry. Access to relevant and accurate information is at the heart of Education.

Here, the proposed project on Hostel Management System tries to bridge the gap by noting what a user/student perceives as relevant. Hence, the aim of this project entails the design and implementation of a platform that will assist both administration and users in gaining access to hostel amenities or access to rooms booking and records of student/users to admin. The project also helped to provide knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future.

This project helps in making paperless activities. It reduces the workload from Administrative/users' perspective. It provides more flexibility and ease to users and admin.

It is developed using, ReactJS, MySQL, bootstrap, Ant Design, Spring boot technology. This work has created a little awareness and promotes the idea that the concept of paperless office is reality.

References

- https://reactjs.org/docs/getting-started.html
- https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/
- https://developer.mozilla.org/en-US/docs/Web/JavaScript
- https://docs.oracle.com/en/java/
- https://dev.mysql.com/doc/
- https://www.eclipse.org/documentation/
- https://code.visualstudio.com/docs/
- https://razorpay.com/docs/#home-payments/
- Ant Design of React - Ant Design