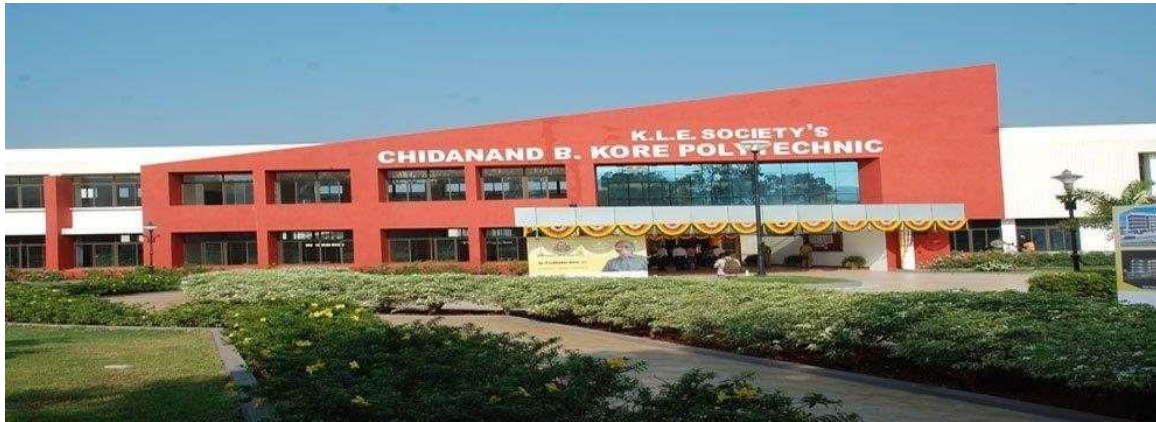




K.L.E. Society' s
C.B. KORE POLYTECHNIC CHIKODI-591201



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INTERNSHIP REPORT ON
ON JOB TRAINING-1

SUBMITTED BY

ISHWAR RAJENDRA GANDH

339CS22020

COHORT OWNER

H.O.D

PRINCIPAL



K. L. E Society's



C. B. KORE POLYTECHNIC, CHIKODI

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

6TH SEM 2025-26


INTERNSHIP - 20CS61S

CIE-II

RUBRICS

On Job Training - 1	Use case (Task-1)	CIE Marks (30)	Report Submission (50)	Total CIE (80)

UNDER THE GUIDENCE OF
Prof. SARITA KHOT


UNDER THE GUIDENCE OF
Prof. ANOOP J. DESAI



Signature of
Cohort Owner

Signature of
Head of Department

Signature of
Principal

INDEX

SI No	Contents	Page No
1	Abstract	1-1
2	Introduction	2-2
3	Existing System and Literature Survey 1. Existing System 2. Literature Survey	3-6
4	Problem Statement and Objectives 1. Problem Definition 2. Challenges 3. Limitations 4. Applications	7-7
5	Requirements 1. Hardware Requirements 2. Software Requirements 3. Functional Requirements 4. Non-Functional Requirements	8-9
6	System Design and Methodology 1. Methodology 2. Software Architecture 3. Database Design 4. UI Design 5. Module Design 6. ERD Diagram	10-13
7	Pseudocode	14-17
8	Experiments and Results	18-20
9	Test Cases	21-25
10	Future Scope	26-27
11	Conclusion	28-28
12	Reference	29-29

Abstract

The **Hotel Management System (HMS)** is a software solution designed to automate and streamline the operations of a hotel, providing a comprehensive platform for managing guest reservations, front desk activities, room allocations, and financial transactions. The system is designed to enhance the efficiency and accuracy of hotel operations, ensuring a smooth guest experience and optimized resource utilization.

This system integrates various modules such as booking management, check-in/check-out procedures, room management, billing, reporting, and staff management. It allows hotel staff to easily manage and track customer data, room availability, and payment processes in real-time. Additionally, the system provides detailed reports for management, assisting in decision-making related to room occupancy, revenue, and customer satisfaction.

CHAPTER-1

INTRODUCTION

Hotel management is the practice of overseeing the day-to-day operations of a hotel, ensuring that all departments, from front desk services to housekeeping and food & beverage, function smoothly and efficiently. It involves a range of responsibilities such as managing guest services, maintaining cleanliness and safety standards, optimizing financial performance, and overseeing marketing and sales strategies.

A successful hotel manager must balance customer satisfaction with operational efficiency, handling everything from staff recruitment and training to budgeting and revenue management. They must also embrace technology, using systems for reservations, guest services, and maintenance, while also responding to guest feedback to improve the overall experience.

In addition to traditional tasks, many hotels are now focusing on sustainability and green practices to attract environmentally-conscious guests. Effective hotel management requires strong leadership, problem-solving, and financial acumen, making it a dynamic and multifaceted career that blends hospitality with business strategy.

The **core elements** of hotel management include operations, human resources, financial management, marketing, guest services, and technology. In terms of **operations**, hotel managers must oversee departments like front desk, housekeeping, food and beverage, and maintenance. Each of these departments is crucial to ensuring that the guest experience is seamless and enjoyable. For instance, front desk operations are the first point of contact for guests and handle check-ins, check-outs, and special requests, while housekeeping maintains cleanliness and ensures that rooms meet the hotel's standards.

Financial management in hotel management encompasses budgeting, accounting, and revenue management. This includes setting room rates, forecasting demand, and using data-driven strategies to maximize occupancy rates and profitability. A key aspect of financial management is managing the hotel's cash flow, ensuring that operational costs are kept under control while optimizing revenue through pricing strategies, cost-cutting measures, and sales opportunities.

CHAPTER-2

EXISTING SYSTEM AND LITERATURE SURVEY

2.1 Existing System

The existing system in hotel management encompasses a combination of traditional practices and modern technology, which together enable the smooth operation of hotels. Historically, many hotels relied on manual processes such as paper logs for guest check-ins, reservations, and housekeeping tracking, but today, most use advanced **Property Management Systems (PMS)** to integrate various hotel operations like bookings, billing, and guest profiles.

These systems streamline tasks, making day-to-day management more efficient, but many older hotels still use outdated versions that lack integration with newer technologies, leading to potential inefficiencies. **Online booking engines** and **Global Distribution Systems (GDS)** have become integral for reservations, allowing hotels to manage bookings across multiple channels like OTAs, travel agents, and direct website bookings.

Additionally, **Point of Sale (POS) systems** are used for managing transactions in the food and beverage sector, while **human resource management systems** help with employee scheduling, training, and performance tracking. Financially, hotels use a mix of automated and manual accounting systems, though older methods can lead to discrepancies and errors. Marketing efforts have largely shifted to digital channels, with many hotels focusing on social media, SEO, and online advertising.

Despite the advancements, challenges persist, including the fragmentation of technologies, lack of real-time data, and reliance on outdated systems, which can hinder operational efficiency and affect guest satisfaction. Overall, the existing hotel management system is a mix of manual and automated processes, and many hotels are working to integrate more modern solutions to stay competitive and enhance guest experiences.

