**DOCSPOT - Seamless Appointment Booking for Health**

**1. INTRODUCTION**

**1.1 Project Overview**

DocSpot is a full-stack web application that simplifies healthcare access by allowing patients to register, browse available doctors, and book appointments online. Admins can manage and approve/reject bookings, ensuring a streamlined system for both patients and healthcare professionals.

**1.2 Purpose**

To digitize the healthcare appointment process, minimizing manual work and improving accessibility for patients and operational efficiency for administrators.

**2. IDEATION PHASE**

**2.1 Problem Statement**

Patients face difficulty scheduling doctor appointments due to lack of online systems, resulting in long waiting times and miscommunication.

**2.2 Empathy Map Canvas**

* **Think & Feel:** Wants quick doctor access. Worries about long queues.
* **Hear:** Feedback from friends on delayed appointments.
* **See:** Long hospital lines. Complex manual registration forms.
* **Say & Do:** Expresses frustration. Looks for alternatives online.
* **Pain:** Unclear availability, travel hassle, long waiting.
* **Gain:** Quick booking confirmation, visible doctor availability.

**2.3 Brainstorming**

* Booking interface for users
* Filtering doctors by specialization
* Admin approval system
* View personal bookings dashboard
* Email/SMS notifications (future scope)

**3. REQUIREMENT ANALYSIS**

**3.1 Customer Journey Map**

1. User registers on the platform
2. Logs in and browses doctors
3. Books an appointment
4. Admin approves/rejects the booking
5. User views confirmation

**3.2 Solution Requirements**

* User registration/login
* Doctor listing and filtering
* Appointment booking
* Admin dashboard for approvals
* Booking status updates

**3.3 Data Flow Diagram**

****

**3.4 Technology Stack**

* **Frontend:** React.js, Axios, Bootstrap
* **Backend:** Node.js, Express.js
* **Database:** MongoDB (Atlas)
* **Others:** Thunder Client, Mongoose

**4. PROJECT DESIGN**

**4.1 Problem-Solution Fit**

* **Problem:** Patients face hurdles in appointment booking.
* **Solution:** Online booking system with user/admin flows and doctor listings.

**4.2 Proposed Solution**

Provide user-friendly online portal to book and manage healthcare appointments with real-time updates.

**4.3 Solution Architecture**

Client (React.js) → Axios → Express.js API → MongoDB

**5. PROJECT PLANNING & SCHEDULING**

**5.1 Project Planning (Sprints & Backlogs)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement** | **User Story No** | **User Story / Task** | **Points** | **Priority** |
| 1 | User Registration | US-1 | Register with email/password | 2 | High |
| 1 | User Login | US-2 | Login with email/password | 1 | High |
| 2 | Doctor Listing | US-3 | View doctors by specialization | 2 | High |
| 3 | Book Appointment | US-4 | Book doctor slot | 3 | High |
| 4 | Admin Panel | US-5 | Approve/Reject booking | 3 | Medium |

