**MySQL Table Creation Script**

CREATE DATABASE IF NOT EXISTS hospital\_db;

USE hospital\_db;

-- Staff Table

CREATE TABLE staff (

staff\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) UNIQUE NOT NULL,

password VARCHAR(100) NOT NULL,

role ENUM('Admin', 'Staff') NOT NULL

);

-- Patients Table

CREATE TABLE patients (

patient\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

age INT,

gender ENUM('Male', 'Female', 'Other'),

phone VARCHAR(15),

address VARCHAR(255)

);

-- Doctors Table

CREATE TABLE doctors (

doctor\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

specialization VARCHAR(100),

phone VARCHAR(15)

);

-- Appointments Table

CREATE TABLE appointments (

appointment\_id INT AUTO\_INCREMENT PRIMARY KEY,

patient\_id INT,

doctor\_id INT,

appointment\_date DATE,

reason VARCHAR(255),

FOREIGN KEY (patient\_id) REFERENCES patients(patient\_id),

FOREIGN KEY (doctor\_id) REFERENCES doctors(doctor\_id)

);

-- Billing Table

CREATE TABLE billing (

billing\_id INT AUTO\_INCREMENT PRIMARY KEY,

patient\_id INT,

appointment\_id INT,

total\_amount DECIMAL(10,2),

discharge\_date DATE,

summary TEXT,

FOREIGN KEY (patient\_id) REFERENCES patients(patient\_id),

FOREIGN KEY (appointment\_id) REFERENCES appointments(appointment\_id)

);

**☕ Java Code Structure**

**1. DBConnection.java**

import java.sql.Connection;

import java.sql.DriverManager;

public class DBConnection {

private static final String URL = "jdbc:mysql://localhost:3306/hospital\_db";

private static final String USER = "root";

private static final String PASSWORD = "root";

public static Connection getConnection() throws Exception {

Class.forName("com.mysql.cj.jdbc.Driver");

return DriverManager.getConnection(URL, USER, PASSWORD);

}

}

**2. LoginService.java**

import java.sql.\*;

import java.util.Scanner;

public class LoginService {

public static boolean login(String username, String password) {

try (Connection con = DBConnection.getConnection()) {

String query = "SELECT \* FROM staff WHERE username = ? AND password = ?";

PreparedStatement ps = con.prepareStatement(query);

ps.setString(1, username);

ps.setString(2, password);

ResultSet rs = ps.executeQuery();

return rs.next(); // login success if a match is found

} catch (Exception e) {

e.printStackTrace();

return false;

}

}

}

**3. PatientService.java**

import java.sql.\*;

import java.util.Scanner;

public class PatientService {

public static void registerPatient(Scanner sc) {

try (Connection con = DBConnection.getConnection()) {

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

String name = sc.nextLine();

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

int age = sc.nextInt();

sc.nextLine();

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

String gender = sc.nextLine();

System.out.print("Phone: ");

String phone = sc.nextLine();

System.out.print("Address: ");

String address = sc.nextLine();

String sql = "INSERT INTO patients (name, age, gender, phone, address) VALUES (?, ?, ?, ?, ?)";

PreparedStatement ps = con.prepareStatement(sql);

ps.setString(1, name);

ps.setInt(2, age);

ps.setString(3, gender);

ps.setString(4, phone);

ps.setString(5, address);

ps.executeUpdate();

System.out.println("✅ Patient registered successfully.");

} catch (Exception e) {

e.printStackTrace();

}

}

public static void viewPatients() {

try (Connection con = DBConnection.getConnection()) {

String sql = "SELECT \* FROM patients";

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery(sql);

System.out.printf("\n%-5s %-20s %-5s %-10s %-15s %-30s\n", "ID", "Name", "Age", "Gender", "Phone", "Address");

while (rs.next()) {

System.out.printf("%-5d %-20s %-5d %-10s %-15s %-30s\n",

rs.getInt("patient\_id"), rs.getString("name"), rs.getInt("age"),

rs.getString("gender"), rs.getString("phone"), rs.getString("address"));

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

**4. AppointmentService.java**

import java.sql.\*;

import java.util.Scanner;

public class AppointmentService {

public static void bookAppointment(Scanner sc) {

try (Connection con = DBConnection.getConnection()) {

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

int patientId = sc.nextInt();