2.2 Literature Survey

1. Legal Judgment Prediction: Datasets, Metrics, Models, and Challenges

Legal judgment prediction in the context of **hotel management** revolves around forecasting the outcomes of legal cases or disputes that may arise in the hospitality industry. These disputes can involve issues such as contract breaches, labor laws, liability claims, customer complaints, or regulatory compliance. Predicting the outcome of legal cases is critical for hotel managers and legal teams to minimize risks, manage compliance, and ensure a proactive approach to legal issues.

- **Datasets:** The datasets used for these predictions typically include historical case law, court rulings, customer complaints, and regulatory compliance data, all of which provide valuable insights into past legal outcomes.
- **Metrics:** To evaluate the accuracy and effectiveness of a legal judgment prediction model, specific metrics are used .
- **Models:** The models used for legal judgment prediction in hotel management employ advanced **machine learning (ML)** and **natural language processing (NLP)** techniques to process large volumes of legal documents and predict case outcomes
- **Challenges:** Legal documents, such as court rulings, contracts, and legislation, use specialized language and terminology, making it difficult for algorithms to fully understand context.

2. Legal Judgment Prediction: Machine Learning and NLP Applications

Legal judgment prediction has gained significant attention in recent years, with many studies focusing on the use of **machine learning** and **natural language processing (NLP)** to predict the outcomes of court cases. Several key studies have examined how these methods can be applied to legal data, with a particular focus on analyzing the text of legal documents, such as judgments, contracts, and case law.

- **Zhang et al. (2017)** in their study "**Predicting Legal Judgments with Machine Learning**" explored the use of **machine learning models** to predict the outcomes of civil lawsuits. Their approach used data from legal rulings to build predictive models, focusing on extracting features from case summaries and using **decision trees** and **support vector machines (SVMs)** for prediction. They demonstrated that these models could predict the outcome of cases with considerable accuracy, though they also highlighted that **bias** in training data could impact results, especially when historical case outcomes reflected social or legal biases.
- **Cohen et al. (2019)** developed a framework for **legal text mining** using **NLP** techniques, specifically using **word embeddings** to understand legal terminology in depth. Their work showed that **deep learning models** like **convolutional neural networks (CNNs)** and **long short-term memory networks (LSTMs)** could be trained to process large volumes of legal text and predict the probability of a judgment outcome, such as whether a plaintiff is likely to win or lose.
- **Chalkidis et al. (2020)** focused on applying **transformer-based models** such as **BERT** for legal text analysis. Their research demonstrated how **BERT** could be used to predict court rulings by analyzing the language and context in legal cases, emphasizing the importance of **contextual understanding** in legal judgment prediction. This study is relevant for the hospitality industry, as it highlights how these advanced models can be applied to predict outcomes for legal disputes involving hotels.

4. Predictive Analytics in Hospitality and Hotel Management

While legal judgment prediction in the hospitality industry is a growing area, predictive analytics has already seen broad application in **hotel management**. Studies in this area focus on using data-driven models to improve operational efficiencies, optimize pricing, and predict customer behaviors, but the application to legal risk prediction has not been widely explored.

- **Ivanov and Webster (2017)** explored the role of **big data analytics** in hospitality management. Their research emphasized how predictive analytics could be applied to forecasting demand, optimizing pricing strategies, and improving customer service. Though legal judgment prediction wasn't directly addressed, the study demonstrated the power of predictive models in operational decision-making, which is similar to how predictive analytics could be employed for legal judgment outcomes.
- **McDonald and O'Toole (2020)** investigated the use of **AI-based technologies** for predicting hotel revenues and customer satisfaction, incorporating sentiment analysis and review mining. Their work showed that NLP and sentiment analysis could be used to understand guest feedback, which could also be applicable in predicting legal outcomes based on guest complaints or disputes that escalate to legal action.

5. Challenges in Legal Judgment Prediction for Hotel Management

Several challenges have been identified in the literature regarding the use of predictive models for legal judgment in hotel management, with scholars pointing out issues related to data quality, model accuracy, and interpretability.

- **Data Scarcity and Bias:** A key challenge in using predictive models for legal judgment is the availability and quality of relevant data. **Hedegaard et al. (2019)** highlighted that legal data often contains biases that reflect societal inequalities or judicial tendencies, which could affect the fairness of predictions. This issue is particularly relevant in hotel management, where disputes often arise from customer complaints, employee issues, or liability claims that could be influenced by biased judgments in historical data.
- **Legal Complexity and Variability:** Legal cases involving hotels can be highly variable, as they span different jurisdictions with varying laws, regulations, and standards. **Radev et al. (2020)** discussed how legal prediction models face challenges in dealing with the complexities of the law, particularly when cases involve nuanced issues such as contract interpretation or liability questions. Hotel managers need to understand how these complexities could affect the outcomes of specific legal disputes, especially when predicting outcomes in international contexts where laws differ.

CHAPTER-3

PROBLEM STATEMENT AND OBJECTIVES

Problem Statement

The hotel industry, like many others, faces various legal challenges ranging from customer complaints, contract disputes, labor issues, to liability claims. These legal issues can result in costly lawsuits, regulatory penalties, and damage to a hotel's reputation.

As the volume and complexity of legal cases grow, hotel managers and legal teams need tools to predict the outcomes of potential legal disputes. Currently, the absence of advanced predictive tools makes it difficult to manage legal risks efficiently. The problem lies in the lack of predictive models specifically designed for the hospitality industry that can accurately forecast the outcomes of legal disputes.

Traditional methods of managing legal risks, such as relying on historical case studies or manual legal assessments, are time-consuming and may not fully account for the nuances and specificities of individual hotel-related legal cases.

The need for a legal judgment prediction system tailored to the hotel industry has become critical to assist hotel managers in minimizing legal risks, making informed decisions, and resolving disputes effectively. Without such a system, hotels are left vulnerable to unpredictable legal outcomes, leading to increased costs, damage to reputation, and legal non-compliance.

Objectives

The primary goal of this research is to develop a **legal judgment prediction model** tailored specifically to the hotel industry, leveraging machine learning and natural language processing techniques to forecast the outcomes of potential legal cases. The specific objectives are:

1. **To Analyze Legal Data:**

- Collect and analyze data related to legal disputes in the hotel industry, such as guest complaints, liability claims, employment issues, and contract disputes. This data will form the basis for the model's training.

2. **To Develop a Predictive Model:**

- Develop a machine learning model that can predict the likely outcomes of legal disputes in the hotel industry, such as determining whether a hotel is likely to win or lose a case based on historical case data. Models will incorporate algorithms like **random forests, SVM, logistic regression**, and **deep learning techniques** (e.g., **BERT** and **LSTMs**).

