REPORT

ON

DATABASE MANAGEMENT SYSTEM

ON

HOTEL MANAGEMENT SYSTEM

Submitted to

NMAM INSTITUTE OF TECHNOLOGY, NITTE

In partial fulfillment of the requirements for the award of the

Degree of Bachelor of Technology
In
Information Science & Engineering

By

SANDEEP	USN: **NNM23IS160**
VIJETH	USN: **NNM23IS154**
ROSHWIN	USN: **NNM23IS152**
ROHAN	USN: **NNM23IS150**

Under the guidance of

Ms. PRATHEEKSHA HEGDE

Professor
Department of Information Science & Engineering

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

NITTE (Deemed to be University) | NMAM INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the "Data Science report" submitted by **SANDEEP** USN **NNM23IS160** *and* **VIJETH** USN **NNM23IS154** *and* **ROHAN** USN **NNM23IS150** *and* **ROSHWIN** USN **NNM23IS152** *of* 4^{th} *semester B.Tech., a bonafide student of NMAM Institute of Technology, Nitte, has undergone Data Science Project on* **HOTEL MANAGEMENT SYSTEM** *at* ***N.M.A.M. INSTITUTE OF TECHNOLOGY*** *during* ***2024-25*** *fulfilling the partial requirements for the award of degree of Bachelor of Technology in* ***Information Science & Engineering*** *at NMAM Institute of Technology, Nitte.*

_____

Name and Signature of Course Instructor

Table of Contents

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on individual efforts but on the guidance, encouragement, and cooperation of intellectuals, elders, and friends. A number of personalities, in their own capacities, have helped me in carrying out this project work. I would like to take this opportunity to thank them all.

First and foremost, I would like to thank **Dr. Niranjan N Chiplunkar**, Principal, and **Dr.I.R Mithanthaya** Vice Principal NMAMIT, Nitte, for their moral support towards completing our Project work.

I deeply express my sincere gratitude to our Project guide **Ms. Pratheeksha Hegde N**, Department of ISE, NMAMIT, Nitte, for his able guidance, regular source of encouragement, and assistance throughout this project work.

Last, but not the least, I would like to thank my peers and friends who provided us with valuable suggestions to improve our project.

ABSTRACT

This project presents the design and implementation of a **Hotel Management System** developed using **PHP** and **MySQL**. The system aims to simplify and streamline various administrative tasks in a hotel, including **customer registration**, **room booking**, **payment recording**, **check-out processing**, and **report generation**. The primary goal is to replace manual record-keeping with an automated and user-friendly solution that improves operational efficiency and data accuracy.

The system supports core functionalities such as adding and managing customers and rooms, booking rooms based on availability, recording payments with receipts, and managing check-outs with automatic room status updates. A secure login system is implemented for admin access, and the user interface is styled for ease of use and navigation.

An important feature of this system is the use of **MySQL triggers**, which help in maintaining data integrity—for example, deleting customer records after checkout if they have no active bookings. Additionally, the system provides a consolidated view of all bookings and supports data reporting to help with decision-making and record auditing.

This Hotel Management System is scalable, maintainable, and secure, making it suitable for small to mid-sized hotels looking to modernize their operations through an efficient and cost-effective solution.

Chapter 1

INTRODUCTION

In today's fast-paced and technology-driven world, automation of business processes is essential to improve efficiency, accuracy, and user satisfaction. The hotel industry, being a service-oriented domain, greatly benefits from digital management solutions that replace traditional, paper-based operations. The **Hotel Management System** (HMS) is a web-based application developed to address this need by streamlining key hotel operations such as customer registration, room booking, payment processing, and room availability tracking.

Manual hotel management can often lead to errors in record-keeping, double bookings, customer dissatisfaction, and poor use of resources. This project aims to solve these issues by implementing an integrated system using **PHP** for server-side programming and **MySQL** for managing the relational database. The system is designed with scalability, user-friendliness, and data consistency in mind.

The HMS allows hotel staff to easily manage customer data, room details, check-in/check-out operations, and payment tracking all from a centralized dashboard. It includes smart features such as real-time availability updates, receipt generation, and automated customer removal through triggers to keep the database clean and optimized.

