## INTRODUCTION

PROJECT TITLE

### FITFLEX

YOUR PERSONAL FITNESS COMPANION

TEAM MEMBERS & THEIR ROLES

#### K.THUSIRA - PROGRAM OUTPUT RUNNER

#### S.VARALAKSHMI- DOCUMENTATION EDITOR

#### S.AMARAVATHI - VOICE OVER

#### C.ROSHINI - CONTENT PROVIDER

## PROJECT OVERVIEW

## PURPOSE :

FitFlex is a responsive fitness web application designed to help users create personalized workout routines, track progress, and stay motivated with virtual fitness coaching. It offers a user-friendly interface, dynamic content, and interactive components to enhance the fitness journey.

# FEATURES:

* User authentication (sign-up/login)
* Personalized workout plans
* Progress tracking dashboard
* Exercise video tutorials
* Nutrition tips and meal plans
* Responsive design for mobile and desktop
* Dark and light mode toggle

## ARCHITECTURE

## COMPONENT STRUCTURE :

* App.js: Root component housing routes and layout
* Navbar.js: Navigation bar with links and user menu
* Home.js: Landing page with feature highlights
* WorkoutPlans.js: Lists and details of workout plans
* ProgressTracker.js: User dashboard displaying

progress

* ExerciseDetail.js: Exercise-specific details with

videos

* Profile.js: User profile settings and preferences
* Footer.js: App footer with additional links

## STATE MANAGEMENT :

* Global state managed using Redux Toolkit
* Authentication state, user progress, and workout plans are stored in Redux slices
* Context API used for theme management (dark/light mode)

## ROUTING:

* React Router DOM is used for navigation
* Routes: /, /login, /signup, /workouts, /progress, /profile, /exercise/:id
* Protected routes are implemented for authenticated user areas

### SETUP INSTRUCTIONS

## PREREQUISITES :

* Node.js (version 16.x or above)
* npm (version 8.x or above)
* Git

### INSTALLATION :

1. Clone the repository:

git clone [https://github.com/username/fitflex -](https://github.com/username/fitflex%20-)

frontend.git

2. Navigate to the project directory:

cd fitflex-frontend

3. Install dependencies:

npm install

4. Create a .env file and add necessary environment

variables (API URLs, keys)

5. Start the development server:

npm start

## FOLDER STRUCTURE

## CLIENT(/src) :

* /components - Reusable components like Navbar, Footer, WorkoutCard
* /pages - Main views (Home, Login, SignUp, Profile, Progress)
* /redux - Redux slices and store configuration
* /utils - Helper functions (API calls, validators)
* /assets - Images, icons, and style files

### UTILITIES :

### api.js: Axios instance with interceptors

* + validators.js: Input validation logic
  + auth.js: Authentication helpers
  + theme.js: Dark/light mode context and provider

## RUNNING THE APPLICATION

**Tools and Technologies Used to Run the Front-end Application in FitFlex:**

* + Node.js: JavaScript runtime environment.
  + npm: Package manager for Node.js.
  + React: JavaScript library for building user interfaces.
  + Webpack: Module bundler and build tool.
  + Babel: JavaScript compiler and transpiler.

## FRONTEND :

1. Navigate to the project directory:

cd fitflex-frontend

2 . Run the frontend server:

npm start

3.Open the app in the browser at http://localhost:3000

## COMPONENT DOCUMENTATION

## KEY COMPONENTS :

* + - Navbar.js: Navigation bar displaying links based on authentication state
    - WorkoutCard.js: Displays individual workout details
    - ProgressTracker.js: Shows charts of user activity and progress
    - ExerciseDetail.js: Video player with exercise instructions

## REUSABLE COMPONENTS :

* + - Button.js: Custom button with variant support
    - InputField.js: Reusable input for forms
    - Modal.js: Generic modal for pop-ups
    - Loader.js: Loading spinner component

