Project Documentation

*Fitflex-Personal fitness*

# 1. Introduction

Project Title: Fitflex   
Team ID: NM2025TMID37765  
Team Leader: M.Gobhiga-gobhiga662@gmail.com  
Team Members:  
- J.Jayalakshmi-jailaksh13@gmail.com  
- [M.Saranya-ms4845854@gmail.com](mailto:M.Saranya-ms4845854@gmail.com)

-P.Manasha-manashap2007@gmail.com

# 2. Project Overview

**Purpose:**  
FitFlex is a fitness and wellness web application designed to help users manage workouts, track progress, and explore healthy routines dynamically.

**Features:**

* Homepage with featured workouts and trending fitness tips
* Workout plans with details (exercise name, duration, difficulty level)
* Search facility to quickly find exercises or plans
* Dynamic updates using JSON server for real-time data

# 3. Architecture

* **Frontend:** React.js with Tailwind CSS and Material UI
* **Backend:** Node.js (used for project setup and runtime environment)
* **Database:** JSON Server (db.json file) for managing workouts, exercises, and user progress

# 4. Setup Instructions

Prerequisites:  
- Node.js (LTS version)  
- Visual Studio Code  
- JSON Server (installed globally)  
  
Installation Steps:  
1. Download and extract the project folder from SmartInternz portal.  
2. Open the folder in Visual Studio Code.  
3. Open a terminal and run:  
 npm install  
 npm run dev  
4. Split the terminal, navigate to the db folder and run:  
 npm i -g json-server  
 json-server --watch db.json --port 3000

# 5. Folder Structure

FitFlex/

|-- client/ # React frontend

| |-- components/ # UI components

| |-- pages/ # Application pages

|-- db/ # JSON server database

| |-- db.json

# 6. Running the Application

**Frontend:**

cd client

npm run dev

Access at → <http://localhost:5173>

**Database:**

cd db

json-server --watch db.json --port 3000

Data available at → <http://localhost:3000>

# 7. API Documentation

**Workouts API:**

* GET /workouts → fetch all workouts
* POST /workouts → add a new workout
* PUT /workouts/:id → update a workout
* DELETE /workouts/:id → remove a workout

**Exercises API:**

* GET /exercises → fetch all exercises
* POST /exercises → add a new exercise
* PUT /exercises/:id → update an exercise
* DELETE /exercises/:id → remove an exercise

## 8. Authentication

Not implemented in this version, planned for future scope (user login and personalized fitness tracking).

## 9. User Interface

* **Homepage:** Navigation bar, featured workouts, and trending fitness tips
* **Workout Page:** List of workout plans with details
* **Exercise Page:** Exercise steps, duration, and difficulty levels
* **Progress Tracker:** Planned for future version

## 10. Testing

* Manual testing during each execution step
* Browser-based testing to check UI and responsiveness
* JSON server tested for live data updates

## 11. Screenshots or Demo

* Homepage screenshot
* Workout plans view screenshot
* Exercise detail screenshot

- Fitness view screenshot