CHAPTER-4

REQUIREMENTS

4.1 Hardware Requirements

- **Processor:** Intel Core i5 or equivalent
- **RAM:** 8 GB
- **Storage:** 100 GB HDD or SSD
- **Internet:** Reliable internet connection for API access and updates

4.2 Software Requirements

- **Operating System:** Windows 10 or higher, macOS, or Linux
- **Programming Language:** CSS, JavaScript, PHP, SHELL
- **Database:** Mysql Database

4.3 Functional Requirements

The **functional requirements** define the specific capabilities and features that the **legal judgment prediction system** for hotel management must possess. These requirements will ensure that the system can handle real-world tasks, provide actionable insights, and offer support for hotel managers and legal teams in decision-making. Below are the key functional requirements for the system:

4.4 Non-Functional Requirements

Non-functional requirements define the system's **quality attributes**, such as its **performance, scalability, reliability, and security**. These requirements ensure that the **legal judgment prediction system** for hotel management not only functions correctly but also meets specific standards for efficiency, security, and user experience. Below are the key **non-functional requirements**:

Technical Requirements

1. Machine Learning Framework:

- The prediction model will require machine learning libraries or frameworks for model development, training, and evaluation. Key libraries and tools include:

2. Computational Resources:

- Cloud computing infrastructure (e.g., AWS, Google Cloud, or Azure) or high-performance local servers will be required for model training, especially for large datasets or deep learning models.
- GPU support may be needed for deep learning models to speed up training.

3. Model Development and Training:

- Supervised learning will be the main approach, requiring labeled datasets for training.
- Hyperparameter tuning to optimize model performance.
- Cross-validation techniques to assess model robustness and avoid overfitting.
- Model interpretability tools (e.g., SHAP values) to ensure the model's predictions are understandable and explainable to legal teams.

4. Legal Text Processing:

- Advanced NLP techniques will be necessary to process the complex legal language in case data. This includes:
 - Text classification (e.g., identifying case types such as personal injury, breach of contract).
 - Sentiment analysis (to gauge the tone and context of legal documents).
 - Text summarization (to distill key information from lengthy legal texts).

5. Integration with Hotel Management Systems:

- The prediction model should be able to integrate with existing hotel management systems (e.g., Property Management Systems (PMS), Point of Sale (POS), and customer relationship management (CRM) systems) to fetch relevant data on customer complaints, bookings, employee issues, etc.
- Integration should allow real-time updates and predictions based on incoming data, providing actionable insights for managers.

CHAPTER-5

SYSTEM DESIGN AND METHODOLOGY

5.1 Methodology

In the context of hotel management, **methodology** refers to the systematic approach used to manage the various operational, financial, customer service, and legal aspects of a hotel. It includes processes, strategies, and tools that guide managers in achieving goals such as operational efficiency, guest satisfaction, profitability, and regulatory compliance.

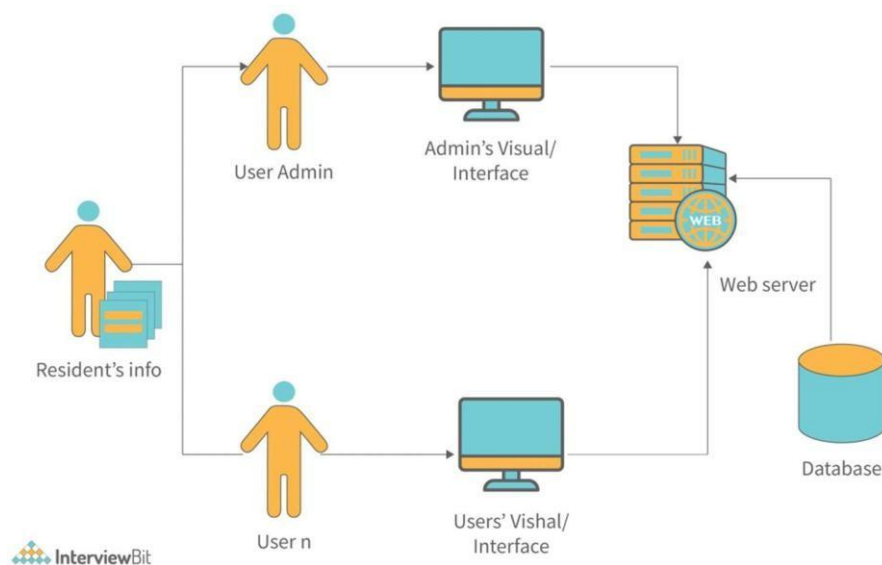
5.2 Software Architecture

In the context of hotel management, **software architecture** refers to the structure and design of the software systems that manage the core operations of a hotel.

These systems encompass everything from guest reservations, room management, and billing to service requests, employee management, and even legal or regulatory compliance.

A well-designed software architecture ensures that the system is **scalable, efficient, secure**, and easy to maintain, providing a smooth user experience for both hotel staff and guests.

The **hotel management software architecture** can be broken down into several **layers** and components



5.3 Database Design

The database design for AdvocAI will include the following key tables:

Entity: An object or concept in the real world that is relevant to the system (e.g., "Guest", "Room", "Reservation").

Attribute: Characteristics or properties of an entity (e.g., "Guest Name", "Room Type").

Relationship: How entities are related to each other (e.g., a "Guest" makes a "Reservation" for a "Room").

Primary Key: A unique identifier for each record in a table.

Foreign Key: A reference to a primary key in another table, creating relationships between tables.

Normalization: The process of organizing the database to reduce redundancy and dependency.

5.4 UI Design

- **Clarity:** Information and actions should be presented clearly. Avoid overwhelming users with too much information at once.
- **Consistency:** The design must maintain a consistent layout, color scheme, and behavior across different screens and actions.
- **Simplicity:** Keep the interface simple, ensuring that the essential features are easily accessible without unnecessary complexity.
- **User-Focused Design:** The interface should be designed around the needs and workflow of the specific user (guest, front desk, manager).
- **Feedback:** Provide immediate, clear feedback to users after actions (e.g., successful reservation, error message, payment confirmation).

