# Create a Hospital Management System

**Develop a sophisticated Java program to manage various aspects of hospital or healthcare facility. This system should include modules for patient registration, appointment scheduling, electronic health records (EHR), billing and invoicing, inventory management for medical supplies and staff management.**

# Introduction

A Hospital Management System (HMS) is a digital solution designed to streamline hospital operations, improve patient care, and enhance administrative efficiency. It integrates various departments such as patient registration, appointment scheduling, doctor management, billing, pharmacy, laboratory, and medical records into a unified platform. HMS automates processes like patient admission, discharge, and treatment history while ensuring data security and accessibility. By reducing paperwork and optimizing workflows, the system enhances hospital productivity and patient experience. Advanced features like electronic health records (EHR), analytics, and real-time reporting help healthcare providers make informed decisions, ultimately improving the quality of medical services.

# Objective

The objective of the Hospital Management System (HMS) is to streamline and automate hospital operations, improving efficiency and patient care. It enables efficient patient registration, appointment scheduling, electronic health record (EHR) management, billing, inventory control, and staff administration. The system reduces paperwork, minimizes human errors, and enhances data security by ensuring role-based access. It provides a user-friendly interface for seamless interaction among patients, doctors, and hospital staff. Additionally, HMS integrates reporting and analytics to support decision-making. By optimizing workflows and resource management, this system enhances healthcare quality, improves hospital efficiency, and ensures better patient experiences.

# Modules

## 1. Patient Registration Module:

- This module allows the hospital to register new patients, store their personal details, contact information, and medical history.
- It assigns a unique patient ID for tracking medical records and enables quick retrieval of information for future visits, ensuring efficient hospital workflow and patient management.

## 2. Appointment Scheduling Module:

- This module facilitates booking, modifying, and cancelling appointments with doctors.
- It prevents scheduling conflicts by checking doctor availability and sending appointment reminders to patients.
- The system optimizes time management for both doctors and patients, reducing waiting times and ensuring a smooth appointment process.

## 3. Electronic Health Records (EHR) Module

- The EHR module securely stores and manages patient medical records, including diagnoses, prescriptions, lab reports, and treatment history.
- It ensures easy access to patient data for doctors, enhances collaboration between departments, and improves treatment accuracy while maintaining compliance with healthcare data security standards.

## 4. Billing and Invoicing Module

- This module automates the billing process, generating invoices for consultations, treatments, and medical services.
- It supports multiple payment methods, tracks pending payments, and integrates with insurance providers for claim processing.

- The system ensures accurate billing, minimizes errors, and provides financial transparency to both patients and hospitals.

### 5. Inventory Management Module

- The inventory module manages the hospital’s medical supplies, tracking stock levels, purchase orders, and usage.
- It prevents shortages by sending alerts for low stock and optimizes procurement processes. This ensures hospitals maintain a steady supply of essential medicines, equipment, and other medical necessities.

### 6. Staff Management Module

- This module handles staff details, including doctors, nurses, and administrative personnel.
- It manages work schedules, payroll, leave requests, and performance tracking.
- The system ensures efficient workforce management by automating administrative tasks, reducing manual work, and improving staff coordination within the hospital.

