Hospital Management System

Student Details:

Name: Maliha Mansuri

Roll Number: 24f2000184

Email: 24f2000184@ds.study.iitm.ac.in

About Me:I am a student in the BS program at IIT Madras, and I have a great interest in learning about and experimenting with new technologies. This is my first project, so I'm excited to learn more about web development and create useful applications. I like learning how things operate, making practical tools, and consistently developing my abilities through practical experience.

Problem Statement:

The objective of this project is to design and develop a comprehensive web-based Hospital Management System that streamlines interactions between administrators, doctors, and patients. The system enables efficient appointment scheduling, patient record management, treatment tracking, and overall hospital operations management, ensuring a smoother and more organized healthcare workflow.

Approach:

The application is built using Flask as the backend framework, with SQLAlchemy handling database management. It features role-based access control for three user types—Admin, Doctor, and Patient—each with clearly defined functionalities. The system supports real-time appointment booking with conflict prevention, treatment history tracking, and management of doctor availability. On the frontend, the application uses Bootstrap 5 for a responsive user interface and Jinja2 for dynamic content rendering.

AI/LLM Declaration: NO use of AI

Technologies and Frameworks Used

Flask	Purpose
Technology / Library	Core backend web framework for routing and application logic
Flask-SQLAlchemy	ORM for database operations and model management
SQLite	Lightweight relational database for data storage
Jinja2	Template engine for rendering dynamic HTML pages
Bootstrap 5	Frontend CSS framework for responsive design
Python datetime	Date and time management for appointments

Database Schema / ER Diagram

Tables:

1. Admin — Hospital administrator credentials (id, username, email, password)
2. Department — Medical specializations (id, name, description)
3. Doctor — Doctor profiles (id, name, email, password, phone, department_id, experience, is_active)
4. Patient — Patient information (id, name, email, password, phone, age, gender, address, is_active)
5. DoctorAvailability — Available time slots (id, doctor_id, date, start_time, end_time, is_available)
6. Appointment — Bookings (id, patient_id, doctor_id, date, time, status, created_at)
7. Treatment — Medical records (id, appointment_id, diagnosis, prescription, notes, created_at)

Relationships:

- One-to-Many: Department → Doctor
- One-to-Many: Doctor → DoctorAvailability, Doctor → Appointment
- One-to-Many: Patient → Appointment
- One-to-One: Appointment → Treatment

API Resource Endpoints:

Endpoint	**Method**	**Description**
/	GET	Home page
/login	POST	User authentication
/register	POST	Patient registration

/logout	GET	End session
/admin/dashboard	GET	Admin statistics
/admin/doctors	GET	View/manage doctors
/admin/add_doctor	POST	Add doctor
/admin/patients	GET	View patients
/doctor/dashboard	GET	Doctor dashboard
/doctor/availability	POST	Set availability
/doctor/complete_appointment/<id>	POST	Add treatment
/patient/dashboard	GET	Patient dashboard
/patient/book_appointment/<id>	POST	Book appointment
/patient/treatment_history	GET	View records

Architecture and Features

Architecture:

- hospital-management/
- ├── app.py # Main application
- ├── templates/ # HTML templates (19 files)
- ├── static/css/ # CSS styling
- └── hospital.db # SQLite database

Implemented Features:

Admin: Dashboard statistics, doctor management (add/edit/delete), patient management, appointment overview, search functionality

Doctor: Personalized dashboard, set availability (7 days), complete appointments with diagnosis/prescription, view patient history

Patient: Registration/login, search doctors by specialization, book/cancel appointments, view treatment history, update profile

Security: Password hashing, session management, role-based access control, SQL injection prevention

UI/UX: Responsive Bootstrap 5 design, flash messages, real-time conflict detection, intuitive navigation

---

Video Presentation

Drive Link:
https://drive.google.com/file/d/13NkJAF3L_kjuNA1B5v41lTXSFmVQd3xV/view?usp=sharing

---

Submitted by: Maliha Mansuri

Date: 30/11/25

Course: MAD 1