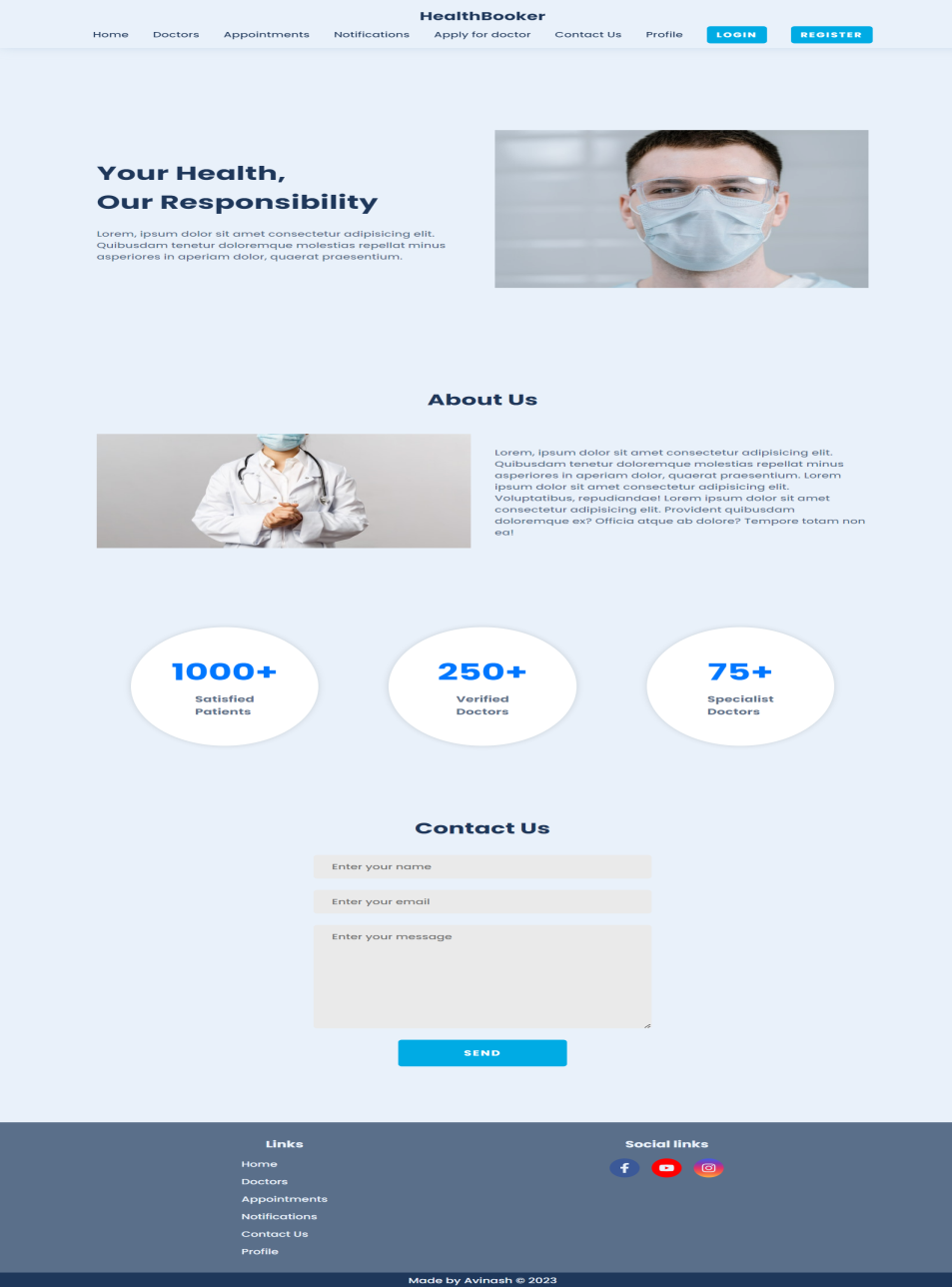
**6. FUNCTIONAL AND PERFORMANCE TESTING**

