

Dharmsinh Desai University, Nadiad
Faculty of Technology



Computer Engineering Department
B.Tech - Semester – VI

Subject: Web Services Development

Project Title: Hotel Management System

Guided by: Prof. Prashant M. Jadav

Bhadresha Priyanshi Rajeshbhai (CE017) [20CEUOG016]

Dharmsinh Desai University, Nadiad
Department of Computer Engineering
Faculty of Technology



CERTIFICATE

This is to certify that the project work carried out
in the subject of
Web Services Development
is the bonafide work of

Bhadresha Priyanshi Rajeshbhai (CE017) [20CEUOG016]

of B. Tech. Semester-VI Computer Engineering during the academic
year **2022-2023**.

Prof. Prashant M. Jadav
Associate Professor
Computer Engg. Department
Faculty of Technology
D.D.U., Nadiad

Dr. C. K. Bhensdadia
Professor and HoD,
Computer Engg. Department
Faculty of Technology
D.D.U., Nadiad

INDEX

| Sr. No. | Topics | Page No. |
|----------------|---|-----------------|
| 1. | Abstract | 4 |
| 2. | Introduction | 5 |
| | ● Technology and tools used | 6 |
| 3. | Software Requirements Specifications | 7 |
| | ● Scope | 7 |
| | ● System Function Requirements | 7 |
| | ● Non-Functional Requirements | 9 |
| 4. | Design: | 10 |
| | ● Use Case Diagram | 10 |
| | ● ER Diagram | 11 |
| | ● Data Dictionary | 12 |
| 5. | System Screenshots | 13 |
| 6. | Conclusion | 22 |
| 7. | Limitations and Future enhancements | 23 |
| 8. | Bibliography | 24 |

1. Abstract

Hotel Management System

This hotel management system, built with React and asp .net core web API.

A hotel management system is a software application that helps hotel administrators and staff manage their day-to-day operations. The system provides tools for managing reservations, check-ins and check-outs, room assignments, housekeeping, billing and payments, inventory management, and other tasks related to running a hotel.

The system typically includes a front-end application for hotel staff to access the system and perform tasks, as well as a back-end database to store data and generate reports. The system may also provide interfaces to external systems such as online booking platforms and payment gateways.

A hotel management system can help improve efficiency, reduce errors, and provide a better guest experience by streamlining operations and providing accurate and timely information to staff. The system can also help hotel administrators make data-driven decisions and improve overall performance.

2. Introduction

2.1 Brief Introduction

This efficient and user-friendly hotel management system is admin side system.

Developed with the powerful combination of React and asp .net core web API, this feature-rich application provides a seamless experience, start using the system without any hassle.

2.2 Technology and Tools used

Technology

- Asp.net web API
- React.js
- C#
- HTML
- CSS
- Bootstrap

Tools

- Git
- Github
- Visual Studio
- Visual Studio Code
- Swagger

3. Software Requirements Specifications

3.1 Scope:

- The scope of this project is to manage hotels , rooms in particular hotel , users info and payment details.

3.2 Functional Requirements:

1. Manage hotels:

1.1 Add hotel

- ❖ Description: Admin can add hotel.
- ❖ Input: Enter hotel id , hotel name and location.
- ❖ Output: hotel will be added to available hotel.

1.2 View hotel

- ❖ Description: Admin can view available hotel.
- ❖ Input: view available hotel
- ❖ Output: Admin redirected to hotel page.

1.3 Update hotel

- ❖ Description: Admin can update hotel.
- ❖ Input: Update hotel name , location.
- ❖ Output: Admin redirected to hotel page and details will be updated.

1.4 Delete hotel

- ❖ Description: Admin can Delete hotel.
- ❖ Input: delete hotel
- ❖ Output: Admin redirected to hotel page and particular hotel will be deleted.

2. Manage rooms:

2.1 Add rooms

- ❖ Description: Admin can add rooms for available hotel.
- ❖ Input: Enter hotel id , room id.
- ❖ Output: room will be added to available hotel.

2.2 View rooms

- ❖ Description: Admin can view available rooms.
- ❖ Input: view available rooms.
- ❖ Output: Admin redirected to rooms page.

2.3 Delete rooms

- ❖ Description: Admin can Delete rooms.
- ❖ Input: Delete room.
- ❖ Output: Admin redirected to rooms page and particular room will be deleted.

3. Manage User info

3.1 Add user

- ❖ Description: Admin can add users for available hotel.
- ❖ Input: Enter hotel id , room id , user name , address.
- ❖ Output: user reserved room successfully.

3.2 View user

- ❖ Description: Admin can view user info.
- ❖ Input: view reserved users.
- ❖ Output: Admin redirected to userinfo page.

3.3 clear user

- ❖ Description: Admin can clear user.
- ❖ Input: drop user.
- ❖ Output: User redirected to userinfo page and particular user will be deleted.

3.3 Non-Functional Requirements:

N.1 : Performance

System should not take more time to load the page or components. Also, the system should be an interactive and should not take long delays.

N.2 : Safety

System should not display/leak any sensitive information of the user and hide from the rest of the users for safety.

