**FITFLEX**

1. **Introduction**

**Project Title:** FitFlex

**Team Members:**

1. **BHAVYA MA**– Lead Developer (Frontend & State Management)
2. **BHAVANA SS**– UI/UX Designer & Frontend Developer
3. **BHAVATHARANI R**– Testing & Quality Assurance
4. **BRAGGATHISWARI B**– Documentation & Deployment Support
5. **Project Overview**

**Purpose:**  
FitFlex is a fitness-oriented application built with React that aims to help users track workouts, monitor progress, and maintain healthy habits. It provides a responsive interface to view fitness routines, log workouts, and monitor stats.

**Features:**

* Display of fitness routines or workout plans
* Ability to log workouts and input progress data
* Dashboard or statistics section showing progress over time
* Responsive UI for mobile and desktop views
* Ability to navigate between different sections (home, routines, profile, etc.)

1. **Architecture**

**Component Structure:**  
The project is built with modular React components. Likely components include RoutineList, WorkoutLog, Dashboard, NavBar, etc. Each component handles a specific piece of UI or functionality, improving separation of concerns.

**State Management:**

* Local state with React hooks (useState, useEffect) for component-level state
* Global or shared state (if present) via Context API or similar

**Routing:**  
React Router is used (or expected) to manage navigation between pages/views such as routines, logs, profile, etc.

1. **Setup Instructions**

**Prerequisites:**

* Node.js
* npm

**Installation Steps:**

1. Clone the repository:
2. git clone https://github.com/bhavyamadhan2006/FitFlex.git
3. Navigate to project folder:
4. cd FitFlex
5. Install dependencies:
6. npm install
7. Start development server:
8. npm start
9. **Folder Structure**

**Client (React App):**

* public/ – static assets and index.html etc.
* src/ – main source code including components, pages, etc.
* .gitignore, package.json, package-lock.json – project config & dependencies

1. **Running the Application**

To run the frontend locally:

npm start

Then open [**http://localhost:3000**](http://localhost:3000/) in your browser to view the app.

1. **Component Documentation**

**Key Components (presumed based on code structure):**

* Routines / RoutineList – to display available workout routines
* WorkoutLog – component to add or view logged workout data
* Dashboard – to show user progress / statistics over time
* NavBar – for navigation between parts of the app
* Possibly Profile or Settings if implemented

1. **Reusable Components:**

* Cards or tiles for routines/workouts
* Form inputs to log progress or add new workouts

1. **State Management**

* **Local State:** via React hooks for individual component logic
* **Global / Shared State:** (if present) via Context API or similar to share data like user info, routines, etc.
* **Persistence:** either in local storage or via some backend (if implemented)

1. **User Interface**

The UI is built to be responsive, suitable for both mobile and desktop screens. Key views/screens probably include:

* Routine listing / selection
* Workout logging screen
* Dashboard / Progress view
* Navigation menu/header

1. **Styling**

**CSS Framework / Styling Mechanism:**  
Based on project files, styling is done using CSS (possibility of custom CSS or CSS modules).

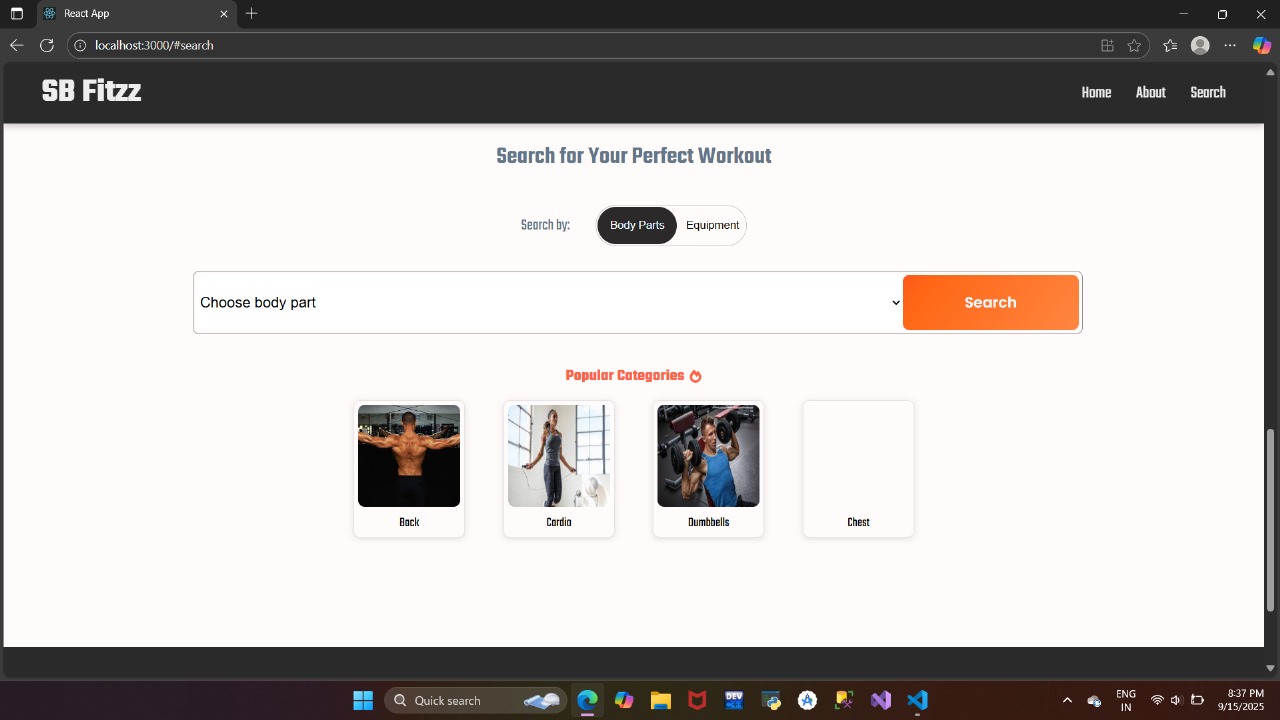
**Theming / UI Consistency:**  
Using style files + CSS to enforce consistent layout, fonts, colors, etc.

1. **Testing**

* **Testing Strategy:** If tests are included, likely using Jest / React Testing Library (otherwise suggested)
* **Code Coverage:** Can focus tests on core functionality: routine display, workout logging, navigation

1. **Screenshots or Demo**

*(screenshots of the major UI screens: home/routines, workout log, dashboard, etc.)*



1. **Known Issues**

* If no backend: data might not persist across sessions
* Possible limited validation on input fields
* UX improvements needed for smaller screens or edge cases

1. **Future Enhancements**

* Integrate backend API for data persistence (user routines, workout logs)
* Add user authentication (login/register) to personalize data
* Enable editing or deleting workout logs or routines
* Add progress charts/graphs for motivation (weight lifted, reps, etc.)
* Dark mode or theming options
* Offline capability or syncing