Internship Project Report

Project Title: DocSpot – Seamless Appointment Booking for Health
Submitted By: Keerthi
Department: Computer Science & Engineering
Internship Domain: Full Stack Web Development (MERN Stack)
Organization: SMARTINTERNZ
Academic Year: 2024–2025

Abstract

DocSpot is a MERN (MongoDB, Express.js, React.js, Node.js) stack-based full-stack web application designed to streamline doctor appointment booking. It provides a seamless interface for three different roles—Users (patients), Doctors, and Admins. The system is built to simplify the healthcare access process by allowing users to book appointments, doctors to manage patient appointments, and admins to oversee platform activities. With its responsive design, role-based login, and secure authentication using JWT, DocSpot addresses real-world pain points in hospital appointment scheduling. The project was developed as part of an internship submission and emphasizes modern web development practices.

Primary Objectives

- To build a role-based appointment booking system for health.
- To allow users to register, log in, and book appointments.
- To enable doctors to register and view patient appointments.
- To provide an admin panel to manage all users and doctors.
- To use JWT for user authentication and secure APIs.
- To separate user, doctor, and admin logins and dashboards.
- To integrate MongoDB to persist data for users, doctors, and appointments.
- To use Ant Design and Bootstrap for a responsive UI.
- To upload doctor verification documents using Multer.
- To validate forms and provide proper success/error notifications.

Introduction

In the current healthcare system, managing doctor appointments is often tedious and unorganized. There is a need for a digitized system that provides better appointment management with transparent communication among users, doctors, and admins. DocSpot is developed to solve this issue using MERN Stack technology.
It incorporates three roles — User, Doctor, and Admin — and provides each with specific features for interaction and management. The system is developed with scalability, security, and usability in mind.

System Requirements

Software Requirements:

- Node.js
- React.js
- MongoDB
- Express.js
- Ant Design / Bootstrap
- VS Code / Any IDE

Hardware Requirements:

- Dual-core Processor
- 4–8 GB RAM
- Internet Connection

Architecture

The system follows a three-tier architecture:

Frontend (React) → Backend API (Express) → MongoDB Database
Role-based login and routing is implemented to direct users to their respective dashboards.

Workflow

1. User, doctor, or admin registers/logins.

2. JWT token is generated upon successful login.

3. Based on the role, user is redirected to their dashboard:

- User: Can book appointments.

- Doctor: Can see booked appointments.

- Admin: Can manage doctors and users.

What We Have Done in Our Project

- Designed UI using React and Ant Design.

- Created backend API with Express and Node.js.

- Developed MongoDB models for users, doctors, and appointments.

- Implemented authentication using JWT.

- Integrated file upload (doctor verification documents).

- Built dashboards for all roles.

- Deployed and tested successfully.

Advantages

- Simple and intuitive user interface
- Role-based secured access
- Real-time appointment booking
- Backend scalability
- Secure password and session handling
- Reusable components and clean code structure

Conclusion

DocSpot is a real-world solution to doctor appointment management. It has been built using modern technologies like MERN stack and successfully provides a scalable, secure, and user-friendly experience. This internship project allowed the developer to gain hands-on experience in full-stack web application development, authentication, routing, and MongoDB-based database operations.