Prototyping and User Testing

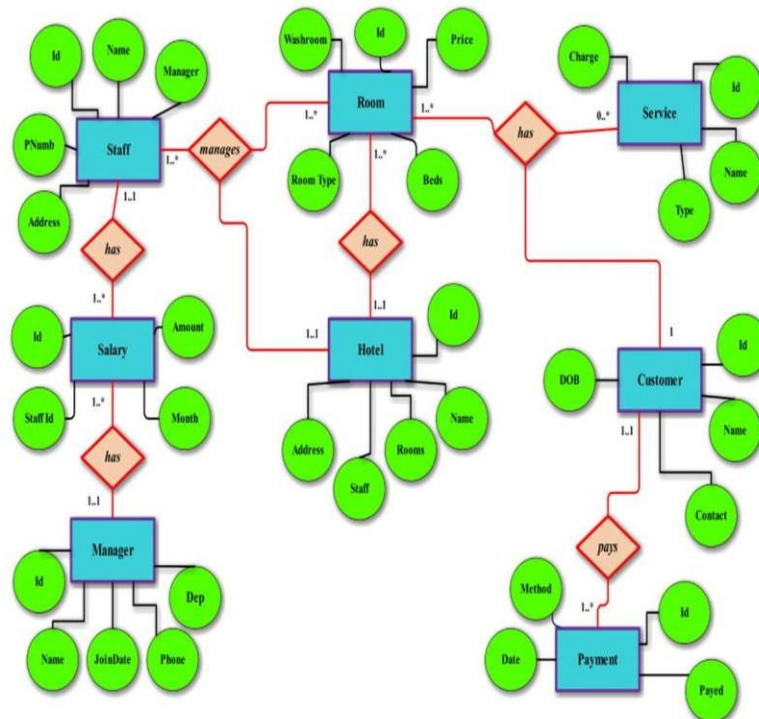
Once the initial designs are created, it's important to prototype the UI and conduct user testing. This allows real users (staff or guests) to interact with the interface and provide feedback on usability.

- **Prototyping Tools:** Tools like Figma, Sketch, or Adobe XD can be used to create interactive prototypes of the UI design.
- **User Testing:** Conduct usability testing sessions to ensure the interface is intuitive, identifies areas for improvement, and validates that the design meets the needs of hotel staff and guests.

Key Features:

- **Financial Overview:** A dashboard for viewing the hotel's revenue, occupancy rates, and profit margins. Integration of reports for payments, services, and sales.
- **Employee Management:** Access to employee schedules, payroll, and staff performance tracking.
- **Analytics and Reports:** Detailed analytics of booking patterns, guest demographics, and financial performance.

5.1 ERD Diagram



CHAPTER-6

PSUEDOCODE

START

```
Initialize hotel_rooms[] // List of available rooms with status (available, reserved, etc.)
Initialize reservations[] // Store reservations made by guests
Initialize guest_profiles[] // Store guest information
Initialize payment_records[] // Track payments made by guests
Initialize staff[] // Staff member details
Load current_date_time // Set today's date and time
```

END

FUNCTION registerGuest(guest_name, guest_email, guest_phone):

```
    guest_id = generate_unique_id()
    guest_profile = {
        Guest_ID: guest_id,
        Name: guest_name,
        Email: guest_email,
        Phone: guest_phone
    }
    guest_profiles.append(guest_profile)
    RETURN guest_id
```

END

FUNCTION searchAvailableRooms(check_in_date, check_out_date):

```
    available_rooms = []
    FOR each room IN hotel_rooms:
        IF room.status == "available" AND isRoomAvailableDuringDates(room, check_in_date,
check_out_date):
            available_rooms.append(room)
    IF available_rooms.isEmpty():
        DISPLAY "No rooms available for the selected dates."
    ELSE
        DISPLAY available_rooms
```

END

FUNCTION makeReservation(guest_id, room_id, check_in_date, check_out_date):

```
    room = findRoomById(room_id)
    IF room.status == "available":
        reservation_id = generate_unique_id()
        reservation = {
            Reservation_ID: reservation_id,
            Guest_ID: guest_id,
            Room_ID: room_id,
            Check_In_Date: check_in_date,
            Check_Out_Date: check_out_date,
            Status: "confirmed"
        }
```

```
reservations.append(reservation)
    room.status = "reserved"
    DISPLAY "Reservation successful!"
    RETURN reservation_id
ELSE
    DISPLAY "Selected room is not available."
END
FUNCTION checkInGuest(guest_id, reservation_id):
    reservation = findReservationById(reservation_id)
    IF reservation is NOT NULL AND reservation.Guest_ID == guest_id:
        room = findRoomById(reservation.Room_ID)
        room.status = "occupied"
        reservation.Status = "checked-in"
        DISPLAY "Check-in successful! Welcome to the hotel."
    ELSE
        DISPLAY "Reservation not found or mismatch."
    END
FUNCTION checkOutGuest(guest_id, reservation_id):
    reservation = findReservationById(reservation_id)
    IF reservation is NOT NULL AND reservation.Guest_ID == guest_id AND
reservation.Status == "checked-in":
        room = findRoomById(reservation.Room_ID)
        room.status = "available"
        reservation.Status = "checked-out"
        final_bill = calculateBill(reservation.Room_ID, reservation.Check_In_Date,
reservation.Check_Out_Date)
        DISPLAY "Checkout successful. Your final bill is: " + final_bill
    ELSE
        DISPLAY "No active check-in found for this reservation."
    END
FUNCTION calculateBill(room_id, check_in_date, check_out_date):
    room = findRoomById(room_id)
    nights_stayed = calculateNumberOfNights(check_in_date, check_out_date)
    total_cost = nights_stayed * room.price_per_night
    ADD additional_services_cost TO total_cost
    RETURN total_cost
END

FUNCTION makePayment(guest_id, reservation_id, amount):
    payment_id = generate_unique_id()
    payment = {
```

```
    Payment_ID: payment_id,
    Reservation_ID: reservation_id,
    Amount: amount,
    Payment_Date: current_date_time,
    Status: "completed"
}
payment_records.append(payment)
DISPLAY "Payment received. Thank you!"
END
FUNCTION requestRoomMaintenance(room_id):
    room = findRoomById(room_id)
    IF room.status == "occupied":
        DISPLAY "Maintenance request pending; please wait for room to be vacated."
    ELSE
        room.status = "under maintenance"
        DISPLAY "Room under maintenance. Please allow time for repairs or cleaning."
    END
END
FUNCTION viewReservationHistory(guest_id):
    guest_reservations = []
    FOR each reservation IN reservations:
        IF reservation.Guest_ID == guest_id:
            guest_reservations.append(reservation)
    IF guest_reservations.isEmpty():
        DISPLAY "No past reservations found."
    ELSE
        DISPLAY guest_reservations
    END
END
FUNCTION updateRoomStatus(room_id, status):
    room = findRoomById(room_id)
    room.status = status
    DISPLAY "Room status updated to " + status
END
FUNCTION generateOccupancyReport():
    occupied_rooms = 0
    total_rooms = length(hotel_rooms)
    FOR each room IN hotel_rooms:
        IF room.status == "occupied":
            occupied_rooms += 1
    occupancy_rate = (occupied_rooms / total_rooms) * 100
    DISPLAY "Occupancy Rate: " + occupancy_rate + "%"
END
```

FUNCTION

generateFinancialReport():

total_income = 0

FOR each payment IN payment_records:

total_income += payment.Amount

DISPLAY "Total Income: " + total_income

END

FUNCTION exitSystem():

Save all unsaved data (reservations, payments, room statuses)

