CHAPTER 1

INTRODUCTION

1.1 Aim

The aim of the Hotel Management DBMS Mini Project is to design and implement a user-friendly and efficient database system to streamline hotel operations, enhance guest experience, and facilitate effective management.

1.2 Problem statement

The Hotel Management DBMS Mini Project aims to address the inefficiencies and complexities inherent in manual hotel management systems by developing a comprehensive database solution. The project seeks to overcome challenges such as manual data entry errors, difficulty in managing reservations and room allocations, inefficient billing processes, and lack of centralized data management.

1.3 Objective of the project

1. **Efficient Reservation Handling:** Enable users to quickly book rooms, modify reservations, and view availability, ensuring seamless guest experiences and optimal room occupancy.
2. **Streamlined Guest Management:** Facilitate easy registration, check-in, and check-out processes, along with maintaining guest profiles and preferences for personalized services.
3. **Effective Resource Management:** Manage inventory, staff schedules, and room allocations efficiently to optimize resource utilization and minimize operational costs.

1.4 Proposed solutions

The proposed solution for the Hotel Management DBMS Mini Project includes designing and implementing a relational database system with tables for guests, rooms, reservations, staff, and billing. The system will automate reservation processes, facilitate efficient room allocation.

CHAPTER 2

SYSTEM DESIGN

2.1 Schema Diagram

A schema diagram for a hotel management system illustrates the structure of its database, depicting entities like Guest, Room, Reservation, and their attributes. Relationships between entities are shown, along with primary and foreign keys for data integrity. It provides a concise overview of the system's data organization.

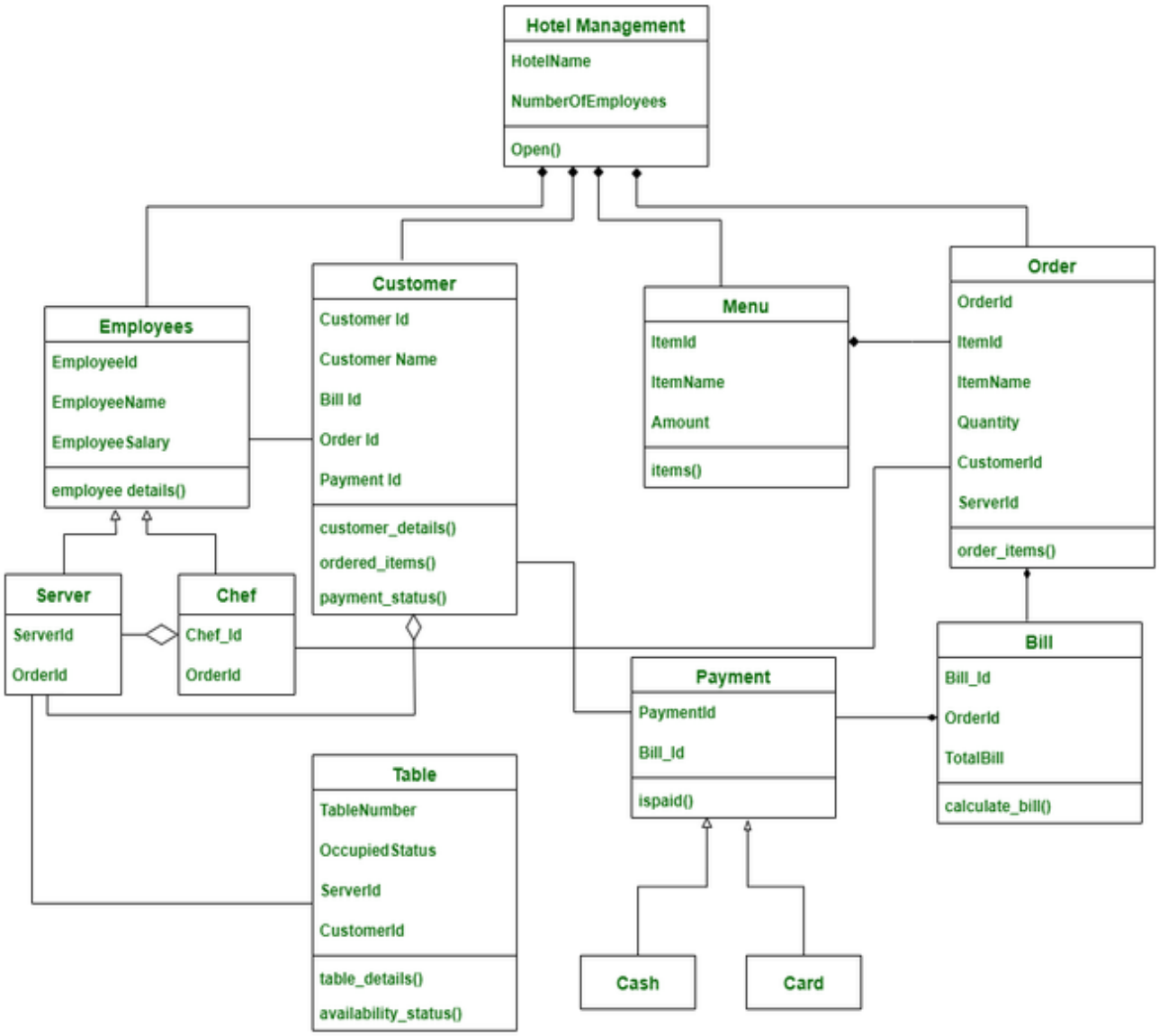

Fig 2.1 Hotel management Scheme diagram

2.2 E-R Diagram

An Entity-Relationship (E-R) diagram for a hotel management system visually represents entities like Guest, Room, Reservation, and their relationships. It illustrates how data is organized within the system, aiding in understanding the structure and flow of information.

