

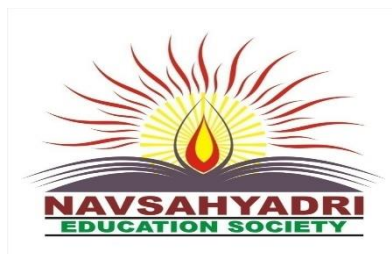
A
SEMINAR REPORT
ON
“Hotel Management System (HMS)”

SUBMITTED BY

TALEKAR YASHRAJ MADHOSHIVRAJ

Under the Guidance of

Prof. M.P.Pujari

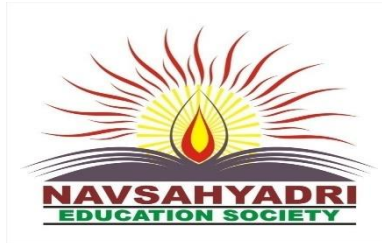


DEPARTMENT OF AIML ENGINEERING

NGI'S NAVSAHYADRI FACULTY OF ENGINEERING, NAIGAON, Savitribai Phule
Pune University

2024-25

**NGI'S NAVSAHYADRI FACULTY OF ENGINEERING,
NAIGAON, PUNE-412213**



CERTIFICATE

This is to certify that the Project report entitled

“Hotel Management System (HMS)”

Is the bonafide work carried out by

Talekar Yashraj Madhoshivraj

And is submitted in partial fulfilment for the award of degree of Bachelor of Engineering in AIML
Engineering of Savitribai Phule Pune University in the academic year 2024-25.

Staff In-Charge

H.O.D.AIML Dept

Principal

INDEX

Chapter	Title	Page no.
1.	Introduction	4
2.	Objectives	5
3.	System Features	6
4.	User Interface Design	7
5.	System Design & Architecture	8
6.	Detailed Code Explanation	9-10
7.	Testing	11
8.	User Experience and Interface Feedback	12
9.	Future Scope	13
10.	Output	14-16
11.	Conclusion	17
12.	References	18

Introduction

Overview

The Hotel Management System (HMS) implemented using HTML, CSS, and JavaScript provides a web-based platform for managing hotel operations. This system allows hotel administrators to manage rooms, check-ins, check-outs, and generate reports such as guest summary reports. This document describes the development of the HMS web application, outlining its architecture, features, and a detailed explanation of its code.

Objectives

The primary objectives of the HMS include:

1. **Room Management:** Administer rooms efficiently, handling room availability, rates, and customer information.
2. **Check-In and Check-Out Management:** Manage customer check-in and check-out seamlessly, ensuring up-to-date room status.
3. **Report Generation:** Provide summary reports for guests, ensuring efficient reporting of hotel occupancy and financials.
4. **Ease of Use:** Develop a user-friendly interface that allows staff to manage hotel operations effectively without deep technical knowledge.

System Features

The core functionalities of the system are as follows:

- **Room Management:** Add and manage room details (room number, AC, comfort level, size, daily rent, etc.).
- **Customer Check-In/Check-Out:** Simplified process to check-in guests by recording customer information, booking details, and payment advances.
- **Available Rooms Display:** View the available rooms for new bookings in real-time.
- **Customer Search:** Search for guests by name to view details of their stay.
- **Bill Calculation:** Calculate and display the total bill based on the room rent and the number of days of stay.
- **Guest Summary Report:** Generate a report on the currently occupied rooms and customer details.

User Interface Design

The User Interface (UI) is designed to be intuitive and easy to navigate. It consists of the following components:

1. **Dashboard with Buttons:**

Users can navigate to various sections such as "Manage Rooms", "Check-In Room", "Available Rooms", "Search Customer", and "Check-Out Room".

2. **Forms for Data Entry:**

Each function (e.g., adding rooms, checking in customers) has a form that collects the necessary data.

3. **Buttons for Interactions:**

Each section includes buttons that trigger JavaScript functions to perform tasks like adding rooms, checking in customers, searching, etc.

4. **Output Sections:**

Each section has a dedicated output area where the result of the action (e.g., the room added, customer details, total bill) is displayed.

System Design & Architecture

The system follows a **client-side** architecture since it is built using HTML, CSS, and JavaScript, with all operations handled in the browser. No backend or server-side technologies are used in this simple system, and data is stored temporarily during the session.

- **HTML:** Defines the structure of the web pages, such as forms for adding room data, checking in guests, and displaying reports.
- **CSS:** Handles the visual design, ensuring the interface is clean, modern, and responsive.
- **JavaScript:** Manages the logic for performing operations, such as storing room information, handling customer check-ins, and generating reports.

Detailed Code Explanation

HTML

The HTML code defines the structure of the system's interface. It contains several sections (divs) representing different parts of the application, such as managing rooms, checking in/out customers, and generating reports.

