Innovation & Entrepreneurship Hub for Educated Rural Youth (SURE Trust – IERY)

Appointment Scheduling System

The domain of the Project- Healthcare

Under the guidance of

Mr. Srihari Poturaju (Full Stack Developer - IBM)

By

Mr. Dharmik Manishkumar Sompura (B.E. IT 3rd Year)

Period of the project

December 2024 to February 2025

SURE TRUST

PUTTAPARTHI, ANDHRA PRADESH

DECLARATION

The project titled **" *Appointment Booking System* "** has been mentored by **Mr. Srihari Poturaju** and organized by SURE Trust from December 2024 to February 2025**.** This initiative aims to benefit educated unemployed rural youth by providing hands-on experience in industry-relevant projects, thereby enhancing employability.

I, **Mr. Dharmik Manishkumar Sompura,** hereby declare that I have solely worked on this project under the guidance of my mentor. This project has significantly enhanced my practical knowledge and skills in the domain.

Name
Mr. Dharmik Manishkumar Sompura

Signature

Mentor
Mr. Srihari Poturaju
Full Stack Developer - IBM

Signature

Seal & Signature

Prof.Radhakumari
Executive Director & Founder
SURE Trust

Table of Contents

Executive Summary

The Healthcare Appointment Scheduling System is a web-based solution designed to streamline, automate, and optimize appointment booking in healthcare facilities, particularly in resource-constrained environments. Built using Flask, FastAPI, PostgreSQL, and AJAX, the system eliminates inefficiencies in traditional scheduling processes by offering real-time booking, doctor availability management, and administrative oversight, all while maintaining strict data security through role-based access control (RBAC).

The system ensures dynamic slot updates and prevents overbooking by leveraging PostgreSQL triggers, enhancing accuracy and reliability. Its AJAX-powered real-time updates provide a seamless user experience, minimizing patient wait times and improving operational workflow. A structured architecture and scalable backend make the platform highly adaptable to evolving healthcare needs.

The system's modular design allows for future enhancements, such as AI-driven doctor recommendations, digital prescriptions, and telemedicine support, positioning it as a scalable, future-ready solution for digitizing healthcare scheduling at scale.

Introduction

Background and Context

Efficient healthcare appointment scheduling is crucial in ensuring seamless patient care, yet traditional systems often suffer from manual inefficiencies, long waiting times, and poor doctor-patient coordination. In resource-constrained environments, these issues are exacerbated, leading to delays, mismanagement, and patient dissatisfaction. The Healthcare Appointment Scheduling System is developed as a web-based solution that digitizes and optimizes appointment management, ensuring real-time slot availability, automated scheduling, and enhanced administrative control.

Problem Statement

Conventional appointment booking methods rely heavily on manual record-keeping, making them prone to errors, overbooking, and inefficiencies in healthcare facilities. This results in long patient waiting times, ineffective resource allocation, and administrative bottlenecks. Furthermore, the lack of dynamic scheduling and real-time updates often causes miscommunication between patients and healthcare providers. This project addresses these challenges by developing an automated, role-based appointment booking system that ensures efficient slot allocation, doctor availability management, and seamless patient experience.

Scope

This project aims to develop a scalable and efficient web-based appointment scheduling system for healthcare facilities. The scope includes:

- **Role-Based Appointment Booking:** Patients can view available doctors and book appointments, while doctors can manage their availability.
- **Real-Time Updates: AJAX-powered** dynamic updates ensure that appointment slots are instantly updated without page reloads.
- **Overbooking Prevention:** PostgreSQL triggers enforce scheduling constraints, eliminating double bookings.
- **User and Doctor Management:** Separate modules for patients, doctors, and administrators to streamline appointment handling.
- **Data Security and Access Control:** Implementing Role-Based Access Control (RBAC) to ensure secure data management.

Limitations

- **Limited AI Functionality:** The current system does not include AI-driven recommendations but allows for future integration.
- **Dependency on Internet Connectivity:** As a web-based platform, the system requires **a stable internet connection** for real-time functionality.
- **Manual Doctor Approvals:** Doctor registrations require manual verification by the administrator.

Innovation

The Healthcare Appointment Scheduling System introduces real-time slot management, overbooking prevention, and an intuitive user interface designed for scalability and efficiency. Unlike traditional methods, it leverages AJAX for real-time updates, PostgreSQL triggers for accuracy, and a secure role-based access system to ensure smooth and error-free appointment scheduling. Future enhancements may include AI-driven doctor recommendations, digital prescriptions, and telemedicine integration, further advancing the platform's capabilities.

Project Objectives

Project Objectives and Expected Outcomes

- **Develop an Efficient Healthcare Appointment Scheduling System**

  Design and implement a web-based platform for healthcare appointment booking, ensuring real-time doctor availability management, automated scheduling, and secure role-based access control (RBAC) to improve efficiency in healthcare facilities.
  Expected Outcome: A functional appointment scheduling system that enables seamless doctor-patient interaction, minimizes manual inefficiencies, and ensures accurate scheduling.

- **Implement Real-Time Updates for Appointment Booking**

  Integrate AJAX-powered dynamic updates to provide instantaneous slot availability without requiring page reloads, enhancing user experience and reducing booking errors.
  Expected Outcome: A responsive system that updates doctor availability and patient bookings in real time, ensuring accurate appointment scheduling.

- **Ensure Overbooking Prevention and Data Integrity**

  Utilize PostgreSQL triggers to automate scheduling constraints, preventing duplicate appointments and ensuring data consistency.
  Expected Outcome: A robust system that eliminates overbooking issues, ensuring smooth doctor-patient coordination.