## STATE MANAGEMENT

## GLOBAL STATE :

**Redux Toolkit manages:**

* + User authentication state
  + Workout plans and progress tracking
  + Notifications and alerts
  + Theme context manages the global dark/light mode
  + toggle

## LOCAL STATE :

* Component-level form handling (e.g., login/signup)
* Modal open/close states
* Toggle switches in profile settings

## USER INTERFACE

**Provide screenshots or GIFs showcasing different UI features:**

* Landing Page with welcome message and CTA
* Workout Plans page with filter and search functionality
* Progress Dashboard with charts and statistics
* Exercise Detail page with embedded video tutorials
* Profile page with editable user information
* Dark and Light Mode toggle screenshots

## STYLING

## CSS FRAMEWORK/LIBRARIES:

* + - Tailwind CSS is used as the primary utility-first CSS
    - framework for rapid UI development and responsive design.
    - Sass is utilized for managing reusable variables,
    - mixins, and nesting styles for complex components when needed.
    - React Icons library is used for consistent and

scalable icons across the UI.

## THEMING :

* + - FitFlex implements a Dark/Light Mode using React Context API.
    - Tailwind CSS custom configurations (tailwind.config.js) are used to define theme colors, font families, and responsive breakpoints.
    - Theming is applied globally with CSS variables for seamless theme switching. Users can toggle themes in their Profile Settings.

## TESTING

## TESTING STRATEGY :

* Unit Testing: Performed using for core utility functions and Redux reducers.
* Component Testing: Implemented using React Testing Library to ensure UI components render correctly and respond to user interactions.
* Integration Testing: Key user flows, such as logging in, accessing workout plans, and updating profiles, are covered.
* End-to-End Testing (Planned): Cypress will be used for end-to-end testing in future updates.

## CODE COVERAGE :

* Jest is configured with coverage reporting (--coverage flag).
* Thresholds are enforced for critical components (80%+ coverage) using jest.config.js.
* Reports are generated in the /coverage directory after each test run.

## SCREENSHOTS OR DEMO

## DEMO LINK :

[Live Demo of FitFlex](https://fitflex-app-demo.vercel.app) (Sample link—replace with your actual URL)

## SCREENSHOTS :

Landing Page: Featuring app introduction and call-to-action buttons.

Login/Register Page: Simple, responsive forms with validation feedback.

Workout Plans Page: Card-based layout of personalized workouts with filtering options.

Progress Tracker Dashboard : Displays interactive charts showing workout stats and progress trends.

Exercise Details Page : Embedded exercise videos with step-by-step instructions.

Dark Mode Preview : UI preview in dark theme for accessibility and user preference.

## KNOWN ISSUES

**Performance Lag on Low-End Devices**:

Some animations and charts may cause slowdowns on lower-end devices, especially in the Progress Tracker section.

**Limited Offline Support:**

Current version does not support offline capabilities; the app requires a stable internet connection.

**Inconsistent Video Playback on Mobile Browsers :**

Some exercise videos may fail to autoplay on certain mobile browsers due to autoplay restrictions.

**Theme Toggle Delay:**

Slight delay experienced when switching themes in large components; optimization is planned.

## FUTURE ENHANCEMENT

**Add Social Integration:**

Enable users to sign in with Google/Facebook and share progress on social media platforms.

**Workout Reminders & Notifications**:

Implement push notifications for workout reminders, streak achievements, and motivational messages.

**Offline Mode & PWA Support:**

Introduce offline capabilities and convert FitFlex into a Progressive Web App (PWA).

**More Comprehensive Nutrition Tracker :**

Expand the nutrition section with personalized meal plans and calorie counters.

**Gamification Features :**

Add badges, achievements, and leaderboards to boost user engagement.

**Multilingual Support :**

Support for multiple languages to expand the app’s reach to a global audience.

**Video Call Integration for Virtual Coaching :**

Allow users to book and attend live coaching sessions directly within the app.