The system also focuses on the separation of concerns by organizing pages for specific tasks—such as adding customers, booking rooms, recording payments, checking out guests, and viewing all bookings—making it modular and easy to maintain.

Chapter 2

OBJECTIVES

The primary goal of this project is to develop an efficient, user-friendly, and reliable **Hotel Management System** that automates and simplifies daily hotel operations. Below are the specific objectives of the system:

2.1 General Objective

- To design and implement a web-based hotel management system that handles customer information, room availability, bookings, payments, and checkout in an organized and efficient manner.

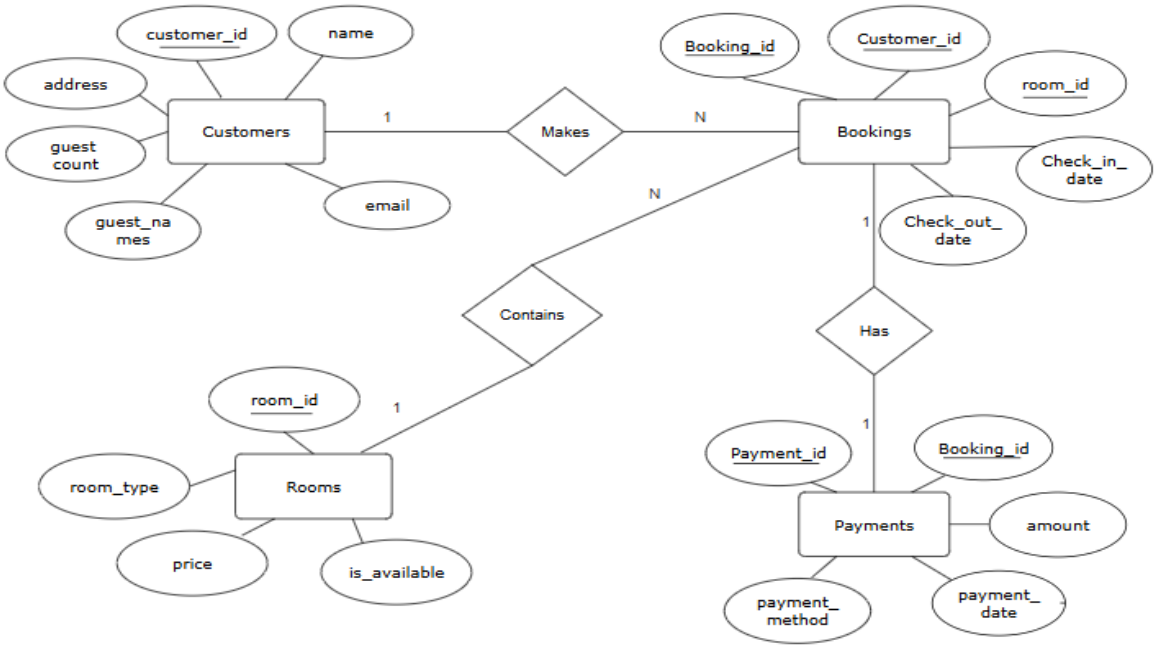
2.2 Specific Objectives

- **Customer Management**: To create a module that allows adding and storing guest information including names, contact details, guest count, and guest names.
- **Room Management**: To maintain an up-to-date list of rooms with types, pricing, and availability status. The system should prevent double bookings.
- **Booking Functionality**: To allow booking of rooms based on availability and customer selection. The booking should be recorded along with check-in/check-out dates.
- **Payment Recording**: To handle payment details for each booking, and generate a receipt with customer, room, and amount information.
- **Checkout Process**: To mark bookings as “Completed” when the customer checks out, and update room availability accordingly.
- **Trigger Usage**: To utilize SQL triggers to automate tasks like deleting customer data once all their bookings are completed.
- **Dashboard Interface**: To provide a clean and interactive dashboard to navigate to key modules such as bookings, rooms, payments, and reports.
- **Report and View Bookings**: To display all bookings with filters, statuses, and essential details for easy monitoring and analysis.
- **Security**: To include admin login authentication and secure password storage using hashing mechanisms like password_hash() and password_verify().
- **Database Efficiency**: To normalize the database, maintain data integrity, and ensure performance using relationships and triggers.

