App Development Project Report

1. Student Details

Name: *Aditya Kadam*
Roll Number: DS*22F3001591*
Email: *22f3001591@ds.study.iitm.ac.in*

About Me:
I am a student of the IIT Madras BS Degree program with a strong interest in full-stackwebdevelopment. This project reflects my understanding of backend development, database design, authentication systems, and role-based access control using Flask.

---

2. Project Details

Project Title:
Hospital Management System (HMS)

Problem Statement:

Hospitals often face challenges in managing doctors, patients, appointments, and treatment records efficiently. Manual handling or poorly integrated systems can lead to scheduling conflicts, lack of transparency, and difficulty in accessing medical history. The goal of this project is to design and develop a web-based Hospital Management System that streamlines hospital operations for administrators, doctors, and patients.

Approach:

The application is developed using the Flask web framework with a modular and role-based architecture. The system supports three types of users, Admin, Doctor, and Patient, each having clearly defined responsibilities. The backend is powered by SQLAlchemy with SQLite as the database, and the frontend uses Jinja2 templates styled with Bootstrap. Secure authentication, appointment scheduling, and treatment record management form the core workflow of the application.

---

3. AI / LLM Declaration

I used ChatGPT (GPT-based model) to assist in:

- Understanding Flask routing structures
- Refining SQLAlchemy relationship queries
- Improving code readability and structure
- Debugging template and routing issues

The use of AI/LLM was limited to **guidance and suggestions (approximately 15 to 20 percent)**.
All core logic, database design decisions, debugging, testing, and final implementation were completed manually by me.

---

4. Technologies and Frameworks Used

Technology / Library	**Purpose**
Flask	Core backend web framework
SQLAlchemy	ORM for database interaction
Flask-Login	Authentication and session management
Jinja2	Server-side HTML templating
Bootstrap 5	Responsive frontend styling
SQLite	Lightweight relational database
Werkzeug	Secure password hashing

---

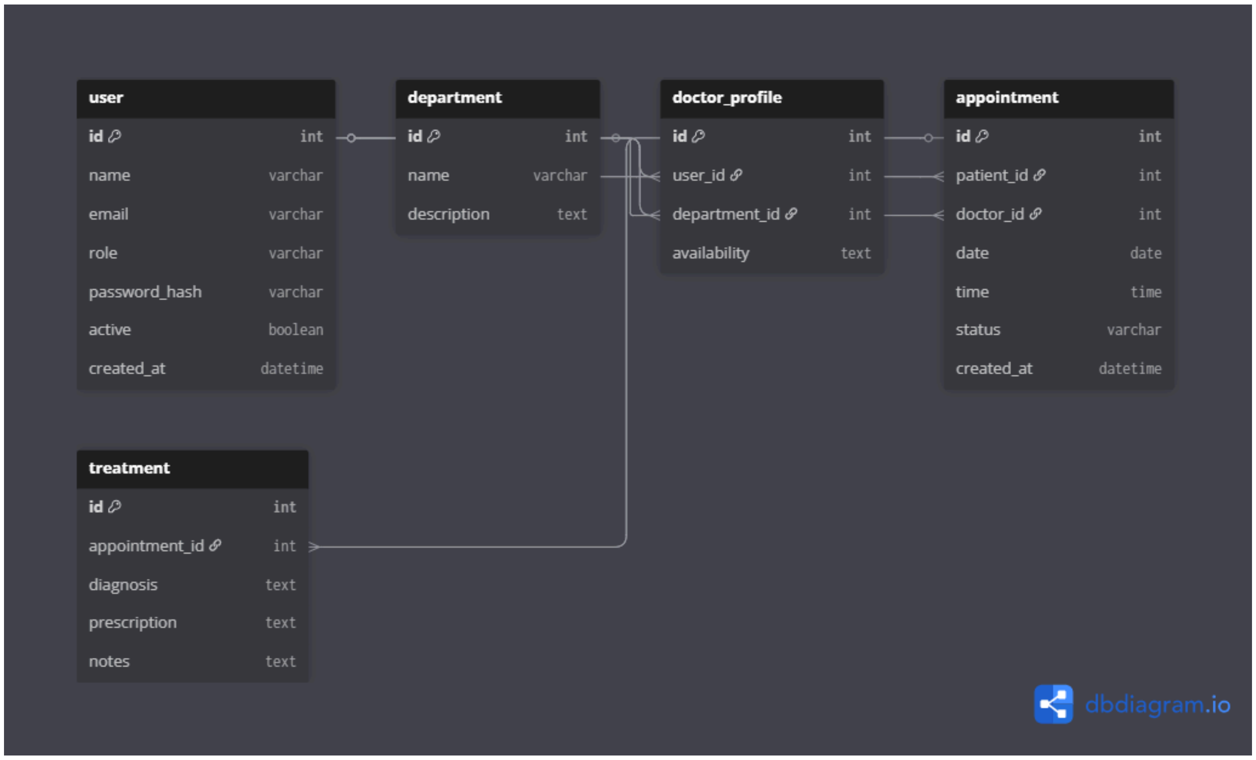
5. Database Schema / ER Diagram

Tables

- **User** — stores all user accounts (id, name, email, password_hash, role, active)
- **Department** — stores medical specializations (id, name, description)
- **DoctorProfile** — stores doctor-specific details (id, user_id, department_id, availability)
- **Appointment** — stores appointment details (id, patient_id, doctor_id, date, time, status)
- **Treatment** — stores medical records for completed appointments (id, appointment_id, diagnosis, prescription, notes)

Relationships

- **One-to-One** → User → DoctorProfile
- **One-to-Many** → Department → DoctorProfile
- **One-to-Many** → User (Patient) → Appointment
- **One-to-Many** → User (Doctor) → Appointment
- **One-to-One** → Appointment → Treatment



6. API / Route Endpoints

Endpoint	Method	Description
/login	GET / POST	User authentication
/register	GET / POST	Patient self-registration
/admin/dashboard	GET	Admin overview dashboard
/admin/add-doctor	GET / POST	Add doctor account
/admin/search	GET / POST	Search patients and doctors
/doctor/dashboard	GET	Doctor dashboard
/doctor/complete-appointment	GET / POST	Add treatment record
/patient/dashboard	GET	Patient dashboard
/book-appointment	GET / POST	Book appointment

---

7. Architecture and Features (Optional)

Architecture Overview:

- `run.py` – Application entry point and admin creation
- `config.py` – Application configurations
- `models.py` – Database models using SQLAlchemy
- `routes.py` – Role-based Flask routes using Blueprint
- `/templates` – Jinja2 HTML templates
- `/static` – CSS files for UI styling

Implemented Features:

- Role-based authentication (Admin / Doctor / Patient)

- Admin management of doctors, patients, and departments
- Doctor appointment handling and treatment recording
- Patient appointment booking, rescheduling, and cancellation
- Appointment conflict prevention
- Treatment history management
- Responsive UI using Bootstrap
- Secure password hashing

---

8. Video Presentation

Drive Link: **mad1_presentation.mkv**

https://drive.google.com/file/d/1qi-aPfEgrh50YE98N4cuM1cXgBughY_J/view?usp=sharing