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

int doctorId = sc.nextInt();

sc.nextLine();

System.out.print("Enter Appointment Date (YYYY-MM-DD): ");

String date = sc.nextLine();

System.out.print("Reason: ");

String reason = sc.nextLine();

String sql = "INSERT INTO appointments (patient\_id, doctor\_id, appointment\_date, reason) VALUES (?, ?, ?, ?)";

PreparedStatement ps = con.prepareStatement(sql);

ps.setInt(1, patientId);

ps.setInt(2, doctorId);

ps.setDate(3, Date.valueOf(date));

ps.setString(4, reason);

ps.executeUpdate();

System.out.println("✅ Appointment booked.");

} catch (Exception e) {

e.printStackTrace();

}

}

}

**5. BillingService.java**

import java.sql.\*;

import java.util.Scanner;

public class BillingService {

public static void generateBill(Scanner sc) {

try (Connection con = DBConnection.getConnection()) {

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

int appointmentId = sc.nextInt();

sc.nextLine();

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

int patientId = sc.nextInt();

sc.nextLine();

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

double amount = sc.nextDouble();

sc.nextLine();

System.out.print("Enter Discharge Date (YYYY-MM-DD): ");

String date = sc.nextLine();

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

String summary = sc.nextLine();

String sql = "INSERT INTO billing (patient\_id, appointment\_id, total\_amount, discharge\_date, summary) VALUES (?, ?, ?, ?, ?)";

PreparedStatement ps = con.prepareStatement(sql);

ps.setInt(1, patientId);

ps.setInt(2, appointmentId);

ps.setDouble(3, amount);

ps.setDate(4, Date.valueOf(date));

ps.setString(5, summary);

ps.executeUpdate();

System.out.println("✅ Billing generated.");

} catch (Exception e) {

e.printStackTrace();

}

}

}

**6. Main.java**

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("👤 Username: ");

String username = sc.nextLine();

System.out.print("🔒 Password: ");

String password = sc.nextLine();

if (!LoginService.login(username, password)) {

System.out.println("❌ Login failed. Exiting...");

return;

}

while (true) {

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

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

System.out.println("2. View Patients");

System.out.println("3. Book Appointment");

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

System.out.println("5. Exit");

System.out.print("Choose option: ");

int choice = sc.nextInt();

switch (choice) {

case 1 -> PatientService.registerPatient(sc);

case 2 -> PatientService.viewPatients();

case 3 -> AppointmentService.bookAppointment(sc);

case 4 -> BillingService.generateBill(sc);

case 5 -> {

System.out.println("🔚 Goodbye!");

return;

}

default -> System.out.println("❓ Invalid option");

}

}

}

}

**1. Set Up MySQL Database**

Run the **SQL script** to create the database and tables:

-- Paste this into MySQL Workbench or CLI

CREATE DATABASE IF NOT EXISTS hospital\_db;

USE hospital\_db;

CREATE TABLE staff (

staff\_id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) UNIQUE NOT NULL,

password VARCHAR(100) NOT NULL,

role ENUM('Admin', 'Staff') NOT NULL

);

CREATE TABLE patients (

patient\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

age INT,

gender ENUM('Male', 'Female', 'Other'),

phone VARCHAR(15),

address VARCHAR(255)

);

CREATE TABLE doctors (

doctor\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100),

specialization VARCHAR(100),

phone VARCHAR(15)

);

CREATE TABLE appointments (

appointment\_id INT AUTO\_INCREMENT PRIMARY KEY,

patient\_id INT,

doctor\_id INT,

appointment\_date DATE,

reason VARCHAR(255),

FOREIGN KEY (patient\_id) REFERENCES patients(patient\_id),

FOREIGN KEY (doctor\_id) REFERENCES doctors(doctor\_id)

);

CREATE TABLE billing (

billing\_id INT AUTO\_INCREMENT PRIMARY KEY,

patient\_id INT,

appointment\_id INT,

total\_amount DECIMAL(10,2),

discharge\_date DATE,

summary TEXT,

FOREIGN KEY (patient\_id) REFERENCES patients(patient\_id),

FOREIGN KEY (appointment\_id) REFERENCES appointments(appointment\_id)

);