Full Stack MERN Project Documentation

# Introduction

* + Project Title: docspot: seamless appointment booking for health
  + Team Members:  
    - Pajjuri Saranya Durga – Team Leader  
    - Ponnuru Giridhara Prasad – Team Member  
    - Rishmitha Bandi – Team Member  
    - Saragada Tirupathi Venkata Mohan Reddy – Team Member

# Project Overview Purpose: This project aims to provide a seamless appointment booking platform for patients to connect with doctors and healthcare providers efficiently. The goal is to simplify healthcare scheduling, reduce waiting times, and improve patient-doctor communication. Features: - User registration and authentication - Search and filter doctors by specialization, location, and availability - Real-time appointment booking and cancellation - Doctor dashboard to manage schedules - Notifications and reminders - Secure patient-doctor communication Architecture Frontend: Built using React.js for a responsive, user-friendly interface. Backend: Node.js with Express.js for RESTful APIs. Database: MongoDB with Mongoose for schema modeling.

* + Purpose: Describe the purpose and goals of the project.
  + Features: - User authentication - CRUD functionality - Admin panel - Responsive UI - API integration

# Architecture

* + Frontend: Built using React with routing and state management.
  + Backend: Node.js with Express.js for RESTful APIs.
  + Database: MongoDB with Mongoose for schema modeling.

# Setup Instructions Prerequisites: - Node.js - MongoDB - Git Installation: - Clone the repository - Install dependencies using npm - Configure environment variables in a .env file - Start frontend and backend servers Running the Application: Frontend: Run `npm start` in the client directory. Backend: Run `npm start` in the server directory.

* + Prerequisites: - Node.js - MongoDB - Git
  + Installation: - Clone the repository - Install dependencies using npm - Set up .env file - Start frontend and backend servers

# Folder Structure

* + Client: - /src/components - /src/pages - App.js, index.js
  + Server: - /controllers - /models - /routes - server.js, .env

# Running the Application

Frontend: Run `npm start` in the client directory. Backend: Run `npm start` in the server directory.

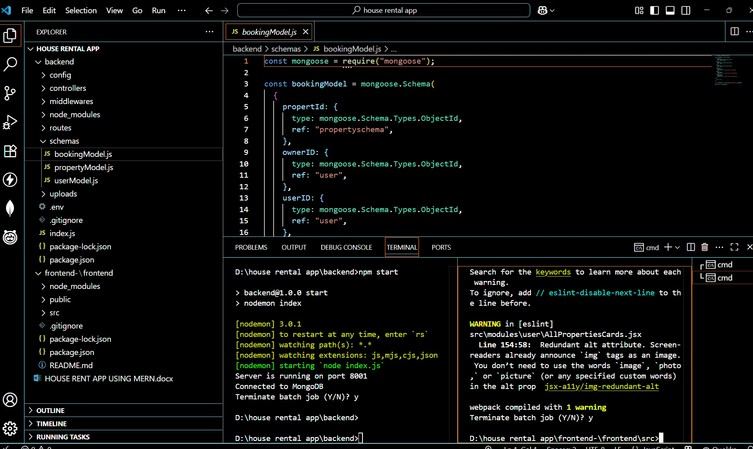
# API Documentation Example: POST /api/users/register Request Body: { "name": "John Doe", "email": "john@example.com", "password": "securepassword" } Authentication: JWT-based authentication with protected routes.