N.3 : Reliability

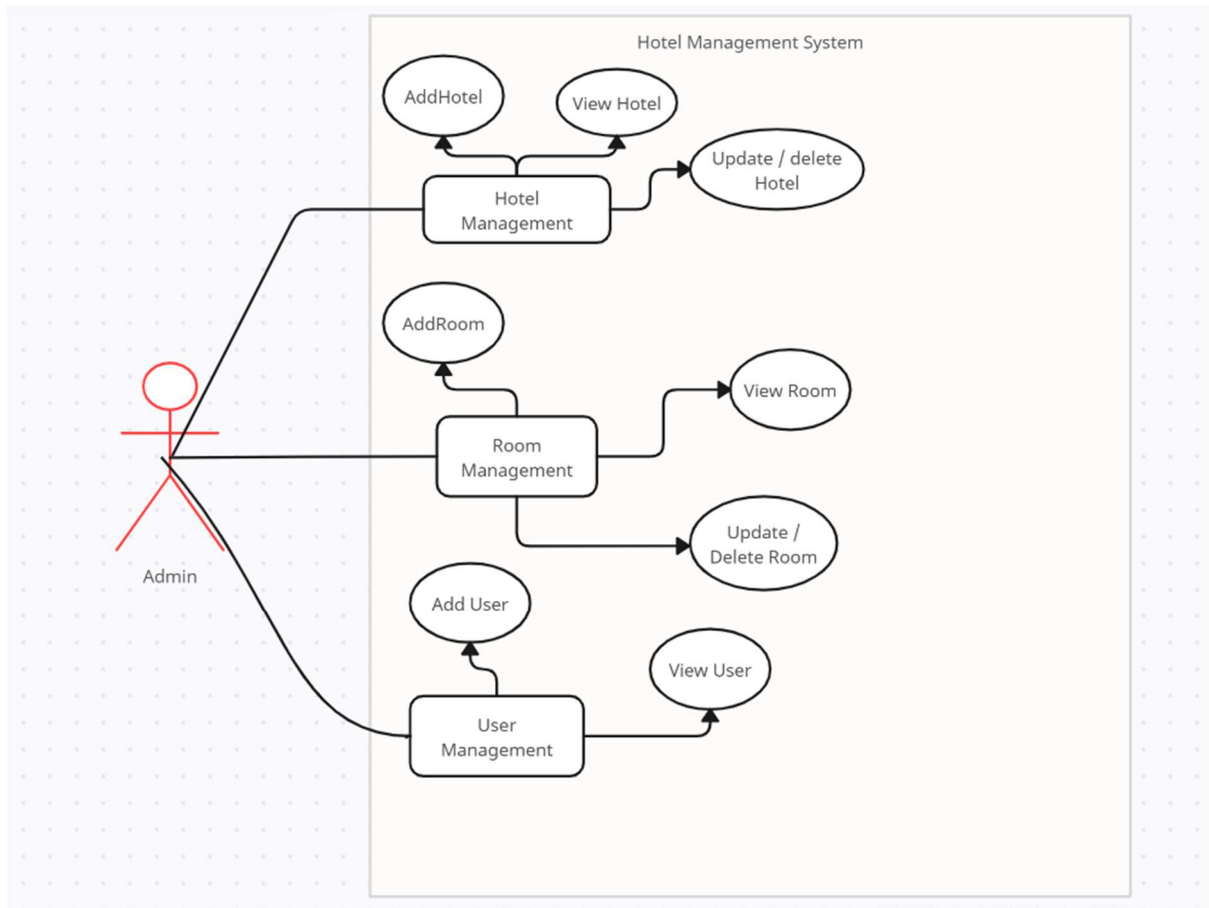
system should handles all unusual cases and activities which a normal authenticated user cannot do, hence our system doesn't crash on any boundary case is much more reliable.

N.4 : Database

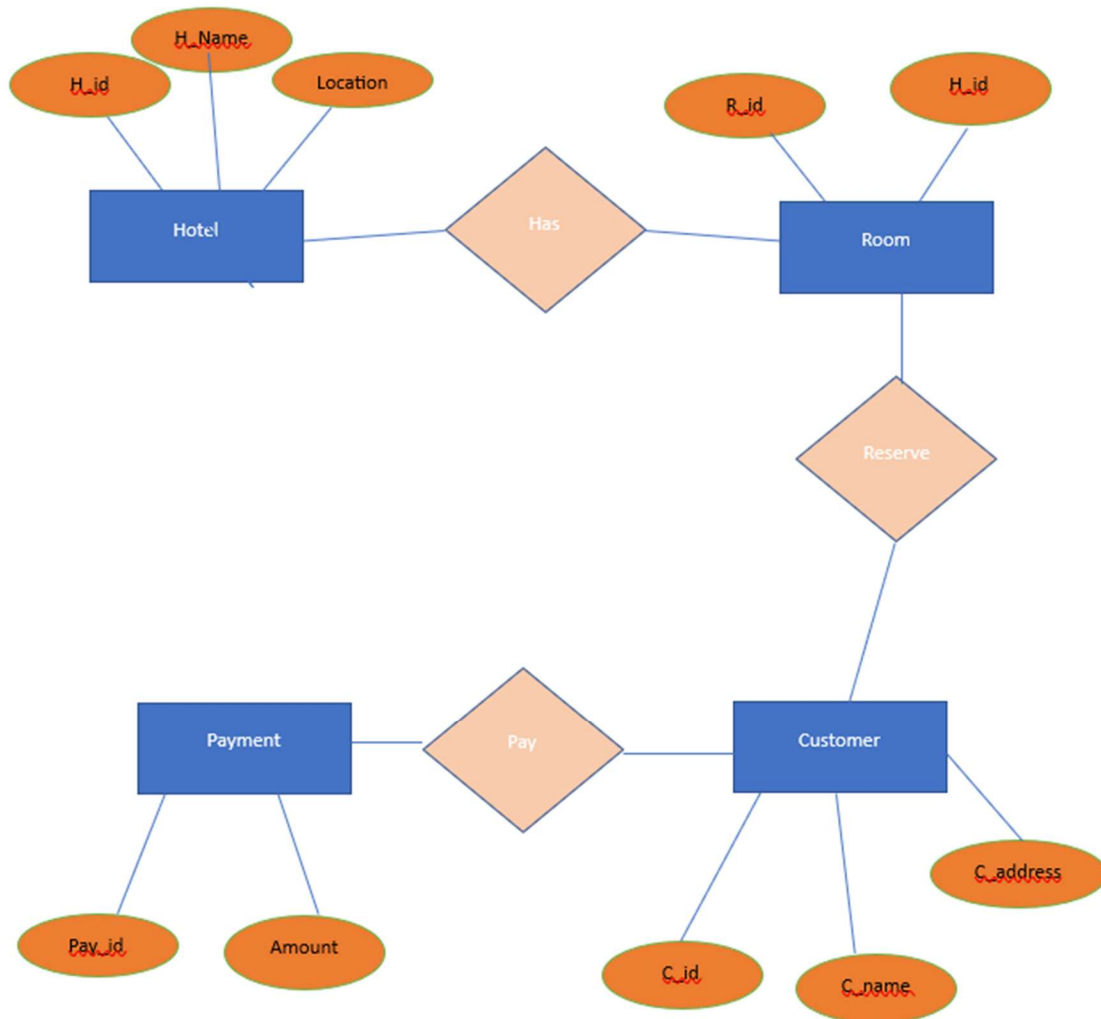
Any kind of activity related to the database follows Atomicity, meaning either action takes place, or no action will be taken. Hence, our database will always be in a consistent state.

