FitFlex: Your Personal Fitness Companion

# Introduction

**Team Members**

KRISHNA KANTH V (TEAM LEADER)

AJAI BALAJI S

AMRISH DEV K

GOWDAM M M

HARSHAN R

# Project Overview

## Purpose :

FitFlex is a fitness web application designed to help users plan, track, and improve their workout routines. The platform allows users to explore different exercises, create personalized workout plans, and monitor their fitness progress. With an intuitive design and responsive interface, FitFlex caters to both beginners and experienced fitness enthusiasts.

FitFlex is a cutting-edge fitness web application designed to empower users to take control of their workout routines. Our platform offers a comprehensive suite of features, including:

* Workout Library: Explore a vast collection of exercises categorized by type (strength, cardio, flexibility, etc.)
* Personalized Workout Plans: Create customized plans tailored to your fitness goals and preferences
* Progress Tracking: Monitor your progress with dynamic dashboards and analytics
* Responsive Design: Seamlessly access FitFlex on both desktop and mobile devices

## Features:

* + Browse and explore categorized workouts (strength, cardio, flexibility, etc.)
  + Search workouts by keyword or muscle group
  + Personalized workout plans and user profiles
  + Save favorite exercises for quick access
  + Track progress with dynamic dashboards
  + Responsive design for mobile and desktop

# Architecture

## Component Structure

* App.js – Main entry point with routes and layout
* Header – Navigation bar with links to Dashboard, Workouts, and Profile
* WorkoutFeed – Displays available workouts with filters
* WorkoutCard – Individual exercise details with name, category, and instructions
* SearchBar – Enables keyword and category search
* UserProfile – Manages user details, preferences, and saved plans
* ProgressTracker – Displays fitness progress and statistics

## State Management

* Uses **Context API** to manage authentication, saved workouts, and user preferences globally.
* useState is used for local form states and search inputs.

## Routing

Implemented with **React Router v6**:

* / – Dashboard with quick stats and recommended workouts
* /workouts – Full workout library with search and filter
* /profile – User profile, preferences, and saved exercises

# Setup Instructions

## Prerequisites

* Node.js version 18+
* npm version 9+
* MongoDB Atlas (cloud database)

## Installation

# Clone the repository

git clone: <https://github.com/krisha-hp/Gym-fitness.git>

# Navigate to client (frontend) cd fitflex/client

npm install

# Navigate to server (backend) cd ../server

npm install

## Environment Variables (.env)

MONGODB\_URI=your\_mongo\_connection JWT\_SECRET=your\_secret\_key PORT=5000

## Start Development Servers

# Start backend cd server

npm start

# Start frontend cd client

npm start

This launches the frontend at: [http://localhost:3000](http://localhost:3000/)

# Folder Structure

fitflex/

├── client/ (React frontend)

│ ├── public/

│ ├── src/

│ │ ├── components/

│ │ │ ├── Header.js

│ │ │ ├── WorkoutFeed.js

│ │ │ ├── WorkoutCard.js

│ │ │ ├── SearchBar.js

│ │ │ └── ProgressTracker.js

│ │ ├── context/

│ │ │ └── AppContext.js

│ │ ├── pages/

│ │ │ ├── Dashboard.js

│ │ │ ├── Profile.js

│ │ │ └── Workouts.js

│ │ ├── App.js

│ │ └── index.js

│ ├── package.json

│

├── server/ (Node.js backend)

│ ├── models/

│ │ └── Workout.js

│ ├── routes/

│ │ └── workouts.js

│ ├── controllers/

│ │ └── workoutController.js

│ ├── server.js

│ ├── package.json

│

├── README.md

# Running the Application

# Run backend cd server

npm start

# Run frontend cd client

npm start

# Component Documentation

## WorkoutFeed

* Displays exercises filtered by category.
* Props: filter, searchQuery.

## WorkoutCard

* Displays a single workout (title, image, instructions).
* Props: workout object.

## UserProfile

* Displays and updates user details, preferences, and saved workouts.

## ProgressTracker

* Shows charts/metrics for workout completion and progress.

## Reusable Components

* SearchBar → handles keyword input.
* Header → navigation across app pages.

# State Management

## Global State (Context API):

* Stores user authentication, saved workouts, and preferences.

## Local State (useState):

* Handles form inputs, search bar, and temporary UI states.

# User Interface

## Screenshots (examples to include):

* Dashboard Page → quick stats and recommended workouts
* Workouts Page → list of exercises with filters
* Profile Page → saved workouts and preferences