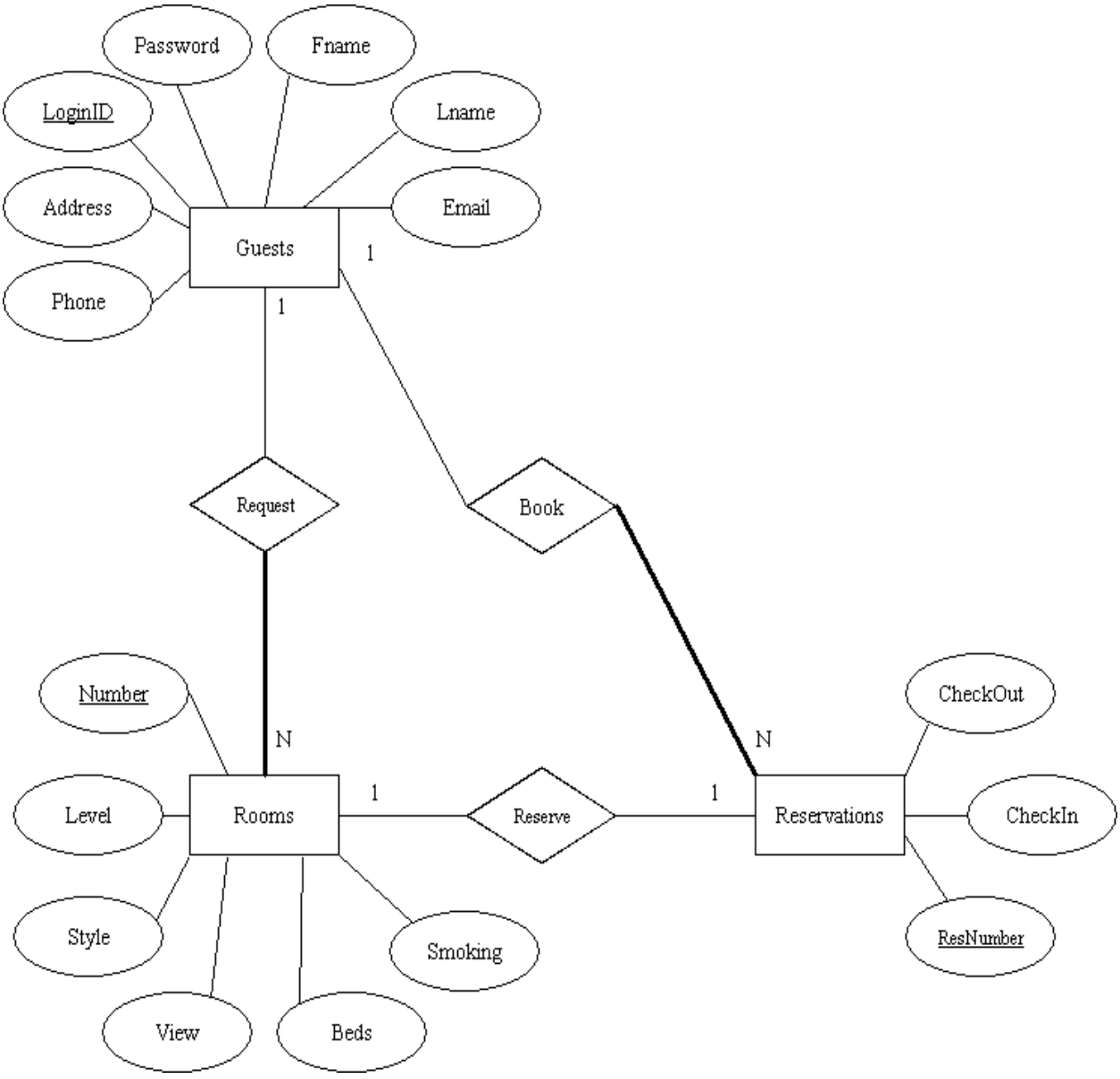

Fig 2.2 E-R Diagram for Hotel Management System

CHAPTER 3

IMPLEMENTATION

3.1 Languages used for implementation

- **HTML:** HTML to create a structure of the user interface.
- **CSS:** CSS to design the layout, to make it visually appealing
- **PHP:** PHP is a popular programming language that is commonly used to create web applications. It can be used to develop an attendance system that is accessible via a web browser. And also, to handle back-end tasks such as database interactions and user authentication.
- **JavaScript:** JavaScript is a programming language primarily used to create interactive effects within web browsers. It is used to implement an application that runs in a web browser or as a mobile app.
- **SQL :** For hotel database management system.

3.2 Platform used for implementation

For the implementation of the Hotel Management DBMS Mini Project, a robust and versatile platform such as MySQL will be utilized. MySQL offers a powerful relational database management system that is widely used for its reliability, scalability, and performance. With its support for SQL queries, transactions, and data integrity constraints, MySQL provides an ideal platform for designing and implementing the complex data structures required for hotel management systems. Additionally, MySQL's compatibility with various programming languages and frameworks allows for seamless integration with the project's user interface, enabling efficient interaction between the database and the frontend application.