- **Enhance Data Security and Role-Based Access Control (RBAC)**

  Implement secure authentication and authorization mechanisms to restrict access based on user roles (patients, doctors, and administrators), ensuring data privacy and system integrity.
  Expected Outcome: A secure, role-based access system that protects sensitive medical data and allows controlled system usage.

- **Optimize System Performance for Scalability**

  Develop an efficient backend architecture using Flask, FastAPI, and PostgreSQL to ensure scalability and seamless handling of increasing users in healthcare facilities.
  Expected Outcome: A high-performance system capable of handling multiple concurrent users, ensuring fast response times and smooth operation.

Deliverables

- **Fully Functional Healthcare Appointment Scheduling System:**

  A complete web-based platform with real-time booking, role-based access control, and doctor availability management.

- **Overbooking Prevention and Secure Access Control Implementation**

  A working system demonstrating PostgreSQL-based scheduling constraints and RBAC security features.

- **Codebase and Documentation**

  A well-structured code repository with detailed documentation covering system architecture, database design, and deployment instructions for future enhancements.

Methodology and Results

Methods/Technology Used

Technologies Implemented:

- **Frontend:** HTML, Jinja2 Template, Bootstrap5, JavaScript, jQuery, AJAX
- **Backend:** Flask, FastAPI, Python3
- **Database:** PostgreSQL
- **API Documentation:** Swagger

Tools/Software Used:

- **Visual Studio Code (VS Code)** – Used for developing, testing, and debugging the project.
- **Swagger**– Used for API testing and validation.
- **PgAdmin** – Used for managing and querying the PostgreSQL database.
- **GitHub** – Version control and collaboration.

Project Architecture

User Registration & Role-Based Access

- Patients, doctors, and administrators register and log in with secure authentication mechanisms.
- Role-based access ensures that each user only accesses authorized functionalities.

Appointment Booking & Doctor Availability Management

- **Patients** can view available doctors, check real-time slot availability, and book appointments.
- **Doctors** manage their schedules and confirm or modify appointments.
- **Admins** oversee the entire system, ensuring smooth operations.

Dynamic Slot Management

- The system prevents double bookings and ensures conflict-free appointment scheduling.
- AJAX-based dynamic updates allow users to see the latest appointment status without reloading the page.

Secure Database & API Integration

- PostgreSQL is used to store and manage patient and doctor data.
- APIs built with FastAPI and Flask handle data processing, appointment validation, and retrieval.

Results

- **Reduced Booking Conflicts:** Implementing real-time updates significantly decreased appointment clashes.
- **Efficient Doctor Scheduling:** Doctors were able to optimize their time with dynamic appointment adjustments.
- **Improved User Experience:** AJAX-powered instant updates resulted in faster booking interactions and minimal waiting times.
- **Enhanced System Security:** Secure authentication and role-based access control ensured data privacy and restricted unauthorized access.

The system successfully optimized healthcare appointment scheduling, demonstrating efficient real-time booking, reduced patient waiting times, and improved operational workflow for healthcare facilities.

Final Project Hardware and Working Screenshots

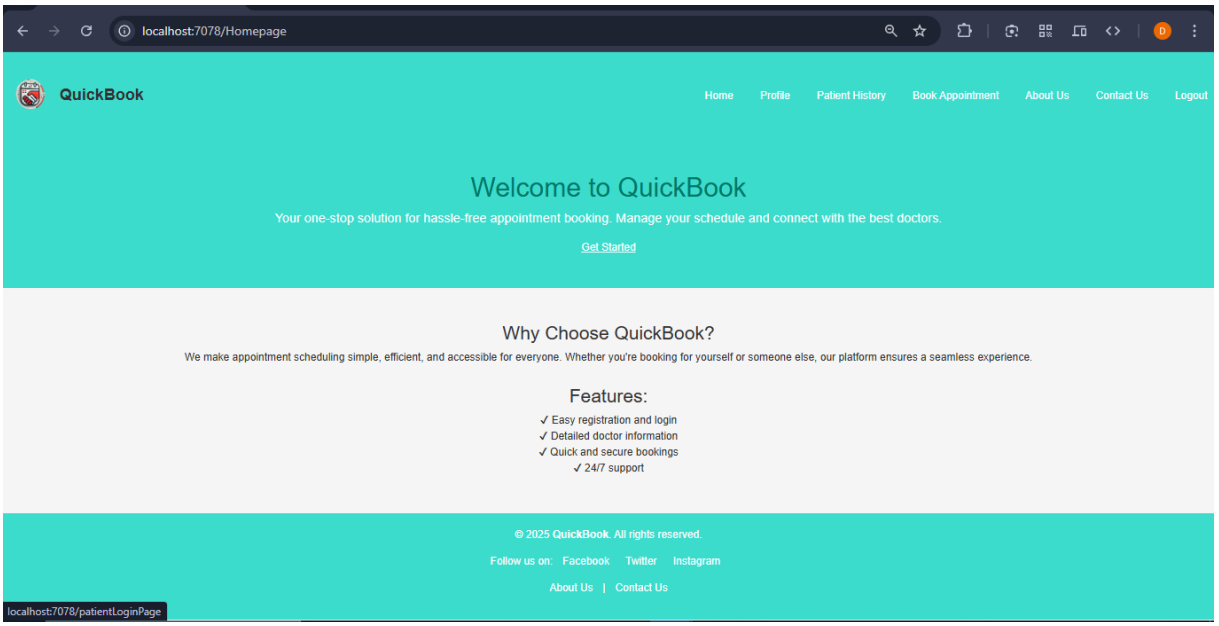

Figure 1: Home Page

Figure-1 : The landing page provides an overview of the healthcare appointment scheduling system, offering navigation to key sections like booking, doctor details, and user accounts. Accessible to all visitors, it ensures a seamless experience for both new and returning users.