Example:POST /api/users/registerRequest Body:{ "name": "John", "email": "[john@example.com](mailto:john@example.com)", "password": "123456"}

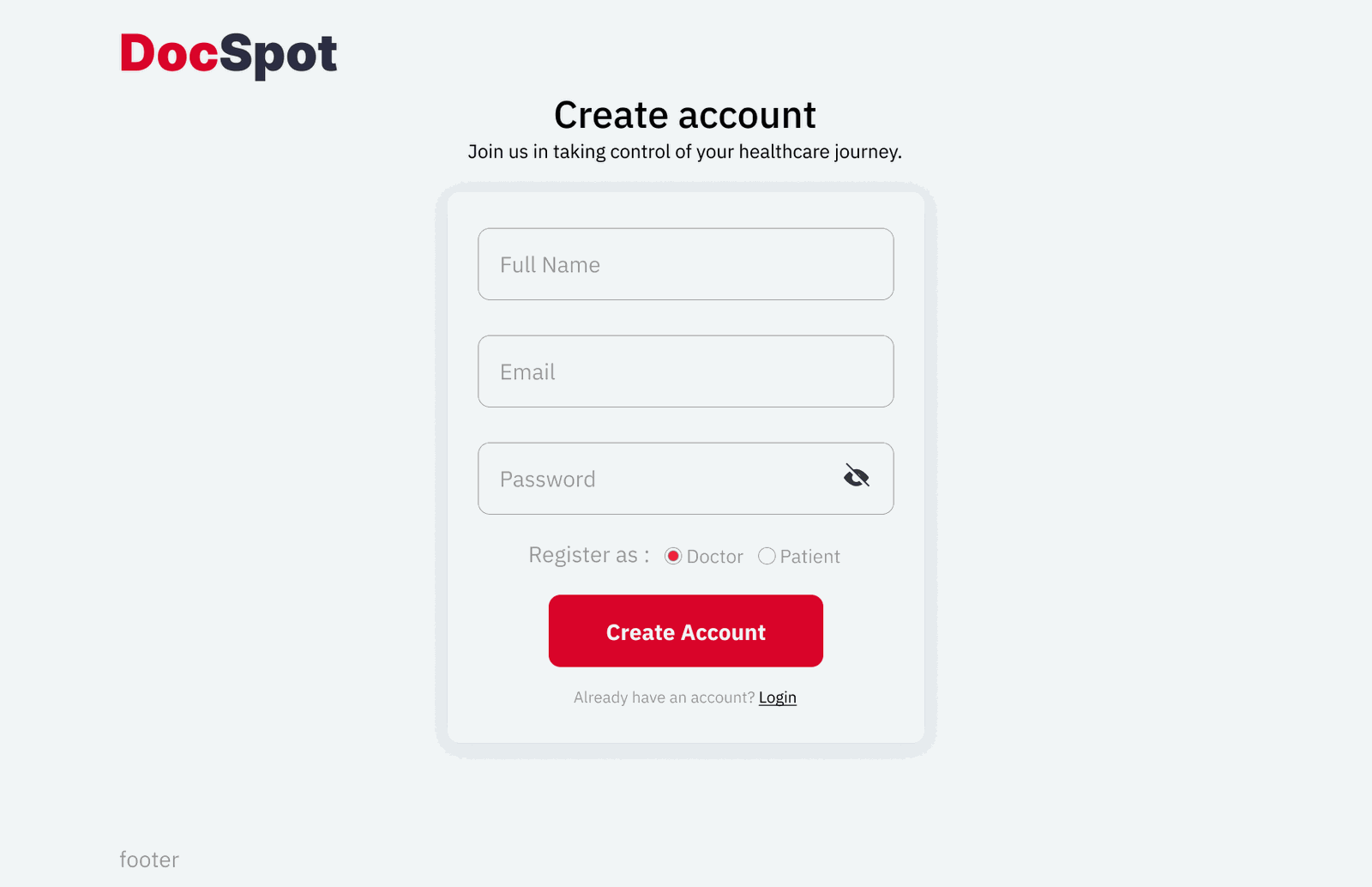
# Authentication

JWT-based authentication with protected routes.