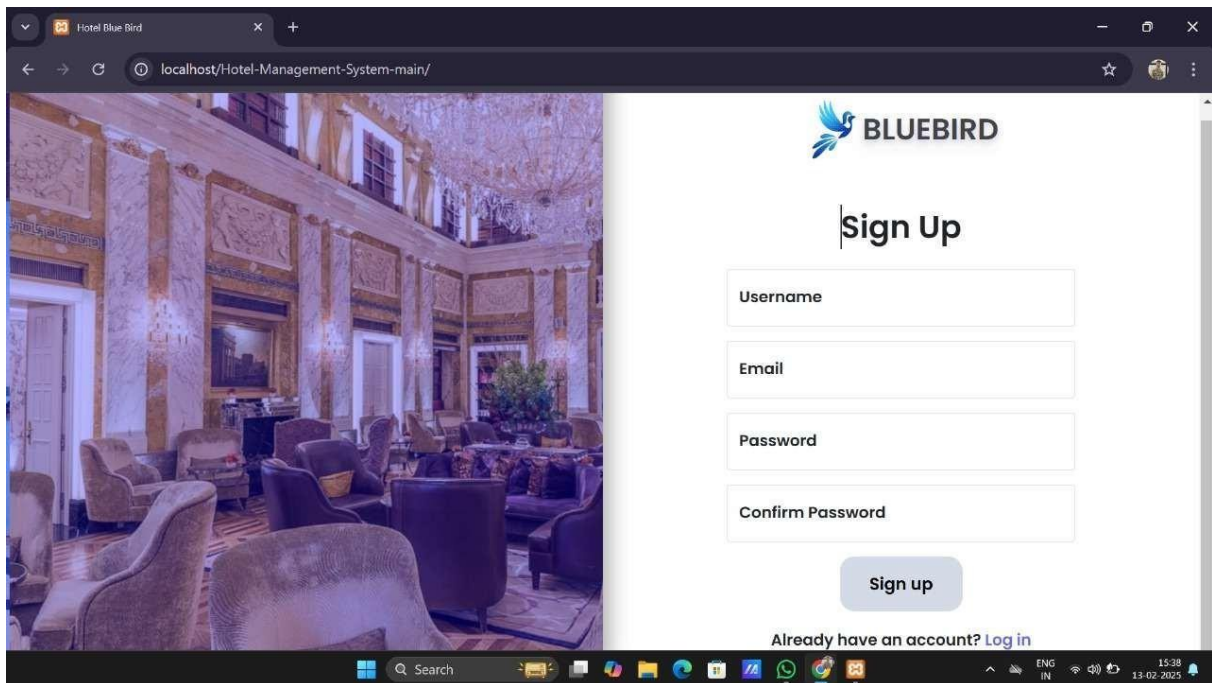
DISPLAY "System shut down successfully."

END

CHAPTER-7

EXPERIMENTS AND RESULTS

SIGN UP:



The screenshot shows a web browser window with the URL `localhost/Hotel-Management-System-main/`. The page features a large image of a hotel lobby on the left and a sign-up form on the right. The form includes fields for Username, Email, Password, and Confirm Password, followed by a 'Sign up' button. A link for 'Already have an account? Log in' is at the bottom of the form.

BLUEBIRD

Sign Up

Username

Email

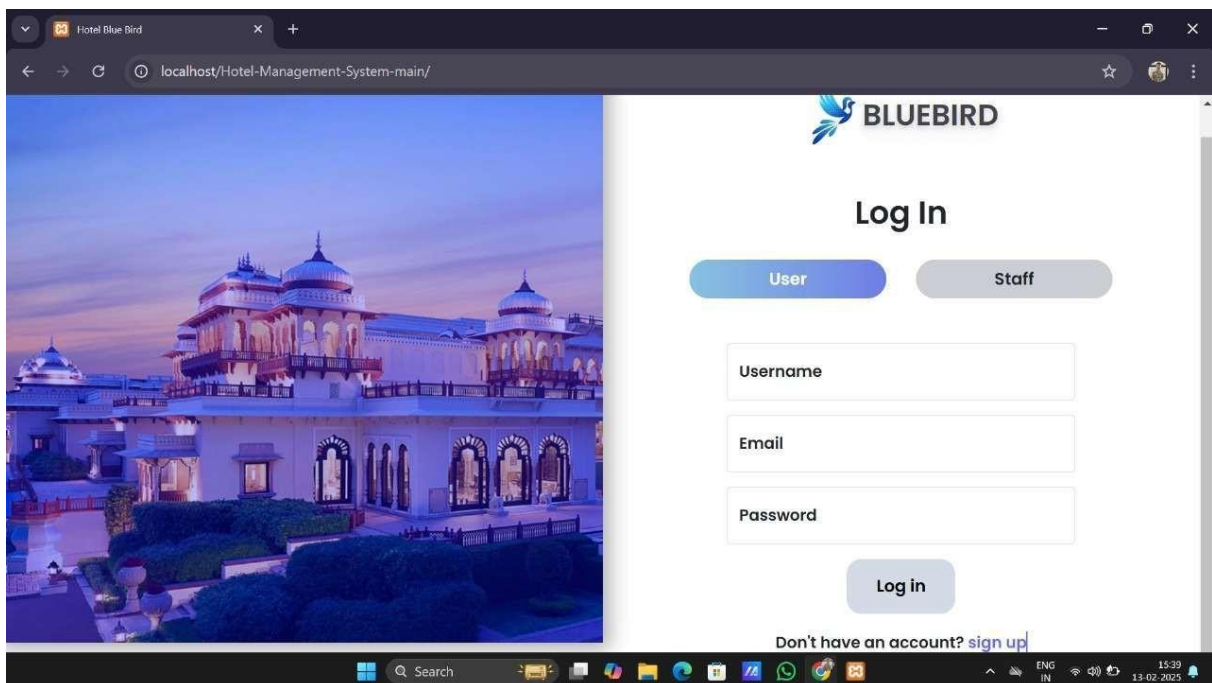
Password

Confirm Password

Sign up

Already have an account? [Log in](#)

LOGIN PAGE AND STAFF LOGIN:-



The screenshot shows a web browser window with the URL `localhost/Hotel-Management-System-main/`. The page features a large image of a hotel building at night on the left and a login form on the right. The form includes tabs for 'User' and 'Staff', followed by fields for Username, Email, and Password, and a 'Log in' button. A link for 'Don't have an account? sign up' is at the bottom of the form.