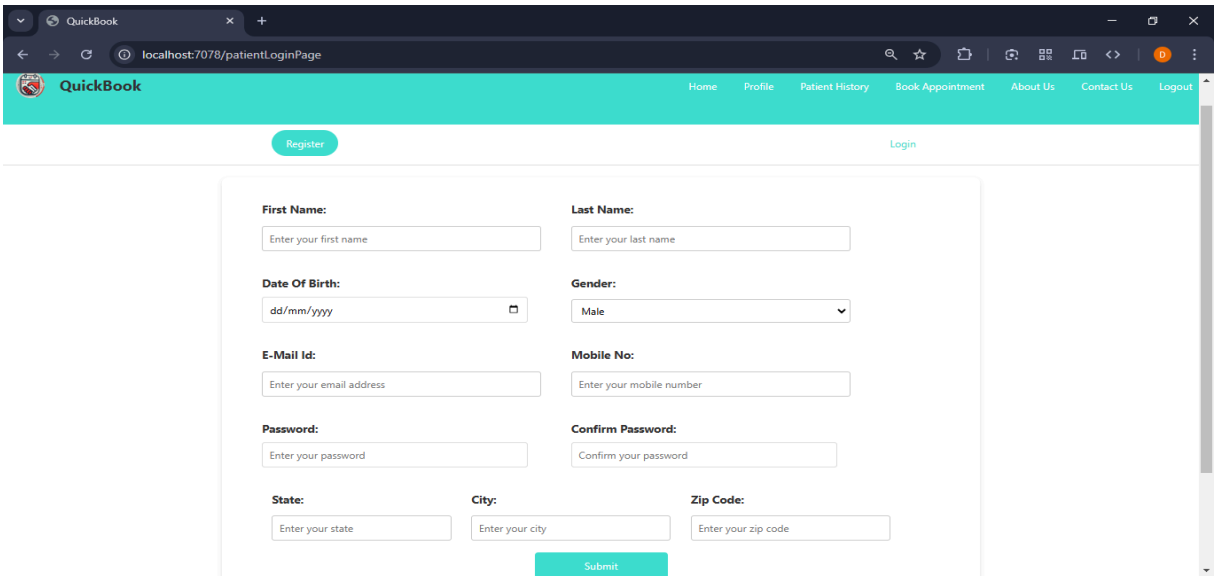

Figure 2: User Registration Page

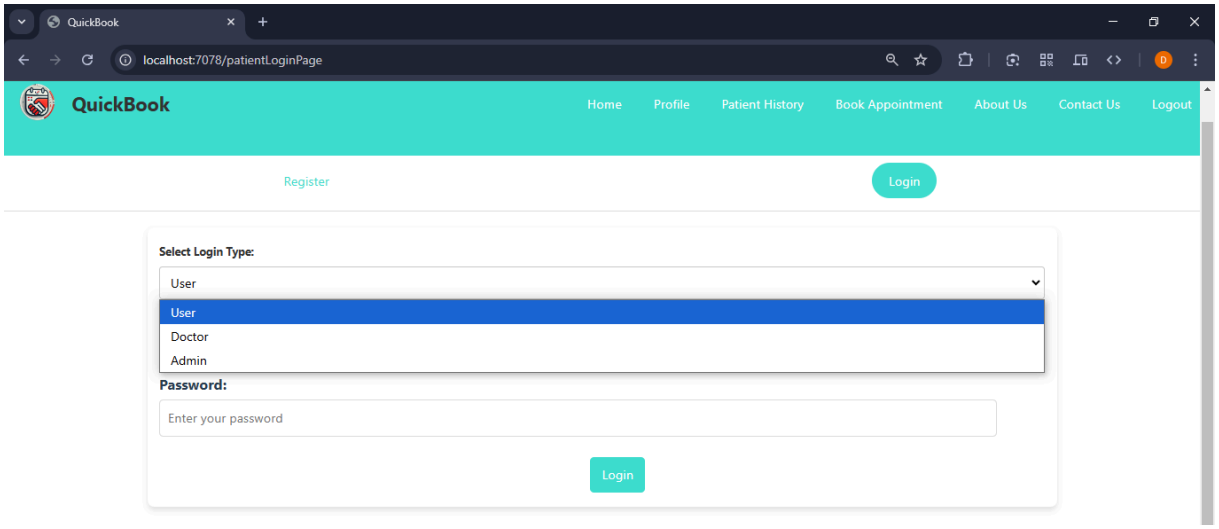

Figure 3: User Login Page(RBA)

Figure-2,3 : The authentication page allows new users to register and existing users to log in. Patients can book appointments, while doctors and admins can manage schedules upon successful login.