4. Design

4.1 Use Case diagram



4.2 ER diagram



4.3Data Dictionary:

Entity - hotels

| Column | Data type | Constraint |
|---------------|------------------|------------------------------|
| hotelId | int | PK, Not null, Auto increment |
| h_Name | string | Not null |
| location | string | Not null |

Entity - rooms

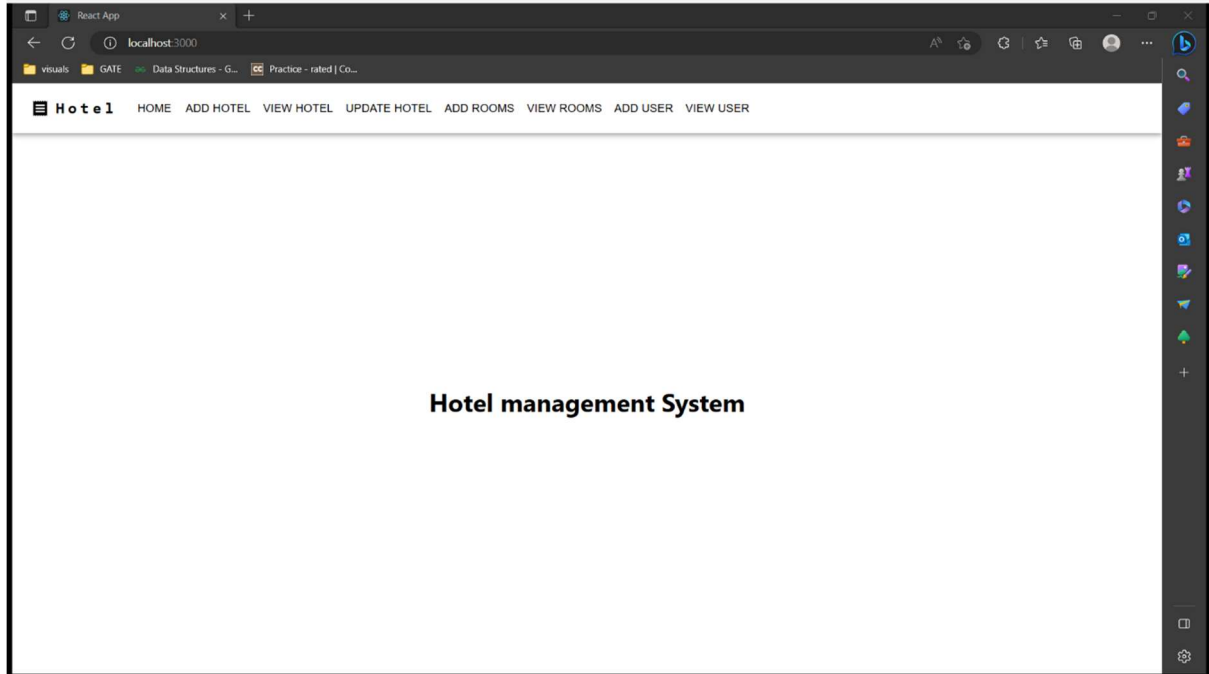
| Column | Data type | Constraint |
|---------------|------------------|------------------------------|
| roomsId | int | PK, Not null, Auto increment |
| hotelId | int | Not null |

Entity – user information

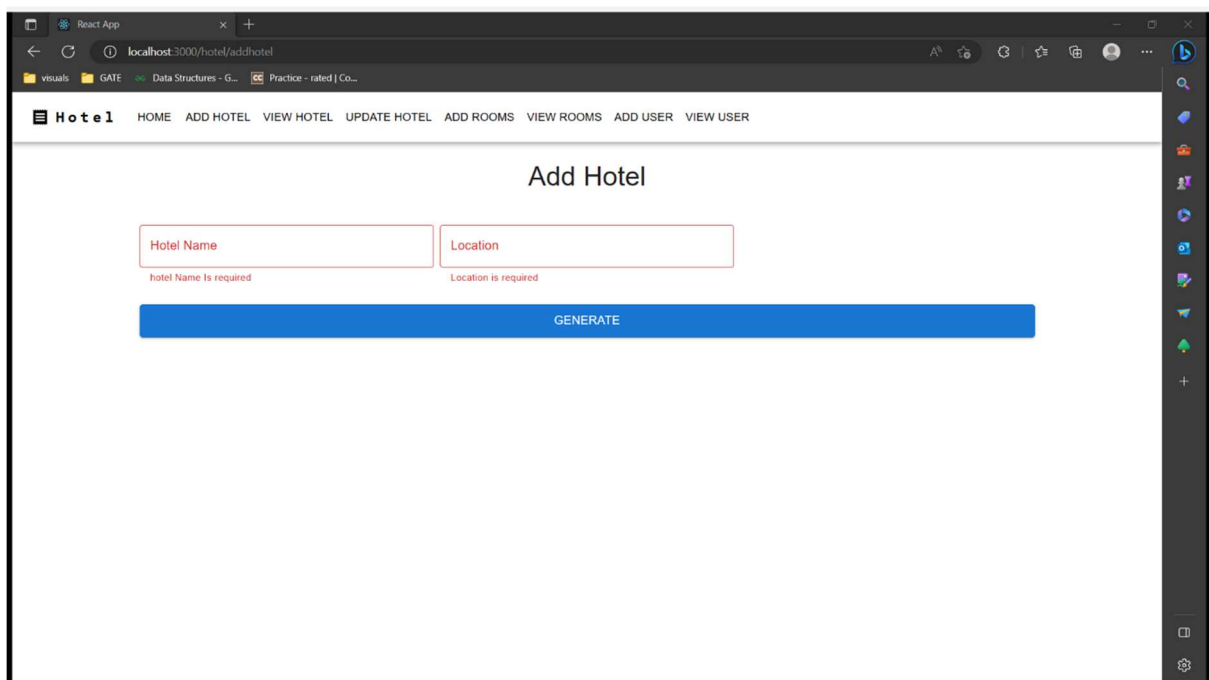
| Column | Data type | Description |
|---------------|------------------|------------------------------|
| userinfoId | int | PK, Not null, Auto increment |
| u_Name | int | Not null |
| address | date | Not null |
| hotelId | int | Not null |
| roomsId | int | Not null |