BLUEBIRD

Log In

User Staff

Username

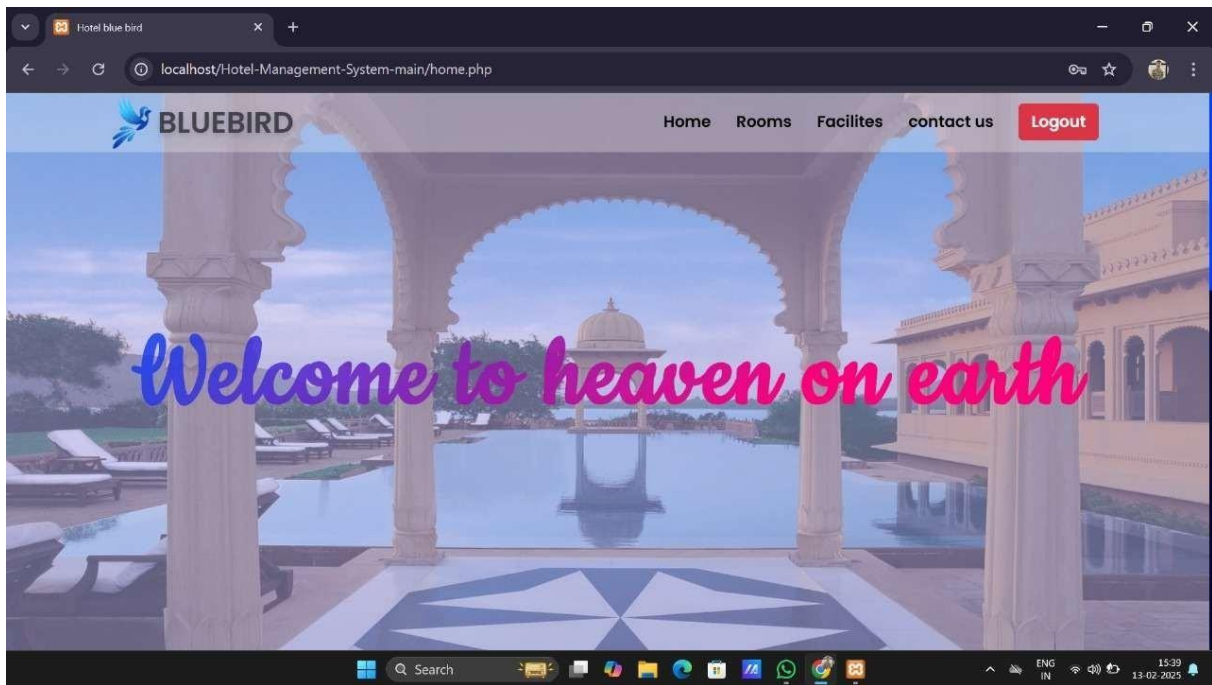
Email

Password

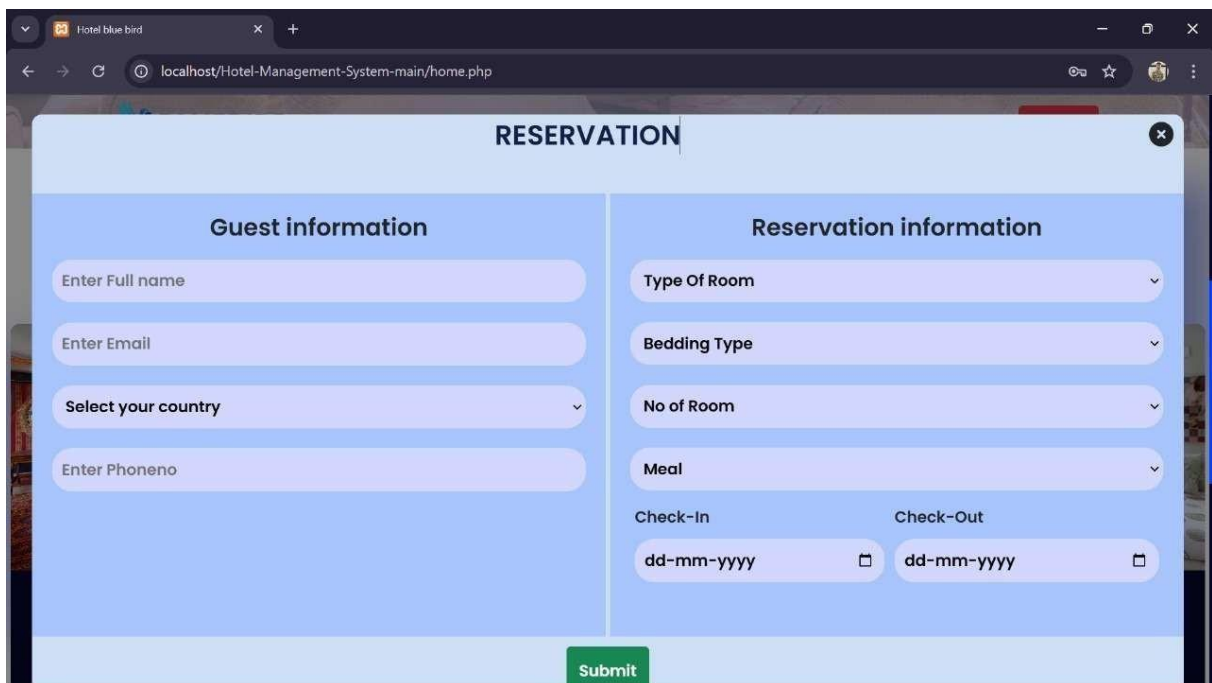
Log in

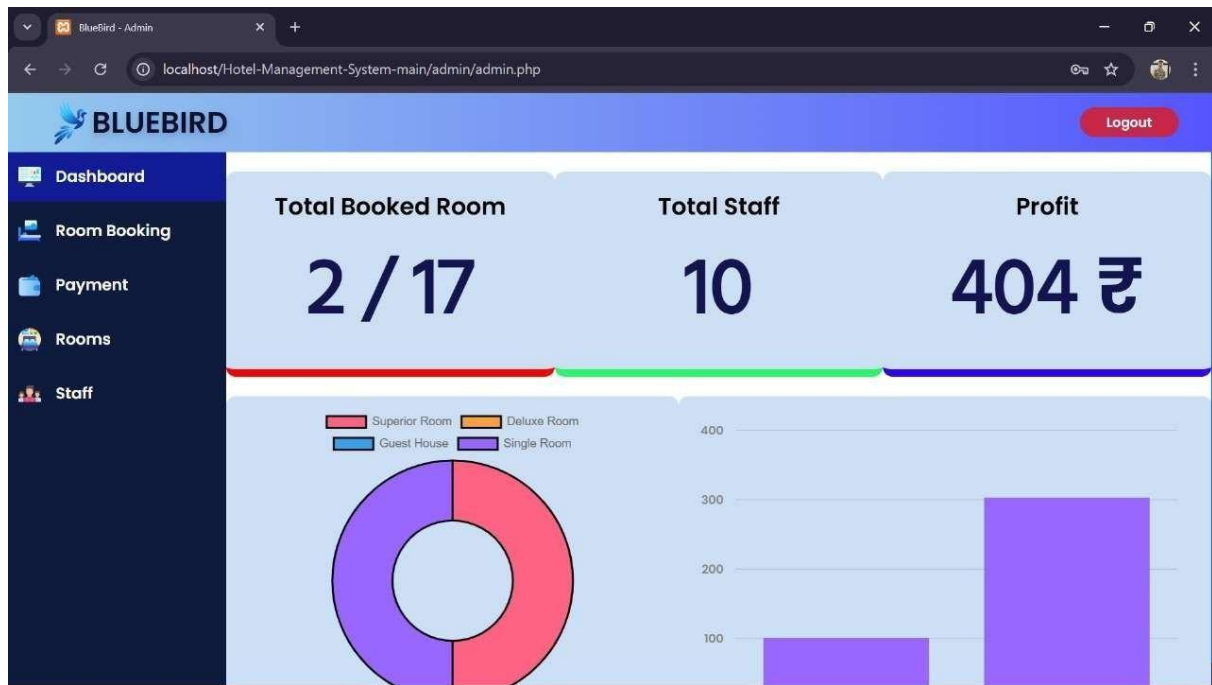
Don't have an account? [sign up](#)

