## Project Documentation: Fit flex - Your Personal Fitness Companion

### ​1. Introduction

​**Project Title:** Fit Flex: Your Personal Fitness Companion

**Team ID:** NM2025TMID47785

**Team Leader:**SABARINATHAN S(sabarish2006pc@gmail.com)

**Team Members:**

* ​PRAVEEN RAJ M (praveenpraveen9919@gmail.com)
* ​SANJAY KUMAR T(shyamsanjay007@gmail.com)
* ​SRIDHAR S(sumathi2142003@gmail.com)

​**Roles and Responsibilities:**

* ​**Team Leader – RESHMA M:** Oversees project progress and task allocation, coordinates team communication and final integration, and ensures timely submission and quality of deliverables.
* ​**Team Member – HARINI S:** Designs and develops UI/UX, implements workout and progress tracking features, and integrates frontend with backend APIs.
* ​**Team Member – NANDHINI N:** Develops server-side logic and database, creates APIs for user, workout, and progress tracking, and manages authentication and data flow.
* ​**Team Member – MADHUMITA R:** Conducts testing and debugging, ensures application quality and performance, and prepares research inputs and documentation.

### ​2. Project Overview

​**Purpose:**

**Fit Flex: Your Personal Fitness Companion** is an interactive fitness application designed to help users track their workouts, monitor progress, and achieve their health goals efficiently. It aims to elevate the fitness experience by offering curated workout routines, personalized progress tracking, and smart goal management features.

### ​3. Features

* ​**Workout and Exercise Guidance:** Guides users through various exercises and workout routines.
* ​**Personalized Workout Recommendations:** Suggests workouts based on user goals and preferences.
* ​**Smart Progress Tracking:** Tracks user progress with metrics like reps, sets, weight, and calories burned.
* ​**Search and filter by muscle group or exercise type:** Helps users find specific exercises.
* ​**User profile and saved workout management:** Allows users to save and manage their favorite workout routines.
* ​**Admin panel for content moderation:** Provides tools for administrators to manage workout content and user submissions.

### ​4. Architecture and Component Structure

​The application follows a modular React component architecture, promoting reusability and separation of concerns.

​**Major components include:**

* ​**Header & Navigation:** Provides site-wide navigation and access to user profile and settings.
* ​**Workout List:** Displays a grid or list of available workouts with filtering options.
* ​**Workout Detail:** Shows detailed information about a selected workout, including exercises and instructions.
* ​**User Profile:** Manages saved workouts, preferences, and fitness goals.
* ​**Admin Panel:** A restricted component for managing workout content and user submissions.

​**State Management:**

The application uses the Context API for global state management. Key state domains include:

* ​User Authentication and Profile Data
* ​Workout Collection and Favorites
* ​Progress Tracking
* ​Goal Setting

​**Routing:**

Routing is the process of defining how your server responds to different HTTP requests (like GET, POST, etc.) at specific URLs. The application uses Express.js, a popular framework built on Node.js that simplifies routing.

### ​5. Setup Instructions

​**Prerequisites:** Node js ,MongoDB, Git, React.js, Express.js, Visual Studio Code.

​**Installation:**

1. ​Clone the repository.
2. ​Install client dependencies: cd client && npm install.
3. ​Install server dependencies: cd ../server && npm install.

​**Folder Structure:**

The React app is organized into clear folders:

* ​components/: Reusable UI parts like buttons, headers, and workout cards.
* ​pages/: Full screens like Home, Workouts, Progress, and Profile.
* ​assets/: Images, icons, and other static files.
* ​styles/: CSS files or styled-components for design.
* ​routes/: Handles navigation between pages.
* ​context/: Manages shared data like user info and progress items.
* ​utils/: Helpful code for filtering workouts, formatting data, etc.
* ​hooks/: Custom React hooks like useAuth.
* ​services/: API calls for fetching workouts or updating user data.

### ​6. Running the Application

* ​**Frontend:** cd client && npm start
* ​**Backend:** cd server && npm start
* ​**Access:** Visit http://localhost:3000

### ​7. Component Documentation

​**Key Components:**

* ​**Workout List:** Displays all available workouts.
* ​**Workout Detail:** Shows full details of a selected workout.
* ​**Progress Tracker:** Allows users to log their progress for various exercises.
* ​**Goal Manager:** Helps users set and track personal fitness goals.
* ​**User Profile:** Manages user settings and saved workouts.

​**Reusable Components:**

* ​**Header:** Top navigation bar with links and user info.
* ​**WorkoutCard:** Compact workout preview used in lists.
* ​**Button:** Custom button with styling options.
* ​**Modal:** Popup for forms or messages.

### ​8. Authentication

​JWT-based authentication for secure login. Middleware is used to protect user and admin routes.

### ​9. User Interface

* ​**Home Page:** Displays featured workouts and navigation options.
* ​**Workout Detail Page:** Shows exercises, instructions, and duration.
* ​**Progress Page:** Allows users to track their progress and visualize data.
* ​**Goal Manager:** Allows users to set fitness goals and generate progress reports.
* ​**Forms:** Includes login, workout submission, and profile update forms.

### ​10. Testing

* ​**Manual Testing:** Performed at key development milestones to ensure core features like workout browsing, progress updates, and goal setting worked as expected.
* ​**Tools Used:**
  + ​**Postman:** Used to test and verify API endpoints for workout data, user authentication, and progress management.
  + ​**Chrome Dev Tools:** Used for debugging UI components, inspecting network requests, and monitoring performance.

### ​13.Screenshot / Demo

### 

### 12. Known Issues

* ​**Progress sync delay:** Updates to progress tracking may take a few seconds to reflect across all components.
* ​**Mobile responsiveness:** Some UI elements may not display correctly on smaller screens and require layout adjustments.

### ​13. Future Enhancements

* ​**New Components:** Add features like a workout rating system, comment section, and guided workout timers.
* ​**Enhanced Styling:** Improve visual design with animations, transitions, and theme customization options.
* ​**Mobile Optimization:** Refine layout and responsiveness for better performance on smartphones and tablets.
* ​**Notifications:** Introduce reminders for workout scheduling, progress tracking, and goal milestones.