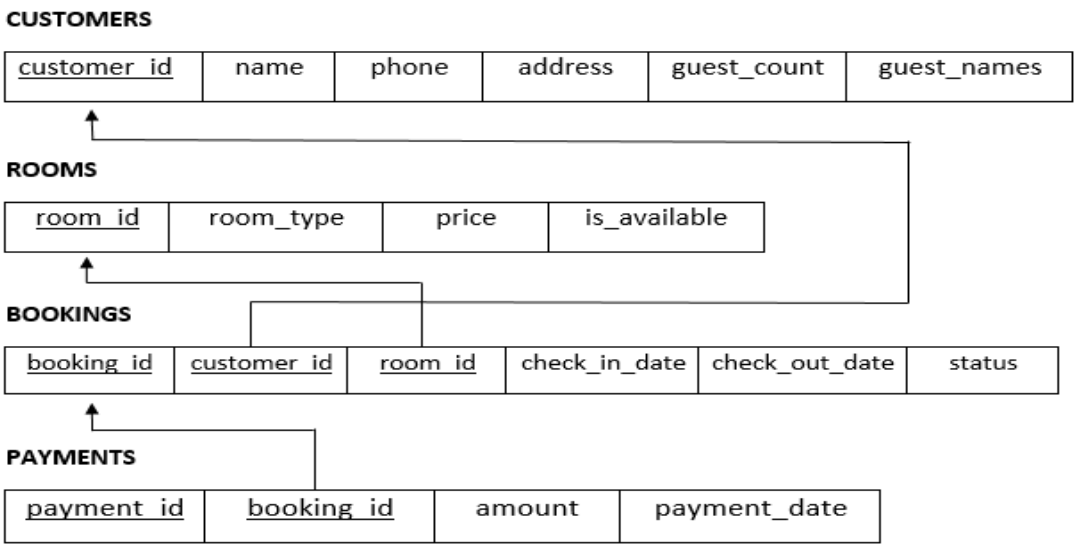
Chapter 3

DIAGRAMS

3.1. ER DIAGRAM



3.1. Schema Diagram



Chapter 4

System Requirements

This section outlines the essential **hardware and software requirements**, along with the **technologies** used to build and run the Hotel Management System.

4.1. System Requirements

4.1.1 Hardware Requirements

Component	Minimum Requirement
Processor	Intel Core i3 or equivalent
RAM	4 GB
Hard Disk	500 GB HDD or 128 GB SSD
Input Devices	Keyboard and Mouse

4.1.2 Software Requirements

Software	Description
Operating System	Windows 10 / Linux / macOS
Web Server	XAMPP (Apache + MySQL + PHP)
Database Server	MySQL 5.7 or higher
Programming Language	PHP 7.x or higher
Web Browser	Chrome / Firefox / Edge
Text Editor/IDE	VS Code

Chapter 5

SQL QUERIES

5.1. Table Creation

```
CREATE TABLE admin (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    password VARCHAR(255) NOT NULL
);

CREATE TABLE rooms (
    room_id INT AUTO_INCREMENT PRIMARY KEY,
    room_type VARCHAR(100),
    price DECIMAL(10,2),
    is_available BOOLEAN DEFAULT 1
);
CREATE TABLE customers (
    customer_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    phone VARCHAR(20),
    address TEXT,
    guest_count INT,
    guest_names TEXT
);
CREATE TABLE bookings (
    booking_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    room_id INT,
    check_in_date DATE,
    check_out_date DATE,
    status VARCHAR(20) DEFAULT 'Booked',
     FOREIGN KEY (customer_id) REFERENCES customers(customer_id) ON DELETE
CASCADE,
    FOREIGN KEY (room_id) REFERENCES rooms(room_id) ON DELETE CASCADE
);

CREATE TABLE payments (
    payment_id INT AUTO_INCREMENT PRIMARY KEY,
    booking_id INT,
    amount DECIMAL(10,2),
    payment_date DATETIME,
    FOREIGN KEY (booking_id) REFERENCES bookings(booking_id) ON DELETE CASCADE
);
```