3.3 SQL Commands and Queries

- **Creation of Table structure for table "employee login"**

```
CREATE TABLE `emp_login` (
  `empid` int(100) NOT NULL,
  `Emp_Email` varchar(50) NOT NULL,
  `Emp_Password` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- **Insertion of values into "Employee Login"**

```
INSERT INTO `emp_login` (`empid`, `Emp_Email`, `Emp_Password`) VALUES
(1, 'Admin@gmail.com', '1234');
```

- **Creation of Table structure for table "payment"**

```
CREATE TABLE `payment` (`id` int(30) NOT NULL,`Name` varchar(30) NOT NULL,`Email` varchar(30) NOT NULL,`RoomType` varchar(30) NOT NULL,`Bed` varchar(30) NOT NULL,`NoofRoom`int(30) NOT NULL,`cin` date NOT NULL,`cout` date NOT NULL,`noofdays` int(30) NOT NULL,`roomtotal` double(8,2) NOT NULL,`bedtotal` double(8,2) NOT NULL,`meal` varchar(30) NOT NULL, `mealtotal` double(8,2) NOT NULL,`finaltotal` double(8,2) NOT NULL) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- **Insertion of values into "Payment"**

```
INSERT INTO `payment` (`id`, `Name`, `Email`, `RoomType`, `Bed`,
`NoofRoom`, `cin`, `cout`, `noofdays`, `roomtotal`, `bedtotal`, `meal`,
`mealtotal`,`finaltotal`)VALUES(41, 'Tushar pankhaniya',
'pankhaniyatushar9@gmail.com', 'Single Room', 'Single', 1, '2022-11-09',
'2022-11-10', 1, 1000.00, 10.00, 'Room only', 0.00, 1010.00);
```

- **Creation of Table structure for table "room"**

```
CREATE TABLE `room` (`id` int(30) NOT NULL,`type`
varchar(50) NOT NULL,`bedding` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- **Insertion of data for table "room"**

```
INSERT INTO `room` (`id`, `type`, `bedding`) VALUES
(4, 'Superior Room', 'Single'),
(6, 'Superior Room', 'Triple'),
(7, 'Superior Room', 'Quad'),
(8, 'Deluxe Room', 'Single'),
(9, 'Deluxe Room', 'Double'),
(10, 'Deluxe Room', 'Triple'),
(11, 'Guest House', 'Single'),
(12, 'Guest House', 'Double'),
(13, 'Guest House', 'Triple'),
(14, 'Guest House', 'Quad'),
(16, 'Superior Room', 'Double'),
```

```
(20, 'Single Room', 'Single'),
(22, 'Superior Room', 'Single'),
(23, 'Deluxe Room', 'Single'),
(24, 'Deluxe Room', 'Triple'),
(27, 'Guest House', 'Double'),
(30, 'Deluxe Room', 'Single');
```

- **Creation of Table structure for table "room_book"**

```
CREATE TABLE `roombook` (
  `id` int(10) NOT NULL,
  `Name` varchar(50) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Country` varchar(30) NOT NULL,
  `Phone` varchar(30) NOT NULL,
  `RoomType` varchar(30) NOT NULL,
  `Bed` varchar(30) NOT NULL,
  `Meal` varchar(30) NOT NULL,
  `NoofRoom` varchar(30) NOT NULL,
  `cin` date NOT NULL,
  `cout` date NOT NULL,
  `nodays` int(50) NOT NULL,
  `stat` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- **Insertion values into table "roombook"**

```
INSERT INTO `roombook` (`id`, `Name`, `Email`, `Country`, `Phone`, `RoomType`, `Bed`, `Meal`, `NoofRoom`, `cin`, `cout`, `nodays`, `stat`) VALUES
(41, 'Tushar pankhaniya', 'pankhaniyatushar9@gmail.com', 'India', '9313346569', 'Single Room', 'Single', 'Room only', '1', '2022-11-09', '2022-11-10', 1, 'Confirm');
```

- **Creation of Table structure for table `signup`**

```
CREATE TABLE `signup` (
  `UserID` int(100) NOT NULL,
  `Username` varchar(50) NOT NULL,
  `Email` varchar(50) NOT NULL,
  `Password` varchar(50) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- **Inserting data for table `signup`**

```
INSERT INTO `signup` (`UserID`, `Username`, `Email`, `Password`) VALUES
(1, 'Shreyas Ugra', 'shreyasugra38@gmail.com', '123');
```

- **Creation of Table structure for table `staff`**

```
CREATE TABLE `staff` (
  `id` int(30) NOT NULL,
  `name` varchar(30) NOT NULL,
  `work` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- **Insertion of data for table `staff`**

```
INSERT INTO `staff` (`id`, `name`, `work`) VALUES(1, 'Tushar pankhaniya', 'Manager'),
(3, 'rohit patel', 'Cook'),(4, 'Dipak', 'Cook'),(5, 'tirth', 'Helper'),(6, 'mohan', 'Helper'),
(7, 'shyam', 'cleaner'),(8, 'rohan', 'weighter'),(9, 'hiren', 'weighter'),(10, 'nikunj', 'weighter'),
(11, 'rekha', 'Cook');
```

- **Adding primary key' s for the above created tables**

```
--Indexes for table `emp_login`

ALTER TABLE `emp_login`
  ADD PRIMARY KEY (`empid`);

-- Indexes for table `payment`

ALTER TABLE `payment`
  ADD PRIMARY KEY (`id`);

-- Indexes for table `room`

ALTER TABLE `room`
  ADD PRIMARY KEY (`id`);
```

```
-- Indexes for table `roombook`

ALTER TABLE `roombook`
  ADD PRIMARY KEY (`id`);

--Indexes for table `signup`

ALTER TABLE `signup`
  ADD PRIMARY KEY (`UserID`);

-- Indexes for table `staff`

ALTER TABLE `staff`
  ADD PRIMARY KEY (`id`);
```