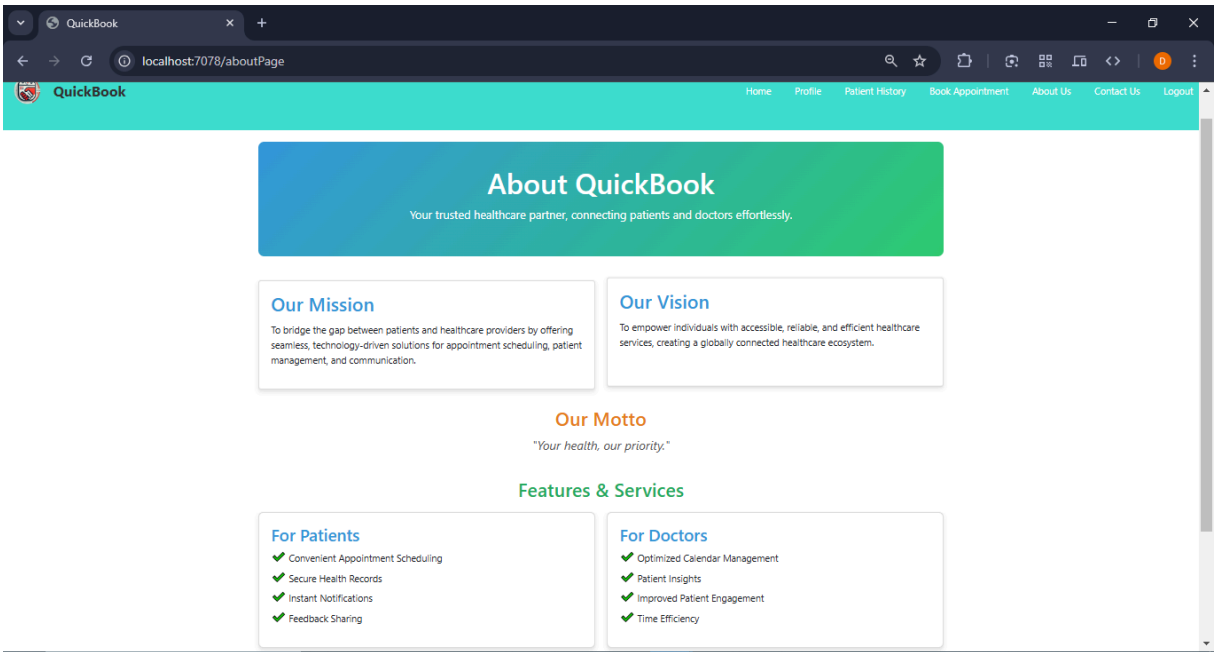

Figure 4: About Us Page

Figure-4 : This page highlights the platform's purpose, mission, and benefits for patients and doctors. It provides insights into how the system improves appointment management. Anyone visiting the platform can access this section.

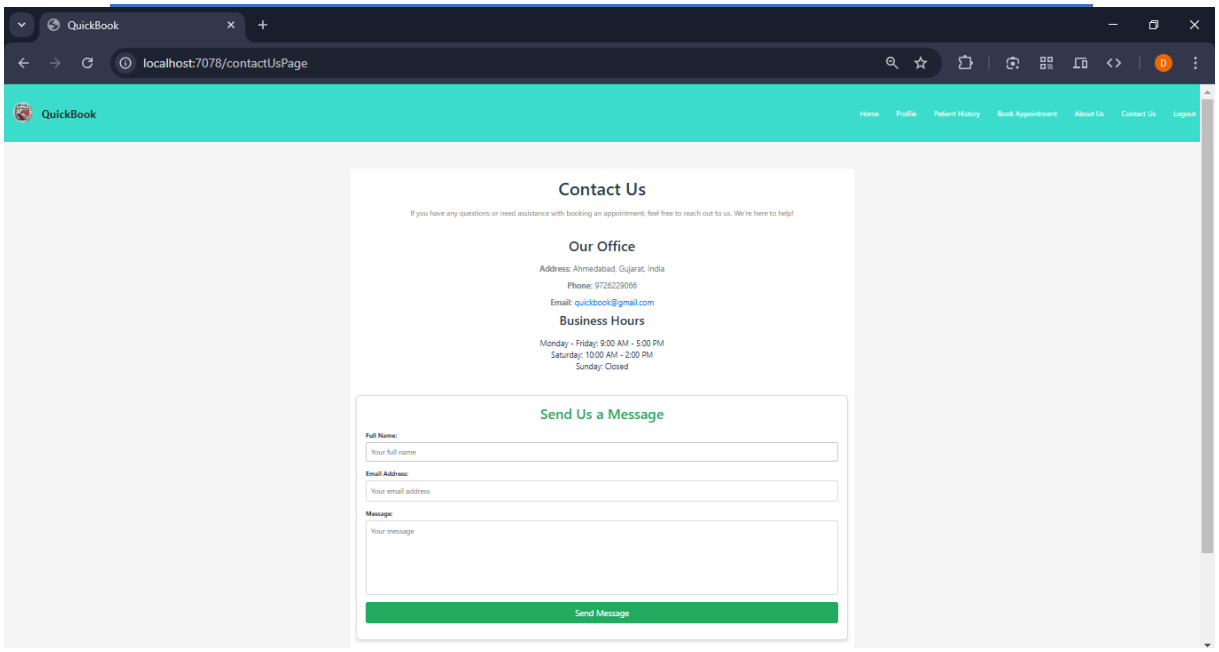

Figure 5: Contact Us Page

Figure-5 : A dedicated page where users can reach out for report an issue,support, inquiries, or feedback. It includes a form for queries and essential contact details. Accessible to all users, including patients, doctors, and administrators.