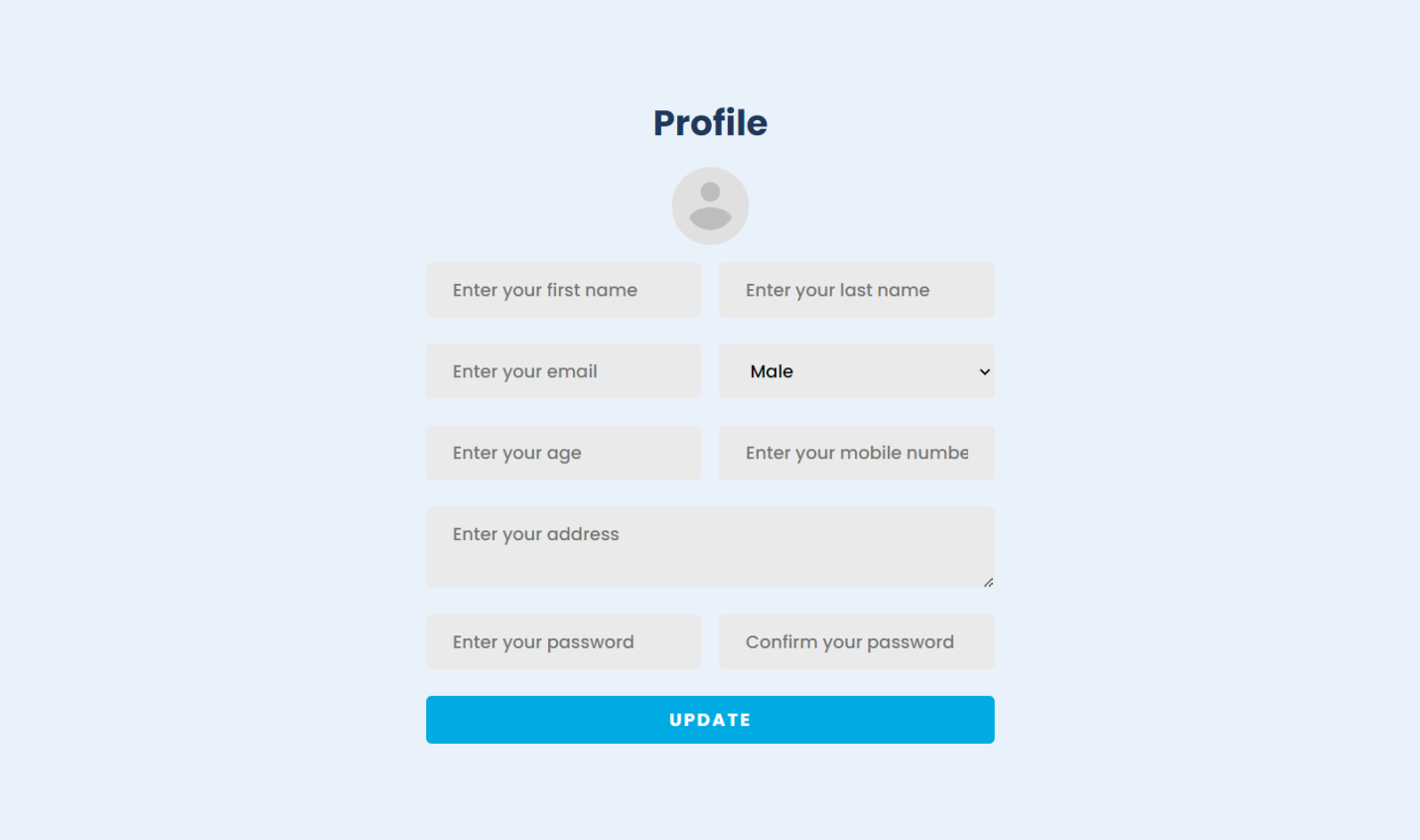
**6.1 Performance Testing**

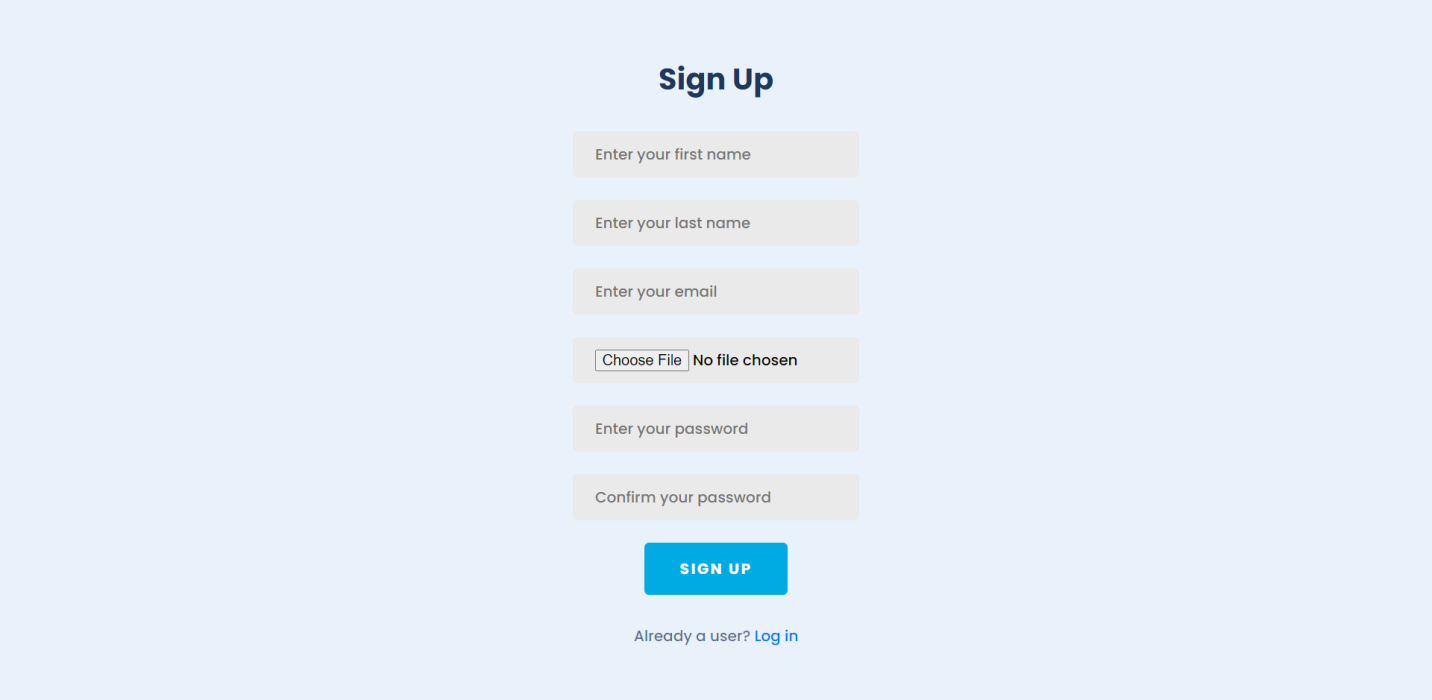
* Load tested API using Thunder Client
* Confirmed MongoDB query times remain optimal (<150ms)