HOME PAGE:



BOOKING PAGE:

A screenshot of the 'RESERVATION' form on the Hotel Blue Bird website. The form is divided into two main sections: 'Guest information' and 'Reservation information'. The 'Guest information' section includes input fields for 'Enter Full name', 'Enter Email', 'Select your country' (a dropdown menu), and 'Enter Phoneno'. The 'Reservation information' section includes dropdown menus for 'Type Of Room', 'Bedding Type', 'No of Room', and 'Meal'. It also features 'Check-In' and 'Check-Out' date pickers, both with the format 'dd-mm-yyyy'. A green 'Submit' button is located at the bottom center of the form. The browser's address bar and taskbar are also visible.

STAFF DASHBOARD:**PAYMENT :**

BLUEBIRD Logout

Dashboard
Room Booking
Payment
Rooms
Staff

search...

Id	Name	Room Type	Bed Type	Check In	Check Out	No of Day	No of Room	Meal Type	Room Rent	Bed Rent	Meals	Total Bill	Action
41	Tushar Pankhaniya	Single Room	Single	2022-11-09	2022-11-10	1	1	Room only	1000.00	10.00	0.00	1010.00	Print Delete
51	kishan	Superior Room	Single	2025-02-20	2025-02-21	1	1	Room only	3000.00	30.00	0.00	3030.00	Print Delete

CHAPTER-8

TEST-CASES

Guest Registration

Test Case ID: TC_Guest_01

Title: Register a New Guest

Preconditions: The hotel system is up and running.

Test Steps:

1. Open the hotel management system.
2. Navigate to the guest registration form.
3. Enter valid guest details (name, email, phone number).
4. Submit the registration form.

Expected Result:

- A new guest profile is created.
- The system assigns a unique Guest_ID.
- A success message is displayed: "Registration successful."

Postconditions: The new guest profile is stored in the database.

Test Case 2: Search Available Rooms

Test Case ID: TC_Room_01

Title: Search for Available Rooms

Preconditions: The hotel system is up and running. The guest is registered.

Test Steps:

1. Open the hotel management system.
2. Login as a guest or visitor.
3. Enter check-in and check-out dates.
4. Apply room filters (e.g., room type, price range).
5. Click on the "Search" button.

Expected Result:

- The system displays a list of available rooms for the selected dates.
- Each room shows basic details (e.g., price per night, room type).

Postconditions: A list of available rooms is displayed for the guest.

Test Case 3: Make a Reservation

Test Case ID: TC_Reservation_01

Title: Make a Room Reservation

Preconditions: The guest has already registered. A room is available for the selected dates.

Test Steps:

1. Select an available room from the search results.
2. Click on "Book Now" or similar option.
3. Enter guest details (if not logged in already).
4. Confirm check-in and check-out dates.
5. Submit the reservation.

Expected Result:

- The room is reserved for the guest.
- The system updates the room status to "reserved".
- A reservation ID is generated and stored.
- The guest receives a confirmation message with reservation details.

Postconditions: A new reservation record is stored in the database.

Test Case 4: Check-in Guest

Test Case ID: TC_Checkin_01

Title: Guest Check-in Process

Preconditions: The guest has a confirmed reservation.

Test Steps:

1. Open the hotel management system.
2. Navigate to the check-in page.
3. Search for the guest by name, reservation ID, or guest ID.
4. Select the guest's reservation.
5. Click on "Check-in" button.

Expected Result:

- The guest is checked in successfully.
- The room status is updated to "occupied".
- A success message is displayed: "Check-in successful."

Postconditions: The reservation status is updated to "checked-in" in the system.

Test Case 5: Check-out Guest

Test Case ID: TC_Checkout_01

Title: Guest Check-out Process

Preconditions: The guest has already checked in and is staying in an assigned room.

Test Steps:

1. Open the hotel management system.
2. Navigate to the check-out page.
3. Search for the guest by name, reservation ID, or guest ID.
4. Select the guest's reservation.
5. Click on "Check-out" button.

Expected Result:

- The guest is checked out successfully.
- The room status is updated to "available".
- A final bill is calculated and displayed to the guest.
- A payment option is offered.

Postconditions: The reservation status is updated to "checked-out", and the room is marked as "available".

Test Case 6: Payment Process

Test Case ID: TC_Payment_01

Title: Process Payment for Guest Check-out

Preconditions: The guest has checked out, and a final bill is calculated.

Test Steps:

1. Open the hotel management system.
2. Navigate to the payment page for the guest.
3. Enter payment details (credit card, cash, etc.).
4. Submit the payment.

Expected Result:

- Payment is successfully processed.
- A payment confirmation message is displayed: "Payment successful".
- The payment record is stored in the database.

Postconditions: The payment is recorded in the system, and the guest's bill is marked as "paid".

Test Case 7: Request Room Maintenance

Test Case ID: TC_Maintenance_01

Title: Request Room Maintenance

Preconditions: The guest is staying in the hotel, and the room is in an occupied or available state.

Test Steps:

1. Open the hotel management system.
2. Navigate to the "Maintenance Request" page.
3. Select the room requiring maintenance.
4. Specify the type of maintenance required (e.g., cleaning, repair).
5. Submit the request.

Expected Result:

- The system marks the room as "under maintenance".
- A confirmation message is displayed: "Maintenance request submitted."

Postconditions: The room status is updated to "under maintenance" in the system.

Test Case 8: Update Room Status by Admin

Test Case ID: TC_Admin_RoomStatus_01

Title: Update Room Status (e.g., Occupied to Available)

Preconditions: The system is logged in by an admin. The room is either occupied or reserved.

Test Steps:

1. Open the hotel management system as an admin.
2. Navigate to the "Room Status" management section.
3. Select the room whose status needs to be updated.
4. Change the status from "occupied" to "available".
5. Submit the changes.

Expected Result:

- The room status is updated successfully.
- A success message is displayed: "Room status updated to available."

Postconditions: The room status is updated in the database.

Test Case 9: View Reservation History (Guest)

Test Case ID: TC_ReservationHistory_01

Title: View Guest Reservation History

Preconditions: The guest has made previous reservations.

