**Frontend Development with React.js**

**Project Documentation format**

1. **Introduction**
   * **Project Title**: FitFlex – Fitness & Workout Tracking App

**Team Leader**: Pragasheetha S ([prabashanmugasundaram@gmail.com](mailto:prabashanmugasundaram@gmail.com))

**Team member:**

* + - * Preetha R ([preetharavikumar26@gmail.com](mailto:preetharavikumar26@gmail.com))
      * Preethi K ([pk9465789@gmail.com](mailto:pk9465789@gmail.com))
      * Pavithra S([pavi85106@gmail.com](mailto:pavi85106@gmail.com))

1. **Project Overview**
   * **Purpose**FitFlex is a React-based frontend application that provides a clean and engaging interface for fitness tracking. It allows users to monitor workouts, manage goals, and visualize progress.
   * **Features**:

-Dashboard with fitness overview

- Workout planner and activity log

- Responsive UI optimized for desktop and mobile

- Reusable React components (tiles, buttons, cards)

- Animated interactions for better UX

1. **Architecture**
   * **Component Structure**:
     + - App.js – Root component, sets up routing and layout
       - Header – Displays logo, navigation bar, and user info
       - Dashboard – Main landing page showing workout summary
       - WorkoutTile – Reusable tile component for workout entries
       - Footer – Displays app footer and quick links
   * **State Management**: Local state with useState/useEffect, shared state with Context API
   * **Routing**: Implemented with react-router-dom (/, /dashboard, /workouts, /profile)
2. **Setup Instructions**
   * **Prerequisites**: Node.js >= 18.x, npm >= 9.x
   * **Installation**:
     + git clone https://github.com/your-username/FitFlex.git
     + cd FitFlex/code
     + npm install
     + Environment Variables: Create a .env file for API keys if needed
3. **Folder Structure**
   * **Client**:

FitFlex/

├── public/

│ ├── index.html

│ ├── favicon.ico

│ └── manifest.json

├── src/

│ ├── components/ (Header.js, WorkoutTile.js, Footer.js)

│ ├── pages/ (Dashboard.js, Workouts.js, Profile.js)

│ ├── assets/ (images, icons, styles)

│ ├── App.js, index.js, App.css

└── package.json

* + **Utilities**: Custom hooks (useAuth, useWorkoutData), helper functions

1. **Running the Application**
   * Development: npm start
   * Production build: npm run build
2. **Component Documentation**
   * **Key Components**: Document major components, their purpose, and any props they receive.
   * **Reusable Components**: Detail any reusable components and their configurations.
3. **State Management**
   * **Global State**: Managed with Context API (UserContext, WorkoutContext)
   * **Local State**: Managed with useState in components
4. **User Interface**
   * Responsive UI with centered texts, left-aligned tiles, animated buttons
5. **Styling**

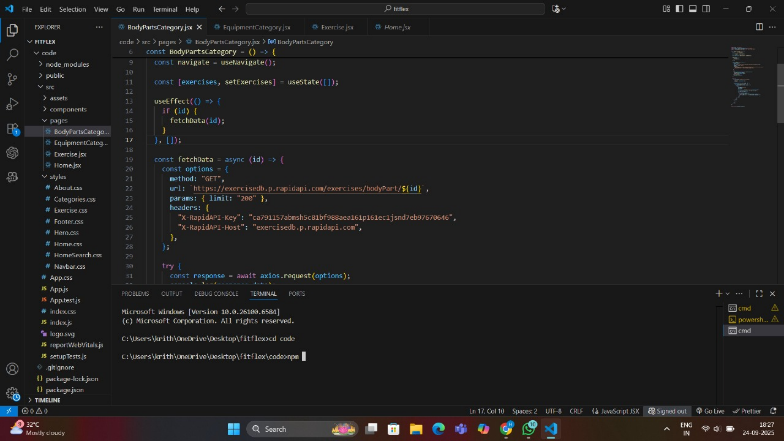
* **CSS Frameworks/Libraries**:
  + - * + CSS Modules + Tailwind CSS
        + Framer Motion animations
        + Light theme with particles
* **Theming**: Light Theme with Particles

