

# **HOSPITAL MANAGEMENT SYSTEM**



## **A PROJECT REPORT**

*Submitted by*

**KARTHIKA R**

*in partial fulfillment of requirements for the award of the course*

**EGB1221 - DATABASE MANAGEMENT SYSTEM**

*in*

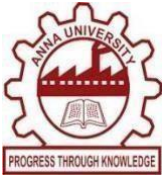
**ELECTRICAL AND ELECTRONICS ENGINEERING**

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY**

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

**SAMAYAPURAM – 621 112**

**JUNE - 2025**



# **HOSPITAL MANAGEMENT SYSTEM**



## **A PROJECT REPORT**

*Submitted by*

**KARTHIKA R (2303811710522028)**

*in partial fulfillment of requirements for the award of the course*

**EGB1221 - DATABASE MANAGEMENT SYSTEM**

*in*

**ELECTRICAL AND ELECTRONICS ENGINEERING**

**K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY**

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

**SAMAYAPURAM – 621 112**

**JUNE - 2025**

# **HOSPITAL MANAGEMENT SYSTEM**

## **K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)**

**SAMAYAPURAM – 621 112**

### **BONAFIDE CERTIFICATE**

Certified that this project report on “**HOSPITAL MANAGEMENT SYSTEM**” is the bonafide work of **KARTHIKA R (2303811710522028)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

#### **SIGNATURE**

Dr. A.T.SANKARA SUBRAMANIAN, M.E.,phd.,  
**HEAD OF THE DEPARTMENT**

ASSOCIATE PROFESSOR

Department of EEE

K. Ramakrishnan College of Technology

(Autonomous)

Samayapuram—621112.

#### **SIGNATURE**

Ms. P. KARTHIKA, M.E.,  
**SUPERVISOR**

ASSISTANT PROFESSOR

Department of CSE

K. Ramakrishnan College of Technology

(Autonomous)

Samayapuram—621112.

Submitted for the viva-voce examination held on .....

INTERNAL EXAMINER

EXTERNAL EXAMINER

## **DECLARATION**

I declare that the project report on **“HOSPITAL MANAGEMENT SYSTEM”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **EGB1221 – DATABASE MANAGEMENT SYSTEM**.

**Signature**

---

KARTHIKA R

Place: Samayapuram

Date: 04.06.2025

## ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution “**K.Ramakrishnan College of Technology (Autonomous)**”, for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN, B.E.**, for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.**, for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.**, Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A.T. SANKARA SUBRAMANIAN, M.E., Ph.D.**, Head of the department, **ELECTRICAL AND ELECTRONICS ENGINEERING** for providing his encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Ms. P. KARTHIKA, M.E.**, Department of **COMPUTER SCIENCE AND ENGINEERING**, for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

## **VISION OF THE INSTITUTION**

To serve the society by offering top-notch technical education on par with global standards.

## **MISSION OF THE INSTITUTION**

- M1: Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- M2: Be an institute with world class research facilities
- M3: Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

## **VISION OF DEPARTMENT**

To develop globally competent Electrical Engineers with expertise in education and cutting edge research technologies thereby contribute value to their career and society.

## **MISSION OF DEPARTMENT**

**M1: Knowledge:** To bestow quality education in Electrical and Electronics Engineering and prepare the students for career development and higher studies.

**M2: Skill:** To Excel in Contemporary core and Interdisciplinary areas with Prime Research Facilities and Industrial collaborations.

**M3: Attitude:** To augment student competency along with moral and ethical values through academics to serve the society

## **PROGRAM EDUCATIONAL OBJECTIVES**

### **PEO 01 – Knowledge**

To empower graduates with high standards of technical knowledge making them readily employable or well prepared for pursuing higher education to thrive in their career development.

### **PEO 02 – Skills**

To produce graduates with interdisciplinary skills who can contribute meaningfully to cutting- edge research and innovation in emerging areas, thereby making a significant impact on their respective industries and the global community.

### **PEO 03 - Attitude**

To develop highly competent professionals who are also committed to uphold the highest standards of ethical conduct and moral consciousness in the society.

## **PROGRAM SPECIFIC OUTCOMES (PSOs)**

### **PSO 01**

To demonstrate proficiency in Core Electrical areas such as Electrical Circuits, Electromagnetic fields, Control Engineering, Instrumentation, Electrical Drives, Power System and Power Electronics to Solve Practical Engineering Problems.

### **PSO 02**

To analyze, design and develop Electronic circuits and systems through insights acquired in Integrated circuits, Embedded systems, Analog and Digital Electronics.

### **PSO 03**

To apply technical competency, management and interdisciplinary skills for developing socially acceptable solutions to complex emerging area problems and its applications.

## PROGRAM OUTCOMES (POs)

Engineering students will be able to:

- 1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the



engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

## **ABSTRACT**

The Hospital Management System is a web-based application developed to enhance the efficiency, accuracy, and organization of hospital operations. It replaces traditional manual processes with a streamlined digital platform that automates essential tasks such as patient registration, doctor scheduling, medical records management, billing, and inventory tracking. The system aims to minimize human error, reduce administrative workload, and provide fast and reliable access to critical data, enabling healthcare professionals to deliver better patient care.

Developed using PHP and MySQL for backend functionality and HTML, CSS, and JavaScript for the frontend, the system offers an intuitive, user-friendly interface for seamless interaction. It consists of integrated modules that allow users to add, edit, delete, and view data related to patients, staff, appointments, medical history, and billing. A centralized relational database ensures real-time updates and synchronized data across departments. Additional features such as invoice generation, reporting tools, and secure user authentication further strengthen the system's utility and reliability.

This project is especially useful for small to mid-sized hospitals and clinics seeking to digitize their administrative and clinical workflows. By automating routine processes, the Hospital Management System supports improved patient management, efficient staff coordination, and timely service delivery. Ultimately, it enhances operational performance, improves decision-making, and provides a scalable solution for modern healthcare environments.

