Team ID : SWTID1743701170

Project Report: Book A Doc

---

1. INTRODUCTION

1.1 Project Overview

The healthcare industry is evolving rapidly, yet a significant gap still exists in how patients connect with doctors, especially when it comes to scheduling consultations. "*Book A Doc*" aims to fill that gap by providing a reliable, intuitive, and fully digital appointment booking platform.

This MERN Stack web application allows patients to search for doctors by specialization, location, or availability, and schedule appointments without needing to physically visit a clinic or hospital beforehand. At the same time, doctors get a centralized dashboard to manage their schedules, patient appointments, and logs.

With a focus on accessibility and user-friendliness, Book A Doc ensures that even users with minimal tech knowledge can book appointments with ease. It's a step toward building a smarter, faster, and patient-centered healthcare ecosystem.

1.2 Purpose

The core purpose of Book A Doc is to **reduce the friction** in healthcare access. Long queues, manual appointment entries, and lack of doctor availability information create inefficiencies in the current system. Our project aims to:

- Enable online doctor appointment scheduling
- Provide doctors with a personalized dashboard to manage their slots
- Offer administrators tools to oversee operations efficiently
- Improve patient experience by cutting wait time and confusion

Ultimately, this platform aspires to bring transparency and convenience to both patients and healthcare providers.

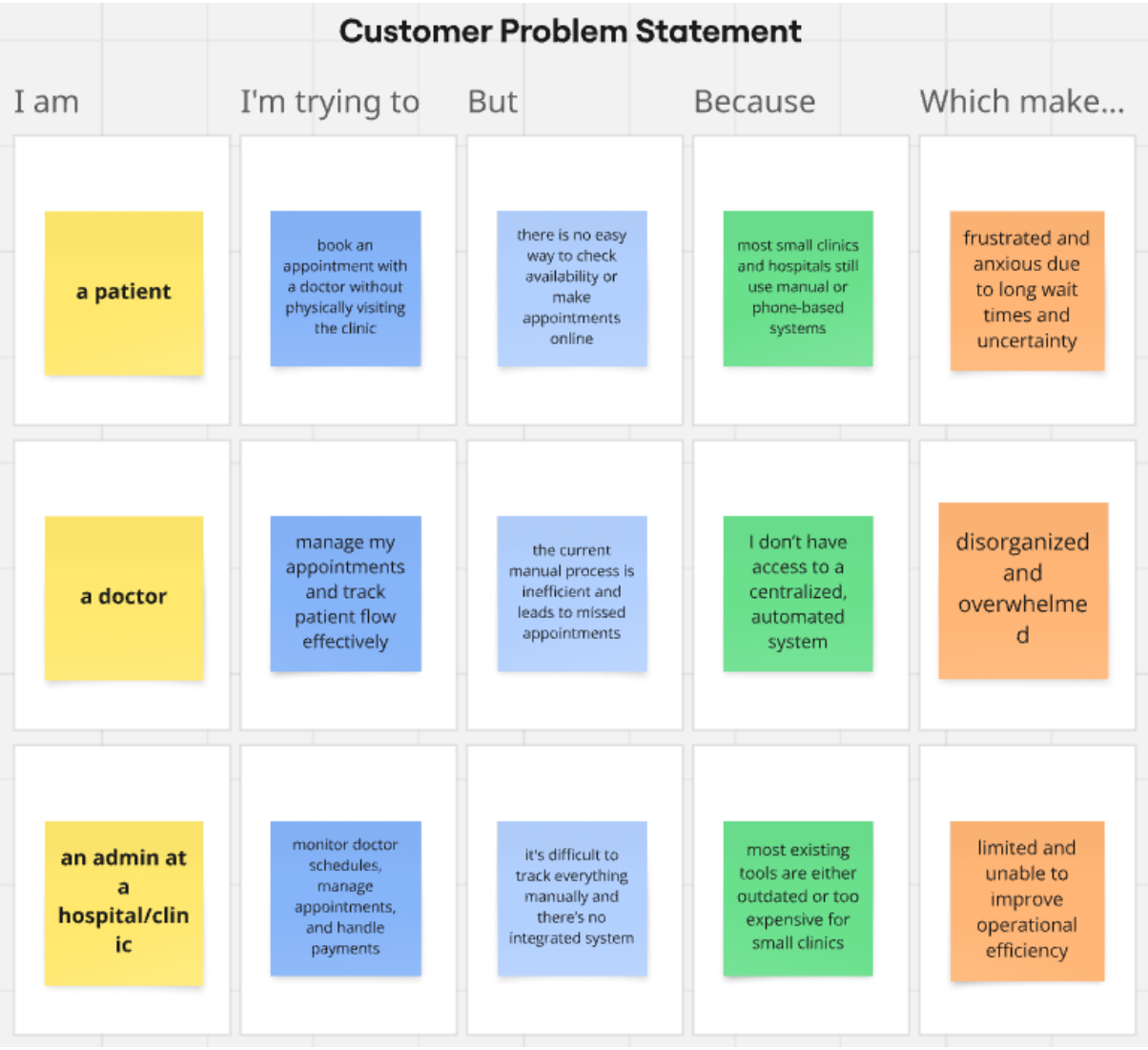
Team ID : SWTID1743701170

2. IDEATION PHASE

2.1 Problem Statement

The traditional healthcare appointment system still relies heavily on offline methods—patients must physically visit clinics to get appointments, often encountering long queues, mismatched schedules, or unavailability of doctors.

Moreover, there's no efficient way for doctors to manage their appointments or patient logs in real-time. This disconnect causes stress for patients and reduces operational efficiency for healthcare professionals.

