**1.INTRODUCTION**

PROJET TITLE

**FITFLEX**

**YOUR PERSONAL FITNESS COMPANION**

TEAM MEMBERS:

**K.KEERTHANA(212205254)**

**A.HARIPRIYA(212205232)**

**N.RUBIKA(212205291)**

**D.YAMINI(212206651)**

**2.PROJECT OVERVIEW**

* **PROJECT NAME:** Fitflex
* **PROJECT TYPE:** Fitness & Wellness

Platform

**PURPOSE:**

Fitflex aims to provide comprehensive

fitness solution that caters to individual needs and preferences, helping users eve their fitness achieves goals in a flexible and convenient manner.

**FEATURES:**

* Customised Workout plans
* Personalized Training
* Extensive Video Library
* Progress Tracking

**3.ARCHITECTURE**

**STRUCTURE COMPONENT:**

* Responsible for for providing an intuitive and user-friendly interface users to interact with the platform.
* Includes features such as navigation, search, and dashboard.

**STATE MANAGEMENT**

* The process of managing the state of the application, including user data, workout plans, and other relevant information. Mean state management is a specific approach to state management that uses a centralized store to manage the application's state.

**ROUTING**

* Routing in FitFlex refers to the process of navigating between different screens or pages within the application.

**4.SETUP INSTRUCTIONS**

**PREREQUISITES:**

**HARDWARE PREREQUISITES:**

* **Device:**Smartphone or tablet with

Android or iOS operating system

* **Processor:** Quad-core processor or higher
* **RAM:**2GB or higher
* **Storage:**10GB or higher

**SOFTWARE PREREQUISITES:**

* **Operating System**: Android 5.0 or higher, or iOS 11.0 or higher
* **Browser:** Google Chrome, Mozilla Firefox, or Safari
* **Internet Connection**: Stable internet connection with minimum speed of 1 Mbps

**OTHER PREREQUISITES**:

* **Fitness Goals**: Clear fitness goals and motivation to use the app.
* **Basic Computer Skills**: Basic computer skills and familiarity with mobile apps

**5.FOLDER STRUCTURE**

**CLIENT:**

**Client Folder**:

* **public**: Public assets, such as index.html, images, and other static files.
* **src**: Source code for the front-end application, including:
* **components**: Reusable UI components.
* **App js**: The main application component.
* **index.js**: The entry point for the front-end application.

**UTILITIES:**

* In FitFlex, utilities refer to a set of tools, functions, or classes that provide additional functionality and support to the application. These utilities can be used to perform various tasks, such as:Types of Utilities in FitFlex:
* Helper Functions, Data Processing, Error Handling,security.

6.**RUNNING THE APPLICATION**

**FRONTEND:**

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.

7**.COMPONENT DOCUMENTATION**

**KEY COMPONENTS:**

**Front-end Components**:

* User Interface (UI): The visual elements and interactions that users interact with.
* Client-side Logic: JavaScript code that runs on the client-side, handling user input, and updating the UI.
* Responsive Design: Ensures the application adapts to different screen sizes and devices.

**REUSABLE COMPONENTS**:

* Buttons: Reusable button components with consistent styling and behavior.
* Forms: Reusable form components for user input, including text fields, checkboxes, and dropdowns.
* Cards: Reusable card components for displaying workout information, user profiles, or other data.
* Navigation: Reusable navigation components, such as menus, tabs, or breadcrumbs.

8. **STATE MANAGEMENT**

**GLOBAL STATE**:

* In FitFlex, global state refers to a centralized repository that stores data that is accessible and shared across the entire application. This data is not specific to a particular component or feature, but rather is used throughout the app to manage its behavior, appearance, and functionality**.**

**LOCAL STATE:**

* Local state refers to the data that is specific to a particular component or feature and is not shared with other parts of the application. Local state is used to manage the behavior, appearance, and functionality of a specific component or feature**.**

**9.USER INTERFACE**

* The User Interface (UI) refers to the visual elements and interactions that users interact with to use the application. The UI is responsible for presenting information to the user, receiving user input, and providing feedback to the user.

Components of User Interface in FitFlex:

* Layout: The overall organization and structure of the UI elements.
* Navigation: The menus, buttons, and other controls that allow users to move through the application.
* Input Controls: The text fields, checkboxes, dropdowns, and other elements that allow users to enter data.
* Display Elements: The labels, images, and other elements that display information to the user.
* Feedback Mechanisms: The loading animations, error messages, and other elements that provide feedback to the user.

**STYLING**

**CSS FRAMEWORK/LIBRARIES:**

* Bootstrap: A widely-used framework for building responsive, mobile-first designs.
* Tailwind CSS: A utility-first framework for building custom designs without writing custom CSS.
* Materialize: A framework based on Google's Material Design principles, providing a responsive and consistent design.
* Bulma: A modern framework for building responsive, mobile-first designs with a focus on simplicity and ease of use.

**THEMING:**

* Theming refers to the process of customizing the visual appearance and layout of the application to create a unique and consistent brand identity. Theming involves modifying the colors, typography, imagery, and other visual elements to create a cohesive and engaging user experience**.**

**11. TESTING**

**TESTING STRATEGY:**

In FitFlex, a testing strategy refers to a comprehensive plan that outlines the approach, methods, and techniques used to test the application. The goal of a testing strategy is to ensure that the application meets its requirements, works as expected, and is free from defects or bugs.

**CODE COVERAGE:**

In FitFlex, code coverage refers to the percentage of code that is executed during automated testing. It measures how much of the application's code is covered by tests, including unit tests, integration tests, and other types of automated tests.

**12.SCREENSHOT OR DEMO**

In FitFlex, a screenshot or demo refers to a visual representation of the application's user interface (UI) or a demonstration of its functionality.

**Types of Screenshots or Demos in FitFlex**:

1. UI Screenshot: A static image of the application's UI, showcasing its layout, design, and features.

2. Interactive Demo: A live, interactive demonstration of the application's functionality, allowing users to explore and test its features.

3. Video Demo: A recorded video showcasing the application's features and functionality.

4. Prototype Demo: A demo of a prototype or a minimum viable product (MVP) to test and validate assumptions.

**13.KNOWN ISSUES**

In FitFlex, known issues refer to problems, bugs, or limitations that have been identified and documented during the development, testing, or production phases of the application. These issues are typically tracked and monitored to ensure they are addressed and resolved.

Types of Known Issues in FitFlex:

1. Bugs: Errors or defects in the code that cause the application to malfunction or produce unexpected results.

2. Limitations: Restrictions or constraints in the application's functionality, performance, or compatibility.

3. Compatibility Issues: Problems that arise when the application is used with different browsers, devices, or operating systems.

4. Performance Issues: Problems that affect the application's speed, responsiveness, or overall performance.

**14.FUTURE ENHANCEMENTS**

FitFlex, future enhancements refer to planned or proposed updates, upgrades, or additions to the application's features, functionality, or performance. These enhancements aim to improve the user experience, increase efficiency, or address emerging needs and trends.

Types of Future Enhancements in FitFlex:

1. New Features: Additional functionality or capabilities that expand the application's scope or usefulness.

2. Performance Optimizations: Improvements to the application's speed, responsiveness, or scalability.

3. User Interface (UI) Updates: Enhancements to the application's layout, design, or usability.

4. Integration with Other Systems: Connections to external systems, services, or platforms to expand the application's capabilities.