### ABSTRACT WITH POs AND PSOs MAPPING

#### CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
The Hospital Management System is a web-based application developed to streamline core hospital functions such as patient registration, appointment scheduling, doctor and staff management, billing, and medical records. Created using PHP, MySQL, HTML, CSS, and JavaScript, it transitions hospitals from manual paperwork to a digital interface, enhancing efficiency and reducing administrative errors. The system features modules for admitting and discharging patients, assigning doctors, updating medical history, and handling payments, all integrated into a centralized database. With secure user authentication and a responsive design, it serves as a comprehensive and reliable tool for small to medium-sized hospitals aiming to improve operational workflow and patient care services.	<b>PO1 -3</b> <b>PO2 -3</b> <b>PO3 -3</b> <b>PO4 -3</b> <b>PO5 -3</b> <b>PO6 -3</b> <b>PO7 -3</b> <b>PO8 -3</b> <b>PO9 -3</b> <b>PO10 -3</b> <b>PO11-3</b> <b>PO12 -3</b>	<b>PSO1 -3</b> <b>PSO2 -3</b> <b>PSO3 -3</b>

Note:1-Low,2-Medium,3-High

## TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>ABSTRACT</b>	Xi
	<b>LIST OF FIGURES</b>	Xiv
	<b>LIST OF ABBEVIATIONS</b>	xv
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 Objective	1
	1.2 Overview	1
	1.3 SQL and Database concepts	2
	1.3.1 Database Concepts	2
	1.3.2 Normalization	2
	1.3.3 Structured Query Language	2
	1.3.4 Advanced SQL Operations	2
	1.3.5 SQL and Database concept used in project	3
<b>2</b>	<b>PROJECT METHODOLOGY</b>	
	2.1 Proposed Work	4
	2.2 Block Diagram	5
<b>3</b>	<b>MODULE DESCRIPTION</b>	
	3.1 Patient Management Module	6
	3.2 Staff & Doctor Management Module	7
	3.3 Appointment & Scheduling Module	8
	3.4 Billing & Payment Module	9

	3.5 Medical Records & Reporting Module	10
<b>4</b>	<b>CONCLUSION &amp; FUTURE SCOPE</b>	
	4.1 Conclusion	11
	4.2 Future Scope	11
<b>5</b>	<b>APPENDIX A SOURCE CODE</b>	12
	<b>APPENDIX B SCREENSHORTS</b>	25
	<b>REFERENCES</b>	28

## LIST OF FIGURES

FIGURE.NO	TITLE	PAGE.NO
2.2	BLOCK DIAGRAM	5
5.1	HOME PAGE	25
5.2	ADMIN DASHBOARD	25
5.3	DOCTOR SCHEDULE	26
5.4	PATIENT REGISTRATION	26
5.5	BILLING & PAYMENT	27

## **LIST OF ABBREVIATIONS**

**DB - DATABASE**

**SQL - STRUCTURED QUERY LANGUAGE**

**N - NORMALIZATION**

# **CHAPTER 1**

## **INTRODUCTION**

### **1.1. Objective**

The objective of the Hospital Management System is to digitize and streamline core administrative and medical operations within a hospital setting. This system replaces traditional paper-based methods with a comprehensive web-based solution that efficiently handles patient registration, appointment scheduling, medical records, doctor and staff management, and billing. It enables healthcare staff to carry out critical tasks such as adding, editing, and retrieving data through an intuitive user interface. The platform ensures accurate data handling by utilizing a structured relational database model with primary and foreign keys to maintain integrity and relational consistency. Real-time data access allows for better coordination between departments, and automated features help track appointments, room availability, and billing information. The system also incorporates secure user authentication and role-based access control to protect sensitive health and personal data. With features such as reports on admissions, discharges, and financials, the platform aids in strategic planning and operational efficiency. Overall, the system is designed to reduce administrative burden, enhance workflow, and improve patient care— particularly beneficial for small to mid-sized hospitals looking to adopt a digital transformation strategy.

### **1.2 Overview**

The Hospital Management System is a web-based application developed to automate hospital operations and improve service delivery. It helps manage patient information, appointments, doctor schedules, medical history, billing, and hospital staff data through a centralized digital interface. By reducing paperwork and manual errors, the system offers real-time access to critical information, enhancing efficiency in both clinical and administrative areas. Developed using PHP and MySQL for the backend and HTML, CSS, and JavaScript for the frontend, it ensures a smooth and interactive user experience. The system consists of modules for patient admission, discharge, scheduling, staff allocation, and payment processing. All modules are interconnected through a relational database that ensures synchronized data updates and consistency. Role-based login access secures the platform, allowing only authorized personnel to perform specific functions. Other features include bed and room availability tracking, case history management, and billing summaries. These functionalities collectively support seamless hospital administration, improved patient monitoring, and data-driven decision-making.



## **1.3 SQL and Database Concepts**

### **1.3.1 DATABASE CONCEPTS:**

A database is a systematically organized collection of digital data, managed through a Database Management System (DBMS), which allows for efficient storage, access, and manipulation of structured information. In the context of hospital management, relational databases are typically employed. These databases organize data into related tables, each comprising rows (records) and columns (fields). Unique identification of each record is ensured using primary keys, while foreign keys establish logical links between different tables—for instance, connecting a patient record with a corresponding doctor or appointment entry.

### **1.3.2 NORMALIZATION:**

Normalization is a database design process that reduces data redundancy and dependency by dividing larger tables into smaller, related ones. This helps to organize data efficiently and prevent issues like inconsistent data entries and update anomalies. For the Hospital Management System, normalization ensures that patient records, staff information, billing data, and appointments are stored in a structured and optimized manner, improving database performance and maintainability.

### **1.3.3 STRUCTURED QUERY LANGUAGE:**

Structured Query Language (SQL) is the standard language used to communicate with and manipulate relational databases. SQL allows for defining database structure (using CREATE and ALTER), inserting and modifying data (using INSERT, UPDATE, DELETE), and querying data (using SELECT). These operations are essential for managing a hospital's diverse datasets, from patient logs and treatment histories to payment records and staff schedules.

### **1.3.4 ADVANCED SQL OPERATIONS:**

Advanced SQL techniques include the use of JOINS to fetch related data from multiple tables, such as retrieving all appointments associated with a specific patient. SQL also includes aggregate functions like COUNT(), AVG(), SUM(), and MAX(), which are useful for generating hospital statistics such as patient count, average billing, and occupancy rates. Constraints like NOT NULL, UNIQUE, and CHECK ensure the validity and consistency of data, such as ensuring that patient IDs are unique or that billing values fall within a logical range.

### **1.3.5 SQL and DATABASE Concept used in project:**

In the Hospital Management System, SQL and database concepts are vital for efficient and secure handling of large volumes of sensitive healthcare data. The system's relational database includes tables such as Patients, Doctors, Appointments, Rooms, Bills, and Staff. Each table has a unique primary key (e.g., `patient_id`, `doctor_id`) to identify records, while foreign keys help establish relationships—such as linking appointments with doctors and patients.

Normalization techniques are applied to reduce redundancy—for example, storing doctor details separately from patient data to avoid repetitive entries. SQL commands like `SELECT`, `INSERT`, `UPDATE`, and `DELETE` are used to manage data transactions, while `JOIN` queries are used to generate reports that combine information across different modules.