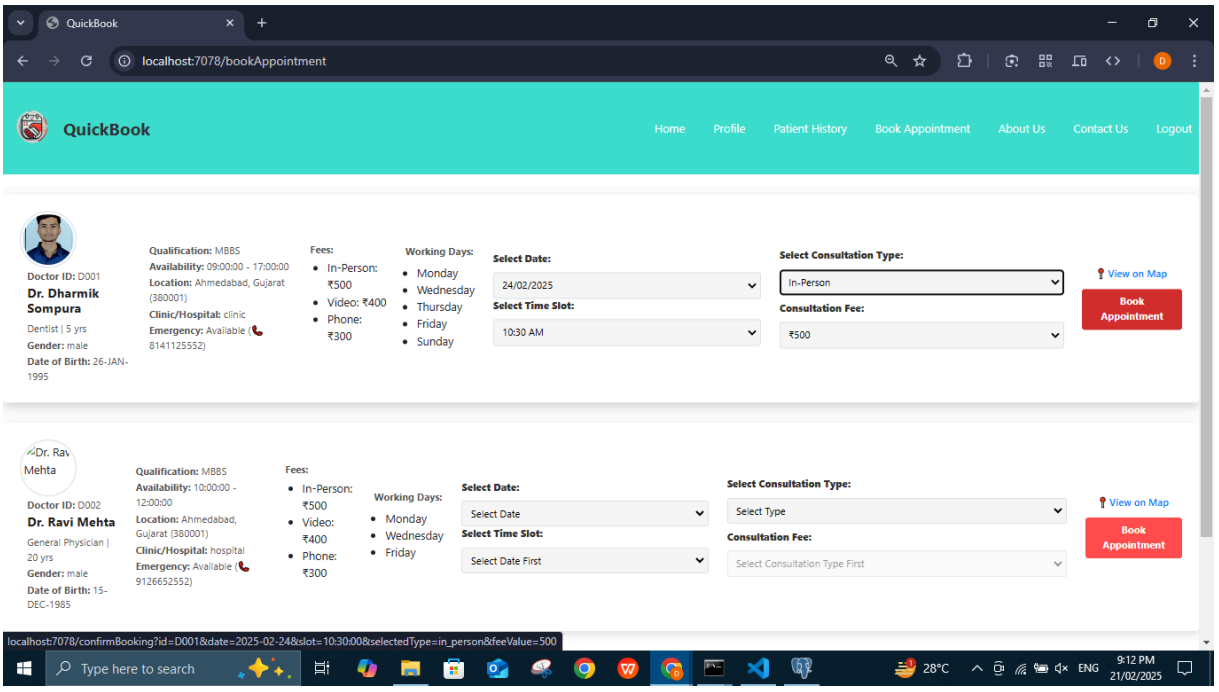

Figure 6: Book Appointment Page (Doctor's List)

Figure-6 : Users can search for available doctors based on specialization and book an appointment from this page. It displays doctor availability, helping patients make informed choices. Only registered users can book appointments.

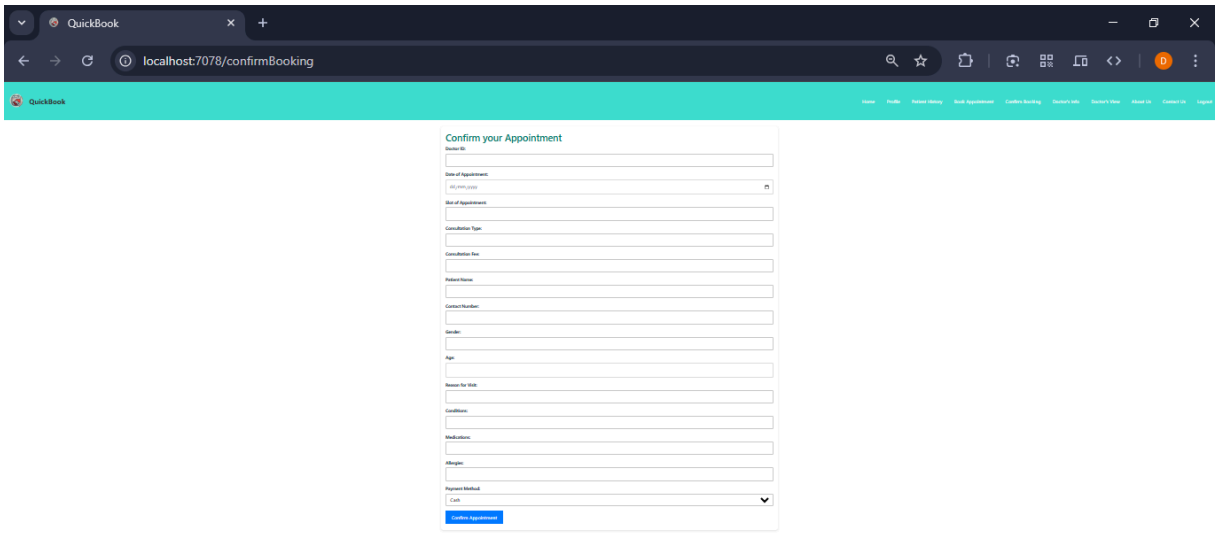

Figure 7: Confirm Appointment Page

Figure-7 : This page allows patients to review their appointment details—selected doctor, date, and time—before finalizing their booking. It ensures accuracy before confirmation. Accessible after selecting an appointment slot.

