**Fitness Tracking System Project Guidelines (MERN Stack)**

**Introduction**

**Purpose**:  
The objective of this project is to develop a lightweight fitness tracking web application using the MERN stack. This project will demonstrate your ability to work with modern web technologies while solving a real-world problem of tracking and improving physical fitness.

**Background**:  
As people strive to adopt healthier lifestyles, fitness tracking has become a necessity. By leveraging technology, this project aims to help users monitor their daily activities and stay motivated in achieving fitness goals.

**Scope**:  
The application will focus on basic features such as activity tracking, goal-setting, and data visualization using the MERN stack (MongoDB, Express.js, React, and Node.js). Advanced features like integration with wearable devices can be explored as optional enhancements.

**Problem Statement**

**Definition**:  
Many people find it challenging to track their fitness activities and measure progress effectively. Existing solutions often lack simplicity or affordability for regular users.

**Importance**:  
By building a web-based fitness tracking system, users can log their activities, monitor progress, and achieve their fitness goals using a simple, cost-effective platform.

**Objectives**

1. Develop a web application using the MERN stack to allow users to log daily fitness activities (steps, distance, calories, etc.).
2. Implement a dashboard for users to visualize their fitness data (e.g., progress charts).
3. Enable goal-setting functionality for users to define and monitor fitness targets.
4. Create a clean and responsive user interface with React.
5. Build secure API endpoints with Express.js and Node.js to interact with a MongoDB database.

**Methodology**

**Steps to Execute the Project**:

1. **Frontend Development**:
   * Use React for building the user interface.
   * Implement a responsive design using Tailwind CSS or Bootstrap.
2. **Backend Development**:
   * Use Node.js and Express.js for building RESTful API endpoints.
   * Design endpoints for user authentication, activity logging, and data retrieval.
3. **Database Design**:
   * Use MongoDB to store user information, activity logs, and goals.
   * Structure collections for scalability (e.g., users, activities, goals).
4. **Visualization**:
   * Use libraries like Chart.js or Recharts to create visual representations of fitness data.
5. **Implementation Steps**:
   * Set up the MERN stack environment.
   * Create RESTful APIs for user authentication and activity management.
   * Integrate the frontend with the backend APIs.
   * Test the system thoroughly and refine based on feedback.

**Project Plan**

**Timeline**:

* **Week 1**:
  + Finalize project requirements.
  + Set up MERN environment and create the basic project structure.
* **Week 2**:
  + Design and implement the database schema.
  + Develop APIs for user authentication and activity management.
* **Week 3**:
  + Build the frontend UI using React.
  + Integrate the frontend with backend APIs.
* **Week 4**:
  + Add data visualization and goal-setting features.
  + Perform end-to-end testing and prepare final deliverables.

**Milestones**:

1. Backend API completion.
2. Fully functional UI with basic features.
3. Integration of data visualization and goal-setting.

**Expected Deliverables**

1. A web-based fitness tracking application built with the MERN stack.
2. Source code hosted on GitHub with proper documentation.
3. Presentation summarizing the project, its features, and outcomes.

**Evaluation Criteria**

1. **Frontend Design and Functionality**: 30%
   * User-friendly interface and responsive design.
2. **Backend API Implementation**: 30%
   * Secure, scalable, and efficient APIs.
3. **Database Design and Integration**: 20%
   * Proper schema and seamless data flow.
4. **Presentation and Report**: 20%
   * Clear documentation and explanation of the project.

**Submission Guidelines**

* **Format**:
  + Source code: GitHub repository.
  + Presentation: PowerPoint slides.
* **Deadline**: Specify a realistic date.
* **Mode of Submission**: Submit the GitHub link and presentation via email or an online portal. https://github.com/PrinceMertiya/MeanStack-project.git

**References**

1. MongoDB Documentation: <https://www.mongodb.com/docs/>
2. Express.js Guide: <https://expressjs.com/>
3. React Official Docs: <https://reactjs.org/>
4. Node.js Documentation: <https://nodejs.org/>