Advanced SQL functions such as `COUNT()` and `SUM()` are utilized in administrative reports to provide insights into patient visits, financial summaries, and room usage. Constraints ensure that critical fields (like patient names or bill amounts) are entered accurately. These database principles collectively ensure the system is robust, maintainable, and capable of delivering secure, efficient hospital operations.

## CHAPTER 2

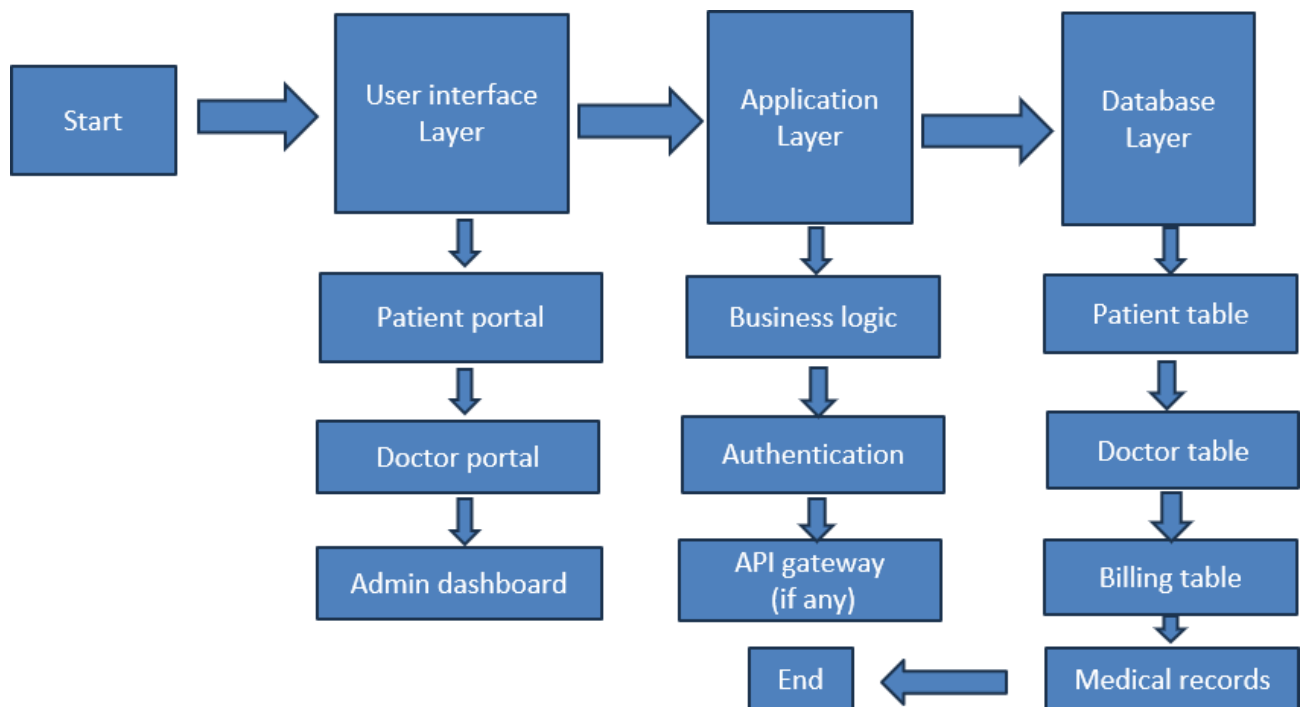
### PROJECT METHODOLOGY

#### 2.1 Proposed Work

The proposed Hospital Management System aims to develop a secure, scalable, and user-friendly web application to automate hospital operations. It will manage core functions such as patient registration, appointment scheduling, doctor and staff assignments, billing, and electronic medical records. Built using PHP and MySQL for the backend and HTML, CSS, and JavaScript for the frontend, the system offers an intuitive interface for smooth use across departments. A relational database model will organize data for efficient retrieval and management. Key features include modules to add, edit, view, and delete records related to patients, staff, appointments, and billing. SQL queries will manage data operations, and normalization will maintain data consistency and reduce redundancy. Role-based access and secure login will protect sensitive information by limiting access to authorized users. All modules will be integrated for real-time updates and smooth data flow. Reporting tools will generate summaries for admissions, discharges, and billing. Alerts for appointments, overdue payments, and room availability will improve task management and service efficiency.

- ☐ **Patient Management** – Digital storage and retrieval of patient records, history, and treatment plans.
- ☐ **Appointment Scheduling** – Online booking and automated scheduling for patients and doctors.
- ☐ **Billing & Insurance** – Integration with payment systems and insurance processing.
- ☐ **Inventory Management** – Tracking medicines, medical equipment, and stock levels.
- ☐ **Doctor & Staff Management** – Work schedules, payroll, and performance tracking.
- ☐ **Reports & Analytics** – Insights on hospital performance and patient trends.
- ☐ **Security & Compliance** – Ensuring data privacy, encryption, and compliance with health regulations.
- ☐ **Telemedicine & EHR Integration** – Remote consultations and seamless electronic health records (EHR) accessibility.

## 2.2Block Diagram



## **CHAPTER 3**

### **MODULE DESCRIPTION**

#### **3.1 Patient Management**

The Patient Management Module is a central component designed to handle all patient-related data and interactions efficiently. It allows for the registration of new patients, maintaining records such as personal information, medical history, and treatment details. This module ensures quick retrieval of patient data, enabling timely diagnosis and care. It supports appointment scheduling, visit tracking, and medical record updates. With secure data storage and structured profiles, it improves patient experience and streamlines communication between departments. Ultimately, it contributes to better healthcare delivery, continuity of care, and operational efficiency.

The Module in a Hospital Management System efficiently handles patient records, medical history, and treatment plans. It ensures seamless registration, storing details like demographics, allergies, prescriptions, and prior visits. This system minimizes paperwork, reducing errors and improving accessibility for doctors and staff.

Additionally, it integrates appointment scheduling, enabling patients to book consultations online and track upcoming visits. With features like automated reminders and digital reports, it enhances patient care while optimizing hospital workflow. Secure access to medical records ensures compliance with healthcare regulations and data privacy standards.

It also facilitates real-time access to patient information, ensuring healthcare providers can make informed decisions quickly. It streamlines communication between departments, allowing seamless coordination in treatment plans, lab results, and medication prescriptions. The system also supports patient feedback and engagement, enabling hospitals to improve their services based on real-time insights. With automation and digital records, it enhances efficiency, reduces administrative burden, and improves overall patient care.

