

App Dev Project Report

1. Student Details

Name: S GANESH

Roll Number: 22f3002929

Email: 22f3002929@ds.study.iitm.ac.in

About Me:

I am an IIT Madras BS Degree student with a strong interest in backend development, databases, and full-stack applications. I enjoy building real-world systems that combine clean UI, structured data flow, and practical usability. This project helped me strengthen my Flask, SQLAlchemy, and Bootstrap skills while designing a professional medical appointment workflow.

2. Project Details

Project Title: Hospital Management System

Problem Statement:

Hospitals face difficulty managing doctor availability, appointments, patient records, and treatment history manually.

The goal of this project is to create a **complete web-based platform** for:

- Patients to book appointments efficiently
- Doctors to manage availability & treat patients
- Admins to manage users, doctors, and analytics
- Simplified medical history tracking

Approach:

The application is built using **Flask** as the backend framework with a modular structure and a clean Bootstrap-based UI.

Key design principles:

- Role-based access system (Admin, Doctor, Patient)
- Real-time availability management
- Secure appointment booking with conflict checks
- Clean medical record structure (diagnosis, prescription, notes)
- Consistent UI with professional dashboards (Admin, Doctor, Patient)
- Dedicated pages for extended reports (e.g., treatment view pages)

The system ensures correctness, simplicity, and smooth user experience.

3. AI/LLM Declaration

I used **ChatGPT (GPT-5.1)** to assist with:

- HTML and Bootstrap improvements
- Ensuring UI consistency
- Writing documentation sections
- Identifying a bug involving appointment cancellation and slot re-booking

The usage was around **20–25%**, limited to:

- UI layout suggestions
- Code formatting
- Bug explanations

All backend logic, database design, API integrations, and final implementation were done manually.

4. Technologies and Frameworks Used

Technology / Library	Purpose
Flask	Core backend web framework
SQLAlchemy	ORM for SQLite database
Jinja2	Dynamic HTML templating
Bootstrap 5	Responsive UI and styling
Flask-Login	Authentication and session management
SQLite	Lightweight database
JavaScript	Dynamic loading of time slots, validation
Fetch API	AJAX calls for admin search

5. Database Schema / ER Diagram

Tables:

User — stores all user-related information, including login credentials and the role (admin, doctor, or patient).

Department — stores the list of all medical departments available in the hospital.

