Student Details

Name: Shresth Kasera
Roll Number: 24F3004285
Email: 24f3004285@ds.study.iitm.ac.in
About: I am shresth currently pursuing Data science from IIT Madras, really like to do web development, also good in python.

Project Details

Project Understanding:

- Develop a web-based Hospital Management System for managing appointments between patients and doctors
- Implement role-based access control (Admin, Doctor, Patient) with distinct functionalities for each role
- Create a complete appointment booking workflow with doctor availability management and treatment records

AI/LLM Usage

Tools Used: Google Gemini

- Take help to create Patient Appointment Booking according api to the Doctor Availability
- Take help to Plot live chart by matplotlib
- Take help in Styling of home page by bootstrap

Technologies Used

- **Backend**: Flask, Flask-SQLAlchemy, Flask-Login , Flask-RESTful
- **Frontend:** Html, css, Jinja2, Bootstrap, JavaScript
- **Database:** SQLite

DB Schema Design

Tables and Structure:

1. DEPARTMENT

2. USER

- **Relationships:**
    - Many-to-One with Department (for doctors)
    - One-to-Many with Appointment (as patient and doctor)

    - One-to-Many with DoctorAvailability

3. DOCTOR_AVAILABILITY

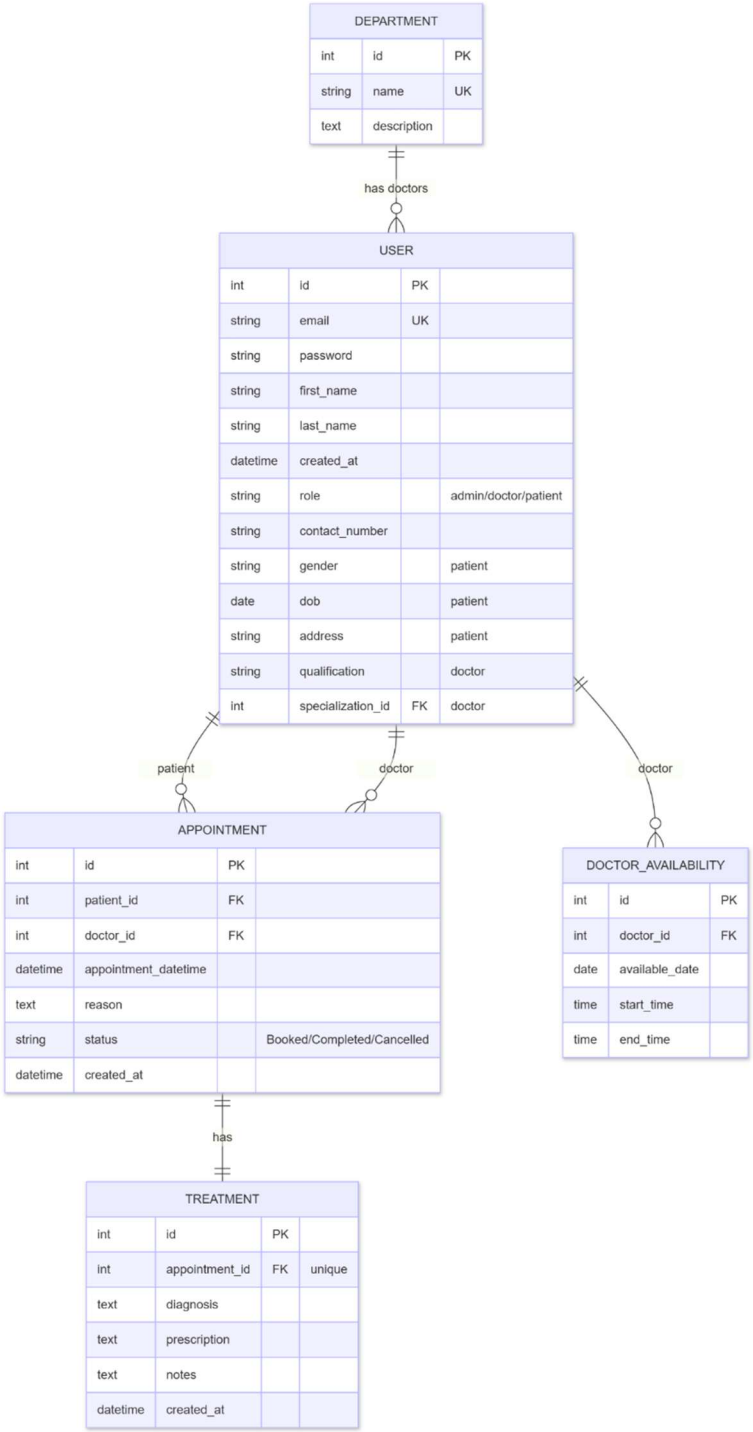
- **Relationships:** Many-to-One with User (doctor)