### **3.2 Staff & Doctor Management**

The Staff & Doctor Management Module handles records related to medical and administrative personnel. It enables hospitals to maintain up-to-date profiles, schedules, specializations, and availability of doctors and staff. This module supports assigning duties, tracking attendance, and managing leaves, ensuring proper workforce allocation. It also facilitates communication between staff members and management for better coordination. The system allows performance tracking and document management, making it easier for administrators to oversee human resources. By improving organization and transparency, this module enhances service quality and staff productivity.

The management is crucial for ensuring smooth hospital operations by handling workforce scheduling, payroll, and performance management. It enables hospitals to efficiently assign shifts, manage leave requests, and optimize staffing levels based on patient demand. By automating administrative processes, it reduces workload and ensures that healthcare providers focus on delivering quality care. Additionally, it integrates digital attendance tracking, staff credentials, and compliance monitoring to maintain operational efficiency.

This module also plays a vital role in facilitating collaboration between doctors, nurses, and administrative teams. It provides a centralized platform for communication, allowing real-time updates on patient conditions, test results, and medical consultations. Doctors can access medical records instantly, ensuring seamless coordination and informed decision-making for treatments. With integrated messaging and task management features, hospital staff can stay organized, reducing delays and miscommunication.

Beyond workforce management, this system enhances hospital transparency and accountability. Hospitals can track employee performance, generate reports on efficiency, and monitor adherence to medical protocols. Additionally, it enables training and professional development by keeping records of certifications and workshops attended by healthcare providers. By streamlining operations, the module improves patient experiences, minimizes errors, and strengthens the overall functionality of the hospital system.

### 3.3 Appointment & Scheduling

The Appointment & Scheduling Module allows patients to book, modify, and cancel appointments with doctors through a user-friendly interface. It manages time slots, doctor availability, and department-specific queues to minimize waiting times. The module sends automated notifications and reminders to both patients and doctors, reducing missed appointments and improving punctuality. It also allows for emergency or walk-in scheduling. With an organized calendar view, it aids receptionists and administrators in efficient planning and resource management. This module improves patient satisfaction and optimizes hospital workflow.

The a Hospital Management System (HMS) is designed to efficiently manage patient bookings and streamline the scheduling of medical services. It provides an organized platform for both patients and hospital staff to schedule consultations, tests, and follow-up visits. By automating the appointment process, it reduces waiting times, eliminates scheduling conflicts, and improves the overall flow of patients within the hospital. This module is especially beneficial in handling large volumes of patients in a systematic and error-free manner.

The system allows patients to book appointments through various channels, including online portals, mobile apps, or hospital reception desks. It displays real-time availability of doctors, consultation rooms, and medical equipment, ensuring accurate scheduling. The module supports different types of appointments, such as general check-ups, specialist consultations, and diagnostic services, and it can also handle recurring appointments and emergency slots. Integration with the hospital's electronic health records (EHR) ensures that doctors have immediate access to a patient's medical history during appointments.

To enhance communication and reduce missed visits, the module includes automated reminders via SMS, email, or app notifications. It also provides hospital administrators with analytical tools to monitor patient flow, doctor utilization, and appointment trends. Overall, the Appointments and Scheduling Module plays a critical role in improving operational efficiency, enhancing patient satisfaction, and ensuring better resource management within the hospital environment.

### **3.4 Billing & Payment**

The Billing & Payment Module manages financial operations including invoice generation, payment processing, and account summaries. It tracks services rendered, calculates charges, applies taxes or discounts, and supports multiple payment options such as cash, card, and insurance claims. The system ensures billing accuracy and transparency with real-time updates and printed or digital receipts. It also provides financial reports, outstanding balances, and daily revenue summaries. This module reduces billing errors, enhances compliance, and supports efficient hospital revenue management.

The vital component that manages all financial transactions related to patient care. It automates the process of generating bills for outpatient, inpatient, emergency, and diagnostic services, ensuring accurate and transparent billing. This module calculates charges based on consultations, treatments, medications, laboratory tests, room stays, surgeries, and other hospital services. By reducing manual work, it minimizes errors and improves the efficiency of the hospital's finance operations.

This module is typically integrated with other systems such as electronic health records (EHR), pharmacy, diagnostics, and appointments, allowing it to pull real-time data to generate itemized and up-to-date invoices. It supports various billing types including cash, credit, insurance-based, and corporate billing. Additionally, it can manage advance payments, partial payments, refunds, and discounts based on hospital policies. The system also supports multiple currencies and tax configurations, making it adaptable for hospitals of all sizes and locations.

The payment section of the module enables secure processing through various methods such as cash, card, UPI, net banking, or mobile wallets. It generates payment receipts automatically and maintains a detailed transaction history for every patient, which can be accessed for audits and financial reporting. The module also provides dashboards and reports that help hospital management monitor revenue, track outstanding payments, and improve financial decision-making. In summary, the Billing and Payment Module ensures smooth, accurate, and transparent financial operations in a hospital setting.



### **3.5 Medical Records & Reporting**

This module maintains detailed medical records, including diagnostic reports, prescriptions, lab results, and treatment plans. It supports uploading and accessing documents for easy reference by healthcare professionals. The system enables longitudinal tracking of a patient's medical journey, ensuring informed clinical decisions. Custom reports can be generated for individual patients, departments, or overall hospital performance. Integration with diagnostic equipment and labs allows for real-time report updates. This enhances data accuracy, aids in research, and supports better patient outcomes.

The responsible for securely storing, managing, and retrieving patient health information. This module maintains a comprehensive electronic medical record (EMR) for each patient, which includes personal details, medical history, diagnosis, treatment plans, lab results, prescriptions, surgical notes, and discharge summaries. By digitizing records, it eliminates the need for physical paperwork, reduces the risk of data loss, and ensures quick access to patient information by authorized healthcare providers.

The module allows doctors and medical staff to update patient records in real time, enhancing collaboration and continuity of care across departments. It supports integration with diagnostic equipment and laboratory modules to automatically receive and store test results, making the clinical workflow more efficient. Advanced search and filtering options enable quick retrieval of past records, which is crucial during emergencies or follow-up treatments. Additionally, role-based access control ensures that sensitive patient information is protected and only accessible to authorized personnel.

On the reporting side, the module can generate a wide range of clinical, operational, and statistical reports. These reports help hospital administrators and medical staff analyze trends in patient visits, treatment outcomes, disease patterns, and resource utilization. Reports can also be customized to meet regulatory and accreditation requirements. Overall, the Medical Records and Reporting Module not only improves patient care and safety but also supports informed decision-making and compliance with healthcare standards.

