CHITKARA UNIVERSITY

DocEase

1. Project Statement:

Patients often face difficulties in scheduling medical appointments, resulting in delays and miscommunication, while healthcare providers struggle to manage bookings efficiently. This project aims to create a full-stack doctor appointment booking system using the MERN stack to streamline the scheduling process, allowing patients to book, view, and track appointments easily and enabling doctors to manage their availability and schedules effectively. The system seeks to improve patient satisfaction, reduce wait times, and enhance communication between patients and healthcare providers.

2. Approximate duration (in hours) to complete the project:

Weeks	Time	Work to be done
Week 1	5 hrs	Project Discussion & Setup
Week 2	10 hrs	Profile Management
Week 3	5 hrs	Appointment Booking
Week 4	3 hrs	Search & Filter
Week 5	10 hrs	Database Integration

Table1: Timeline

3. Proposed Project In-charge:

Ms. Preenu Mittan

4. Team Members, along with roll no's:

Bhuvesh Mittal: 2210991450
Daanushi Sharma: 2210991469

5. Check Points:

a. Does the project statement result in a product? If yes, what type of product?

Yes, the project statement results in a product: a **web-based doctor appointment booking system**. This product would be a **full-stack application** where patients can browse doctors, schedule appointments, and receive notifications, while doctors can manage their schedules and view patient histories. The platform could be used by clinics, hospitals, and healthcare centers to manage appointments, track patient visits, and analyze scheduling data.

b. If it is a product, can a prototype be made, if not, what is it, which we can produce that our teachers can evaluate.

Yes, a prototype can definitely be made for this project. You could develop a **functional prototype** with core features like user registration, appointment scheduling, and a simple interface for both patients and doctors. This prototype would allow teachers to evaluate the design, user flow, and basic functionality of the booking system before full development.

c. Does the project statement use multiple concepts to achieve the outcome?

Yes, the project statement uses multiple concepts to achieve the outcome. These include:

a. Web Development: Frontend (React.js) and Backend (Node.js, Express.js)
b. Database Management: PostgreSQL or MongoDB
c. Authentication and Authorization: To securely manage patient and doctor logins, ensuring data privacy.
d. Real-Time Notifications: For sending appointment reminders and updates to users.
e. Scheduling and Time Management: To enable doctors to manage their availability and prevent double-booking.
f. User Experience: Designing user interfaces and user experience flows

d. Does it have enough for our team members to do enough work?

Yes, the project has enough scope for team members to engage in a sufficient amount of work. The tasks can be divided into different modules such as:

a. Frontend Development: Designing and implementing the user interface
b. Backend Development: Setting up the server, database interactions, and API development
c. Authentication To securely manage patient and doctor logins, ensuring data privacy.
d. Real-Time Notifications: For sending appointment reminders and updates to users.
e. Scheduling and Time Management: To enable doctors to manage their availability and prevent double-booking.
f. Database Design: Designing and managing the database schema

6. Technical Nodes

Subject / Area / Topic	Technical Nodes
Frontend Development	React
Backend Development	Node.js, Express.js
Database Management	MongoDB
Authorization and Authentication	JWT

Table 2: Technical Nodes

7. Prerequisites:

a. Knowledge

1. Familiarity with frontend frameworks (React.js)
2. Knowledge of backend development (Node.js, Express.js)
3. Understanding of relational and non-relational databases (PostgreSQL, MongoDB)
4. Basic concepts of authentication and authorization
5. Familiarity with RESTful API design

b. Concepts

1. MVC (Model-View-Controller) architecture
2. RESTful APIs
3. CRUD operations
4. Asynchronous programming and Promises
5. State management in frontend frameworks
6. Responsive design principles

c. Materials

1. Tutorials and documentation for React.js, Node.js, Express.js, PostgreSQL, and MongoDB
2. Example projects and source code for similar applications
3. Books and online courses on full-stack web
4. Design resources for creating user interfaces (wireframes, mockups)

8. Material that may be required to make the project and where it might be available

a. Development Tools

1. IDE/Text Editor: Visual Studio Code
2. Version Control: Git, GitHub (available at Git and GitHub)
3. Package Manager: npm (included with Node.js installation from Node.js)

b. FrontEnd

1. React: Reactjs

c. BackEnd

1. Node.js & Express.js: Node.js, Express.js
2. JWT: jwt

d. DataBase

1. MongoDB: MongoDB

9. Features

a. User (Patient) Interface

1. Doctor Directory and Search
2. Appointment Booking
3. Appointment History

b. Admin Interface

1. Doctor and Appointment Management
2. Payment Management
3. Reports

c. Doctor Profile Interface

1. Profile Management
2. Appointment Management
3. Earnings overview

10. What could the total cost of the project?

N/A

11. Resources available to us:

a. Computer/Laptops
b. Development Tools
c. Internet Access
d. Guidance from faculty
e. Access to online tutorials & documentation