5.2. Insert Queries

```
INSERT INTO admin (username, password) VALUES
('admin', 'hashed_password');
INSERT INTO rooms (room_type, price) VALUES
('Deluxe', 3000);
INSERT INTO customers (name, email, phone, address, guest_count, guest_names)
VALUES
('John Doe', 'john@example.com', '1234567890', '123 Street', 2, 'John, Jane');
INSERT INTO bookings (customer_id, room_id, check_in_date, check_out_date,
status) VALUES
(1, 2, '2025-04-01', '2025-04-05', 'Booked');
```

5.3. Select Queries

```
SELECT * FROM rooms WHERE is_available = 1;
SELECT b.booking_id, c.name, r.room_type
FROM bookings b JOIN customers c ON b.customer_id = c.customer_id
JOIN rooms r ON b.room_id = r.room_id;
SELECT * FROM payments WHERE booking_id = 1;
```

5.4. Triggers

```
DELIMITER $$
CREATE TRIGGER `make_room_available_after_checkout` AFTER UPDATE ON `bookings`
FOR EACH ROW BEGIN
    IF OLD.check_out_date IS NULL AND NEW.check_out_date IS NOT NULL THEN
        UPDATE rooms
        SET is_available = 1
        WHERE room_id = NEW.room_id;
    END IF;
END
$$
DELIMITER ;
```

```
DELIMITER $$
CREATE TRIGGER `mark_room_unavailable_after_booking` AFTER INSERT ON `bookings`
FOR EACH ROW BEGIN
    UPDATE rooms
    SET is_available = 0
    WHERE room_id = NEW.room_id;
END
$$
DELIMITER ;
```

Chapter 6

SNAPSHOTS

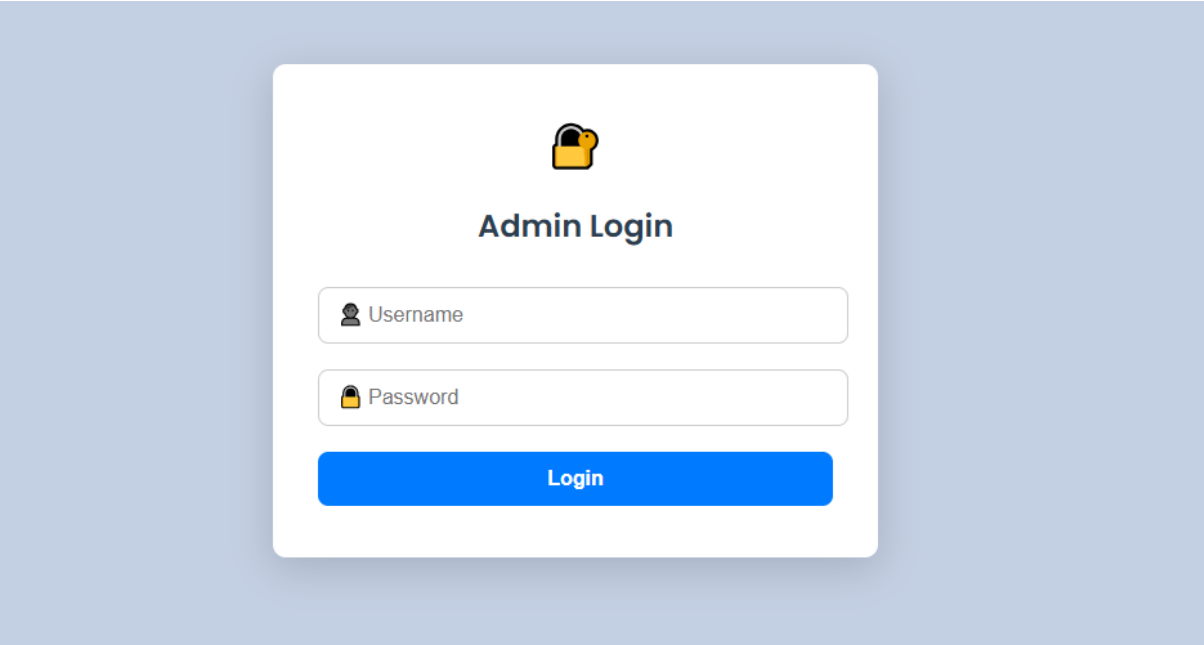


Fig 1: LOGIN PAGE

This page allows the administrator to securely log into the hotel management system using a username and password. Passwords are verified using hashing techniques for added security.