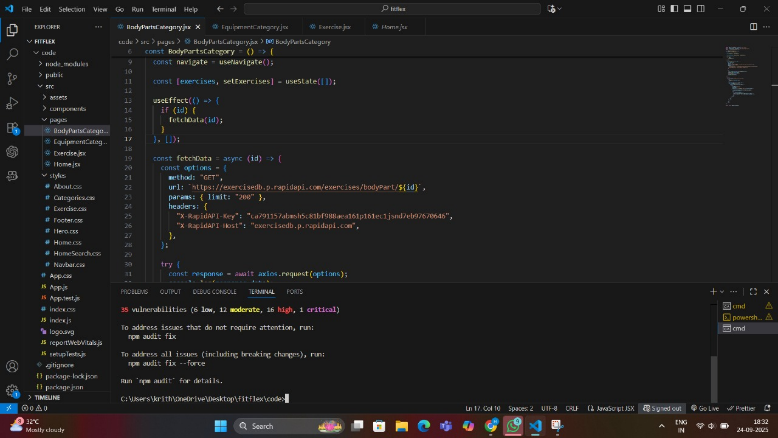
1. **Testing**

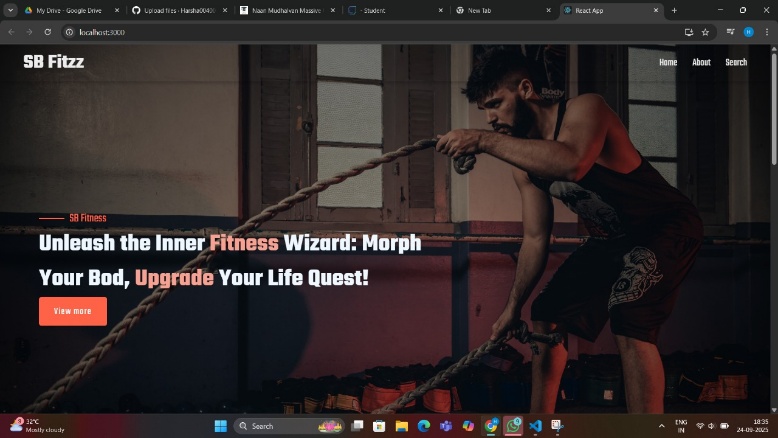
* **Testing Strategy**:
  + Unit tests with Jest + React Testing Library
  + Integration tests for routing and data flow
* **Code Coverage**: Cold Coverage with jest (--Coverage)

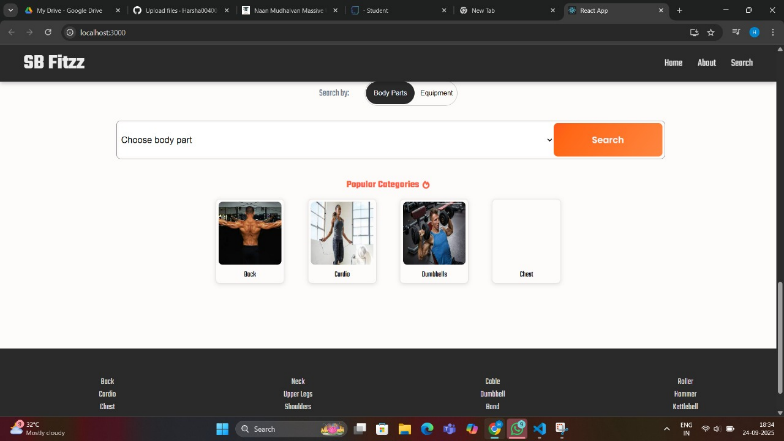
1. **Screenshots or Demo**

* **Provided Link :** <https://drive.google.com/drive/folders/1TSWk_wXESN8Q6R34qTj_m6ec4FpT52Yn?usp=sharing>
* **Screenshots :**









1. **Known Issues**

* Missing react-scripts error (fix with npm install react-scripts)
* Animations may lag on low-end devices

1. **Future Enhancements**

* Add authentication (JWT/Google OAuth)
* Analytics with progress charts
* Dark mode toggle
* PWA support