Key components of the HTML include:

- **Menu Section:** A list of buttons for navigating between different functions.

```
html
Copy code
<div class="menu">
  <button onclick="showSection('manageRooms') ">Manage Rooms</button>
  <button onclick="showSection('checkInRoom') ">Check-In Room</button>
  <button onclick="showSection('availableRooms') ">Available
Rooms</button>
  <button onclick="showSection('searchCustomer') ">Search
Customer</button>
  <button onclick="showSection('checkOutRoom') ">Check-Out Room</button>
  <button onclick="showSection('guestSummaryReport') ">Guest Summary
Report</button>
</div>
```

- **Forms for Data Entry:** HTML forms are used to collect data such as room details and customer information.

```
html
Copy code
<form id="addRoomForm">
  <label>Room Number:</label>
  <input type="number" id="roomNumber" required>
  <!-- More fields here -->
  <button type="button" onclick="addRoom() ">Add Room</button>
</form>
```

CSS

The CSS ensures the layout is user-friendly and responsive, with components like buttons and forms styled for ease of use.

```
css
Copy code
.menu button {
  margin: 5px;
  padding: 10px 20px;
  border: none;
  background-color: #007BFF;
  color: white;
  border-radius: 5px;
  cursor: pointer;
}
```

JavaScript

JavaScript is responsible for managing the application's logic and interactions.

- **Room Management:** The `addRoom` function stores room details in an array and updates the UI to display confirmation messages.

```
javascript
Copy code
let rooms = [];
function addRoom() {
  const roomNumber = document.getElementById('roomNumber').value;
  const room = { roomNumber, status: 0 }; // 0 = available
  rooms.push(room);
  document.getElementById('manageRoomOutput').innerText = `Room
${roomNumber} added!`;
}
```

- **Check-In Functionality:** This function updates room availability, assigns customer data to a room, and outputs success or failure messages.

```
javascript
Copy code
function checkIn() {
  const roomNumber = document.getElementById('checkInRoomNumber').value;
  let room = rooms.find(r => r.roomNumber == roomNumber);
  if (room && room.status === 0) {
    room.status = 1; // reserved
    room.customer = { name: customerName };
    document.getElementById('checkInOutput').innerText = `Checked in
successfully!`;
  }
}
```

- **Bill Calculation:** The check-out function calculates the bill based on the room rent and days of stay.

Testing

Unit Testing:

Each feature (e.g., adding rooms, check-in, check-out) was tested individually to ensure functionality.

Integration Testing:

Once all features were developed, integration testing ensured that the different functions worked seamlessly together. For instance, checking room availability after check-in and generating accurate bills at check-out.

User Experience and Interface Feedback

The design is kept minimalistic and straightforward, with intuitive buttons for navigation. Feedback from users showed that the system was easy to navigate and responsive. Some feedback suggested improving the visuals by adding room images or integrating dynamic charts for data reporting.

Future Scope

1. **Database Integration:** Currently, the system stores data only in the session. Future versions should integrate a backend database (e.g., MySQL) for persistent data storage.
2. **Advanced Reporting:** Include advanced analytics for room occupancy trends, guest feedback, and financial data.
3. **Mobile-Friendly Design:** Enhance the CSS to provide a fully responsive experience for mobile and tablet users.

OUTPUT:

Hotel Management System

Manage Rooms

Check-In Room

Available Rooms

Search Customer

Check-Out Room

Guest Summary Report

Manage Rooms

Room Number:

106

AC (A/N):

N

Comfort (S/N):

N

Size (B/S):

S

Daily Rent:

8000

Add Room

Room 106 added successfully!

Hotel Management System

Manage Rooms

Check-In Room

Available Rooms

Search Customer

Check-Out Room

Guest Summary Report

Check-In Room

Room Number:

105

Booking ID:

15

Customer Name:

Yashraj

Address (City):

wing

Phone:

08485071819

From Date:

02-10-2024



To Date:

08-10-2024



Advance Payment:

7000

Check-In

Customer Aditya checked in successfully!

Hotel Management System

Manage Rooms

Check-In Room

Available Rooms

Search Customer

Check-Out Room

Guest Summary Report

Search Customer

Customer Name:

Customer Name: Yashraj

Room Number: 101

From Date: 2024-10-20

To Date: 2024-10-26

Hotel Management System

Manage Rooms

Check-In Room

Available Rooms

Search Customer

Check-Out Room

Guest Summary Report

Check-Out Room

Room Number: Number of Days:

Total Bill for Room 101: \$50000

Conclusion

The Hotel Management System provides a robust, browser-based solution for handling key hotel functions like room management, customer check-ins, and generating reports. The use of front-end technologies such as HTML, CSS, and JavaScript ensured a lightweight and fast application, though it can be enhanced with the integration of a database for persistent storage.

References

- <https://www.geeksforgeeks.org/>
- <https://www.w3schools.com/>