- **Alteration of Created Table Attributes**

```
ALTER TABLE `emp_login`
  MODIFY `empid` int(100) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;


-- AUTO_INCREMENT for table `room`

ALTER TABLE `room`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=31;


-- AUTO_INCREMENT for table `roombook`

ALTER TABLE `roombook`
  MODIFY `id` int(10) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=51;


-- AUTO_INCREMENT for table `signup`

ALTER TABLE `signup`
  MODIFY `UserID` int(100) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;


-- AUTO_INCREMENT for table `staff`

ALTER TABLE `staff`
  MODIFY `id` int(30) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=12;
COMMIT;
```

- **Creation separate login for “user” and “admin”**

```
<?php

include '../config.php';
session_start();

// page redirect
$usermail="";
$usermail=$_SESSION['usermail'];
if($usermail == true){

}else{
  header("location: http://localhost/hotelmanage_system/index.php");
}

?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="./css/admin.css">
    <!-- loading bar -->
    <script src="https://cdn.jsdelivr.net/npm/pace-js@latest/pace.min.js"></script>
    <link rel="stylesheet" href="../css/flash.css">
    <!-- fontowesome -->
```

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.0/css/all.min.css" integrity="sha512-
xh6O/CkQoPOWDdYTDqeRdPCVd1SpvCA9XXcUnZS2FmJNp1coAFzvtCN9BmamE+4aH
K8yyUHUSCcJHgXloTyT2A==" crossorigin="anonymous" referrerpolicy="no-referrer"/>
<title>Real Madrid - Admin</title>
</head>

<body>
<!-- mobile view -->
<div id="mobileview">
<h5>Admin panel doesn't show in mobile view</h4>
</div>

<!-- nav bar -->
<nav class="uppernav">
<div class="logo">
<img class="bluebirdlogo" src="../image/bluebirdlogo.png" alt="logo">
<p>REAL MADRID</p>
</div>
<div class="logout">
<a href="../logout.php"><button class="btn btn-primary">Logout</button></a>
</div>
</nav>
<nav class="sidenav">
<ul>
<li class="pagebtn active"><img src="../image/icon/dashboard.png">   
Dashboard</li>
<li class="pagebtn"><img src="../image/icon/bed.png">    Room
Booking</li>
<li class="pagebtn"><img src="../image/icon/wallet.png">    Payment</li>
```

```
        <li class="pagebtn"><img src="../image/icon/bedroom.png">   
Rooms</li>
        <li class="pagebtn"><img src="../image/icon/staff.png">   
Staff</li>
      </ul>
   </nav>

   <!-- main section -->
   <div class="mainscreen">
      <iframe class="frames frame1 active" src="./dashboard.php" frameborder="0"></iframe>
      <iframe class="frames frame2" src="./roombook.php" frameborder="0"></iframe>
      <iframe class="frames frame3" src="./payment.php" frameborder="0"></iframe>
      <iframe class="frames frame4" src="./room.php" frameborder="0"></iframe>
      <iframe class="frames frame4" src="./staff.php" frameborder="0"></iframe>
   </div>
</body>