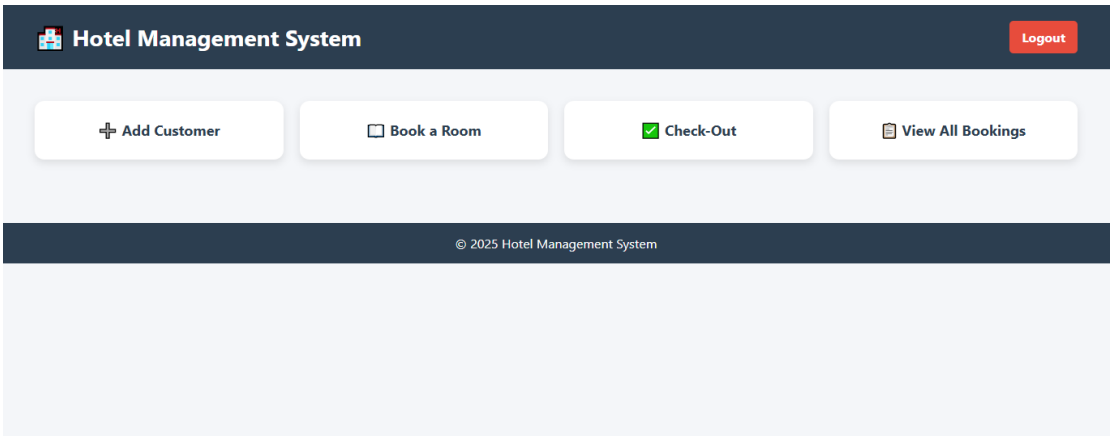


Fig 2: Dashboard

The central hub of the system. From here, the admin can navigate to all major functionalities like adding customers, booking rooms, checking out, viewing bookings, recording payments, and generating receipts.

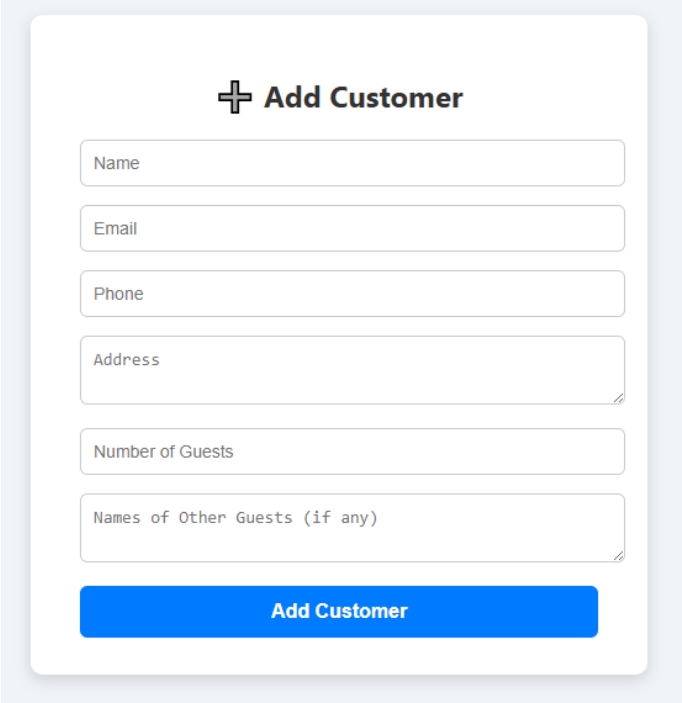


Fig 3 Add Customer

This form collects customer information including name, email, phone, address, number of guests, and guest names. It checks for duplicate entries before inserting new customers.

