Final Report

Full Stack Development with MERN

Project Documentation Format

1. Introduction
2. Project Title: DocSpot-Seamless-Appointment-Booking-For-Health
3. Team ID: LTVIP2025TMID57021
4. Team Size: 4
5. Team Members:
6. Gumpena Hima Vardhini (Team Leader)
7. Gogireddy Gowthami
8. Chitteti Josh Kumar
9. Peddi Ananth Krishna
10. Project Overview
11. Purpose: DocSpot is a full-stack MERN application designed to streamline doctor appointment booking by providing an efficient, digital platform for patients and healthcare providers.
12. Features:
13. Online doctor appointment booking system
14. Doctor dashboard for managing schedules and appointments
15. User authentication, registration, and secure login
16. Real-time appointment tracking and management
17. Search functionality for doctors by specialization
18. Patient profile management
19. Responsive UI for mobile and desktop users
20. Architecture
21. Frontend: Developed using React.js, providing a responsive, single-page application interface with intuitive user experience.
22. Backend: Built with Node.js and Express.js, responsible for handling API requests, user authentication, and business logic.
23. Database: MongoDB is used as the NoSQL database for storing users, doctors, appointments, and system data.
24. Setup Instructions
25. Prerequisites:
26. Node.js (version 16+ recommended)
27. MongoDB (local or cloud-based instance)
28. Git

29. Installation:
30. Clone the repository: https://github.com/joshkumar50/DocSpot-Seamless-Appointment-Booking-for-Health
31. Navigate to both "client" and "server" directories.
32. Run "npm install" in each directory to install dependencies.

33. Create ".env" file in the server directory with required environment variables (e.g., DB connection string, JWT secret).

34. Folder Structure

35. Client:
36. public/: Static assets
37. src/: Components, pages, routing, styles
38. Server:
39. config/: Database configuration
40. models/: Mongoose schemas for users, doctors, appointments
41. routes/: API endpoints
42. controllers/: Request handling logic

43. middleware/: Authentication and error handling

44. Running the Application

45. Frontend:
46. Navigate to "client" directory and run: npm start
47. Backend:

48. Navigate to "server" directory and run: npm start

49. API Documentation

50. Major Endpoints:
51. POST /api/auth/register: Register a new user
52. POST /api/auth/login: Authenticate user and return JWT token
53. GET /api/doctors: Get list of available doctors
54. POST /api/appointments: Book an appointment
55. GET /api/appointments/user: Get logged-in user's appointments

56. GET /api/appointments/doctor: Get appointments for doctor account

57. Authentication

58. JWT (JSON Web Token) based authentication system.
59. Tokens stored securely and validated on each protected route.

60. Role-based access control to distinguish between patients and doctors.

61. User Interface

62. Responsive design using React and CSS.
63. Separate dashboards for doctors and patients.
64. Appointment booking forms with real-time feedback.

65. User-friendly navigation and mobile compatibility.

66. Testing

67. Manual testing conducted for:
68. User registration and login
69. Booking and managing appointments
70. Dashboard functionality for doctors
71. API error handling and validations

72. Future plan to integrate Jest and Mocha for automated testing.

73. Screenshots or Demo

74. Project Demo Video: https://drive.google.com/file/d/17pfzp7NnvDuYvybaG1nk_o39ft7So1nS/view?usp=drivesdk

75. GitHub Repository: https://github.com/joshkumar50/DocSpot-Seamless-Appointment-Booking-for-Health

76. Known Issues

77. Currently supports only registered doctors on the platform.
78. Limited error handling for certain edge cases.

79. Requires stable internet connection for smooth operation.

80. Future Enhancements

81. Integration of real-time video consultation feature.
82. Mobile application for iOS and Android platforms.
83. AI-driven doctor recommendations based on user preferences.
84. Advanced patient history and prescription management system.
85. Admin dashboard for better system control and analytics.