## **CHAPTER 4**

### **CONCLUSION AND FUTURE ENHANCEMENT**

#### **4.1 CONCLUSION:**

In conclusion, the Hospital Management System offers a comprehensive digital solution to streamline the administrative and clinical workflows within a healthcare facility. Through its well-integrated modules—Patient Management, Staff & Doctor Management, Appointment Scheduling, Billing & Payment, and Medical Records—the system ensures improved efficiency, accuracy, and service quality. Each module contributes to automating routine hospital tasks, minimizing manual errors, and enhancing coordination among departments.

The system provides a centralized platform to manage hospital operations, delivering real-time access to patient and staff data. It supports informed decision-making, ensures compliance with medical standards, and improves patient care by enabling faster service delivery. From a financial perspective, the billing and reporting features aid in maintaining transparency, revenue tracking, and cost control.

By modernizing hospital operations, this system benefits administrators, healthcare professionals, and patients alike. It strengthens the overall infrastructure and helps create a data-driven, patient-focused healthcare environment. Ultimately, the system is a scalable and reliable tool for improving the quality in service.

#### **4.2 FUTURE ENHANCEMENT:**

Future improvements to the Hospital Management System could involve incorporating Artificial Intelligence (AI) and Machine Learning (ML) for predictive analysis and clinical decision support. These tools can assist in diagnosing diseases, forecasting patient inflow, and identifying at-risk patients. AI can also automate administrative tasks such as appointment scheduling and resource allocation.

Another major enhancement is the development of a mobile application that enables patients to book appointments, access health records, receive medication reminders, and consult doctors via telehealth features. Cloud integration can support data synchronization across multiple hospital branches, enabling scalability and disaster recovery.

Integration with national health databases and insurance platforms could simplify claim processing and improve compliance. Multi-language support and features for users with disabilities will ensure inclusivity. These advancements will make the system more intelligent, accessible, and adaptable to future healthcare.

## APPENDIX A

### 1. SQL:

```
CREATE DATABASE IF NOT EXISTS hospital_db;
USE hospital_db;
CREATE TABLE IF NOT EXISTS doctors (
    Id INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Specialization VARCHAR(100),
    Phone VARCHAR(20),
    Email VARCHAR(100)
);
CREATE TABLE IF NOT EXISTS patients (
    Id INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Age INT,
    Gender VARCHAR(10),
    Phone VARCHAR(20),
    Email VARCHAR(100)
);
CREATE TABLE IF NOT EXISTS appointments (
    Id INT AUTO_INCREMENT PRIMARY KEY,
    Patient_name VARCHAR(100) NOT NULL,
    Doctor_name VARCHAR(100) NOT NULL,
    Date DATE NOT NULL,
    Time TIME NOT NULL,
    Reason VARCHAR(255)
);
```

### 2. Python:

```
from flask import Flask, render_template, request, redirect
import mysql.connector

app = Flask(__name__)

# Connect to MySQL
db = mysql.connector.connect(
    host="localhost",
    user="your_mysql_username",
    password="your_mysql_password",
    database="hospital_db"
)
```

```

cursor = db.cursor(dictionary=True)
#HomePage
@app.route('/')

def index():
    return render_template("index.html")

# Doctors Page
@app.route('/doctors')
def doctors():
    cursor.execute("SELECT * FROM doctors")
    doctors = cursor.fetchall()
    return render_template("doctors.html", doctors=doctors)

@app.route('/add_doctor', methods=['POST'])
def add_doctor():
    name = request.form['name']
    specialization = request.form['specialization']
    phone = request.form['phone']
    email = request.form['email']
    cursor.execute("INSERT INTO doctors (name, specialization, phone, email) VALUES (%s, %s, %s, %s)",
                   (name, specialization, phone, email))
    db.commit()
    return redirect('/doctors')

# Patients Page
@app.route('/patients')
def patients():
    cursor.execute("SELECT * FROM patients")
    patients = cursor.fetchall()
    return render_template("patients.html", patients=patients)

@app.route('/add_patient', methods=['POST'])
def add_patient():
    name = request.form['name']
    age = request.form['age']
    gender = request.form['gender']
    phone = request.form['phone']
    email = request.form['email']
    cursor.execute("INSERT INTO patients (name, age, gender, phone, email) VALUES (%s, %s, %s, %s, %s)",
                   (name, age, gender, phone, email))
    db.commit()
    return redirect('/patients')

# Appointments Page
@app.route('/appointments')

```

```
def appointments():
```

```
    cursor.execute("SELECT * FROM appointments")
    appointments = cursor.fetchall()
    return render_template("appointments.html", appointments=appointments)
```

```
@app.route('/add_appointment', methods=['POST'])
```

```
def add_appointment():
```

```
    patient_name = request.form['patient_name']
    doctor_name = request.form['doctor_name']
    date = request.form['date']
    time = request.form['time']
    reason = request.form['reason']
```

```
    cursor.execute("INSERT INTO appointments (patient_name, doctor_name, date, time, reason) VALUES
    (%s, %s, %s, %s, %s)",
                    (patient_name, doctor_name, date, time, reason))
    db.commit()
    return redirect('/appointments')
```

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

### 3.Html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Hospital Management System</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body class="bg">
    <div class="container">
        <h1>Welcome to Hospital Management System</h1>
        <div class="nav-links">
            <a href="/doctors">Doctors</a>
            <a href="/patients">Patients</a>
            <a href="/appointments">Appointments</a>
        </div>
    </div>
</body>
</html>
```

#### 4.Html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Doctors – Hospital Management</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body class="bg">
  <div class="container">
    <h1>Doctors</h1>
    <table>
      <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Specialization</th>
        <th>Phone</th>
        <th>Email</th>
      </tr>
      {% for doc in doctors %}
      <tr>
        <td>{{ doc.id }}</td>
        <td>{{ doc.name }}</td>
        <td>{{ doc.specialization }}</td>
        <td>{{ doc.phone }}</td>
        <td>{{ doc.email }}</td>
      </tr>
      {% endfor %}
    </table>

    <h2>Add Doctor</h2>
    <form method="POST" action="/add_doctor">
      <input type="text" name="name" placeholder="Name" required>
      <input type="text" name="specialization" placeholder="Specialization">
      <input type="text" name="phone" placeholder="Phone">
      <input type="email" name="email" placeholder="Email">
```

```

        <button type="submit">Add</button>
    </form>
    <a href="/" class="back-link">Back to Home</a>
</div>
</body>
</html>