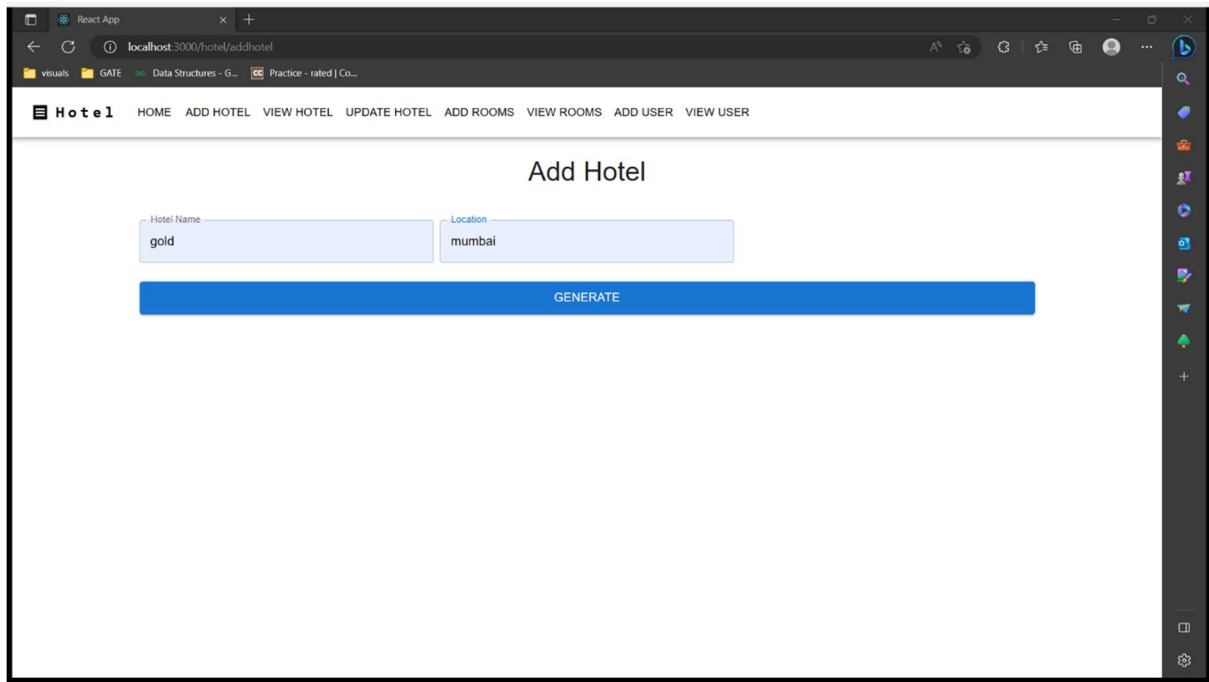
5 Screenshots

Home Page:



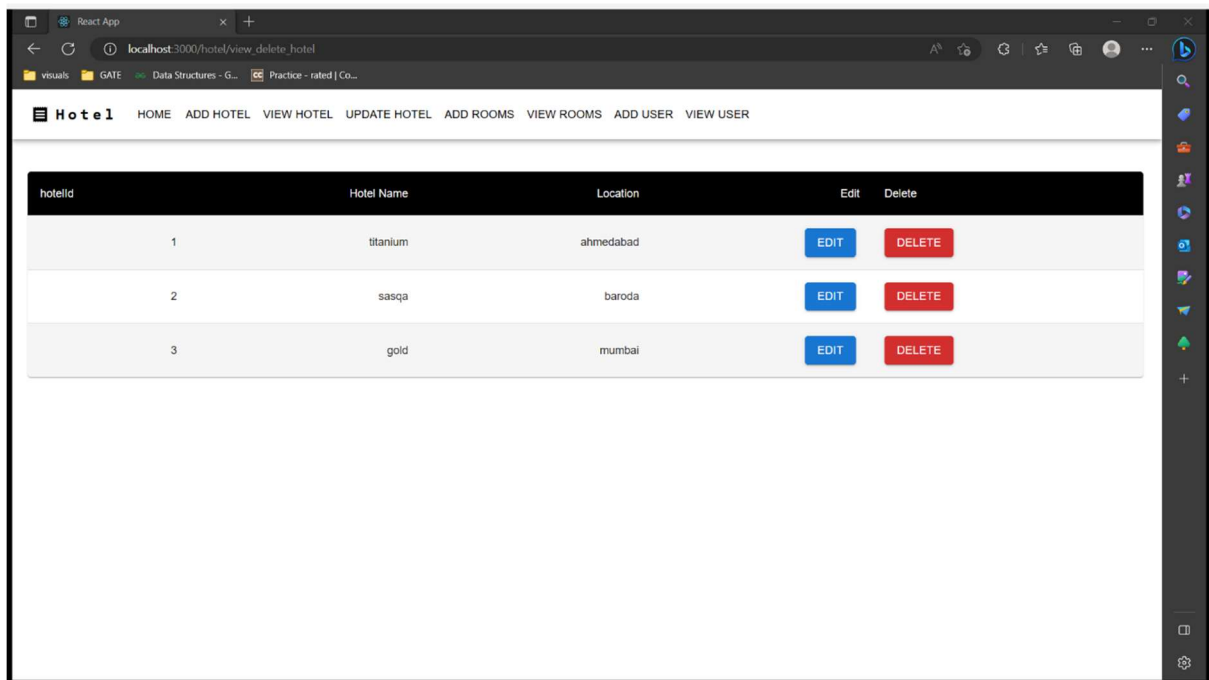
Add hotel:





Hotel added...