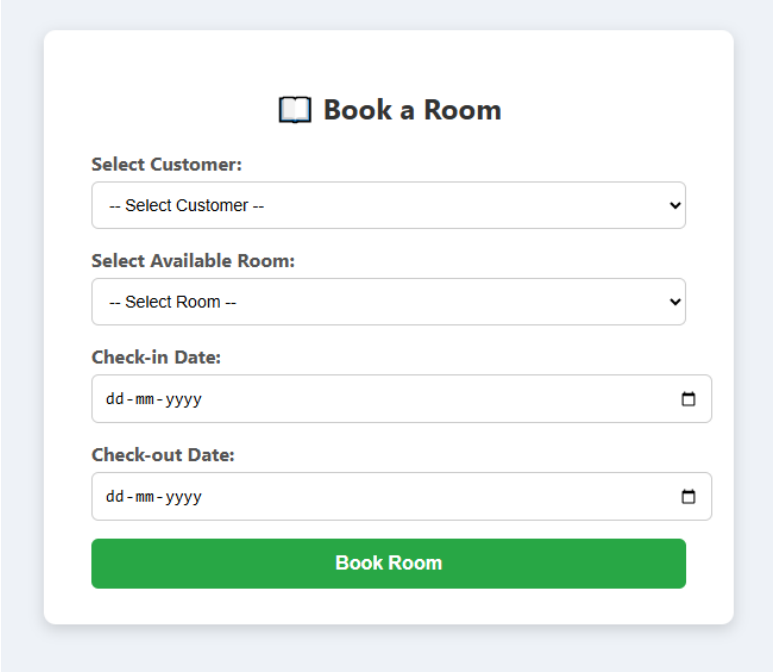


Fig 4 Add Room

This section is used to add hotel rooms with specific room types, prices, and availability status. Rooms can later be assigned to bookings based on availability.

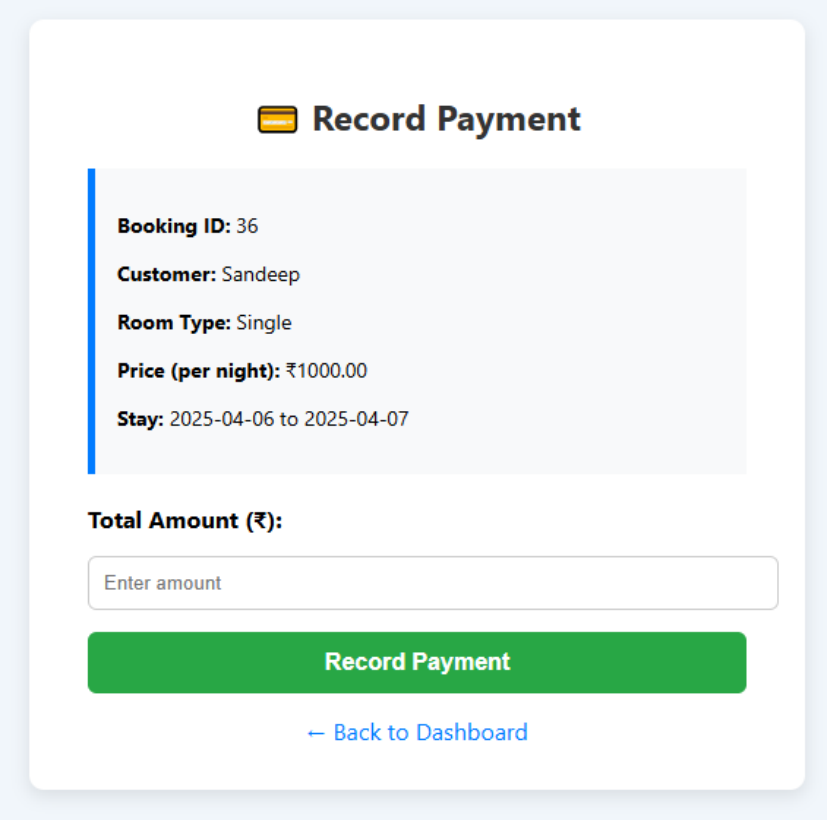


Fig 5 Payment Page

Displays booking and customer details. Admin can enter and record the payment amount. After payment, it automatically shows a styled receipt and provides a print option.