<script src="./javascript/script.js"></script>

</html>
```

3.4 SYSTEM OUTPUT

System testing for a hotel management system DBMS mini project involves validating functions like room booking, check-in/out, and reservation handling. It ensures data integrity, security measures, and seamless integration with external systems. Performance testing assesses system responsiveness and scalability. Regression testing verifies existing features after updates. User acceptance testing involves stakeholders to ensure the system meets business requirements before deployment.

TEST CASE - ID	TEST CASE NAME	TEST CASE DESCRIPTION	INPUT	EXPECTED RESULTS	RESULTS
01	Admin Login	Validate Admin ID and Password	Enter valid Admin ID	Admin Login should be successful	PASS
02	Admin Login	Validate Admin ID and Password	Enter Invalid Admin ID and Password	Invalid username/password Error message should display	FAIL
03	Admin Login	Validate Admin Id	If any fields are left blank	Fields are empty Error message should be display	FAIL
04	User Login	Validate User name	Enter valid signed up user name	User should be valid	PASS
05	User Login	Invalid username	Enter valid user name	User not present	FAIL

CHAPTER 4

SNAPSHOTS

4.1 Results

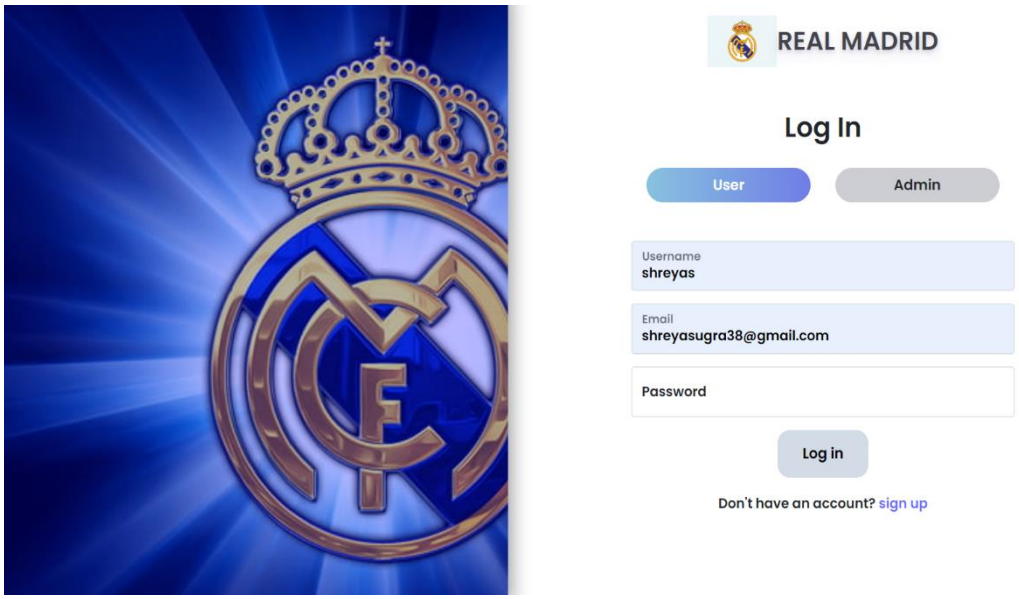

Fig 4.1 User login(Insertion Operation)

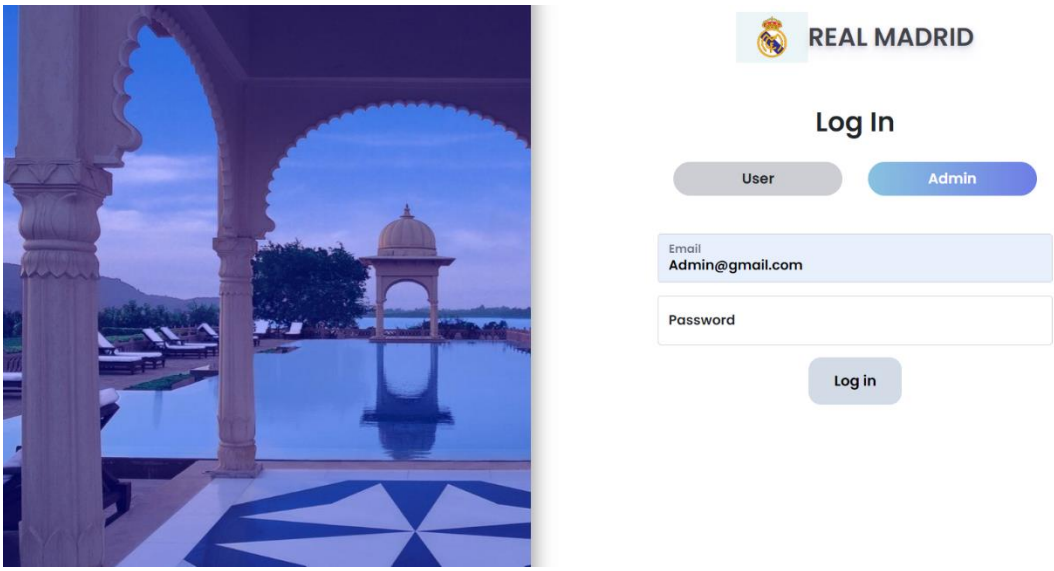

Fig 4.2 Admin login

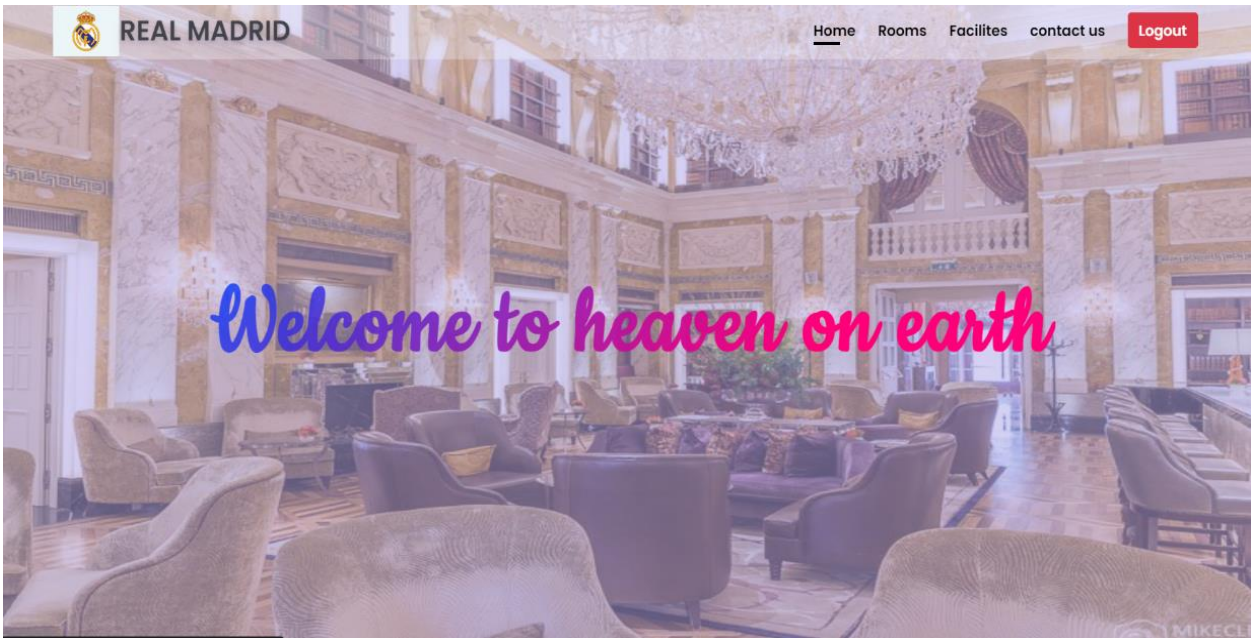

Fig 4.3 Home Window of Hotel Management System

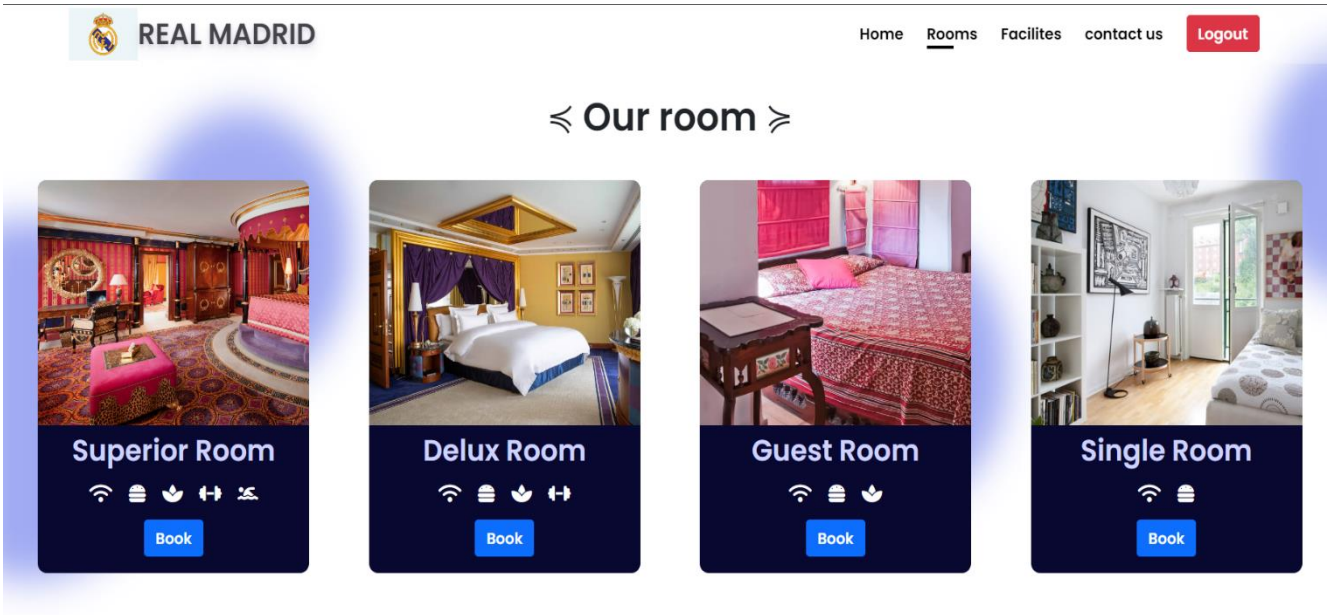

Fig 4.4 Various types of rooms window

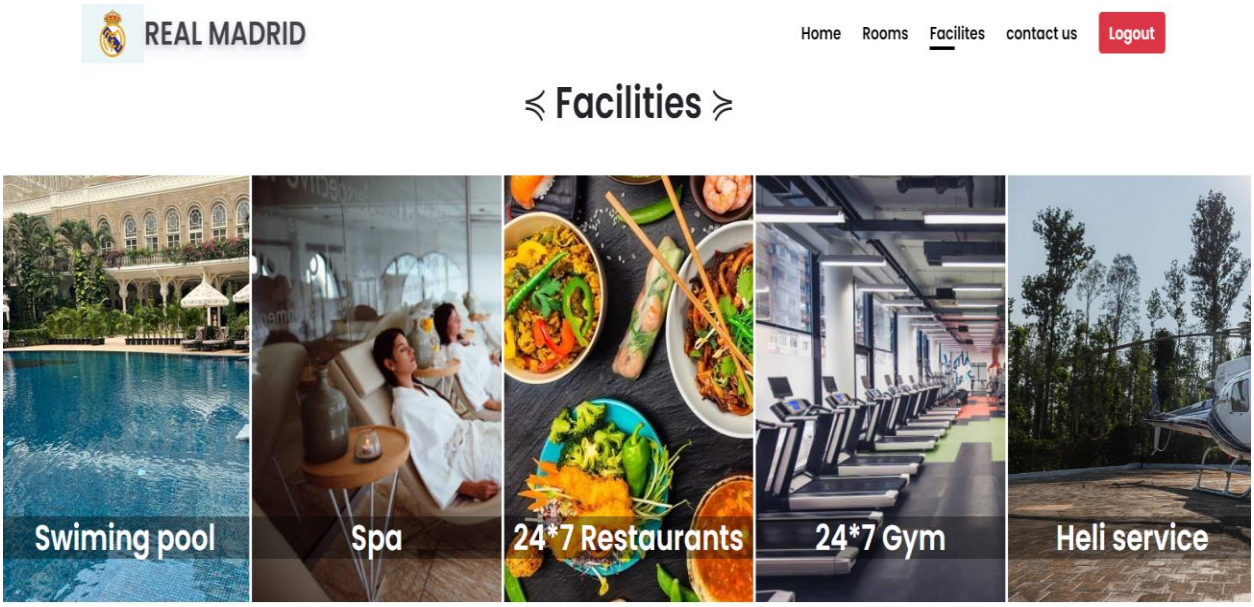

Fig 4.5 Facilities Window

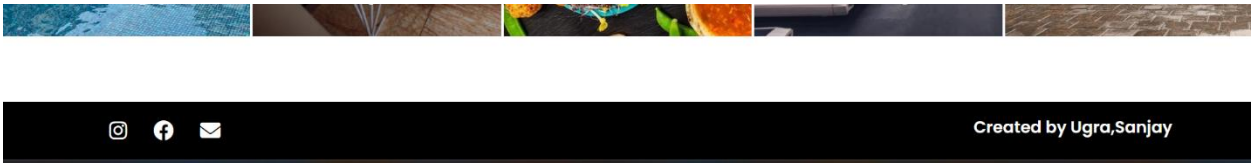

Fig 4.6 Contact Us Window

Fig 4.7 Logout Window

Fig 4.8 Room Reservation Window

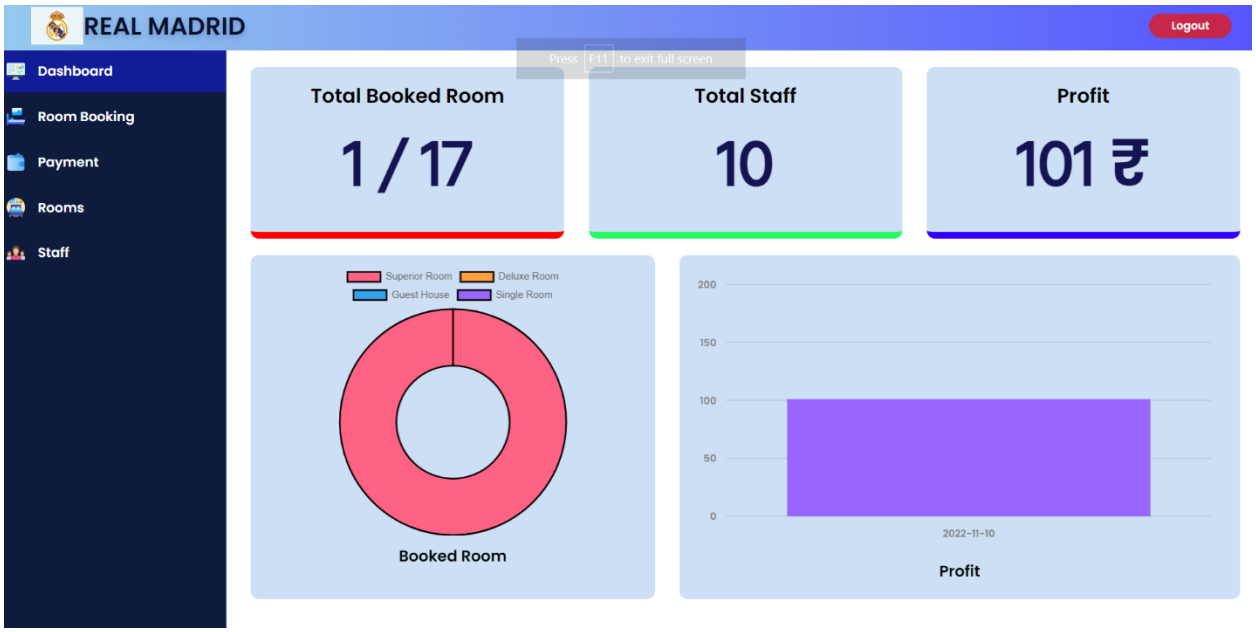

Fig 4.9 Admin Dashboard

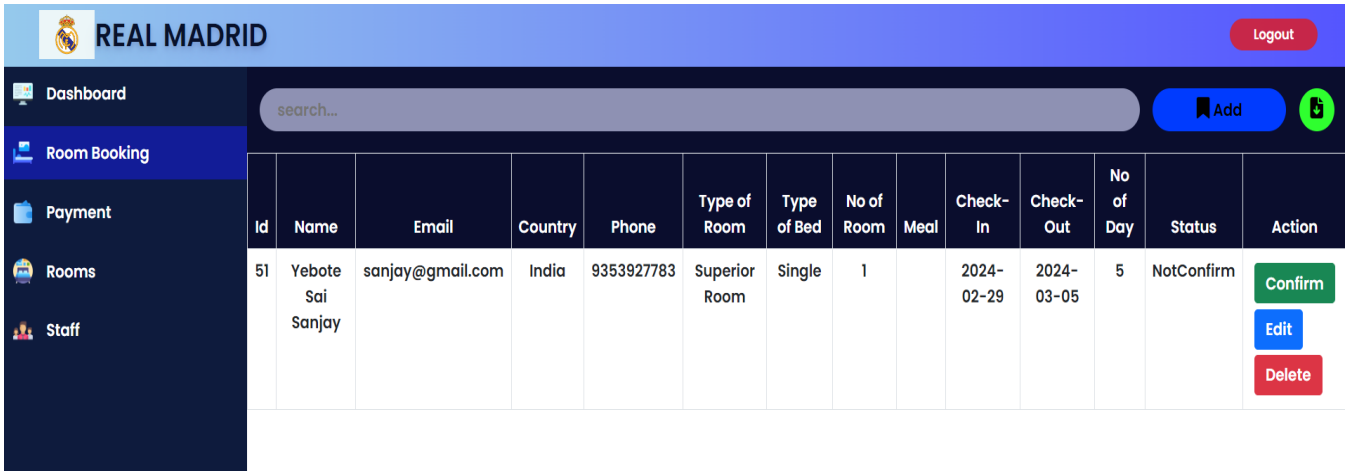

Fig 4.10 Room Booking Request by User(Alter & delete operation)

Fig 4.11 Payment status by user(update operation)

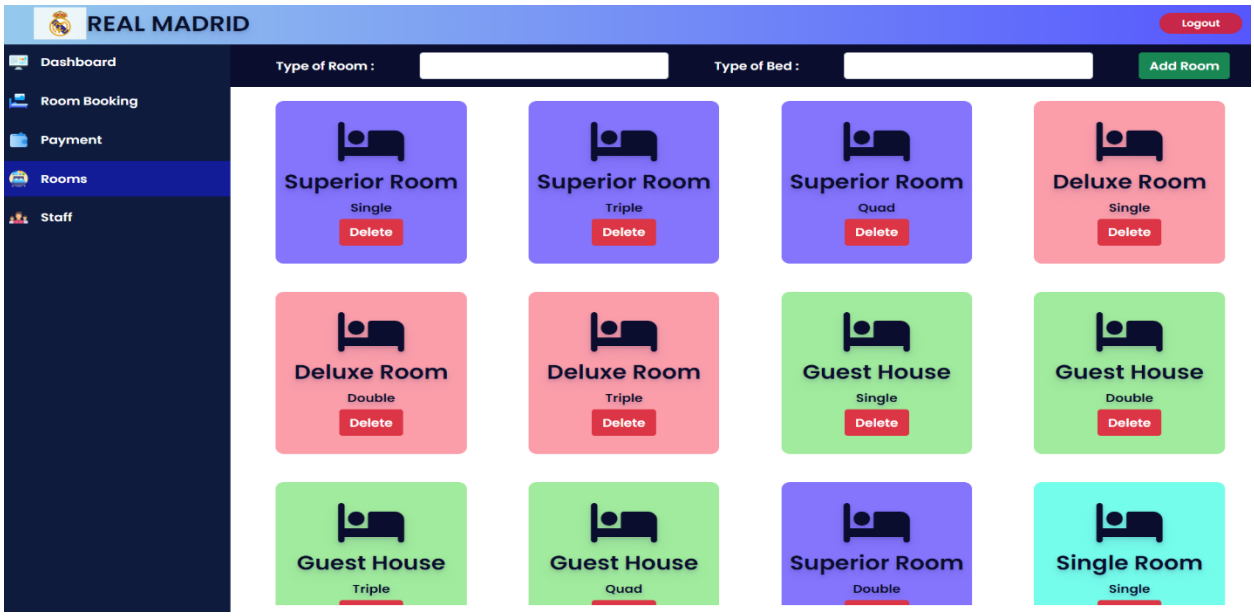

Fig 4.12 Available Rooms in the admin window

Fig 4.13 Number of Staff present in hotel(Insertion of new worker)

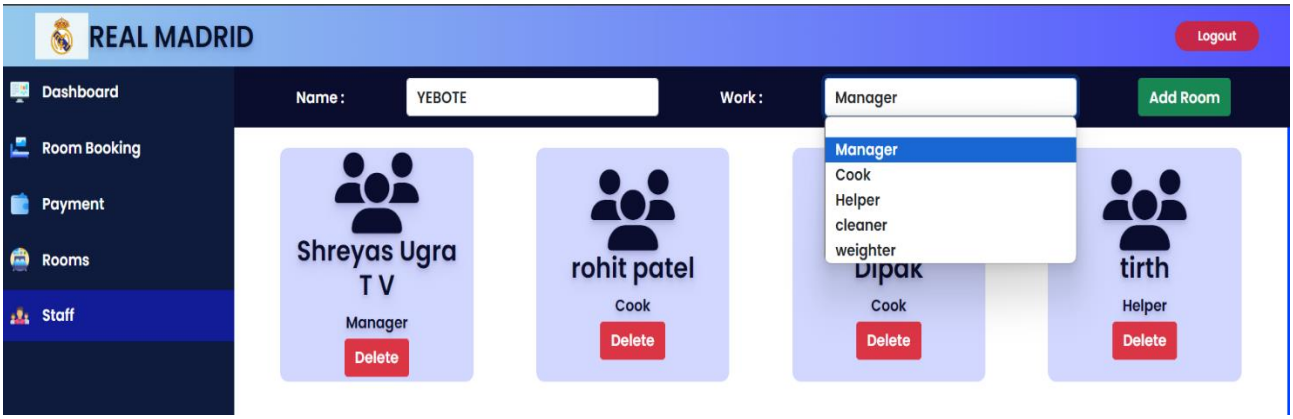

