App Development Project Report

App Dev Project Report

1. Student Details

Name: Saurabh Sunil Jori
Roll Number: 24F1002674
Email: 24f1002674@ds.study.iitm.ac.in
About Me: I am diploma level student in IIT Madras BS Degree program with a deep interest in web application development and data-driven technologies. I enjoy building meaningful applications that combine learning, analytics, and user experience.

2. Project Details

Project Title: HOSPITAL MANAGEMENT SYSTEM

Problem Statement:

Hospitals need efficient systems to manage patients, doctors, appointments, and treatments. Currently, many hospitals use manual registers or disconnected software, which makes it difficult to manage records, avoid scheduling conflicts, and track patient history.

Approach:

The Hospital Management System was developed using Flask with a modular structure to keep the backend clean and organized. The app manages essential workflows such as doctor records, patient details, and appointment scheduling through dynamic routes and templates. SQLite is used for lightweight data storage, while custom HTML/CSS ensures a simple and user-friendly interface.

3. AI/LLM Declaration (3–4 lines)

I used ChatGPT only for minor assistance such as improving wording, refining HTML/CSS structure, and getting suggestions for UI design. The overall implementation, routing logic, database handling, debugging, and project integration were completed manually. The AI usage was limited to around 10–15% and did not contribute to core functionality or decision-making in the system**.**

4. Technologies and Frameworks Used

Technology / Library	**Purpose**
Flask	Core backend framework used to build routes, handle requests, and manage the application structure.
SQLite	Lightweight database used for storing doctor details, patient data, and appointment records.
Jinja2	Template engine used to render dynamic HTML pages and pass backend data to the frontend.
HTML5 & CSS3	Used to design the layout, styling, and responsive frontend UI of the application.
Bootstrap (optional if used)	For improving basic UI responsiveness and layout structure.
SQLAlchemy (optional if used)	ORM used for defining database models and managing data operations.

5. Database Schema / ER Diagram

Tables

1. **Doctor — Stores doctor information such as** ***(doctor_id, name, specialty, phone, email)***

2. **Patient — Maintains patient details including** ***(patient_id, name, age, gender, contact)***

3. **Appointment — Stores appointment records** ***(appointment_id, patient_id, doctor_id, date, time, status)***

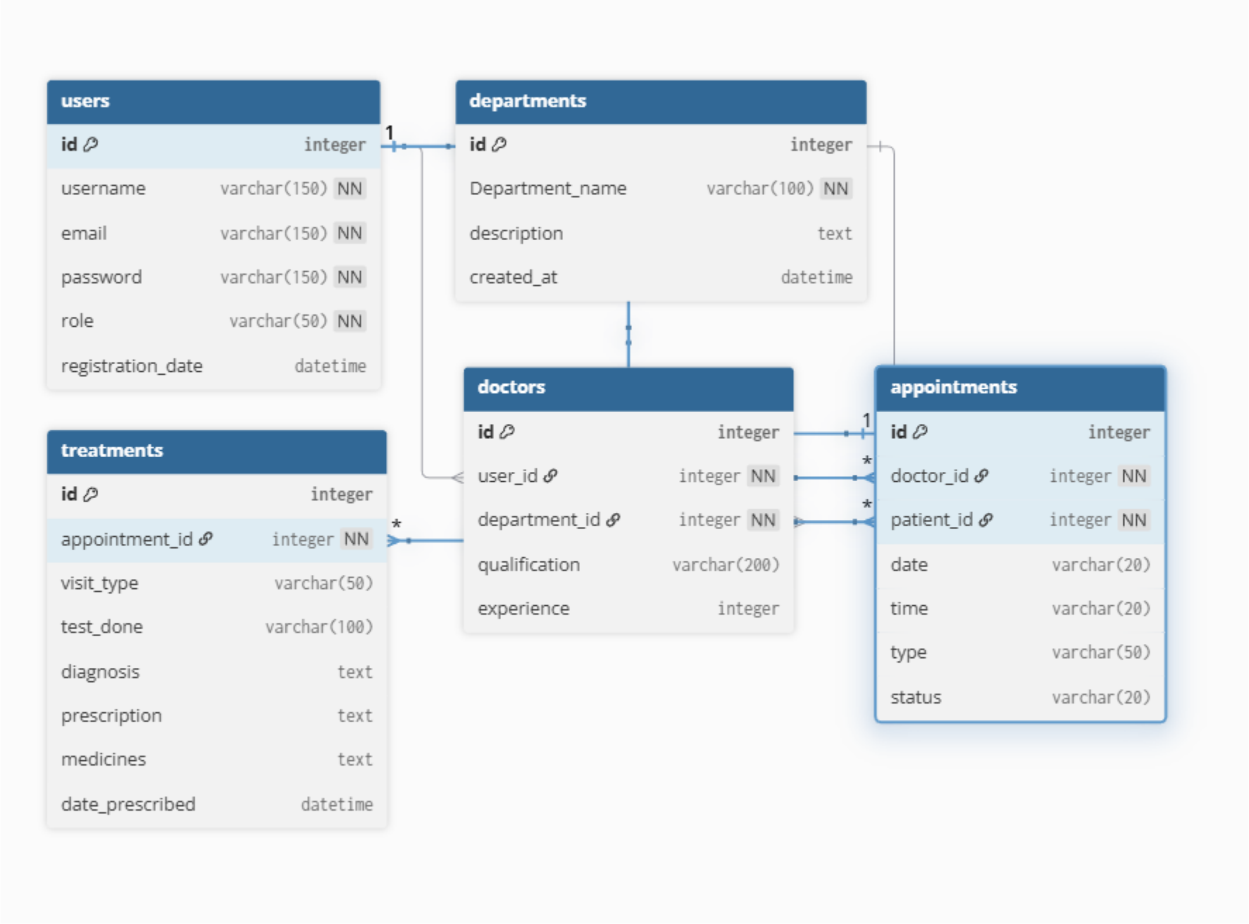
4. **Admin/User (optional) — Handles login or authentication (if implemented)** ***(id, username, password)***

Relationships

- **One-to-Many → Doctor → Appointment**
  A doctor can have multiple appointments.

- **One-to-Many → Patient → Appointment**
  A patient can book many appointments.

- ***(Optional)*** **One-to-Many → Admin/User → Actions**
  If your system tracks user/admin actions.



users		
id	integer	
username	varchar(150)	NN
email	varchar(150)	NN
password	varchar(150)	NN
role	varchar(50)	NN
registration_date	datetime	

departments		
id	integer	
Department_name	varchar(100)	NN
description	text	
created_at	datetime	

doctors		
id	integer	
user_id	integer	NN
department_id	integer	NN
qualification	varchar(200)	
experience	integer	

appointments		
id	integer	
doctor_id	integer	NN
patient_id	integer	NN
date	varchar(20)	
time	varchar(20)	
type	varchar(50)	
status	varchar(20)	

treatments		
id	integer	
appointment_id	integer	NN
visit_type	varchar(50)	
test_done	varchar(100)	
diagnosis	text	
prescription	text	
medicines	text	
date_prescribed	datetime	

8. Video Presentation

Drive Link:
https://drive.google.com/file/d/1b_qNDLBZ36V0cw71IS6qJEWVvfTf4YY6/view?usp=sharing