View available hotel:



Update Hotel Details:

Swagger UI | React App

localhost:3000/hotel/updatehotel/1

visuals | GATE | Data Structures - G... | Practice - rated | Co...

hotelId : 1
Edit Details

| | | |
|----------|------------|----------|
| Hotel Id | Hotel Name | Location |
| 1 | silver | mumbai |

SAVE CANCEL

Swagger UI | React App

localhost:3000/hotel/updatehotel/1

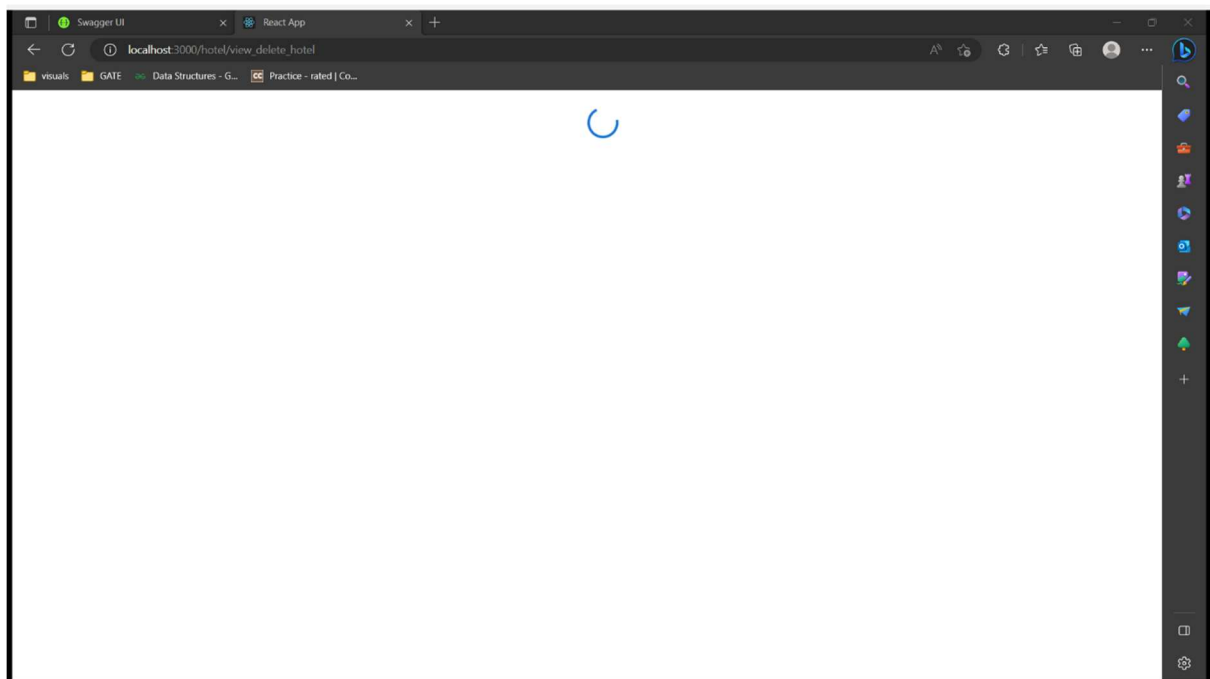
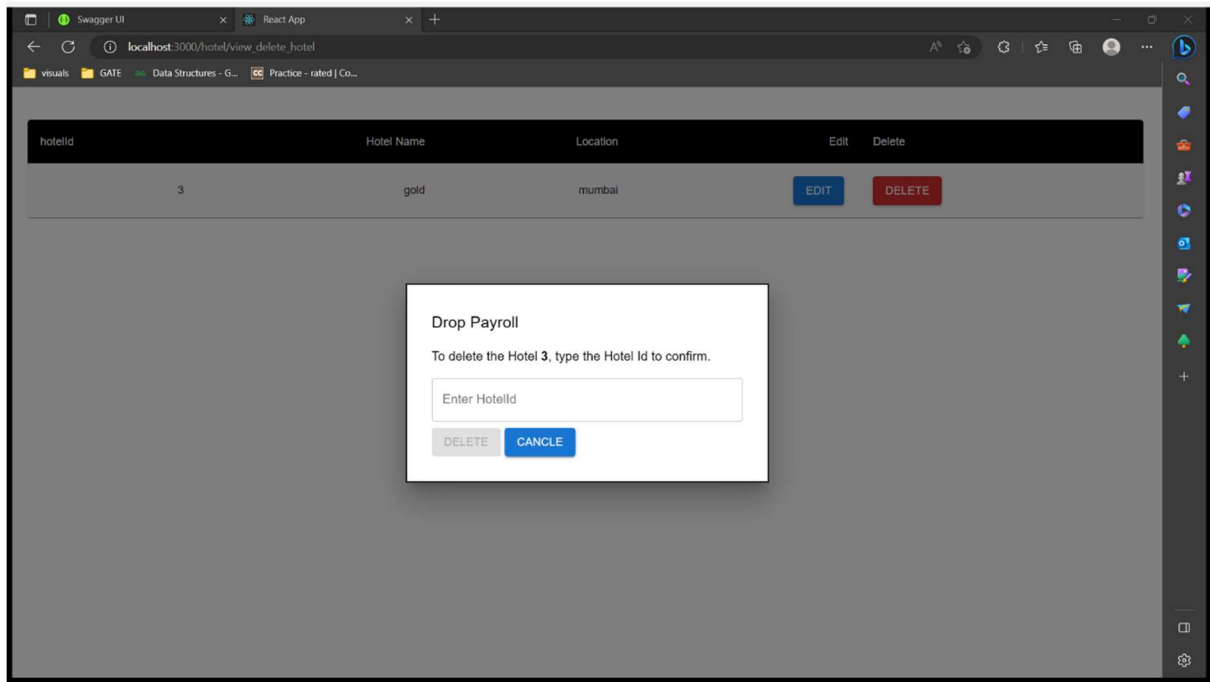
visuals | GATE | Data Structures - G... | Practice - rated | Co...

