Full Stack Development with MERN

**Project Documentation**

# Introduction

Project Title: DocSpot: Seamless Appointment Booking for Health

Team ID: LTVIP2025TMID53942

Team Members:

* + Pajjuri Saranya Durga – Team Leader / Full Stack Developer
  + Ponnuru Giridhara Prasad – Frontend Developer
  + Rishmitha Bandi – Backend Developer
  + Saragada Tirupathi Venkata Mohan Reddy – UI/UX Designer

# Project Overview

Purpose:

DocSpot is developed to streamline the process of booking health appointments online. It empowers users to search, schedule, and manage appointments with doctors effortlessly, thereby improving access to healthcare services and minimizing wait times.

Core Features:

* User registration and login
* Doctor profile management
* Search/filter doctors by specialization, location, and availability
* Appointment scheduling system
* Admin dashboard for managing users and appointments
* Fully responsive user interface

# Architecture

Frontend:

* + Built using React.js
  + Routing handled by React Router
  + State management via Context API
  + API integration with Axios

Backend:

* + Developed with Node.js and Express.js
  + RESTful APIs for business logic and routing
  + Middleware for authentication and error handling

Database:

* + MongoDB with Mongoose

# Setup Instructions

Prerequisites:

* + Node.js
  + MongoDB (local or cloud)
  + npm or yarn
  + Git
  + .env with MONGO\_URI and JWT\_SECRET

Installation:

1. Clone the repo:

git clone [your-repo-link]

1. Install frontend dependencies:

cd client npm install

1. Install backend dependencies:

cd ../server npm install

1. Configure environment variables
2. Start MongoDB server

# Folder Structure

/client

■■■ public/

■■■ src/

* ■■■ components/
* ■■■ pages/
* ■■■ App.js
* ■■■ index.js

/server

■■■ controllers/

■■■ models/

■■■ routes/

■■■ middleware/

■■■ config/

■■■ index.js

# Running the Application

Frontend:

cd client npm start

Backend:

cd server npm start

# API Documentation

User Registration: POST /api/users/register Login: POST /api/users/login

Appointment Booking: POST /api/properties/add (Auth Required) Get Doctors: GET /api/properties

# Authentication

* + JWT-based authentication
  + Passwords hashed using bcryptjs
  + Token verification with middleware
  + Authenticated users can book and manage appointments

# User Interface

Key Screens:

* + Home Page
  + Login/Register Page
  + Doctor Listings
  + Appointment Booking
  + Admin Dashboard

# Testing

Tools:

* + Postman for API testing
  + Jest for backend tests

Strategy:

* + Functional endpoint testing
  + UI and booking flow validation

# Screenshots or Demo

Deployed Site: [Insert URL]

Include screenshots of:

* + Home Page
  + Property Details
  + Admin Panel

# Known Issues

* + Filter options may not work on older browsers
  + Admin panel lacks sorting
  + No 2FA or OTP

# Future Enhancements

* + Payment gateway integration
  + Doctor-patient chat
  + Integration with electronic health records (EHRs)
  + Notifications
  + React Native mobile app