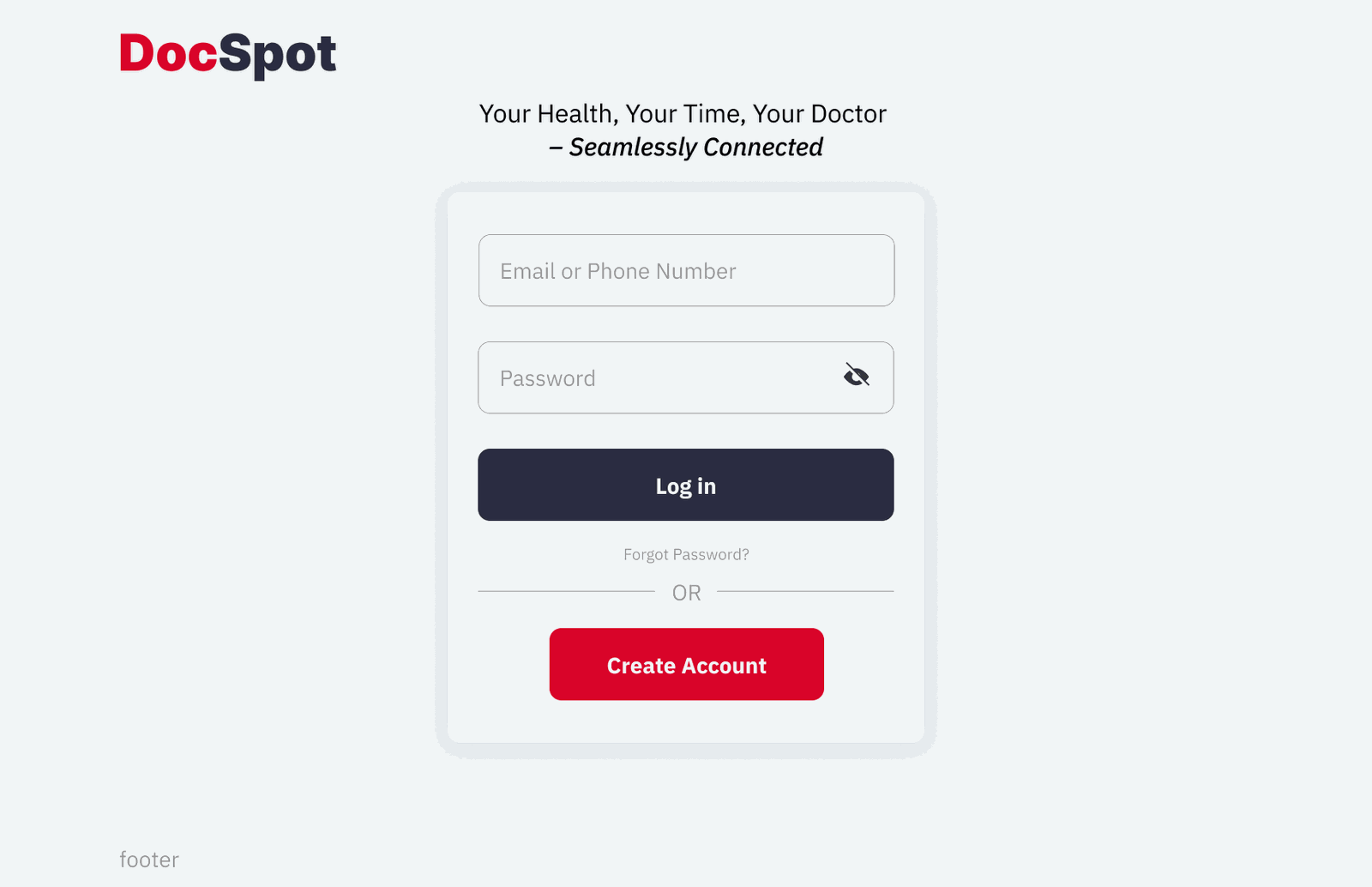
# User Interface



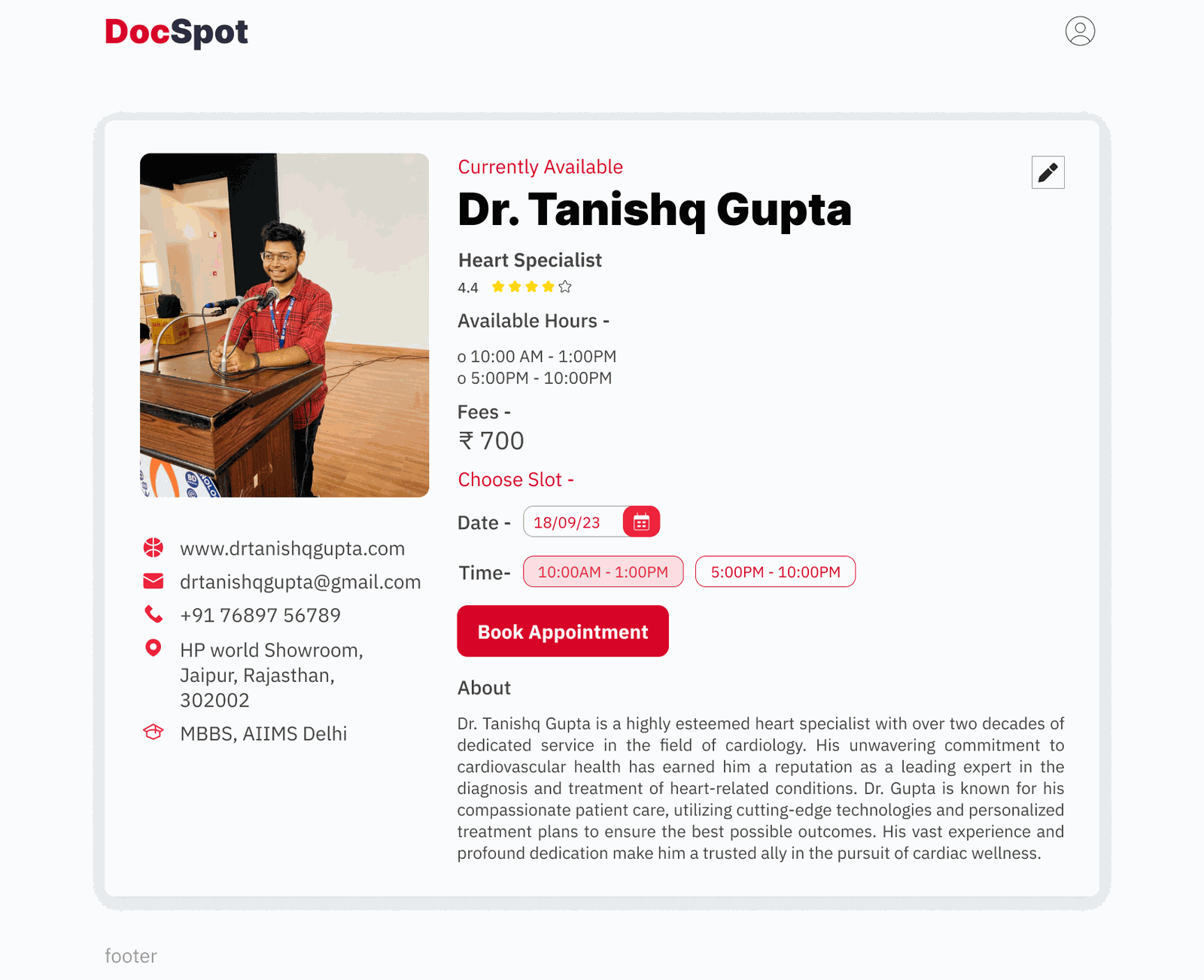
screenshot\_1.png



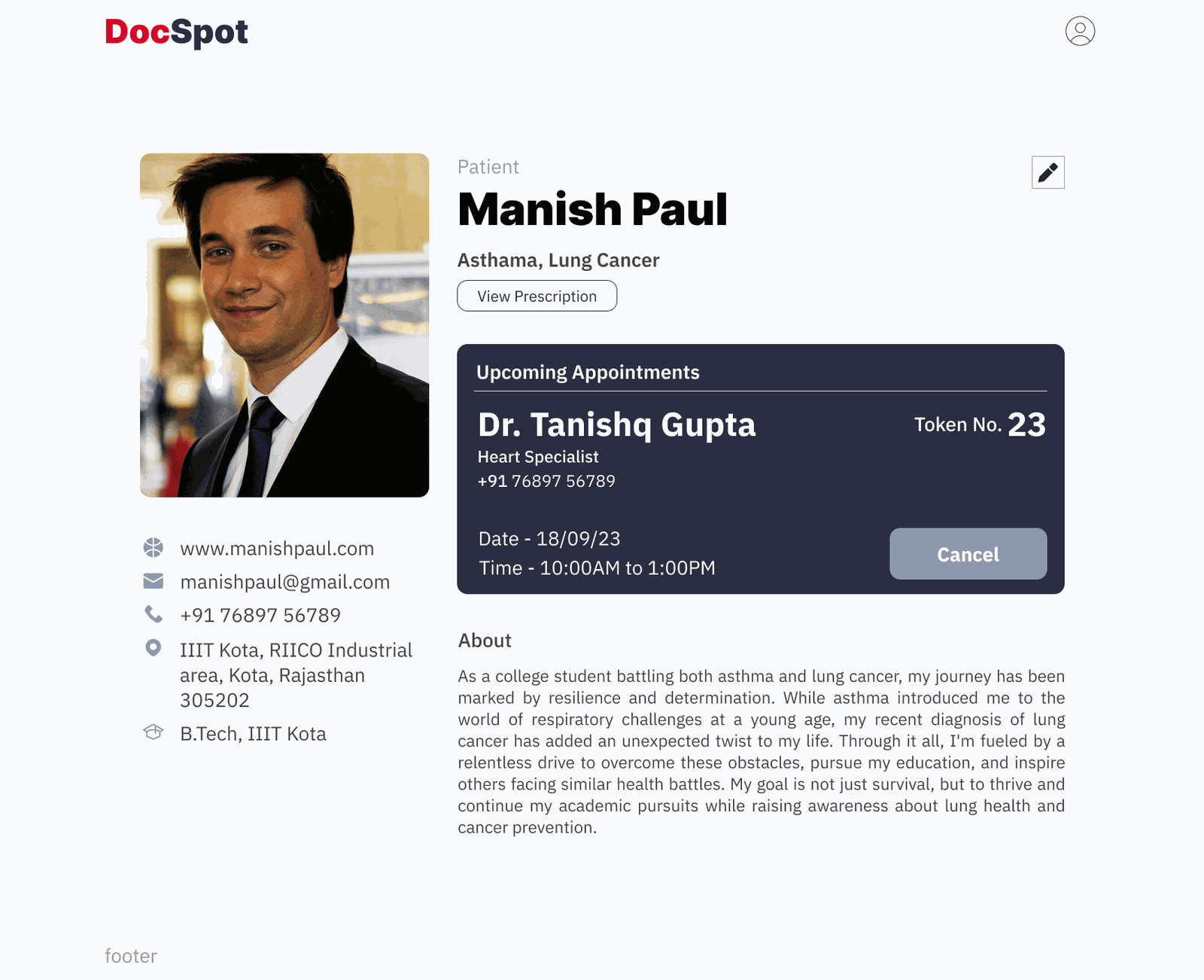
screenshot\_2.png



screenshot\_3.png



screenshot\_4.png



screenshot\_5.png

# Testing

Manual testing using Postman and browser.

# Screenshots or Demo

Refer to the User Interface section above for screenshots.

# Known Issues - Minor responsiveness issues on very small devices - Possible API delays during high traffic

- Responsiveness issues on small devices- API latency on poor networks

# Future Enhancements - Video consultation integration - Online payment gateway for paid consultations - AI-based doctor recommendation system

- Add chat functionality- Implement email verification- Integrate payment gateway