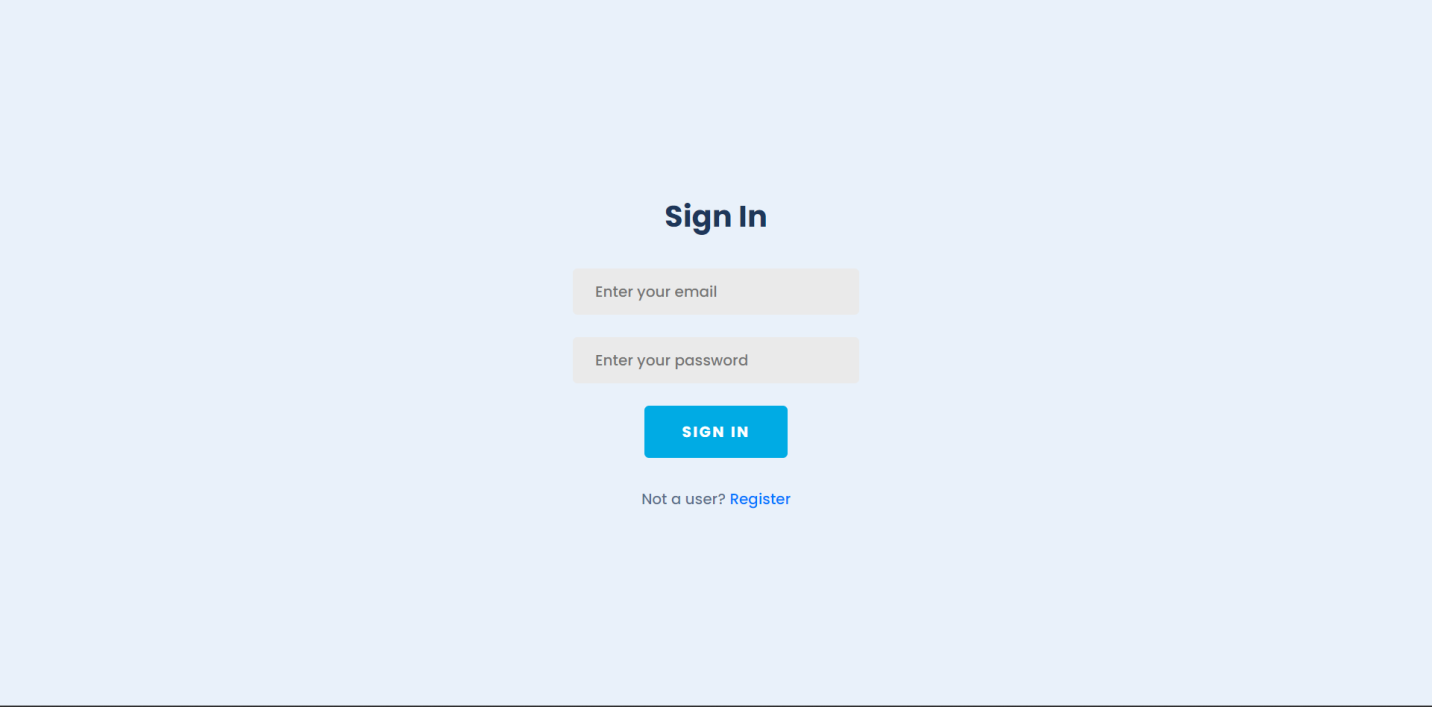
**7. RESULTS**

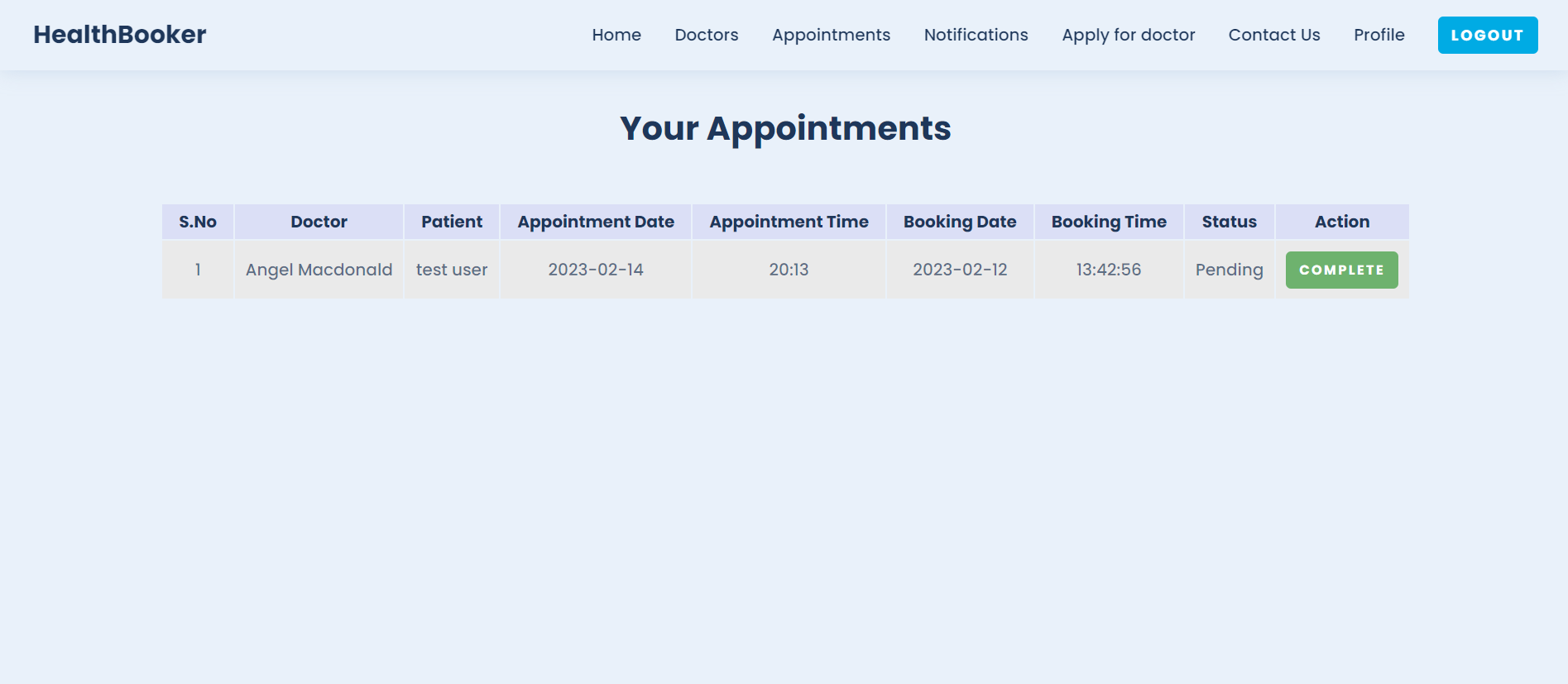
**7.1 Output Screenshots**











**8. ADVANTAGES & DISADVANTAGES**

**Advantages**

* Simplifies doctor access
* Reduces manual effort
* Scalable & modular design

**Disadvantages**

* No email/SMS notification yet
* No real-time doctor calendar integration (future scope)

**9. CONCLUSION**

DocSpot successfully solves the problem of manual appointment handling through an intuitive web interface, empowering both users and admins with easy and efficient booking management.

**10. FUTURE SCOPE**

* Email/SMS confirmation integration
* Doctor-side login & schedule updates
* Calendar integration with Google Calendar
* Payment gateway

**11. APPENDIX**

* **GitHub Link:** [To be added]
* **Demo Video Link:**
* **Source Code:** Available in project files