**FitFlex - Fitness App with React.js**

**Project Documentation**

**Introduction**

* + **Project Title**: FITFLEX
  + **Team Members**:

|  |  |  |
| --- | --- | --- |
| **TEAM MEMBERS LIST** | **NAME** | **EMAIL ID** |
| TEAM LEADER | SURYA R | [suryasurya84172@gmail.com](mailto:suryasurya84172@gmail.com) |
| TEAM MEMBER 1 | VIGNESH B | [vickyb0046@gmail.com](mailto:vickyb0046@gmail.com) |
| TEAM MEMBER 2 | THIRUKKALVARAYAN G | thirukkalvarayan15[@gmail.com](mailto:banupriya07102003@gmail.com) |
| TEAM MEMBER 3 | MADHANRAJ S | madhanraaj2002[@gmail.com](mailto:archu93600@gmail.com) |
| TEAM MEMBER 4 | LITHISH KUMAR S | Lithishkumar68@gmail.com |

**1. Project Overview**

**Purpose**

**FitFlex is a fitness application designed to provide users with exercise routines tailored to specific body parts and available equipment. The goal is to create a personalized fitness experience that helps users achieve their workout objectives efficiently.**

**Features**

* **Exercise recommendations based on body parts and equipment.**
* **User-friendly interface with animations.**
* **Workout tracking and progress monitoring.**
* **Responsive design optimized for mobile devices.**
* **Theming and dark mode support.**

**2. Architecture**

**Component Structure**

* **HomeScreen: Displays an overview of available workouts.**
* **ExerciseList: Lists exercises based on user selection.**
* **ExerciseDetail: Provides detailed instructions and animations.**
* **NavigationBar: Handles navigation between screens.**
* **WorkoutTracker: Logs and tracks user progress.**

**State Management**

**FitFlex uses Context API for global state management, ensuring seamless data flow between components.**

**Routing**

**For navigation, React Navigation is used to handle screen transitions and deep linking.**

**3. Setup Instructions**

**Prerequisites**

* **Node.js (Latest LTS version)**
* **React Native CLI**
* **Android Studio or an emulator**

**Installation**

1. **git link** https://github.com/Surya072003/Fit-flex-React
2. **Navigate to the project directory:**
3. **cd fitflex**
4. **Install dependencies:**
5. **npm install**
6. **Start the development server:**
7. **npm start**

**4. Folder Structure**

**Client**

**fitflex/**

**│-- src/**

**│ │-- assets/ # Images, icons, etc.**

**│ │-- components/ # Reusable UI components**

│ │-- **pages** / # Page-level components

**│ │-- styles/ # Tailwind CSS configurations**

**│ │-- App.css # Global styles**

**│ │-- App.js # Main app component**

**│ │-- App.test.js # Test file for App.js**

**│ │-- index.css # Additional global styles**

**│ │-- index.js # Entry point of the application**

**│ │-- logo.svg # Logo asset**

**│ │-- reportWebVitals.js # Performance measurement**

**│ │-- setupTests.js # Test setup file**

**Utilities**

* **formatDate.js: Utility for formatting dates.**
* **useFetch.js: Custom hook for fetching data.**

**5. Running the Application**

**Start the React Native app:**

**npm run android**

**6. Component Documentation**

**Key Components**

* **ExerciseCard: Displays an exercise preview.** 
  + **Props: title, image, onPress**
* **ProgressTracker: Monitors user activity.**

**Reusable Components**

* **Button: A customizable button component.**
* **Modal: For pop-up dialogues.**

**7. State Management**

**Global State**

* **Managed using Context API for workouts and user data.**

**Local State**

* **Use of useState for temporary UI states.**

**8. User Interface**

**(Screenshots or GIFs showcasing the app will be added here.)**

**9. Styling**

**CSS Frameworks/Libraries**

* **Tailwind CSS for styling.**

**Theming**

* **Dark mode support using React Context.**

**10. Testing**

**Testing Strategy**

* **Jest and React Testing Library for unit tests.**

**11. Screenshots or Demo**

https://drive.google.com/file/d/1EDE3VZkoOqqiIEwLSfrO8eRvqoqryVh8/view?usp=drivesdk

**12. Known Issues**

* **Performance optimization for large exercise lists.**

**13. Future Enhancements**

* **AI-powered workout suggestions.**
* **Social features like challenges and leaderboards.**