hotelId : 1
Edit Details

| | | |
|----------|------------|----------|
| Hotel Id | Hotel Name | Location |
| 1 | gold | mumbai |

SAVE CANCEL

Delete hotel from available hotel list:



Add Rooms:

React App

localhost:3000/hotel/addroom

visuals GATE Data Structures - G... Practice - rated | Co...

Hotel 1 HOME ADD HOTEL VIEW HOTEL UPDATE HOTEL ADD ROOMS VIEW ROOMS ADD USER VIEW USER

Add Room

Hotel Id

hotel id is required

GENERATE

React App

localhost:3000/hotel/addroom

visuals GATE Data Structures - G... Practice - rated | Co...

Hotel 1 HOME ADD HOTEL VIEW HOTEL UPDATE HOTEL ADD ROOMS VIEW ROOMS ADD USER VIEW USER

Add Room

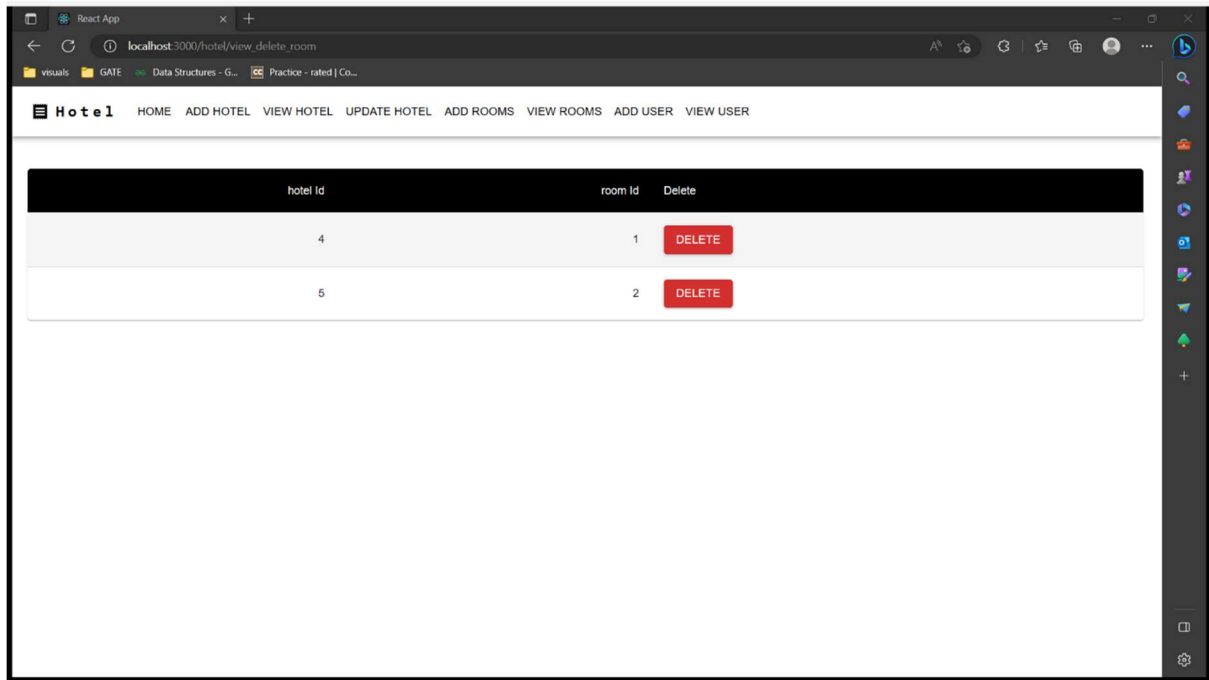
Hotel Id

5

GENERATE

Room is added to hotel id 5...

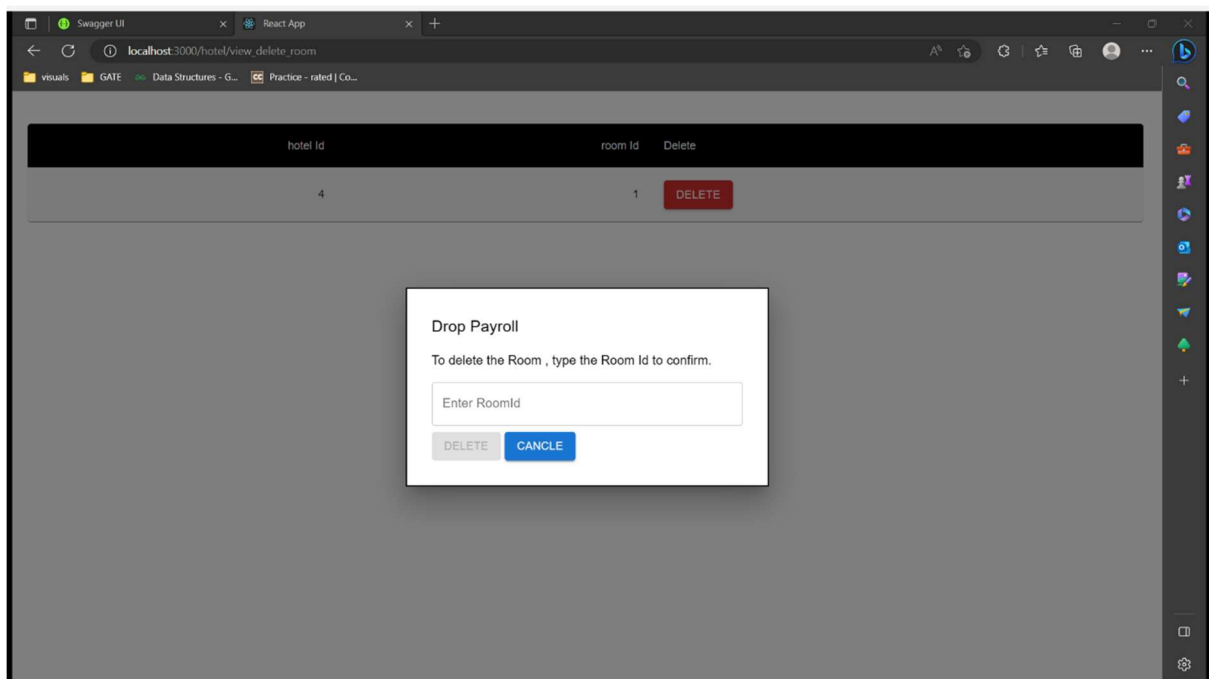
View Rooms:



The screenshot shows a web application interface for a hotel management system. The browser address bar indicates the URL is localhost:3000/hotel/view_delete_room. The application has a navigation bar with links: HOME, ADD HOTEL, VIEW HOTEL, UPDATE HOTEL, ADD ROOMS, VIEW ROOMS, ADD USER, and VIEW USER. Below the navigation bar, there is a table with the following data:

| hotel id | room id | Delete |
|----------|---------|--------|
| 4 | 1 | DELETE |
| 5 | 2 | DELETE |

Delete Room:



The screenshot shows the same web application interface as before, but with a confirmation dialog box displayed in the center. The dialog box has the title "Drop Payroll" and the text "To delete the Room , type the Room id to confirm." Below the text is an input field labeled "Enter Roomid". At the bottom of the dialog are two buttons: "DELETE" and "CANCEL".

Add new User for reservation:

React App

localhost:3000/hotel/adduser

visuals GATE Data Structures - G... Practice - rated | Co...

Hotel1 HOME ADD HOTEL VIEW HOTEL UPDATE HOTEL ADD ROOMS VIEW ROOMS ADD USER VIEW USER

Add User Information

User Name address hotelid

User Name is required Address is required Hotel Id is required

roomsid

Room Id is required

GENERATE

React App

localhost:3000/hotel/adduser

visuals GATE Data Structures - G... Practice - rated | Co...

