Project Documentation: DocSpot – Seamless Appointment Booking for Health

Team ID: LTVIP2025TMID53796

1. Category

Full Stack Development

2. Skills Required

HTML, CSS, JavaScript, Bootstrap, React.js, Node.js, MongoDB

3. Team Details

Team ID: LTVIP2025TMID53796

Team Members:

- **Vallabhanaeni Sivatmika** – rs200221@rguktsklm.ac.in
- **Tupakula Hazeera** – rs200747@rguktsklm.ac.in
- **Sahitya Pedapudi** – sahityapedapudi187@gmail.com
- **Shaik Khalida Farheen** – shaikkhalidafarheen@gmail.com
- **Thalapaneni Harshitha** – thalapaneniharshitha@gmail.com

4. Project Description

Booking a doctor's appointment has never been easier. With DocSpot, users can conveniently schedule appointments from the comfort of their homes. No more phone calls or waiting in lines. The platform enables:

- Real-time doctor availability
- Flexible booking times (morning/evening/weekend)
- Document uploads for insurance or medical history
- User-friendly dashboards for Patients, Doctors, and Admins

5. Scenario-based Case Study

Scenario: Booking an Appointment

- **User Registration:** John signs up as a patient, entering his email and password.
- **Browsing Doctors:** John logs in and filters doctors based on specialty and location.
- **Booking:** John selects a doctor and a time slot. He uploads his insurance file and submits the booking request.
- **Confirmation:** The doctor accepts the request. John gets a scheduled notification with date and time.
- **Dashboard Management:** John can view, cancel, or reschedule his appointments.
- **Admin Approval:** New doctor registrations are verified and approved by the admin.
- **Doctor Management:** Dr. Smith manages appointments and updates status post consultation.
- **Follow-up:** John receives visit summary and prescription through the app.

6. Technical Architecture

DocSpot uses a client-server architecture.

Frontend

- React.js for SPA structure
- Tailwind CSS, Material UI, Bootstrap for styling
- Axios for REST API calls

Backend

- Node.js with Express.js
- JSON Web Tokens (JWT) for secure authentication
- bcrypt for password hashing

Database

- MongoDB with Mongoose
- Stores user data, appointments, doctor profiles

7. Activity Distribution

1. ER Diagram

All team members contributed:

- Sahitya Pedapudi, Thalapaneni Harshitha, Vallabhanaeni Sivatmika, Shaik Khalida Farheen, Tupakula Hazeera

2. Pre-requisites Setup

Installed Node.js, MongoDB, React, Tailwind CSS, Bootstrap

3. Project Structure

- Sahitya Pedapudi – Backend structure and Mongoose models
- Sivatmika – Frontend pages and routing
- Harshitha – API routes and server setup

4. Setup & Configuration

- Sahitya Pedapudi – '.env', Express server
- Sivatmika – React environment and page routing
- Khalida Farheen – Testing configs and test data

5. Backend Development

- Harshitha – Server routes
- Sahitya – MongoDB Models
- Hazeera – Admin controls

6. Frontend Development

- Sivatmika – Landing, Dashboard, Appointment UI
- Hazeera – Styling with Tailwind/Bootstrap

7. Implementation Pages

- **Landing Page:** Sivatmika
- **Register Page:** Farheen
- **Login Page:** Harshitha
- **Admin Dashboard:** Hazeera
- **Doctor Dashboard:** Sahitya
- **User Dashboard:** Sivatmika and Hazeera

8. Demo and Code Reference

Demo Video:
Google Drive Demo Link

GitHub Repository:
https://github.com/SivatmikaVallabhaneni/DocSpot.git

9. Conclusion

DocSpot is a complete solution for doctor appointment booking using the MERN Stack. The platform offers real-time booking, robust admin controls, and responsive dashboards for all roles. It supports future integrations such as:

- Chat systems for doctor-patient interaction
- Payment gateway for online booking
- Advanced analytics for admins

This project proves team collaboration in building and deploying a real-world full-stack application.