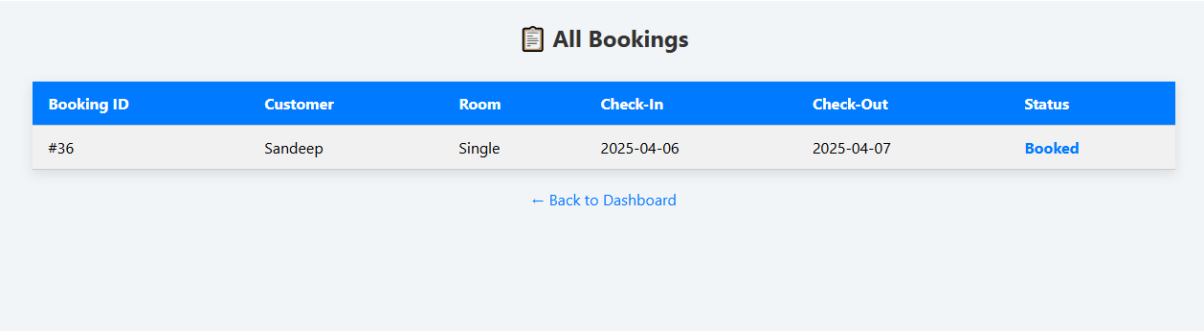


Fig 6 View Bookings

Displays all bookings in a tabular format including booking ID, customer name, room type, check-in and check-out dates, and booking status (Booked or Completed).

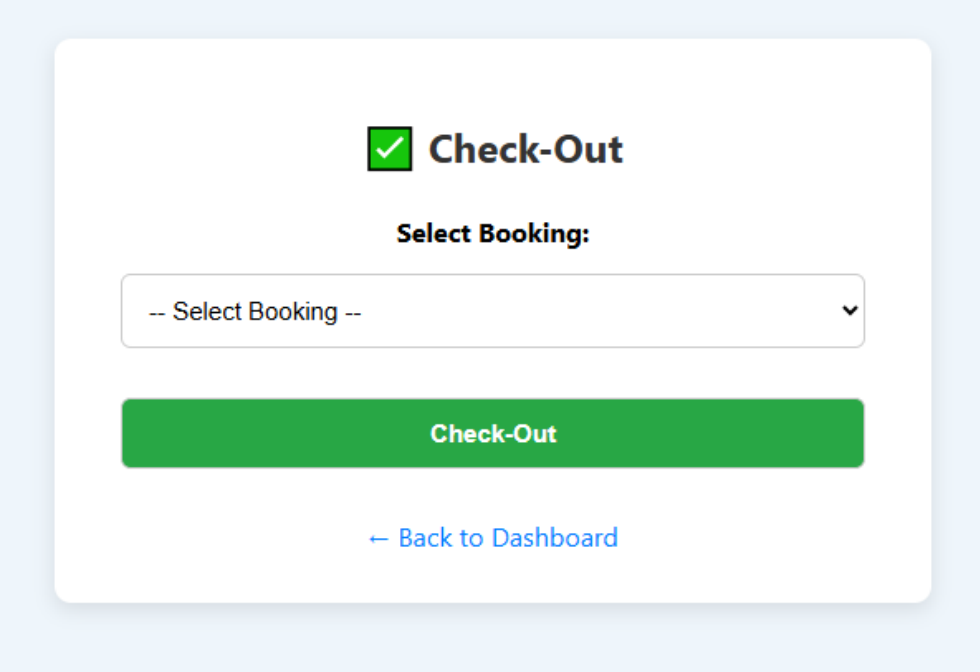


Fig 7 Check-Out

Lets the admin check out a customer. It updates the booking status to "Completed" and makes the room available again. It can optionally trigger customer removal via a MySQL trigger if no more active bookings exist.

Chapter 7

CONCLUSION

The Hotel Management System developed as part of this project successfully automates and streamlines the essential operations of a hotel, including customer registration, room booking, payment handling, and check-out processes. Through the use of PHP and MySQL, we created a dynamic and user-friendly web application that ensures efficiency, accuracy, and ease of use.

This system not only reduces the manual workload of hotel staff but also ensures that data is well-organized, secure, and easily retrievable. Features like real-time room availability, automated receipts, and status updates enhance the overall experience for both the administrators and the customers.

The integration of SQL triggers, ER diagrams, and proper database relationships adds robustness to the backend while ensuring data integrity. The responsive and styled frontend interface contributes to better usability and clarity.

In conclusion, the system achieves its goal of digitizing hotel operations, paving the way for future scalability and enhancements such as online customer self-service portals, analytics, and cloud integration. It reflects a practical application of core web development and database concepts in solving real-world problems.