Hotel1 HOME ADD HOTEL VIEW HOTEL UPDATE HOTEL ADD ROOMS VIEW ROOMS ADD USER VIEW USER

Add User Information

User Name address hotelid

Priyanshi Vrunda girls hostel, college road 1

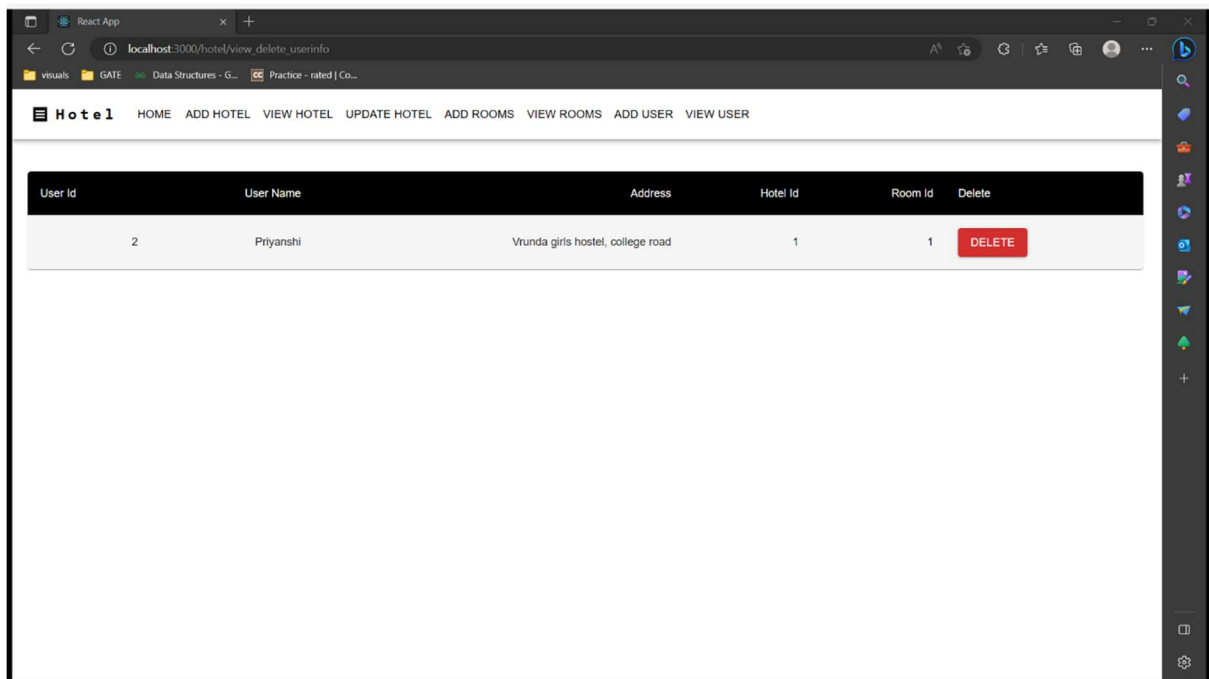
roomsid

1

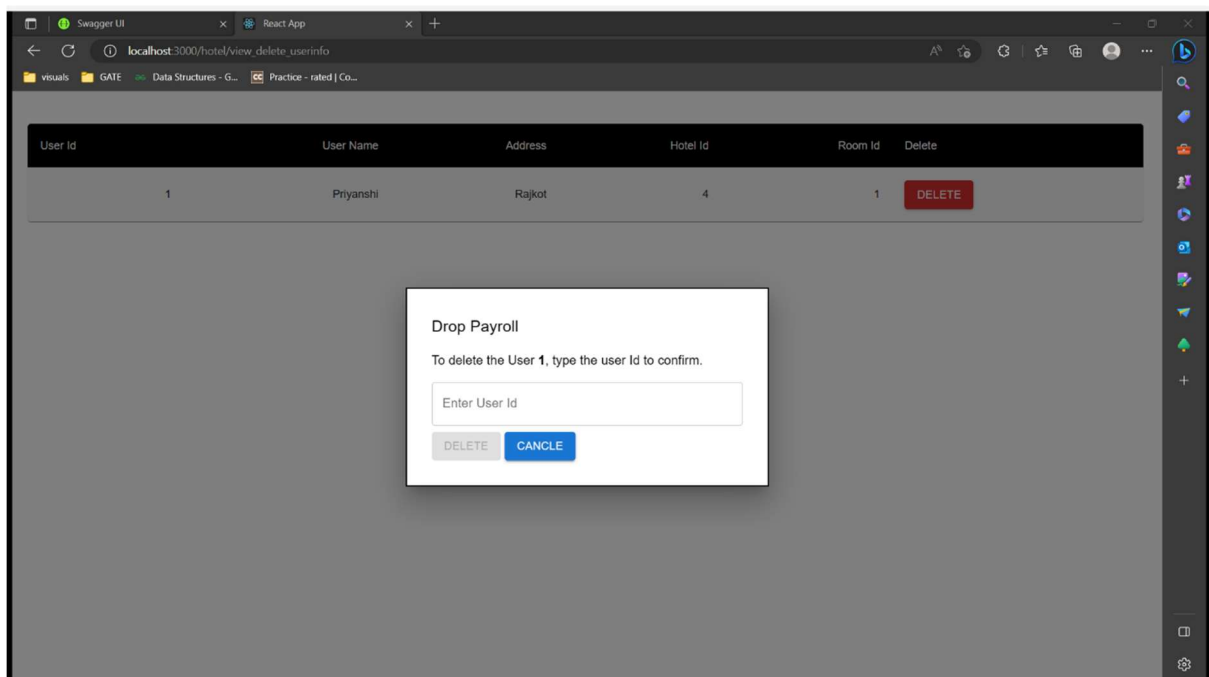
GENERATE

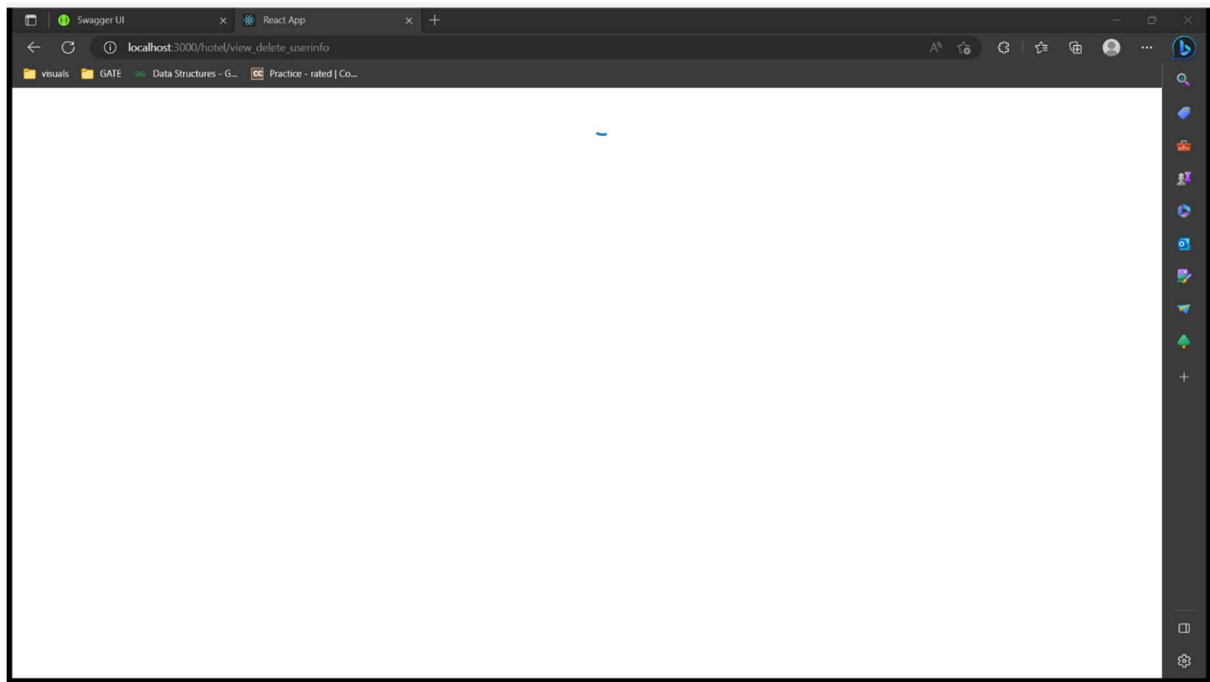
Reservation is successful...

View reserved user information:



Drop User from reserved user list:





6 Conclusion

Hotel Management System is completely free, easy to use and good user interface.

Hotel management website which is created through ASP.NET Core Web API Technology.

The functionalities which are implemented in this project is:

- CRUD operation on Hotel
- CRUD operation on Rooms for particular Hotel
- CRUD operation on User information

7. Limitation And Future Extension

Limitation:

- Current project have no function to make payment using payment gateway.
- Current project have no function to update rooms id for any hotel.
- Current website is not responsive.

Future Extension:

- Functionality for payment gateway will be added.
- making website responsive.
- Functionality for update rooms id will be added.

8 Bibliography

- <https://stackoverflow.com/>