Test Steps:

1. Open the hotel management system.
2. Log in as the guest (or use guest ID to view history).
3. Navigate to the "Reservation History" section.
4. View the past reservations list.

Expected Result:

- The system displays all previous reservations made by the guest, including room type, dates, and status.
- If no reservations exist, the system displays: "No reservations found".

Postconditions: The guest can view all their past reservations.

Test Case 10: Admin View Financial Report

Test Case ID: TC_Admin_FinancialReport_01

Title: Generate and View Financial Report

Preconditions: Admin is logged in and there are completed payments in the system.

Test Steps:

1. Open the hotel management system.
2. Navigate to the "Financial Reports" section.
3. Select the time period for the report (e.g., weekly, monthly).
4. Click "Generate Report".

Expected Result:

- The system generates the financial report showing total income, payment records, and any applicable taxes or discounts.
- A detailed breakdown is displayed with figures for income, expenses, and profits.

Postconditions: The admin can view the generated financial report.

CHAPTER-9

FUTURE SCOPE

Future Scope of Hotel Management System (HMS)

The **Hotel Management System (HMS)** is a continuously evolving domain, and there are numerous areas where future advancements can significantly improve operations, guest experience, and overall efficiency. Below are some key areas where the future scope of HMS can be expanded.

1. Artificial Intelligence (AI) Integration

a. Predictive Analytics for Demand Forecasting

AI and machine learning (ML) can analyze past guest data and trends to predict demand, optimize pricing, and forecast room occupancy rates.

- **Benefits:** Improved decision-making for room rates, better occupancy management, and reduced operational costs.

b. Virtual Concierge Services

AI-driven chatbots or virtual assistants could provide guests with personalized recommendations, concierge services, and real-time support during their stay.

- **Benefits:** 24/7 customer service, improved guest experience, and reduced human resource dependence.

c. Automated Customer Feedback Analysis

Natural language processing (NLP) tools could automatically process guest feedback, reviews, and survey responses to gain actionable insights into service quality and areas of improvement.

- **Benefits:** Real-time analysis, enhanced customer service, and proactive issue resolution.

2. Internet of Things (IoT) Integration

a. Smart Room Technology

Rooms can be equipped with IoT devices that allow guests to control lighting, temperature, entertainment systems, and even curtains using their smartphones or voice assistants (like Alexa or Google Home).

- **Benefits:** Enhanced guest comfort, energy efficiency, and the convenience of personalized room control.

b. Predictive Maintenance Using IoT

Sensors installed in hotel equipment (air conditioning, elevators, plumbing, etc.) can send alerts about potential issues before they escalate, helping maintenance teams act proactively.

- **Benefits:** Reduced downtime, fewer disruptions to guests, and cost-effective maintenance.

c. Smart Inventory Management

IoT devices can track and manage hotel inventory (e.g., linens, toiletries, minibar items) in real-time. This reduces waste and ensures that the hotel is always prepared to meet guest demands.

- **Benefits:** Streamlined operations, reduced operational costs, and minimized stockouts.

3. Mobile App Integration

a. Contactless Check-in/Check-out

Guests can use their smartphones to check in, access their rooms via digital keys, and check out without ever having to visit the front desk.

- Benefits: Improved convenience, enhanced guest safety (especially in post-pandemic times), and reduced operational strain on front desk staff.

b. In-Room Mobile Ordering and Service Requests

Guests can order room service, request housekeeping, or book amenities directly from their mobile phones.

- Benefits: Increased guest satisfaction, better service delivery, and upselling opportunities for the hotel.

c. Loyalty Programs and Personalization

Hotels can integrate personalized loyalty programs via their mobile apps. These programs could reward guests for frequent stays, offer discounts, and tailor services based on preferences and past behaviors.

- Benefits: Increased customer retention, personalized guest experience, and better customer insights.

4. Blockchain for Secure Transactions and Data Management

a. Secure Payment Systems

Blockchain technology can enhance security in payment systems by ensuring that transactions are tamper-proof and reducing fraud.

- Benefits: Increased transaction security, faster payment processing, and reduced payment fraud.

b. Transparent Booking and Reservation Management

Blockchain can ensure transparency in the booking process, enabling guests to verify the authenticity of their reservation and hotel availability.

- Benefits: Reduced booking fraud and improved trust between hotels and guests.

c. Digital Identity Verification

Blockchain can help securely store and verify guest identities, reducing the time spent on check-in and preventing identity fraud.

- Benefits: Faster check-ins, secure guest data management, and reduced chances of identity theft.

CONCLUSION

The **Hotel Management System (HMS)** plays a critical role in optimizing the operations of a hotel by integrating various functions, from guest booking to check-out, room management, and payment processing. As we've explored in this project, an effective HMS enhances operational efficiency, ensures seamless guest experiences, and helps hotels maintain profitability while offering personalized services to guests.

In the context of today's fast-paced, tech-driven world, the future of hotel management lies in embracing emerging technologies. With advancements in **AI, IoT, Blockchain, Cloud Computing**, and **Mobile Integration**, hotels can deliver innovative, personalized, and streamlined services, ensuring they stay ahead of the competition. The system's scalability, flexibility, and automation help minimize human error, reduce operational costs, and improve decision-making.

CHAPTER-10 REFERENCE

1. Ravichandran, R. (2020).

Hotel Management Information Systems: Concept and Application.

International Journal of Advanced Science and Technology, 29(4), 1254-1260.

Available at: <https://www.sci-researcher.org>

2. Sharma, A., & Rani, A. (2018).

The Role of Information Technology in Hotel Management Systems.

International Journal of Computer Applications, 179(12), 34-39.

Available at: <https://www.ijcaonline.org>

3. Vargas, P., & Prieto, M. (2021).

AI and Machine Learning Applications in Hotel Management.

Journal of Tourism Research and Hospitality, 45(6), 547-559.

Available at: <https://www.jtrh.com>

Apps:

1. Google
2. Chat Gpt(AI)
3. You tube



K.L.E SOCIETY'S C.B KORE POLYTECHNIC, CHIKODI
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING.
RUBRICS FOR ASSESSMENT OF INTERNSHIP

NAME: KISHAN NINGANAGAUDA MANGASULE

REG NO: 339CS22024

SUBJECT/CODE: INTERNSHIP/20CS615

DIMENSION	SCALE					STUDENT SCORE
	10	20	30	40	50	
	Beginner	Intermediate	Good	Advanced	Expert	
Organisation	Has not included relevant information	Has included few relevant information	Has included some relevant information	Has included many relevant information	Has included all relevant information	
Full fill Team roles& duties	Does not performanc e duties assigned	performance very little duties	performanc e particular duties	performanc e early all duties	performanc e all duties of assigned team roles	
Conclusion	Poor	Lesse Effective	Partialy Effective	Summarises but not exact	Most Effective	
Conversions	Frequent Error	More Error	Some Error	Occasional Error	No Error	
Average Marks						

Cohort Owner