```

## 5.Html:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Patients – Hospital Management</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body class="bg">
    <div class="container">
        <h1>Patients</h1>
        <table>
            <tr>
                <th>ID</th>
                <th>Name</th>
                <th>Age</th>
                <th>Gender</th>
                <th>Phone</th>
                <th>Email</th>
            <tr>
                <td>{{ pat.id }}</td>
                <td>{{ pat.name }}</td>
                <td>{{ pat.age }}</td>
                <td>{{ pat.gender }}</td>
                <td>{{ pat.phone }}</td>
                <td>{{ pat.email }}</td>
            </tr>
        <tr>
            <td>{{ pat.id }}</td>
            <td>{{ pat.name }}</td>
            <td>{{ pat.age }}</td>
            <td>{{ pat.gender }}</td>
            <td>{{ pat.phone }}</td>
            <td>{{ pat.email }}</td>
        </tr>
    </div>
    <h2>Add Patient</h2>
    <form method="POST" action="/add_patient">
        <input type="text" name="name" placeholder="Name" required>
        <input type="number" name="age" placeholder="Age">
        <input type="text" name="gender" placeholder="Gender">
        <input type="text" name="phone" placeholder="Phone">
        <input type="email" name="email" placeholder="Email">
    </form>

```

```
<button type="submit">Add</button>
</form>

<a href="/" class="back-link">Back to Home</a>
</div>
</body>
</html>
```

## 6.Add\_customer:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Add Customer</title>
<style>
body {
margin: 0;
font-family: Arial, sans-serif;
background-image:
url('images/customer_background.jpg'); /* your image path
*/
background-size: cover;
background-position: center;
height: 100vh;
display: flex;
justify-content: center;
align-items: center;
color: white;
}
```



```
.form-container {  
  background-color: rgba(0, 0, 0, 0.7); /* dark transparent  
background */  
  padding: 30px;  
  border-radius: 10px;  
  width: 400px;  
}
```

```
.form-container h2 {  
  text-align: center;  
  margin-bottom: 20px;  
}
```

```
.form-container input {  
  width:  
  100%;padding:10px;
```

```
margin: 10px 0;  
  border-radius: 5px;  
  border: none;  
}
```

```
.form-container input[type="submit"] {  
  background-color: #28a745;  
  color: white;  
  cursor: pointer;  
}
```

```
.form-container input[type="submit"]:hover {  
  background-color: #218838;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="form-container">
```

```
<h2>Add Customer</h2>
```

```
<form action="save_customer.php" method="POST">
```

```
<label for="name">Customer Name:</label>
```

```

<input type="text" id="name" name="name" required>

<label for="phone">Phone:</label>
<input type="text" id="phone" name="phone" required>

<label for="address">Address:</label>
<input type="text" id="address" name="address"
required>

<input type="submit" value="Save Customer">
</form>
</div>

</body>

</html>

```

## 6.Html:

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Appointments – Hospital Management</title>
  <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body class="bg">
  <div class="container">
    <h1>Appointments</h1>
    <table>
      <tr>
        <th>ID</th>
        <th>Patient</th>
        <th>Doctor</th>
        <th>Date</th>

```

```

        <th>Time</th>
        <th>Reason</th>
    </tr>
    {% for app in appointments %}
    <tr>
        <td>{{ app.id }}</td>
        <td>{{ app.patient_name }}</td>
        <td>{{ app.doctor_name }}</td>
        <td>{{ app.date }}</td>
        <td>{{ app.time }}</td>
        <td>{{ app.reason }}</td>
    </tr>
    {% endfor %}
</table>
<h2>Add Appointment</h2>

```

```

<form method="POST" action="/add_appointment">
    <input type="text" name="patient_name" placeholder="Patient Name" required>
    <input type="text" name="doctor_name" placeholder="Doctor Name" required>
    <input type="date" name="date" required>
    <input type="time" name="time" required>
    <input type="text" name="reason" placeholder="Reason">
    <button type="submit">Add</button>
</form>

```

```

    <a href="/" class="back-link">Back to Home</a>
</div>
</body>
</html>

```

## 7.Css:

```
/* Background */
.bg {
  Background-image: url(../static/hospital_bg.jpg');
  Background-size: cover;
  Background-position: center;
  Background-repeat: no-repeat;
  Min-height: 100vh;
  Font-family: 'Segoe UI', sans-serif;
  Color: #fff;
}

/* Container Styling */
.container {
  Background-color: rgba(0, 0, 0, 0.6);
  Padding: 30px;
  Margin: 50px auto;
  Border-radius: 20px;
  Width: 85%;
  Max-width: 1000px;
  Box-shadow: 0 0 20px #222;
}

/* Headings */
H1, h2 {
  Text-align: center;
  Margin-bottom: 20px;
  Color: #fdd835;
}

/* Links */
.nav-links {
  Text-align: center;
  Margin: 20px 0;
}
.nav-links a {
  Margin: 0 15px;
  Padding: 10px 20px;
  Background-color: #43a047;
  Color: white;
  Text-decoration: none;
  Border-radius: 10px;
}
```

```
.nav-links a:hover {  
    Background-color: #66bb6a;  
}
```

```
.back-link {  
    Display: inline-block;  
    Margin-top: 30px;  
    Background-color: #039be5;  
    Padding: 10px 15px;  
    Color: white;  
    Text-decoration: none;  
    Border-radius: 8px;  
}  
.back-link:hover {  
    Background-color: #29b6f6;  
}
```

```
/* Form Styling */
```

```
Form {  
    Display: flex;  
    Flex-wrap: wrap;  
    Gap: 15px;  
    Justify-content: center;  
}  
Form input, form button {  
    Padding: 10px;  
    Border-radius: 5px;  
    Border: none;  
}  
Form input {  
    Width: 200px;  
}  
Form button {  
    Background-color: #e53935;  
    Color: white;  
    Cursor: pointer;  
}  
Form button:hover {  
    Background-color: #ef5350;  
}
```

```
/* Table Styling */
```

```
Table {  
    Width: 100%;  
    Border-collapse: collapse;  
    Background-color: white;
```

```

    Color: #333;
    Border-radius: 10px;
    Overflow: hidden;
}
Th, td {
    Padding: 12px;
    Text-align: center;
    Border-bottom: 1px solid #ccc;
}
Th {
    Background-color: #3f51b5;
    Color: white;
}
Tr:hover {
    Background-color: #f1f1f1;
}

```

## 8.Python:

From flask import Flask, render\_template, request, redirect  
 Import mysql.connector

```
App = Flask(__name__)
```

```
# Database connection
```

```
Db = mysql.connector.connect(
    Host="localhost",
    User="root",
    Password="your_password", # change this to your MySQL password
    Database="hospital_db"
)
Cursor = db.cursor(dictionary=True)
```

```
# Home page
```

```
@app.route("/")
Def home():
    Return render_template("index.html")
```

```
# Doctors page
```

```
@app.route("/doctors")
Def doctors():
```

```
Cursor.execute("SELECT * FROM doctors") Doctors
= cursor.fetchall()
    Return render_template("doctors.html", doctors=doctors)
```