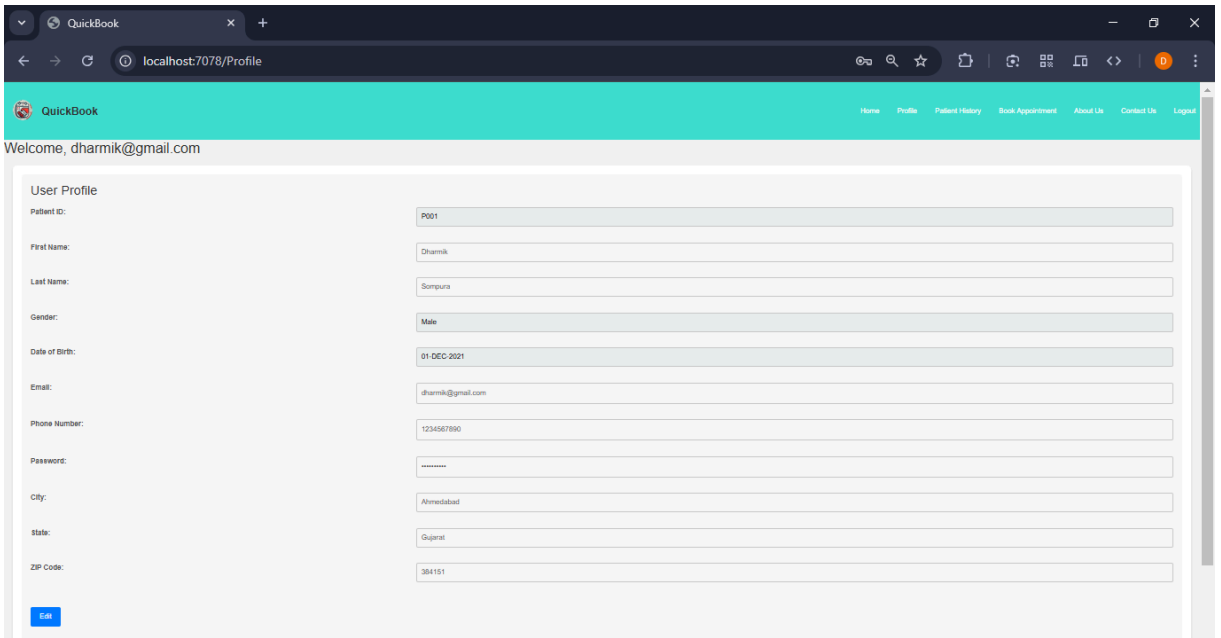

Figure 8: User Profile

Figure-8 : This section allows patients to update their personal details, manage appointments, and access account settings. It provides a streamlined experience for registered users only.

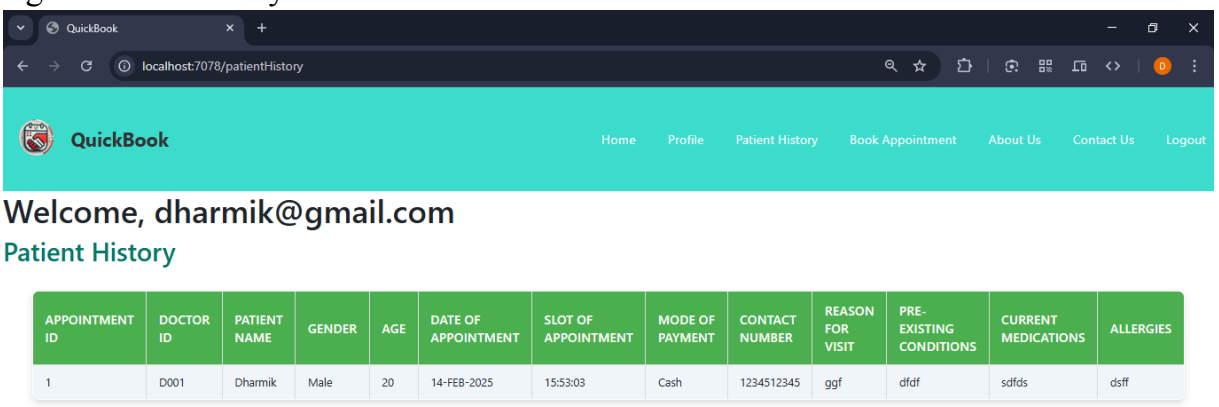

Figure 9: Patient History Page

Figure-9 : A detailed record of past appointments, including doctor visits and medical history. Patients can track their healthcare journey through this section. Accessible only to logged-in users.