## Source code:

```
import java.util.*;

class Patient {

 String id, name, age, gender, contact, medicalHistory;

  Patient(String id, String name, String age, String gender, String contact, String medicalHistory) {

        this.id = id;

        this.name = name;

        this.age = age;

        this.gender = gender;

        this.contact = contact;

        this.medicalHistory = medicalHistory;

    }

     public String toString() {

        return "ID: " + id + ", Name: " + name + ", Age: " + age + ", Gender: " + gender +

                ", Contact: " + contact + ", Medical History: " + medicalHistory;

    }

}

class Appointment {

    String patientId, doctorName, date, time;

    Appointment(String patientId, String doctorName, String date, String time) {

        this.patientId = patientId;

        this.doctorName = doctorName;

        this.date = date;
        this.time = time;
    }
   public String toString() {
   return "Patient ID: " + patientId + ", Doctor: " + doctorName + ", Date: " + date + ", Time: 
" + time;
    }
}
class EHR {
    String patientId, diagnosis, treatment, prescription, labReports;
    EHR(String patientId, String diagnosis, String treatment, String prescription, String 
labReports) {
        this.patientId = patientId;
        this.diagnosis = diagnosis;
        this.treatment = treatment;
        this.prescription = prescription;
        this.labReports = labReports;
    }
   public String toString() {
        return "Patient ID: " + patientId + ", Diagnosis: " + diagnosis + ", Treatment: " + 
treatment +
               ", Prescription: " + prescription + ", Lab Reports: " + labReports;
    }
}
class Billing {
    String patientId;
    double amount;

    Billing(String patientId, double amount) {

        this.patientId = patientId;

        this.amount = amount;

    }

    public String toString() {

        return "Patient ID: " + patientId + ", Bill Amount: $" + amount;

    }

}

class Inventory {

    String itemName;

    int quantity;

    Inventory(String itemName, int quantity) {

        this.itemName = itemName;

        this.quantity = quantity;

    }

    public String toString() {

        return "Item: " + itemName + ", Quantity: " + quantity;

    }

}

class Staff {

    String staffId, name, role;

    Staff(String staffId, String name, String role) {

        this.staffId = staffId;
        this.name = name;

        this.role = role;

    }

    public String toString() {

        return "Staff ID: " + staffId + ", Name: " + name + ", Role: " + role;

    }

}

public class Main {

    static Scanner sc = new Scanner(System.in);

    static List<Patient> patients = new ArrayList<>();

    static List<Appointment> appointments = new ArrayList<>();

    static List<EHR> ehrRecords = new ArrayList<>();

    static List<Billing> bills = new ArrayList<>();

    static List<Inventory> inventoryList = new ArrayList<>();

    static List<Staff> staffList = new ArrayList<>();

  public static void main(String[] args) {

        while (true) {

            System.out.println("\n--- Hospital Management System ---");

            System.out.println("1. Register Patient");

            System.out.println("2. Schedule Appointment");

            System.out.println("3. Add EHR Record");

            System.out.println("4. Generate Bill");

            System.out.println("5. Manage Inventory");

            System.out.println("6. Manage Staff");
```

```
 System.out.println("7. View All Data");
 System.out.println("8. Exit");
 System.out.print("Choose an option: ");
int choice = sc.nextInt();
 sc.nextLine();
 switch (choice) {
     case 1:
         registerPatient();
         break;
     case 2:
         scheduleAppointment();
         break;
     case 3:
         addEHR();
         break;
     case 4:
         generateBill();
         break;
     case 5:
         manageInventory();
         break;
     case 6:
         manageStaff();
         break;
```

```
      case 7:

        viewAllData();

        break;

      case 8:

        System.out.println("Exiting...");

        System.exit(0);

        break;

      default:

        System.out.println("Invalid option! Try again.");

    }

  }

}

 static void registerPatient() {

  System.out.print("Enter ID: ");

  String id = sc.nextLine();

  System.out.print("Enter Name: ");

  String name = sc.nextLine();

  System.out.print("Enter Age: ");

  String age = sc.nextLine();

  System.out.print("Enter Gender: ");

  String gender = sc.nextLine();

  System.out.print("Enter Contact: ");

  String contact = sc.nextLine();

  System.out.print("Enter Medical History: ");
```

```
    String medicalHistory = sc.nextLine();
    patients.add(new Patient(id, name, age, gender, contact, medicalHistory));
    System.out.println("Patient registered successfully.");
}
   static void scheduleAppointment() {
    System.out.print("Enter Patient ID: ");
    String pid = sc.nextLine();
    System.out.print("Enter Doctor's Name: ");
    String doctor = sc.nextLine();
    System.out.print("Enter Date: ");
    String date = sc.nextLine();
    System.out.print("Enter Time: ");
    String time = sc.nextLine();
    appointments.add(new Appointment(pid, doctor, date, time));
    System.out.println("Appointment scheduled successfully.");
}

static void addEHR() {
    System.out.print("Enter Patient ID: ");
    String pid = sc.nextLine();
    System.out.print("Enter Diagnosis: ");
    String diagnosis = sc.nextLine();
    System.out.print("Enter Treatment: ");
    String treatment = sc.nextLine();
    System.out.print("Enter Prescription: ");

    String prescription = sc.nextLine();

    System.out.print("Enter Lab Reports: ");

    String labReports = sc.nextLine();

    ehrRecords.add(new EHR(pid, diagnosis, treatment, prescription, labReports));

    System.out.println("EHR record added successfully.");

}

  static void generateBill() {

    System.out.print("Enter Patient ID: ");

    String pid = sc.nextLine();

    System.out.print("Enter Amount: ");

    double amount = sc.nextDouble();

    sc.nextLine();

    bills.add(new Billing(pid, amount));

    System.out.println("Bill generated successfully.");

}


static void manageInventory() {

    System.out.print("Enter Item Name: ");

    String name = sc.nextLine();

    System.out.print("Enter Quantity: ");

    int quantity = sc.nextInt();

    sc.nextLine();

    inventoryList.add(new Inventory(name, quantity));
    System.out.println("Inventory updated successfully.");
}

   static void manageStaff() {
    System.out.print("Enter Staff ID: ");
    String id = sc.nextLine();
    System.out.print("Enter Name: ");
    String name = sc.nextLine();
    System.out.print("Enter Role: ");
    String role = sc.nextLine();
    staffList.add(new Staff(id, name, role));
    System.out.println("Staff added successfully.");
}

static void viewAllData() {
    System.out.println("\n--- Patients ---");
    patients.forEach(System.out::println);
    System.out.println("\n--- Appointments ---");
    appointments.forEach(System.out::println);
    System.out.println("\n--- EHR Records ---");
    ehrRecords.forEach(System.out::println);
    System.out.println("\n--- Bills ---");
    bills.forEach(System.out::println);
    System.out.println("\n--- Inventory ---");
```

```
        inventoryList.forEach(System.out::println);

        System.out.println("\n--- Staff ---");

        staffList.forEach(System.out::println);

    }

}
```