```
@app.route("/add_doctor", methods=["POST"])
```

```
Def add_doctor():
    Name = request.form["name"]
```

```

Specialization = request.form["specialization"]
Phone = request.form["phone"]
Email = request.form["email"]
Cursor.execute("INSERT INTO doctors (name, specialization, phone, email) VALUES (%s, %s, %s, %s)",
               (name, specialization, phone, email))
Db.commit()
Return redirect("/doctors")

# Patients page
@app.route("/patients")
Def patients():
    Cursor.execute("SELECT * FROM patients")
    Patients = cursor.fetchall()
    Return render_template("patients.html", patients=patients)

@app.route("/add_patient", methods=["POST"])
Def add_patient():
    Name = request.form["name"]
    Age = request.form["age"]
    Gender = request.form["gender"]
    Phone = request.form["phone"]
    Email = request.form["email"]
    Cursor.execute("INSERT INTO patients (name, age, gender, phone, email) VALUES (%s, %s, %s, %s, %s)",
                  (name, age, gender, phone, email))
    Db.commit()
    Return redirect("/patients")

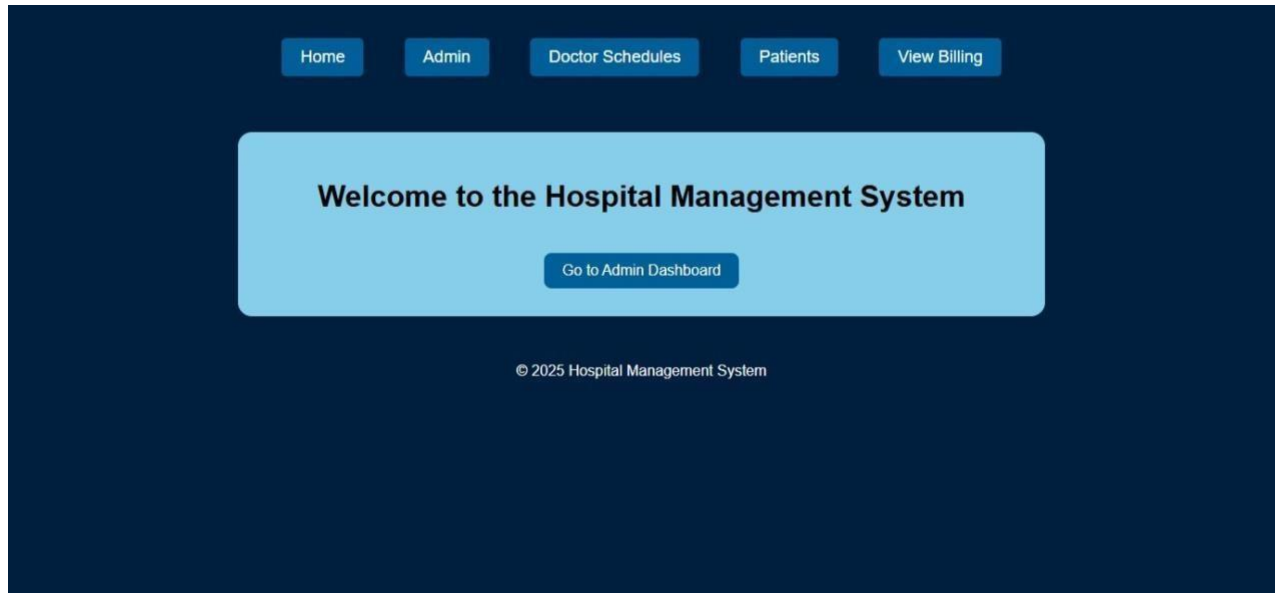
# Appointments page
@app.route("/appointments")
Def appointments():
    Cursor.execute("SELECT * FROM appointments")
    Appointments = cursor.fetchall()
    Return render_template("appointments.html", appointments=appointments)

@app.route("/add_appointment", methods=["POST"])
Def add_appointment():
    Patient_name = request.form["patient_name"]
    Doctor_name = request.form["doctor_name"]
    Date = request.form["date"]
    Time = request.form["time"]
    Reason = request.form["reason"]
    Cursor.execute("INSERT INTO appointments (patient_name, doctor_name, date, time, reason) VALUES (%s,
    %s, %s, %s, %s)",
    (patient_name, doctor_name, date, time, reason)) Db.commit()
    Return redirect("/appointments")