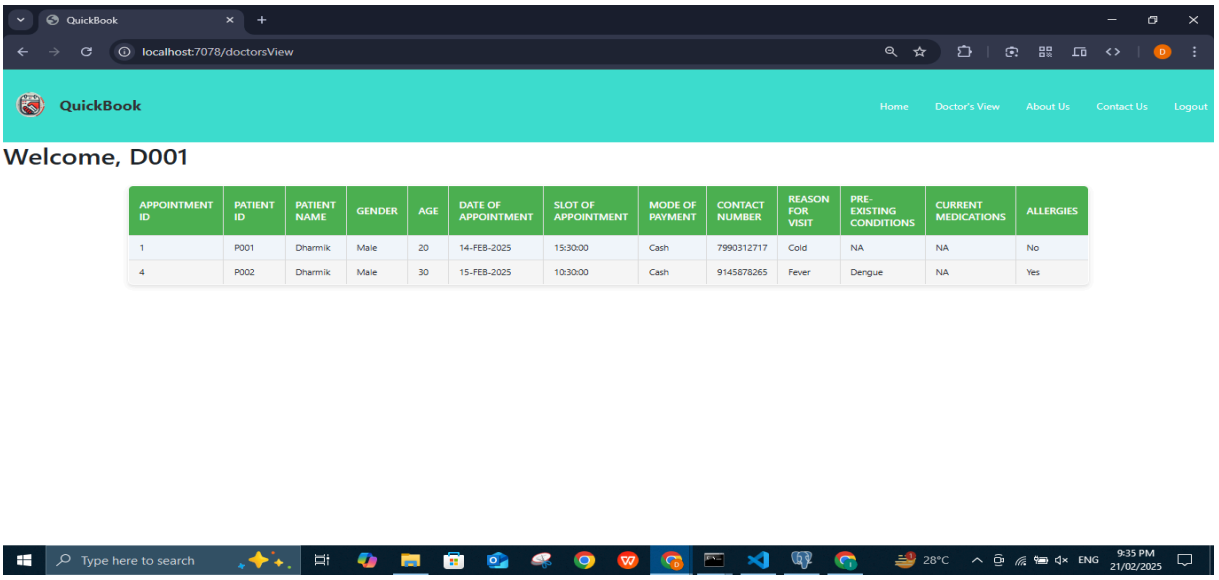

Figure 10: Doctor's View Page (For Appointment List)

Figure-10 : A personalized dashboard for doctors where they can manage their schedules, view upcoming appointments, update availability, and handle patient interactions. Accessible only to logged-in doctors.

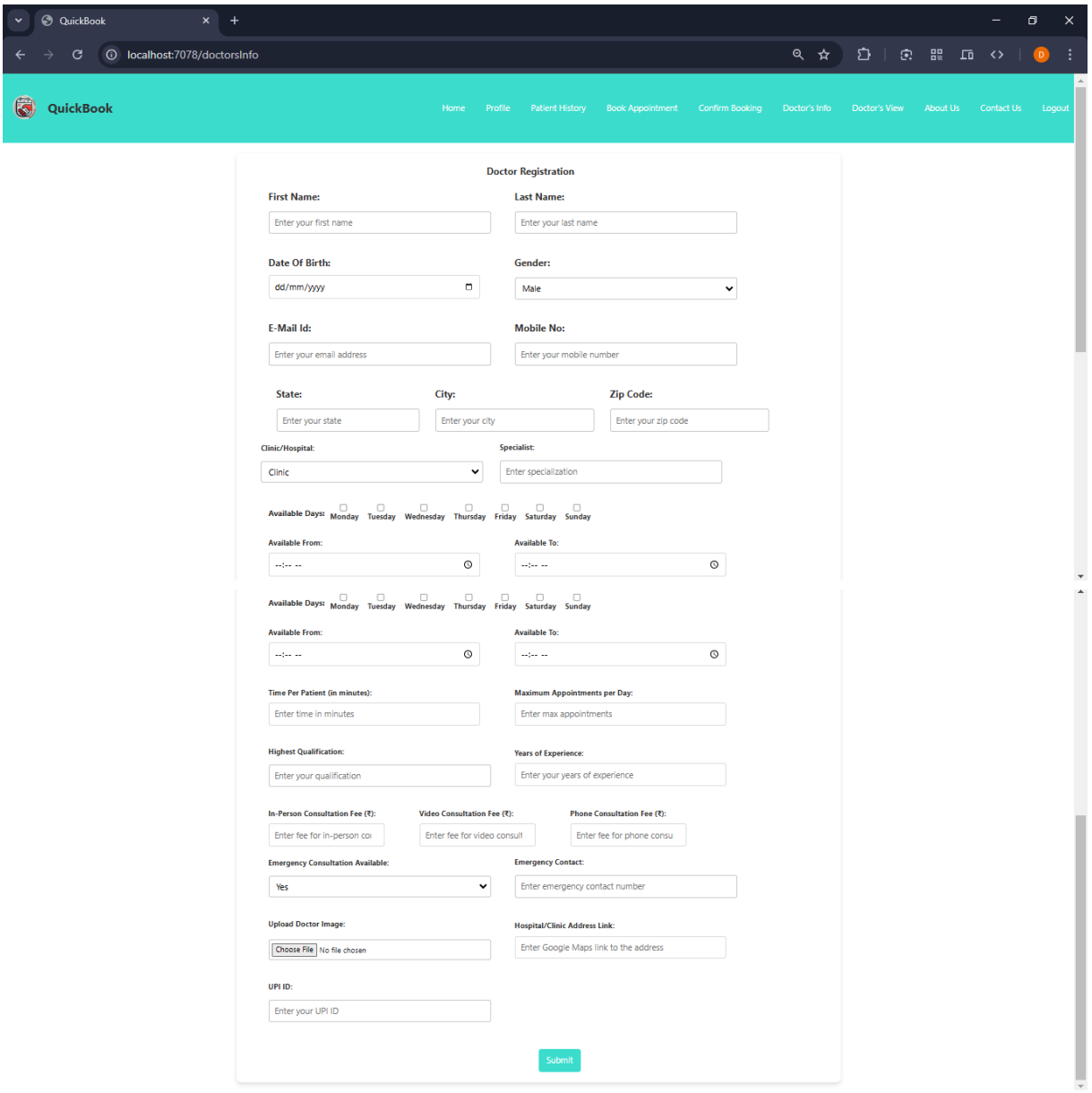

Figure 11,12 : Doctor's Registration Page (Admin only)

Figure-11,12 : A dedicated registration form for doctors, collecting details like specialization, availability, consultation fees, and emergency contacts. Only Admins can fill out this form.