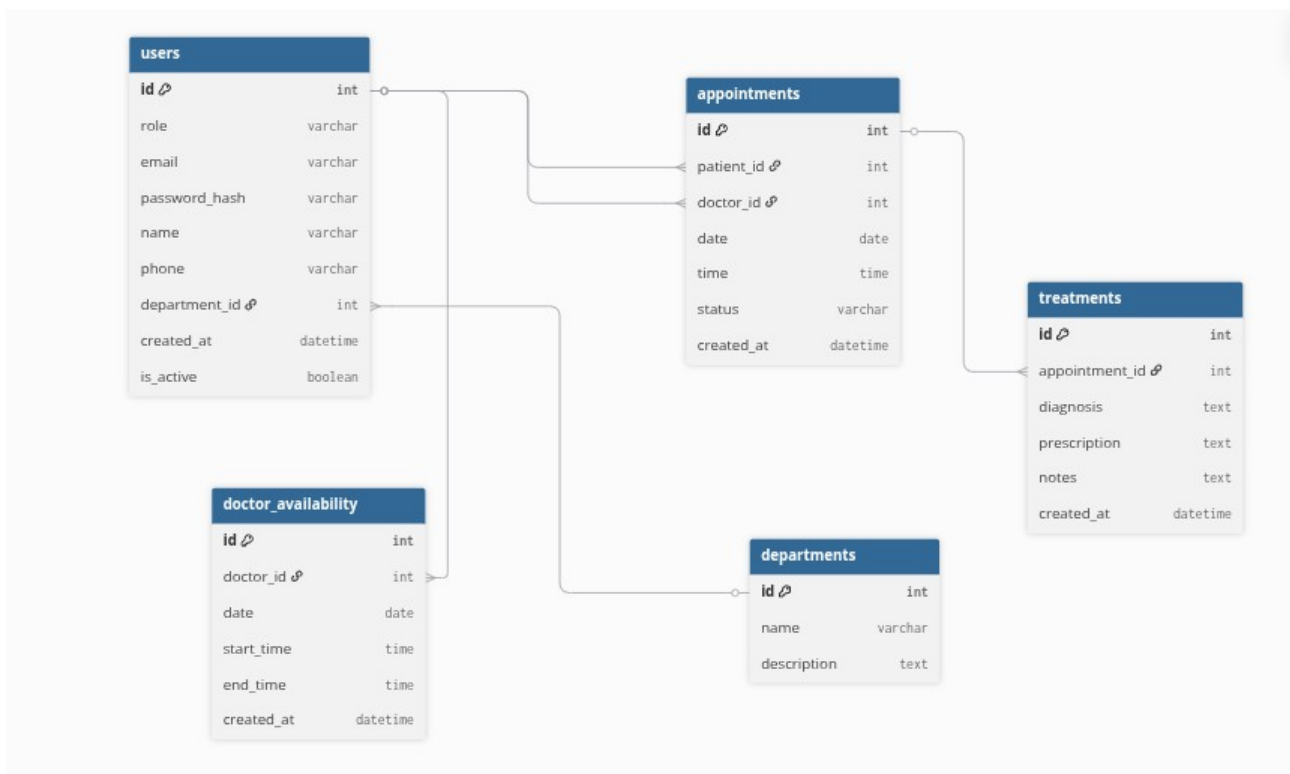
DoctorAvailability — stores the availability slots provided by each doctor for booking appointments.

Appointment — stores all patient–doctor appointment details, including date, time, and status.

Treatment — stores diagnosis, prescriptions, and notes recorded by doctors for completed appointments.

Relationships:

- **User (doctor)** → **DoctorAvailability** → One-to-Many
- **User (doctor)** → **Appointment** → One-to-Many
- **User (patient)** → **Appointment** → One-to-Many
- **Appointment** → **Treatment** → One-to-One
- **Department** → **User (doctor)** → One-to-Many



6. API Resource Endpoints

Endpoint	Method	Description
/login	POST	Authenticate user
/admin/search_appointments	GET	Search appointments dynamically
/doctor/add_availability	POST	Add doctor availability
/doctor/delete_availability/<id>	POST	Delete availability slot
/patient/appointments	GET	View & book appointments
/cancel/<appointment_id>	GET	Cancel a booked appointment

Additional internal routes:

- /admin_dashboard
 - /doctor_dashboard
 - /patient_dashboard
 - /doctor_patients
 - /treatment_report/<appointment_id>
 - More information added as api.yaml inside the zip file.
-

7. Features

Implemented Features

Admin Features

- ✓ Manage doctors
- ✓ Manage users
- ✓ View admin analytics
- ✓ Search appointments dynamically
- ✓ Activate/Deactivate doctors

Doctor Features

- ✓ Manage availability (restricted to today → next 6 days)
- ✓ View appointments
- ✓ Mark treatments (diagnosis, prescription, notes)
- ✓ Generate patient reports

Patient Features

- ✓ Book appointments (conflict-free with real-time slot blocking)
- ✓ Cancel bookings
- ✓ View medical history
- ✓ Open detailed treatment report page

Additional Highlights

- Dynamic time-slot generation
 - Slot disabling after booking and re-enabling after cancellation
 - Sticky table headers
 - Clean dashboard UI
 - Proper flash messages for all actions
-

8. Video Presentation

Drive Link:

https://drive.google.com/file/d/1VsucoOrq686KzvmzB4YqMiHILneSHBy_/view?usp=sharing