Fig 4.14 Addition of New staff into hotel

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

In conclusion, the implementation of a hotel management system within a database management system (DBMS) serves as an efficient solution for streamlining operations, enhancing customer satisfaction, and optimizing resource allocation within the hospitality industry. Through effective utilization of data storage, retrieval, and manipulation capabilities provided by the DBMS, the hotel can manage reservations, room allocations, guest information, billing, and staff scheduling with ease. This system facilitates seamless communication between different departments, enables real-time updates, and empowers decision-makers with comprehensive insights derived from data analysis, ultimately leading to improved operational efficiency and heightened guest experience

5.2 Future Enhancement

1. **Integration of IoT Devices:** Incorporating Internet of Things (IoT) devices such as smart thermostats, occupancy sensors, and energy management systems can enhance efficiency by automating tasks like room temperature control, energy usage optimization, and monitoring of room occupancy in real-time.

2. **Implementing Predictive Analytics:** Leveraging predictive analytics algorithms can help forecast room demand, pricing trends, and guest preferences, enabling the hotel to proactively adjust room rates, inventory levels, and service offerings to meet anticipated demands and maximize revenue.

3. **Enhanced Guest Experience with Mobile Apps:** Developing a dedicated mobile application for guests can provide features such as mobile check-in/check-out.

5.3 References

1. SlideShare--https://www.slideshare.net/KaranamManideep1/hotel-management-dbmsdocx

2. Scribd--https://www.scribd.com/doc/26299255/A-Project-Report-on-Hotel-Management-System

3. Studocu—https://www.studocu.com/in/document/acharya-nagarjuna-university/mca-ii-sem/hotel-management-system-project-documentation/41909511

4. Academic.edu--https://www.academia.edu/2112330/A_SAMPLE_HOTEL_MANAGEMENT_SYSTEM_PROJECT_DOCUMENTATION

5. Github-- https://github.com/topics/hotel-management-system