GitHub Link :

https://github.com/sure-trust/SOMPURA-DHARMIK-MANISHKUMAR-g22-fsd (on SURE Trust Github Account)

https://github.com/Dharmik0712/Healthcare_Appointment_Scheduling_System (on personal Github Account)

Learning and Reflection

New Learnings

- **Full-Stack Web Development with Flask and FastAPI**

  I gained hands-on experience in developing full-stack web applications using Flask and FastAPI, improving my understanding of backend API development, routing, request handling, and data serialization.

- **Database Management and Query Optimization**

  Working with PostgreSQL, I learned about database schema design, indexing, and query optimization to improve data retrieval efficiency. Managing doctor availability, patient history, and appointment bookings required careful database structuring to ensure scalability and performance.

- **Asynchronous Web Updates with AJAX and jQuery**

  To ensure a seamless user experience, I implemented AJAX-based dynamic content updates, which eliminated unnecessary page reloads and improved responsiveness. This was crucial in updating real-time doctor availability and reflecting appointment changes instantly.

- **API Development and Documentation with Swagger**

  Developing APIs for the project taught me how to structure and document them efficiently using Swagger, making them easier to test, debug, and integrate with the frontend.

- **Role-Based Access Control (RBAC) Implementation**

  I implemented role-based authentication for patients, doctors, and administrators, ensuring each user could only access the relevant functionalities. This reinforced my understanding of access control mechanisms, session handling, and security best practices.

- **Error Handling and Debugging**

  During development, I encountered various errors related to database transactions, API response delays, and session timeouts. These challenges taught me the importance of logging, debugging, and handling exceptions to improve system reliability.

- **Performance Optimization for Healthcare Scheduling**

  The project required efficient slot management to minimize booking conflicts and reduce waiting times. I learned how to optimize scheduling algorithms and database queries to ensure faster response times.

Handling Major Errors

- **Database Connection Issues**

  At times, the database connection would time out, causing failed API responses. I resolved this by optimizing connection pooling and implementing automatic reconnections.

- **Session Expiry Handling**

  Users faced unexpected logouts due to session expirations. Implementing session management techniques, including token-based authentication, improved the user experience.

- **AJAX Call Failures**

  AJAX requests sometimes failed due to CORS policy restrictions. I fixed this by configuring CORS settings in Flask and FastAPI to allow cross-origin requests securely.

- **Concurrency Issues in Booking System**

  Multiple users trying to book the same time slot caused conflicts. Implementing transaction locks and real-time availability checks solved this issue.

Experience & Reflection

This project enhanced my full-stack development skills, deepened my understanding of backend APIs, database optimization, and real-time web interactions, and reinforced my problem-solving abilities in handling complex system challenges. Working on a real-world healthcare scheduling system helped me appreciate the impact of technology in improving efficiency and user experience in the medical field.

Conclusion and Future Scope

Objectives

- Develop a robust and efficient **Healthcare Appointment Scheduling System** to streamline doctor-patient interactions.
- Enable **role-based access** for patients, doctors, and administrators, ensuring seamless functionality for all users.
- Provide **real-time availability tracking** of doctors to optimize appointment scheduling.
- Implement a **user-friendly interface** for easy navigation and accessibility across different user roles.
- Ensure **data security and privacy** for patient records, appointments, and doctor details.

Achievements

- Successfully built a **functional web application** that allows patients to book appointments and doctors to manage their schedules.
- Implemented **real-time appointment management**, reducing scheduling conflicts and improving efficiency.
- Developed **role-based authentication**, ensuring that only authorized users can access specific features.
- Enhanced **user experience** by integrating dynamic appointment booking, history tracking, and doctor profile management.
- Laid the foundation for future integration of **AI-driven scheduling** and **ML-based recommendations** for improved efficiency.

Conclusion

The Healthcare Appointment Scheduling System successfully addressed key challenges in appointment management by creating an efficient, structured, and user-friendly platform. Developed using Flask, FastAPI, PostgreSQL, and jQuery, the system streamlines doctor availability management, enhances patient engagement, and improves appointment tracking, significantly reducing wait times and optimizing healthcare resource allocation. With features such as role-based access, real-time appointment updates, and secure user authentication, the system ensures seamless interaction between patients and healthcare providers. The integration of asynchronous web updates and optimized database queries enhances performance, making the system highly responsive and reliable. Overall, the project achieved its objective of minimizing scheduling inefficiencies, improving the patient experience, and providing doctors with an organized and intuitive interface to manage their appointments effectively.

Future Scope

1. AI-Powered Dynamic Appointment Scheduling

Integrating Machine Learning (ML) and Reinforcement Learning to optimize doctor schedules dynamically, reducing patient wait times and improving resource allocation.

2. Symptom-Based Doctor Recommendation System

Implementing a collaborative filtering algorithm to recommend doctors based on patient symptoms, medical history, and specialist availability.

3. Doctor's View Enhancements

Enhancing the Doctor's Dashboard by allowing doctors to write prescriptions, add consultation notes, and mark/cancel/reschedule appointments within the system.

4. Patient Health Record System

Expanding the system to maintain electronic health records (EHRs) for patients, enabling better diagnosis and seamless tracking of medical history.

5. Integration of Payment Gateways: Adding multiple secure payment options for seamless transactions.

6. AI-Powered Chatbot for Appointment Assistance

Developing an AI chatbot to assist patients in booking appointments, providing instant responses to queries, and guiding them to the right specialist.

7. Performance and Security Enhancements

Optimizing database queries, implementing load balancing, and strengthening authentication mechanisms to handle larger user loads while maintaining system security.

By implementing these future enhancements, the Healthcare Appointment Scheduling System can become a comprehensive AI-driven healthcare management platform, improving accessibility and efficiency in medical appointment scheduling.