4. APPOINTMENT

- **Relationships:**
    - Many-to-One with User (patient)
    - Many-to-One with User (doctor)
    - One-to-One with Treatment

5. TREATMENT

- **Relationships:** One-to-One with Appointment



API Design

Authentication:

- GET /auth/login – redirect to login page
- POST /auth/login – send id password to server to login
- GET /auth /register – redirect to patient register page
- POST /auth /register – send data to server to register new patient

Admin:

- GET /admin/ - redirect to admin dashboard
- GET /admin/doctor - view list of all doctor data table
- GET /admin/search_doctor - search doctor
- POST /admin/update_doctor/<id> - update doctor data
- DELETE /admin/delete_doctor/<id> - delete patient data
- GET /admin/patient - view list of all patient data table
- GET /admin/search_patient – search patient
- POST /admin/update_ patient /<id> - update patient data
- DELETE /admin/delete_ patient/<id> - delete patient data
- GET /admin/stats – view statics data chart
- GET /admi /appointments - view all appointments

Doctor:

- GET /doctor/ - redirect to doctor dashboard
- GET /doctor/profile - view doctor profile
- GET /doctor/appointment/update_status/<id> - update booked appointment status
- GET /doctor/appointment/treatment/<id> - redirect to patient treatment page
- POST /doctor/appointment/ treatment/<id> -send treatment data to server
- GET /doctor/patient_history/<id> - view patient medical history
- GET /doctor/availability – redirect to manage availability page
- POST /doctor/availability – send availability slot data to server
- DELETE /doctor/delete_availability/<id> - Delete availability slot
- GET /doctor/stats - view statics data chart

Patient:

- GET /patient/ - redirect to patient dashboard
- GET /patient/profile - view patient profile
- POST /patient/update_profile – send updated patient data to server
- GET /patient/find_doctors – search doctor api
- GET /patient/book_appointment/<doctor_id> - redirect to book appointment page
- POST /patient/book_appointment/<doctor_id> - send booking appointment data
- POST /patient/appointment/cancel/<doctor_id> - cancel booked appointment
- GET /patient/treatment/<id> - view doctor treatment
- GET /patient/stats - view statics data chart

Some RESTful API:

- GET /api/doctors - List all doctors
- GET /api/patients - List all patients
- GET /api/appointments/ - view appointments details
- POST /api/appointments/<id> - Create new appointments
- PUT /api/appointments/<id> - Update appointments
- DELETE /api/appointments/<id> - Delete appointments

Architecture and Features

Architecture Pattern:

- **app.py** – main Flask application entry point
- **models.py** – database models using SQLAlchemy
- **/routes** – Flask Blueprints for user and activity routes
- **/templates** – Jinja2 HTML templates

Features Implemented:

- **User Authentication & Authorization:**
  - Secure registration and login with password hashing
  - Role-based access control (Admin, Doctor, Patient)
  - Session management using Flask-Login
- **Admin Features:**
  - Manage departments (Create, Read, Update, Delete)
  - View all users and appointments
  - Generate system reports and statistics
  - Manage doctor accounts and assignments
- **Doctor Features:**
  - View and manage personal profile
  - Set availability schedules (date, time slots)
  - View assigned appointments
  - Add treatment records (diagnosis, prescription, notes)
  - Update appointment status (Complete/Cancel)
- **Patient Features:**
  - Search doctors by department/specialization
  - View doctor availability in real-time
  - Book appointments with available doctors
  - View appointment history
  - View treatment records from past appointments
  - Cancel upcoming appointments

Video

Video Link:

https://drive.google.com/file/d/1atLrV8untCiX-035r79DMHC663ijmDeV/view?usp=sharing