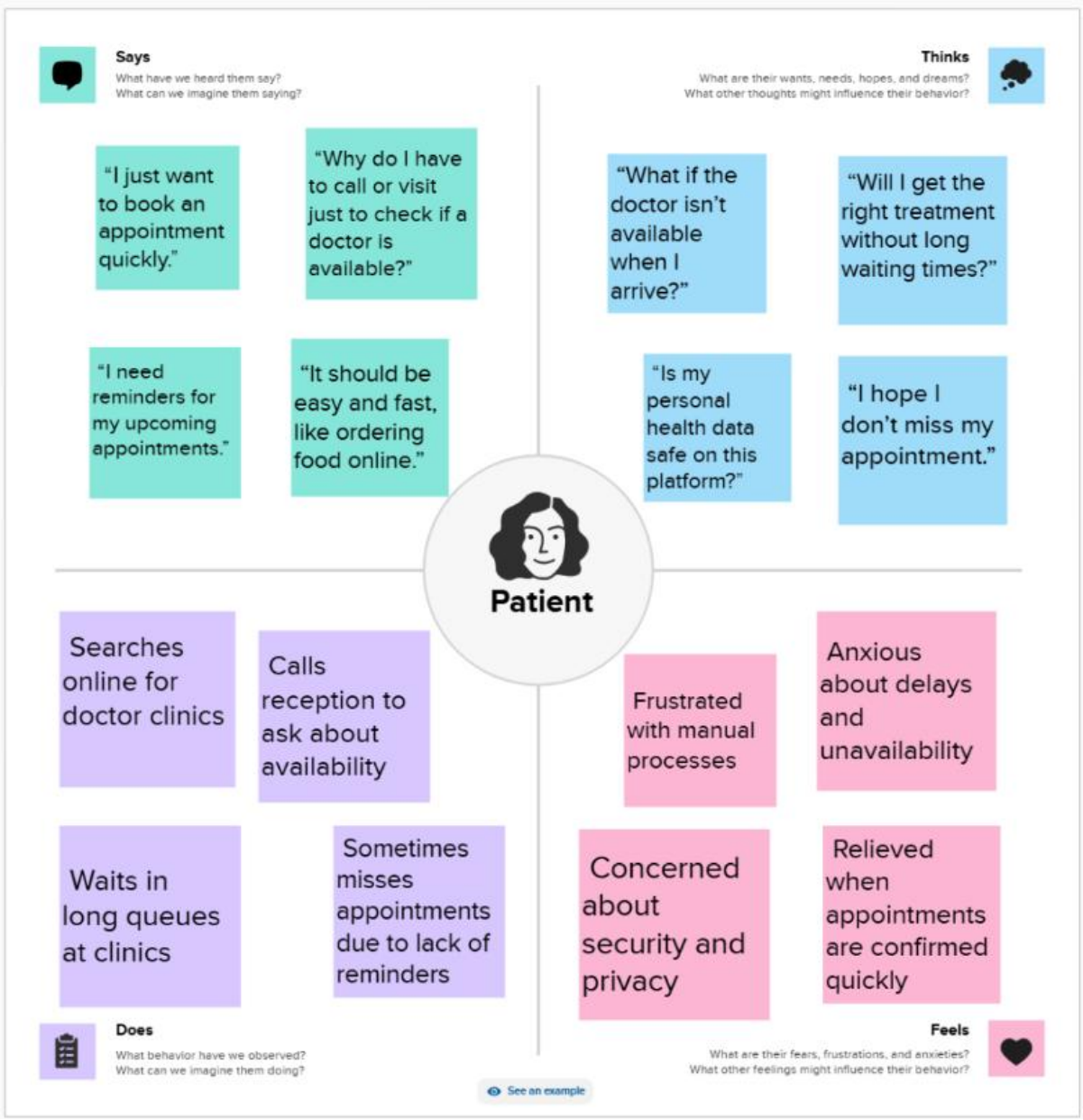


Team ID : SWTID1743701170

2.2 Empathy Map Canvas

Section	Patients	Doctors
Says	"I need a doctor urgently."	"I want a simpler way to manage appointments."
Thinks	"Will I find the right doctor in time?"	"I hope my appointments aren't overlapping."
Does	Calls hospitals, waits in queues	Uses WhatsApp or manual registers
Feels	Stressed, anxious, confused	Overwhelmed, disorganized

This empathy map helped us visualize user frustrations and design a system that meets real needs.

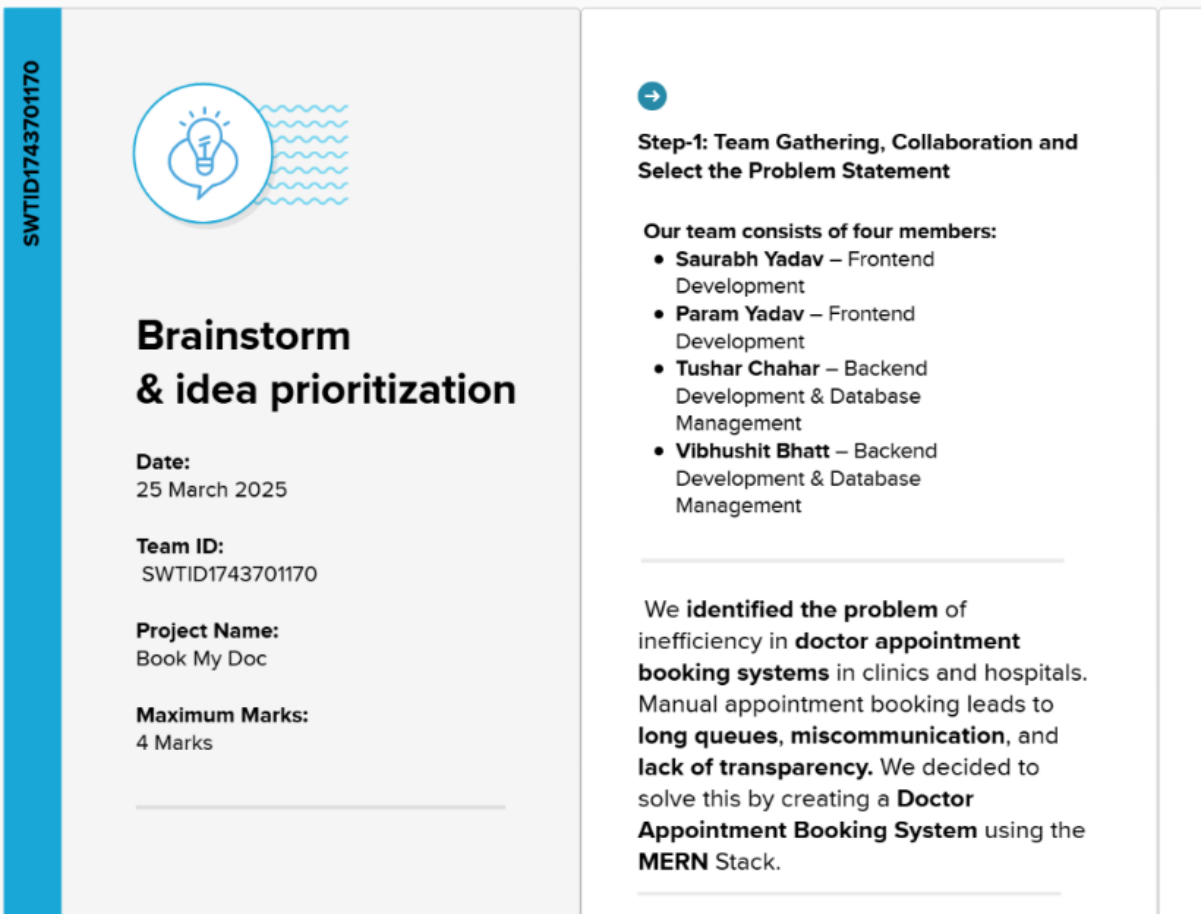


Team ID : SWTID1743701170

2.3 Brainstorming

The team brainstormed pain points and potential solutions in multiple sessions. Key takeaways:

- Patients need **search filters** (specialization, availability).
- Doctors need **appointment dashboards** with real-time updates.
- Admins should be able to **manage users and validate doctors.**
- The UI must be **clean, mobile-friendly, and fast.**
- Use **JWT authentication** for security and **REST APIs** for modular design.