## Output:

```
--- Hospital Management System ---

1. Register Patient

2. Schedule Appointment

3. Add EHR Record

4. Generate Bill

5. Manage Inventory

6. Manage Staff

7. View All Data

8. Exit

Choose an option: 1

Enter ID: 1234

Enter Name: John

Enter Age: 30

Enter Gender: Male

Enter Contact: 9876543210

Enter Medical History: Diabetes

Patient registered successfully.

Choose an option: 2
```

```
Enter Patient ID: 1234
Enter Doctor's Name: Dr. Smith
Enter Date: 2025-04-20
Enter Time: 10:30 AM
Appointment scheduled successfully.
Choose an option: 3
Enter Patient ID: 1234
Enter Diagnosis: Fever
Enter Treatment: Paracetamol
Enter Prescription: Take 500mg every 6 hours
Enter Lab Reports: Normal CBC
EHR record added successfully.
Choose an option: 4
Enter Patient ID: 1234
Enter Amount: 150.75
Bill generated successfully.
Choose an option: 5
Enter Item Name: Paracetamol
Enter Quantity: 100
Inventory updated successfully.
Choose an option: 6
Enter Staff ID: S001
Enter Name: Alice
Enter Role: Nurse
Staff added successfully.
Choose an option: 7
```

```
--- Patients ---

ID: 1234, Name: John , Age: 30, Gender: Male, Contact: 9876543210, Medical History: Diabetes


--- Appointments ---

Patient ID: 1234, Doctor: Dr. Smith, Date: 2025-04-02, Time: 10:30 AM


--- EHR Records ---

Patient ID: 1234, Diagnosis: Fever, Treatment: Paracetamol, Prescription: Take 500mg every 6 hours,
Lab Reports: Normal CBC


--- Bills ---

Patient ID: 1234, Bill Amount: $150.75


--- Inventory ---

Item: Paracetamol, Quantity: 100


--- Staff ---

Staff ID: S001, Name: Alice, Role: Nurse

Choose an option: 8

Exiting...
```