INTRODUCTION

PROJECT TITLE:

FitTrack: Your Personal Fitness Companion

TEAM

Team Leader : Sandhiya.S (Code execution and GitHub management)

Team Member : Syed Ali Fathima Fairose.F(code Execution)

Team Member : Saahithya.G.V(Frontend development)

Team Member : Nishanthi.R(backend development)

Team Member : Pavithra.P(documentation and testing)

PROJECT OVERVIEW

Purpose:

FitTrack is a web application designed to help users achieve their fitness goals. It provides real-time tracking, personalized workout plans, and nutrition insights.

Features:

- **User Profiles**: Create and manage personal fitness goals.
- **Workout Plans**: Access custom and pre-designed workout routines.
- **Activity Tracking**: Log workouts, steps, and calories burned.
- **Diet & Nutrition**: Get meal recommendations based on fitness goals.
- **Progress Reports**: Visualize fitness progress through graphs and statistics.

- **Community Support**: Engage in forums and challenges with other users.
- **Push Notifications**: Set reminders for workouts and diet tracking.

ARCHITECTURE

Component Structure:

- **Navbar Component**: Handles site-wide navigation.
- **Dashboard**: Displays personalized fitness insights.
- **Workout & Nutrition Components**: Fetch and display workout plans and meal recommendations.
- **Progress Tracker**: Graphical representation of fitness improvements.
- **Notifications Component**: Reminders for scheduled workouts and meals.

State Management:

- Managed using React's useState & useEffect for dynamic data updates.
- Backend data is handled using Node.js & Express.js, with a MongoDB database.

Routing:

Implemented using React Router Dom:

/ → Home Page

/dashboard → User Dashboard

/workouts → Workout Plans

/nutrition → Meal Suggestions

/progress → Fitness Progress Reports

SETUP INSTRUCTIONS

Prerequisites:

- Node.js & npm: Required for package management.
- React.js: For building the UI.
- MongoDB: Database for storing user data.
- Postman (optional): For API testing.

Installation:

1. Clone the repository: git clone https://github.com/yourrepo/fittrack.git

2. Install dependencies: npm install

3. Start the development server: npm start

4. Access the application: Open your browser and navigate to:

http://localhost:3000

FOLDER STRUCTURE

/fittrack

|— /src

| ├— /components

| ├— /pages

| ├— /utils

|— /backend

| ├— /models

| ├— /routes

|— package.json

|— README.md

RUNNING THE APPLICATION

Start the app locally: npm start

It will open automatically in your browser at: http://localhost:3000

COMPONENT DOCUMENTATION

Key Components:

- **Navbar**: Manages site navigation.
- **Dashboard**: Displays user fitness stats.
- **Workout List**: Shows available workout plans.
- **Meal Planner**: Provides nutrition recommendations.
- **Progress Tracker**: Logs user fitness progress.
- **Notifications**: Sends reminders for workouts and meals.

Reusable Components:

- **Button**: Customizable for various actions.
- **Card**: Displays workout and nutrition details.
- **Modal**: For pop-ups like workout instructions.

STATE MANAGEMENT

Global State: Managed via React's Context API for storing user data.

Local State: Managed with useState for component interactions.

API Handling: Axios is used for data fetching from the backend.

USER INTERFACE

UI Design:

- **Homepage**: Intro to app features & quick links.
- **Dashboard**: User profile, fitness summary, & quick actions.
- **Workout Page**: List of workouts & exercise details.
- **Nutrition Page**: Diet plans & meal recommendations.
- **Progress Page**: Charts tracking fitness improvements.

Styling:

- **Tailwind CSS**: Modern, responsive design.
- **React Icons**: Intuitive iconography.
- **Custom CSS**: Unique styling elements.

Theming:

- **Light & Dark Mode**: Users can switch themes for better usability.

TESTING

Testing Strategy:

- **Unit Testing**: Jest for individual component testing.
- **Integration Testing**: React Testing Library for component interactions.
- **End-to-End Testing**: Cypress for simulating user flows.

Code Coverage:

- Ensure 80%+ coverage for critical components.
- Use Istanbul for generating reports.

KNOWN ISSUES

- **Syncing delays**: Data updates may take a few seconds.
- **API rate limits**: Could affect data retrieval speed.
- **UI inconsistencies**: Some devices may have minor layout differences.

FUTURE ENHANCEMENTS

User Authentication: Secure login and personalized dashboards.

AI-Based Workout Recommendations: Adaptive plans based on progress.

Wearable Device Integration: Sync data from smartwatches.

Voice Assistant Support: Hands-free workout guidance.

Social Challenges: Compete with friends in fitness goals.