Team ID : SWTID1743701170

1

Step-2: Brainstorm, Idea Listing and Grouping

Raw Ideas Generated:

- Multi-user login system (Admin, Doctor, Patient)
- Real-time appointment booking with time slots
- Doctor profile management
- Patient dashboard for managing appointments
- Admin dashboard for managing doctors and appointments
- Online payment integration for appointment fees
- Responsive frontend using React
- RESTful APIs using Express and Node.js
- MongoDB for database management
- Authentication using JWT
- Profile and earning analytics for doctors
- Email/SMS notification system

Grouped Ideas:

- User Access & Roles: Patient login, Doctor login, Admin login
- Appointments: Booking, cancellation, time slot management
- Payment Integration: Online payment system for appointment fees
- Data Management: MongoDB for storing users, appointments, earnings
- Frontend Design: User-friendly interface using React.js
- Backend System: REST APIs using Express.js and Node.js
- Security: Secure login with JWT, role-based access
- Dashboards: Separate dashboards for Admin, Doctors, and Patients
- Notifications: Email reminders for appointments (future enhancement)

Team ID : SWTID1743701170

2

Step-3: Idea Prioritization

High Priority:
- Patient, Doctor, Admin login system
- Appointment booking & management system
- Doctor profile management
- Admin dashboard with control over all operations
- Secure backend with role-based access
- Online payment integration

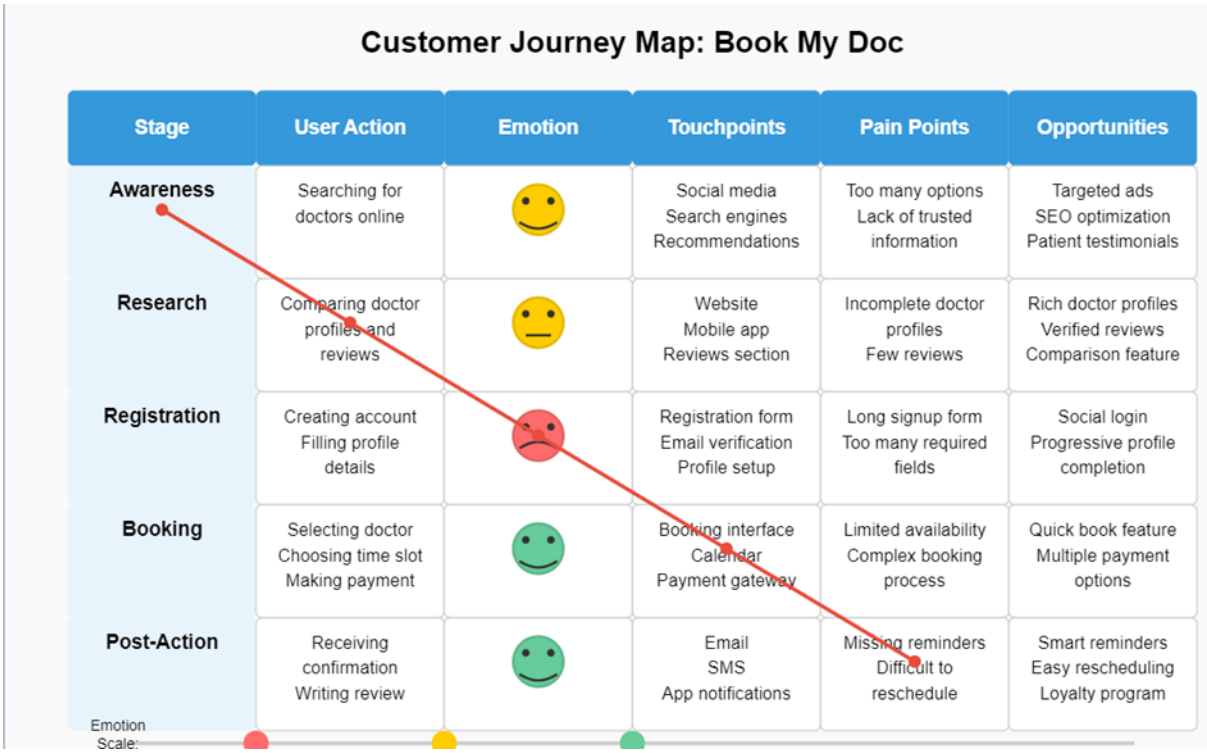
Medium Priority:
- Notification system (Email/SMS)
- Analytics on earnings and appointments
- Detailed appointment history for users

Low Priority:
- Review & Rating system for doctors
- Chat feature between doctor and patient
- Mobile app version

Team ID : SWTID1743701170

3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map



Patient's Journey:

1. Registers/Login
2. Searches for a doctor by specialization or name
3. Views doctor availability
4. Books an appointment slot
5. Receives confirmation
6. Attends the session
7. Optionally receives a visit summary or notes

Doctor's Journey:

1. Registers/Login
2. Verifies patients and manages schedule
3. Views and confirms appointments
4. Updates patient records post consultation

3.2 Solution Requirement

Functional Requirements:

- Registration/Login for doctors, patients, and admins

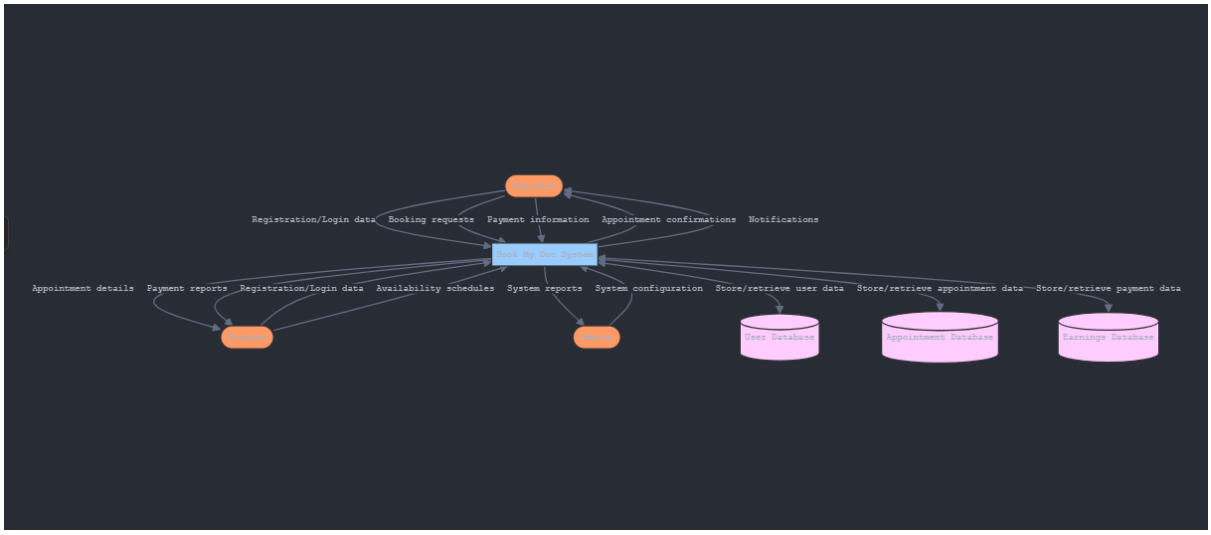
- Profile creation and editing
- Appointment booking with available time slots
- Notifications for appointment confirmation
- Admin approval of doctor accounts

Non-Functional Requirements:

- Responsive web design
- Fast backend API responses
- Secure data handling
- Scalable database structure

3.3 Data Flow Diagram (DFD)

- **Level 0:**
  User → Web Interface → API → MongoDB
  ↳ Outputs: Confirmation, Profiles, Bookings
- **Level 1:**
  - **Auth Module**: Handles login, JWT token generation
  - **Booking Module**: Appointment CRUD operations
  - **Admin Module**: Role-based access and approvals



3.4 Technology Stack

Layer	Tech Used
Frontend	React.js, Tailwind CSS
Backend	Node.js, Express.js
Database	MongoDB

Team ID : SWTID1743701170

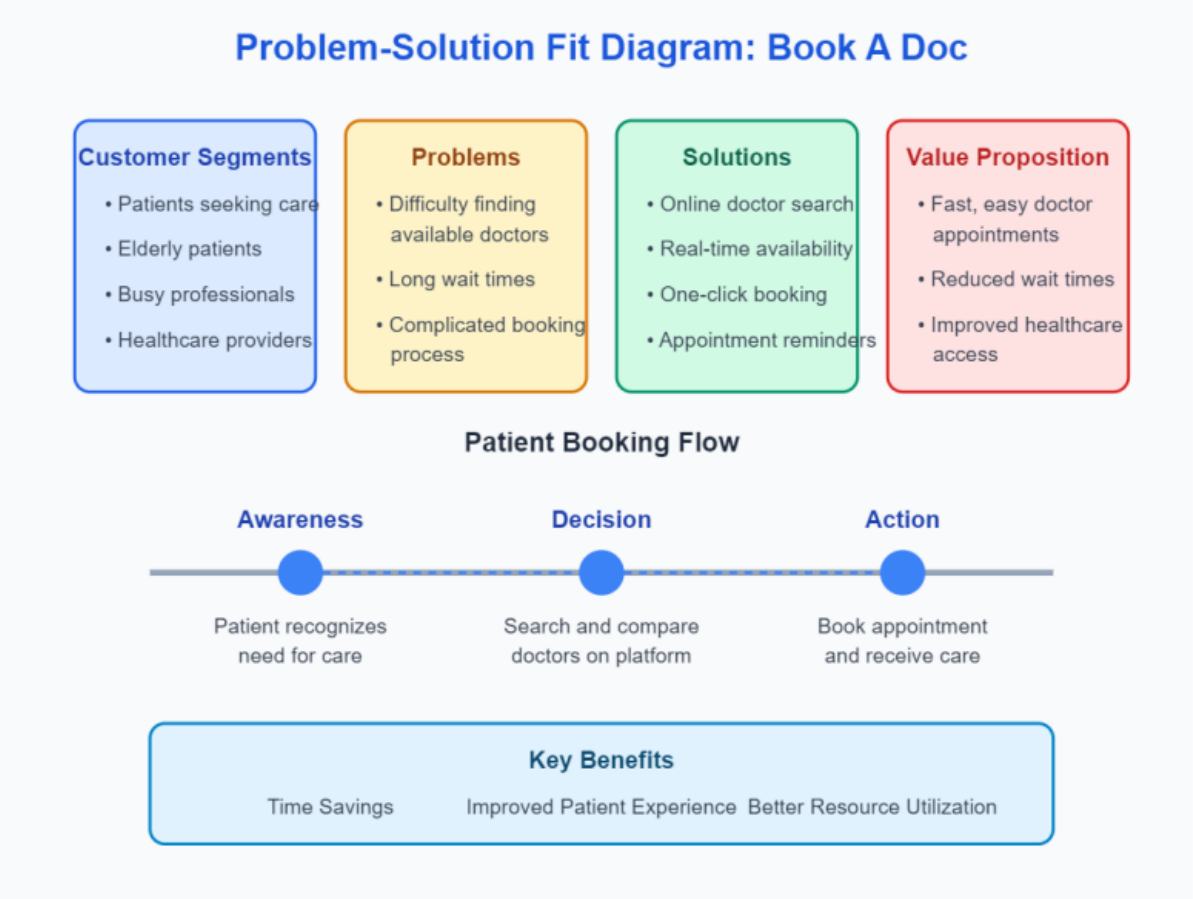
Layer	Tech Used
Authentication	JWT, Bcrypt
Deployment	Render (Backend), Netlify (Frontend)

---

4. PROJECT DESIGN

4.1 Problem-Solution Fit

There is a direct fit between the issues observed in the healthcare appointment flow and what Book A Doc offers. By streamlining booking and scheduling, we reduce wait times, eliminate manual logs, and give control back to both patients and doctors.



4.2 Proposed Solution

Book A Doc features three primary roles:

- **Patient**: Book, view, and cancel appointments
- **Doctor**: Manage slots, view bookings, update patient notes
- **Admin**: Manage users, approve doctors, oversee data consistency

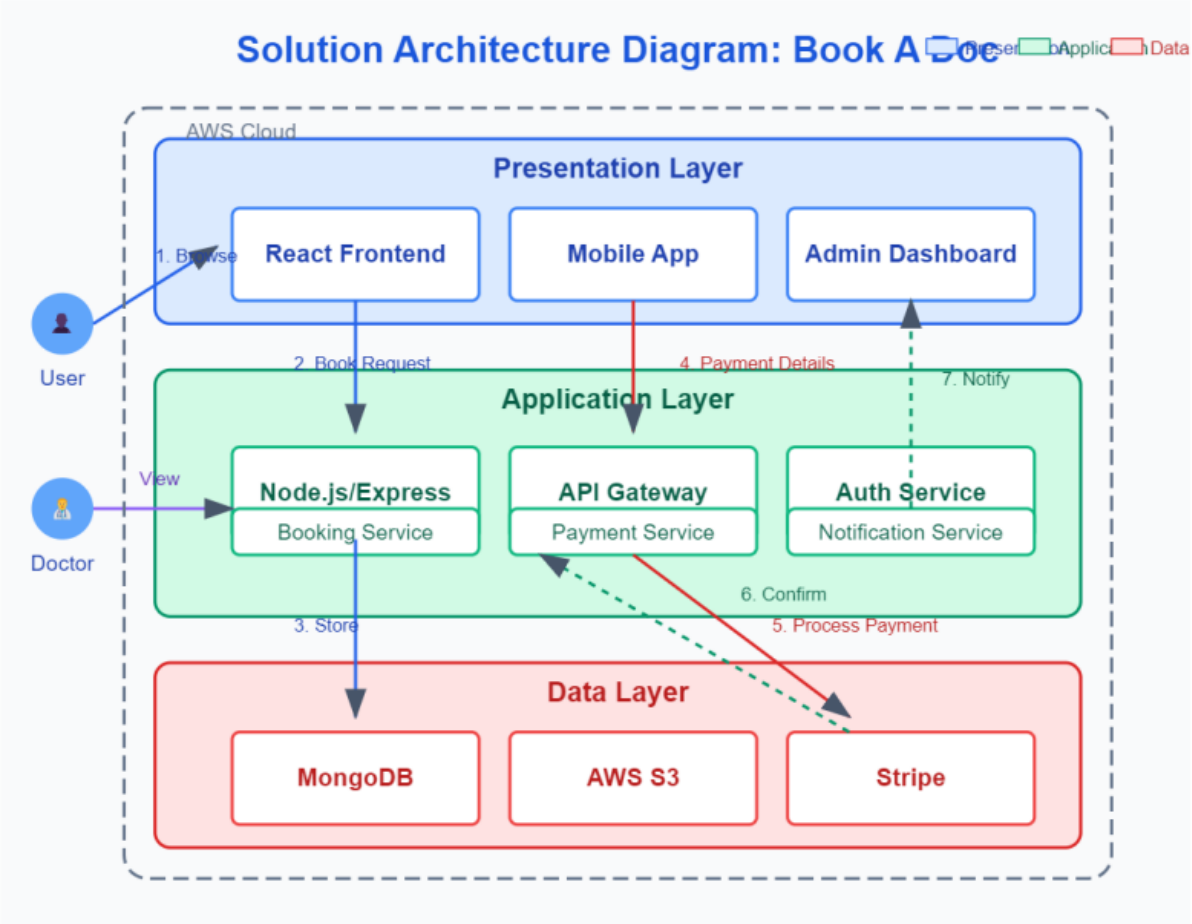
The system also includes:

- Secure session management with JWT

- REST APIs to separate backend logic from frontend
- Responsive design for mobile and desktop

4.3 Solution Architecture

Frontend (React) ⇄ **Backend** (Express API) ⇄ **Database** (MongoDB)
↳ JWT Auth Layer
↳ Role-Based Access



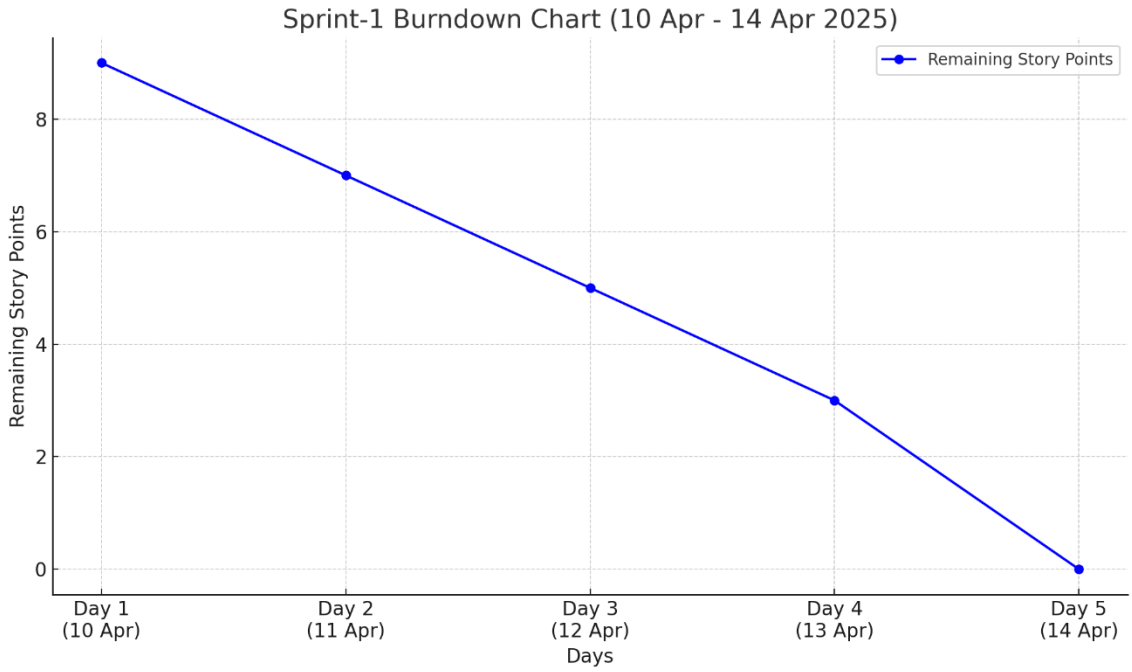
Team ID : SWTID1743701170

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Week	Task	Members
1	Frontend (UI Design, Components)	Saurabh, Tushar
2	Backend API (Routes, DB Models)	Param, Vibhushit
3	Integration (Frontend + Backend)	All
4	Testing, Bug Fixing, Deployment	All

We followed an **Agile-inspired approach**, meeting after each major milestone to review and adjust plans.



---

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

We tested the application using:

- **Postman** for API stress testing
- **Lighthouse** (Chrome DevTools) for frontend performance
- **Manual testing** for all user roles and edge cases

Key Results:

- Login API latency: ~150ms
- Booking module response: ~200ms

Team ID : SWTID1743701170

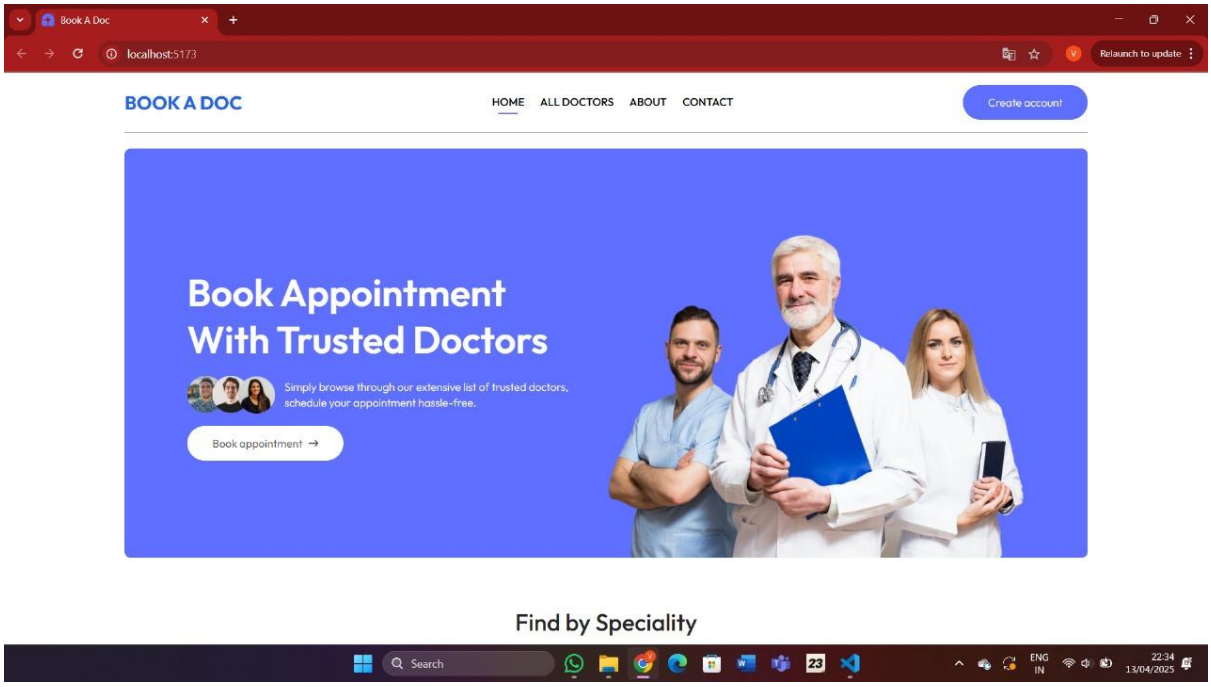
- Lighthouse performance score: 90+

---

7. RESULTS

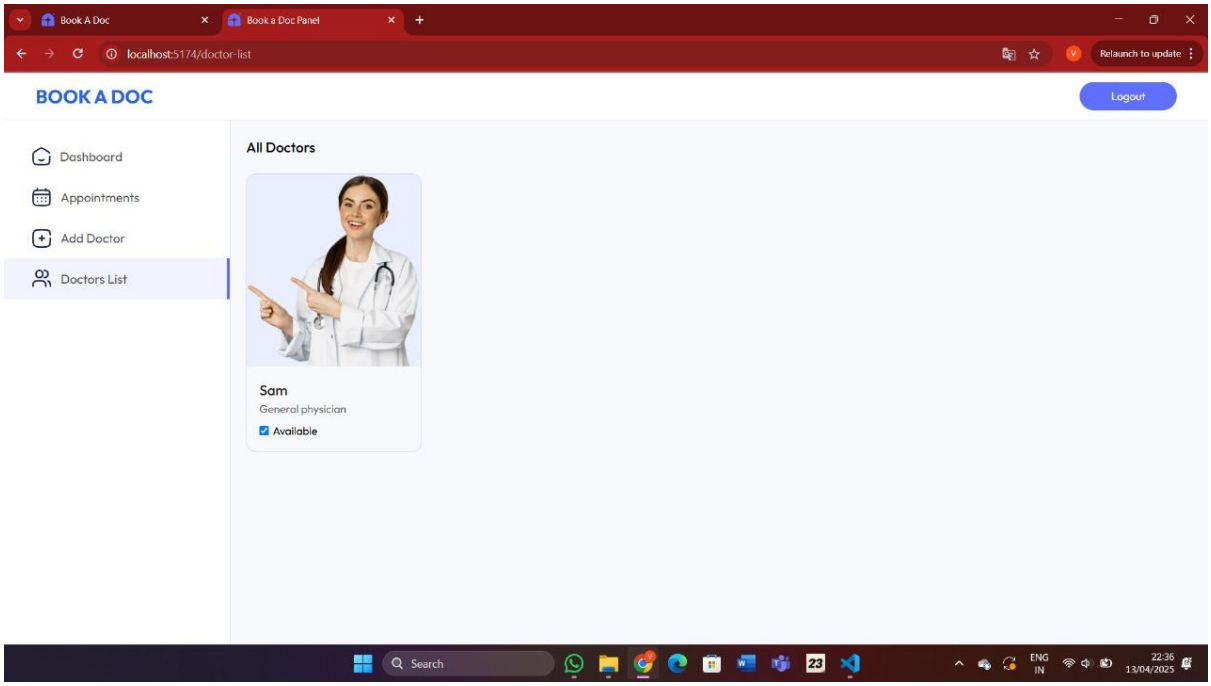
7.1 Output Screenshots

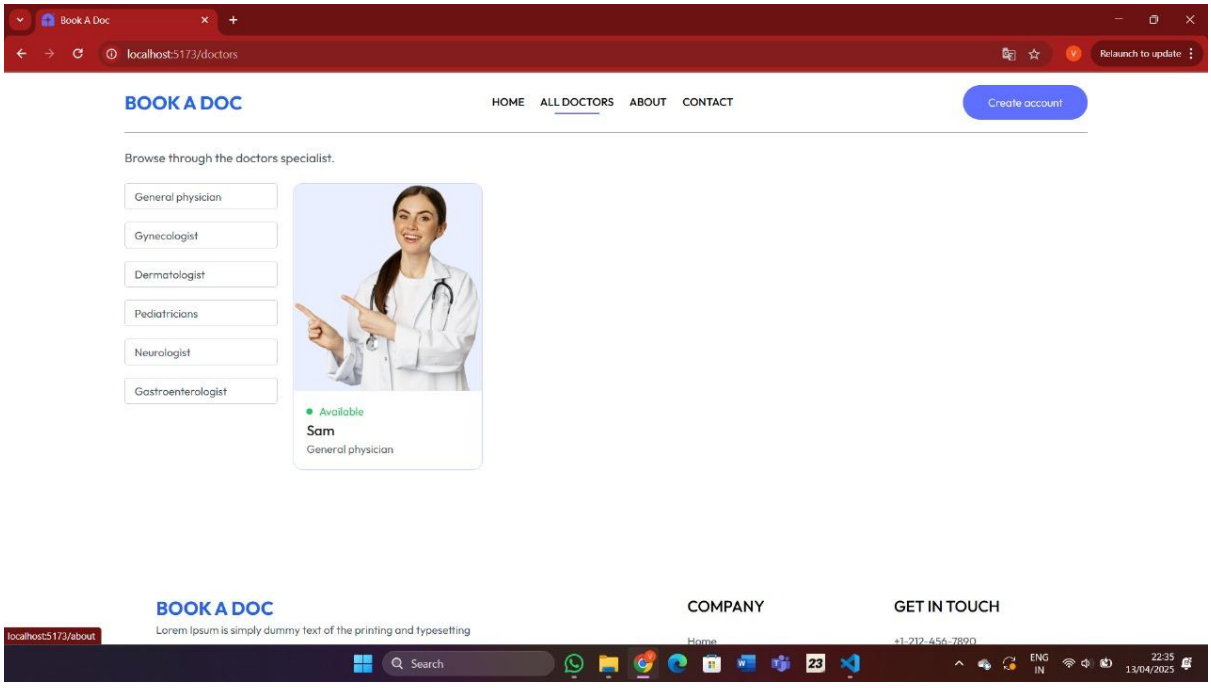
- **Homepage**



Team ID : SWTID1743701170

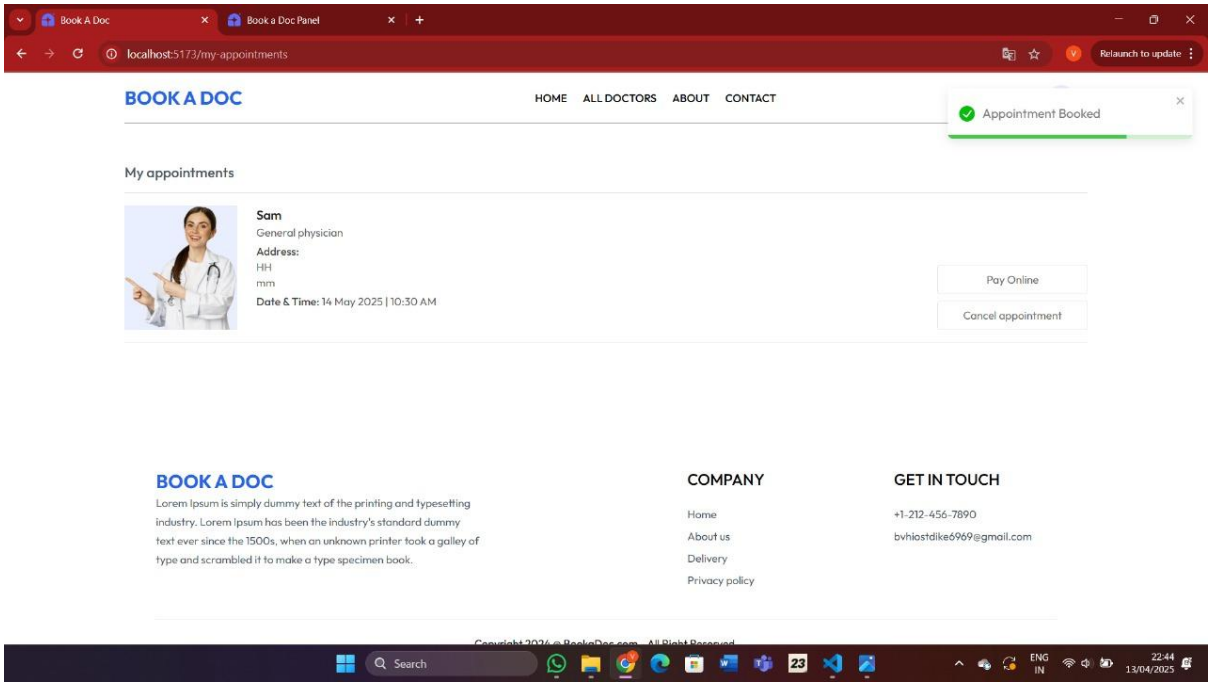
- **Doctor List View**



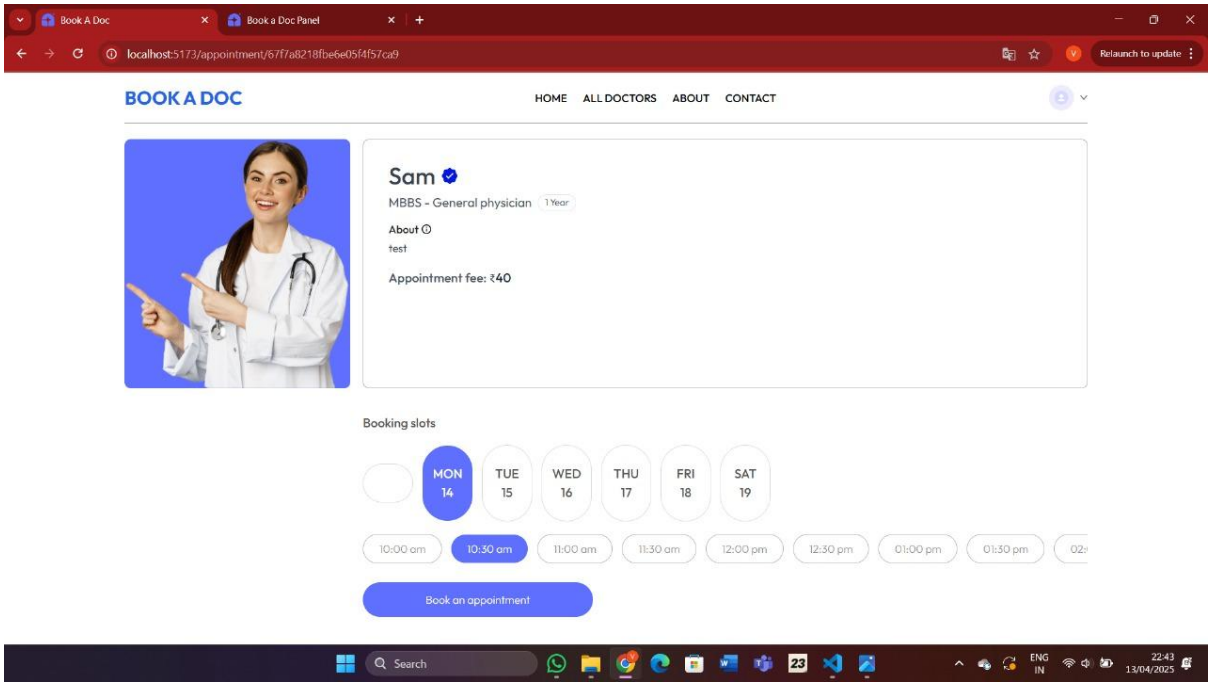


Team ID : SWTID1743701170

- **Patient Appointment Dashboard**

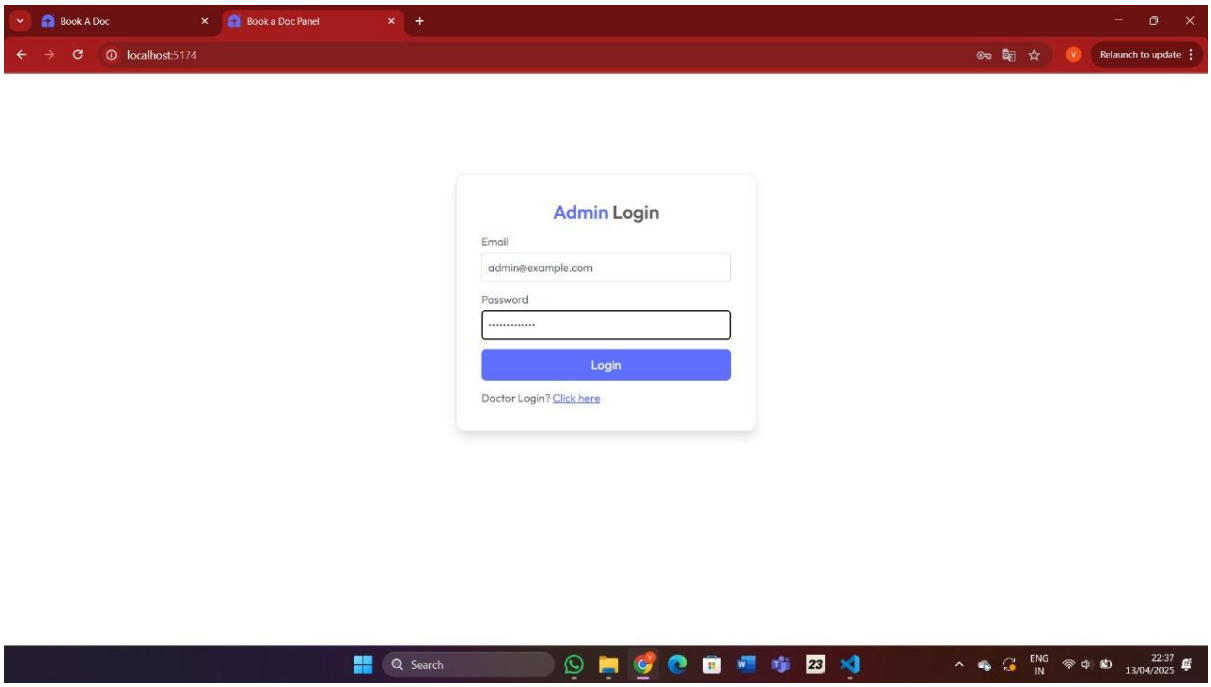


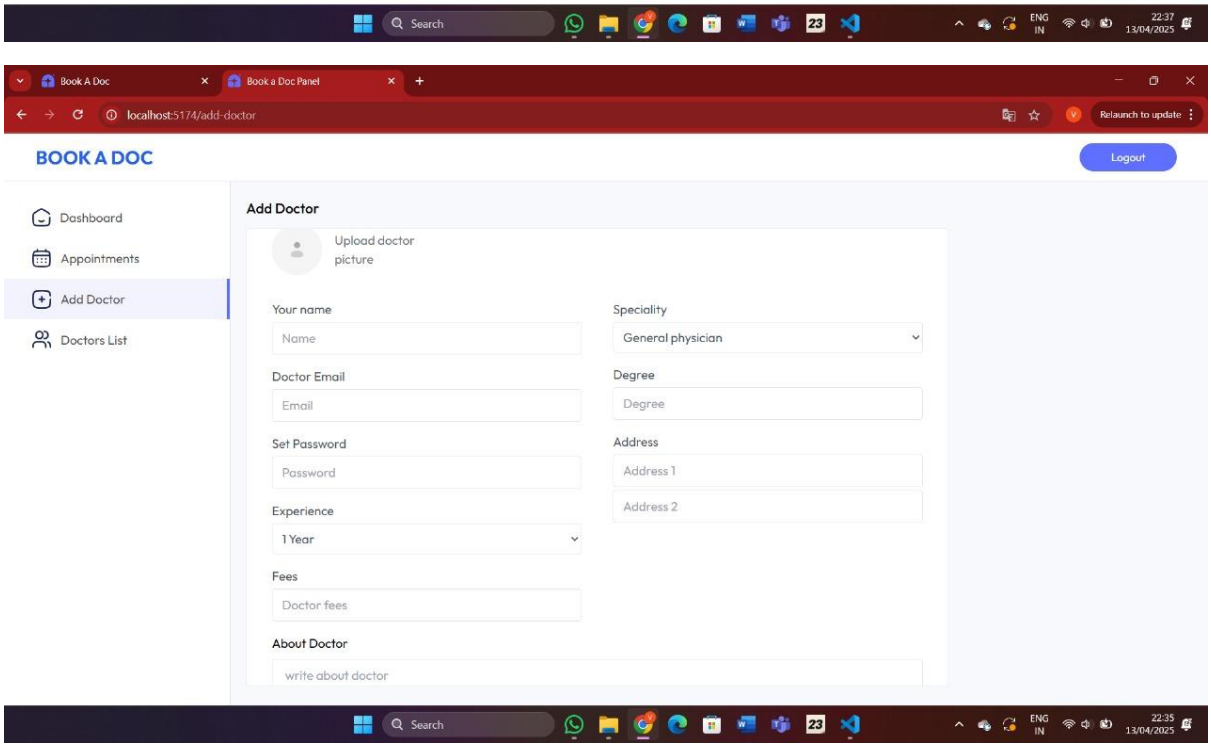
- **Doctor Schedule Dashboard**



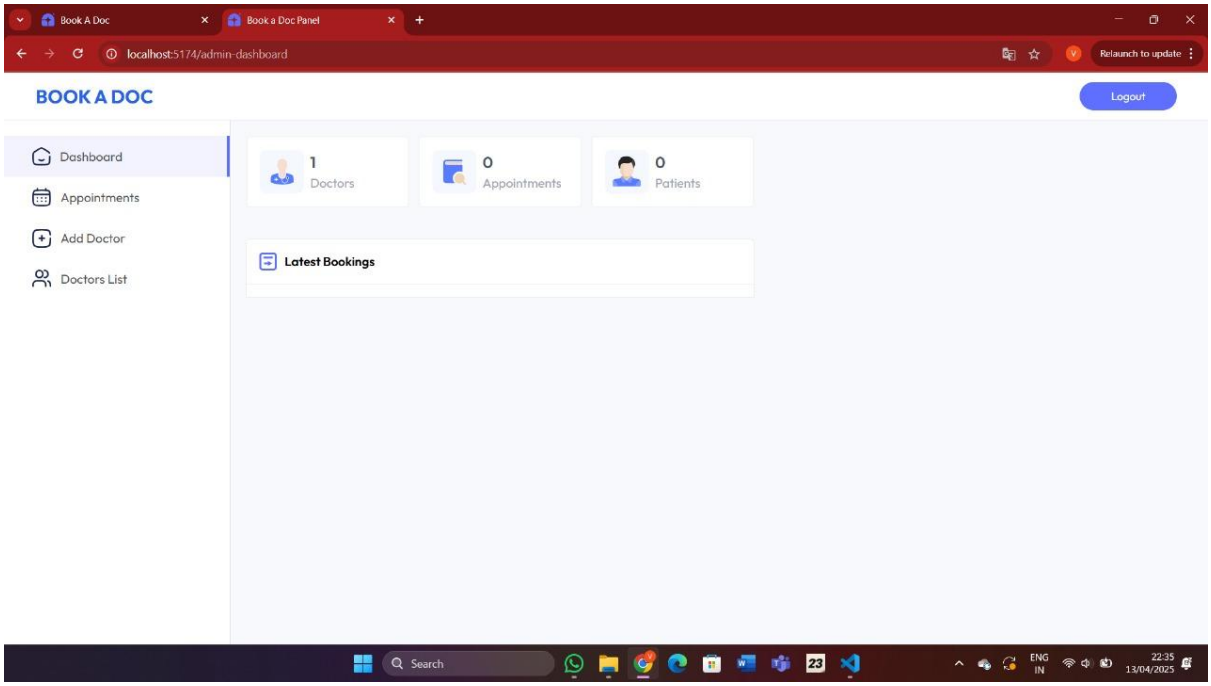
Team ID : SWTID1743701170

- **Admin Approval Panel**





Team ID : SWTID1743701170



---

8. ADVANTAGES & DISADVANTAGES

Advantages

- Reduces manual work and waiting time
- Secure and role-based access
- Responsive and intuitive interface
- Real-time slot availability and booking

Disadvantages

- Internet connection required
- No in-built video consultation (planned)
- Initial manual verification for doctors

Team ID : SWTID1743701170

9. CONCLUSION

"Book A Doc" addresses a real and pressing need in the healthcare industry: **easy and accessible doctor appointments.** By providing a digital solution that balances simplicity with power, we've created a tool that can grow and adapt with time.

This project not only improved our MERN stack development skills but also gave us hands-on experience in building something that could genuinely help people.

---

10. FUTURE SCOPE

- **Telemedicine Integration**: Add video consultation features using WebRTC or third-party APIs
- **Calendar Sync**: Allow doctors to sync schedules with Google Calendar
- **Health Records**: Enable upload and access to medical history
- **AI-powered suggestions**: Smart doctor recommendations based on symptoms
- **Mobile App**: React Native version for Android/iOS

---

11. APPENDIX

- **Source Code**: GitHub Repo – Book A Doc
- **Documentation Link**: Project Doc
- **Demo Link**: Video Link