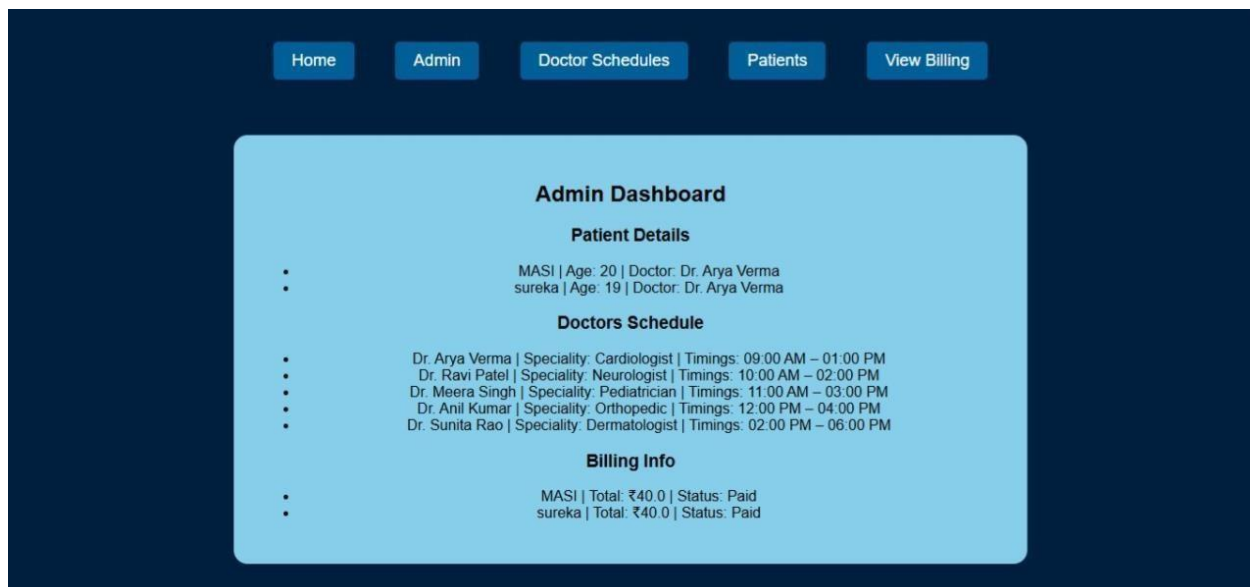
```

## APPENDIX - B

### 6 Home Page

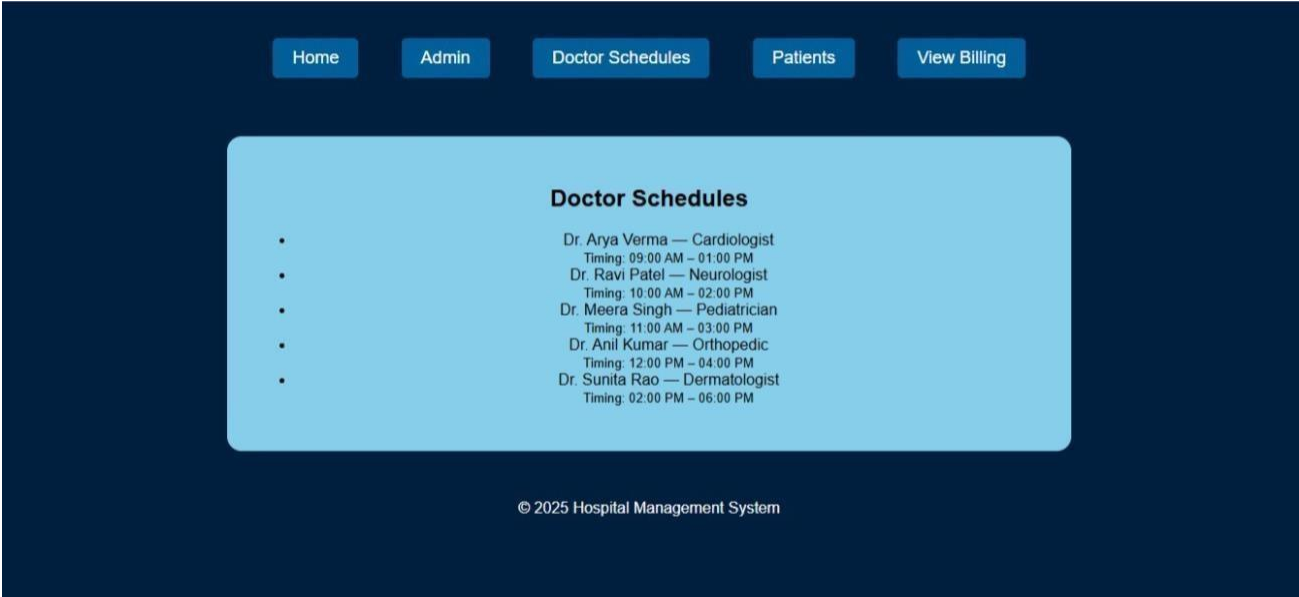


### 7 Admin Dashboard

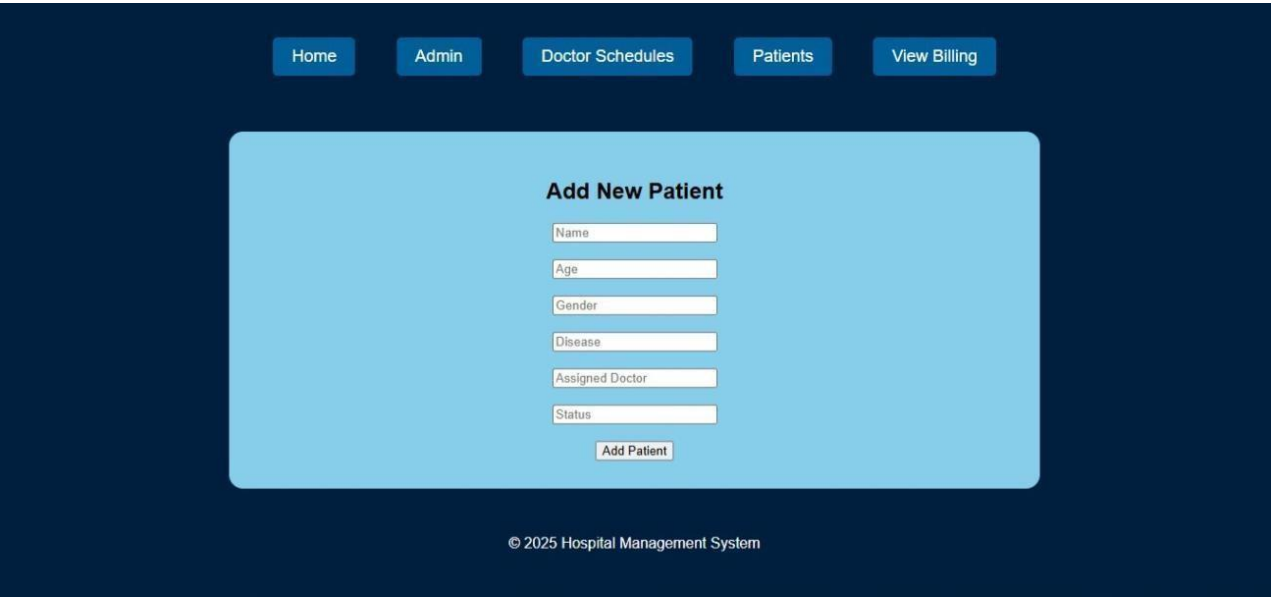




## 8 Doctor Schedule



## 4.Patient Registration



9 Billing & Payment

HomeAdminDoctor SchedulesPatientsView Billing

Add Billing Info

Patient Name

Room Charges

Consultation Fees

Medicine Charges

Lab Tests Charges

Payment Status (Paid/Unpaid)

Add Billing

© 2025 Hospital Management System

## REFERENCES:

1. Sharma, R., & Goyal, R. (2014). Hospital Administration and Human Resource Management  
This book provides foundational knowledge of hospital operations, staff coordination, and resource planning, which are critical components of any Hospital Management System.
2. Winter, A., Haux, R., Ammenwerth, E., & Brigl, B. (2011). Health Information Systems: Architectures and Strategies  
A globally recognized source that explains the architecture and design of healthcare information systems, offering essential theoretical background for HMS software structure.
3. Kumar, S., & Singh, P. (2016). Hospital Management System: Concepts and Applications  
Focuses on practical implementation and real-world scenarios in hospital automation, making it highly relevant for applied HMS development.
4. Patel, J., & Patel, B. (2013). Design and Implementation of Hospital Information Systems  
Provides insights into software development, database design, and workflow integration within hospitals key to building effective HMS platforms.
5. Ramesh, D., & Divya, P. (2018). Healthcare Information Management  
Covers data handling, patient record systems, and compliance with features like EMR and patient tracking.