**Frontend Development with React.js**

**Project Documentation**

* + **PROJECT TITLE**: [ ***FITFLEX*** ]
  + **TEAM MEMBERS**: 1. SWETHAPRIYA. G. (Team Leader)
  + [C24UG206CCS043]
  + 2. DHIVYADHARSHINI. J. (Team Member1)
  + [C24UG206CCS028]
  + 3. MITHRA. M. S (Team Member2)
  + [C24UG206CCS033]
  + 4. KAVIYADHARSHNI. S. (Team Member3)
  + [C24UG206CCS031]

1.**INTRODUCTION:**

* + FitFlex is a single-page fitness web application (SPA) developed using HTML, CSS, and JavaScript.
  + It allows users to search and filter workout programs, view detailed exercise guides, save favorites, and access premium workouts through a simulated unlock system.
  + The app is designed as a front-end only application (no backend), making it easy to run directly in a browser.

1. **PROJECT OVERVIEW:**
   * **Purpose**: To provide users with an all-in-one fitness companion app that blends exercise guidance, lifestyle tips, and personalization in a simple, accessible web interface.
   * **Features**: Workout Library – Beginner, HIIT, Cardio, Strength, Flexibility routines.
   * Search & Filters – Find workouts by category, difficulty level, or keywords.
   * Favorites – Save preferred workouts (stored in localStorage).
   * Premium Content – Some workouts marked premium; users can unlock access.
   * Nutrition & Wellness Tips – Sidebar with healthy lifestyle suggestions.
   * Workout Details & Guided Mode – Modal view with exercises and a step-through timer.
   * Responsive UI – Works smoothly on desktop and mobile devices.
2. ARCHITECTURE:
   * **STATE COMPONENT**: 1. Workout List: Displays a list of workouts
   * 2. Filters: Allows filtering by category and level
   * 3. Search: Searches workouts based on user input
   * 4. Favorites: Manages and displays favorite workouts
   * 5. Detail Modal: Displays workout details and allows starting a guided session
   * 6. Guided Session: Provides step-through workout guidance with a timer
   * **STATE MANAGEMENT**:1. Global Variables: state object that stores the current search query (q), category, and level.
   * 2. Local Storage: Used to store favorite workouts (FAV\_KEY) and premium workout access (PREMIUM\_KEY).
   * The state object is updated when the user interacts with the application, such as searching, filtering, or saving favorites. The application then uses this state to render the workout list and other components accordingly.
   * Here's a breakdown of the state management:
   * -state object:
   * - q: search query
   * - category: selected category
   * - level: selected level
   * - Local Storage:
   * - FAV\_KEY: stores favorite workouts
   * - PREMIUM\_KEY: stores premium workout access status
3. **SETUP INSTRUCTIONS:**
   * **PREREQUITES**: 1. Basic HTML, CSS, and JavaScript knowledge: Understanding of HTML structure, CSS styling, and JavaScript programming is necessary to comprehend and modify the code.
   * 2. Familiarity with front-end development: Knowledge of front-end development concepts, such as DOM manipulation, event handling, and local storage, is required.
   * 3. Understanding of single-page application (SPA) architecture: Familiarity with SPA architecture and how to manage state and render components dynamically is beneficial.
   * 4. Basic understanding of fitness and workout concepts: Knowledge of fitness and workout concepts, such as exercises, categories, and levels, is necessary to understand the application's purpose and functionality.
   * **INSTALLATION**:
   * 1. Create a new folder: Create a new folder for your project, e.g., fitness-app.
   * 2. Create files: Inside the folder, create the following files:
   * - index.html
   * - styles/style.css
   * - scripts/app.js
   * 3. Copy code: Copy the provided HTML, CSS, and JavaScript code into their respective files.
   * 4. Open in browser: Open index.html in a web browser to view and interact with the application.
4. INSTALLATION:

This structure includes:

- index.html: The main HTML file that contains the application's structure and content.

- styles/: A folder containing CSS files, with style.css being the primary stylesheet.

- scripts/: A folder containing JavaScript files, with app.js being the primary script file.

- README.md: A file containing information about the application, its purpose, and usage.

This is a simple structure, and depending on the complexity and requirements of the application, additional folders and files might be necessary. For example

- assets/: For images, icons, or other media files.

- components/: For reusable UI components.

- utils/: For utility functions.

1. **RUNNING APPLICATION:**

1. Open index.html in a web browser: Double-click on the index.html file or right-click and select "Open with" to choose your preferred web browser.

2. View the application: The application will load, and you'll see the fitness app interface.

3. Interact with the application: You can search for workouts, filter by categoryandlevel, save favorites, and start guided sessions.

1. **COMPONENT DOCUMENTATION:**
   * **KEY COMPONENTES**:
   * 1. Workout List: Displays a list of workouts that can be searched, filtered, and interacted with.
   * 2. Search Functionality: Allows users to search for workouts based on keywords.
   * 3. Filtering: Enables users to filter workouts by category and level.
   * 4. Favorites: Allows users to save and manage their favorite workouts.
   * 5. Guided Sessions: Provides step-through workout guidance with a timer.
   * 6. Modal Window: Displays detailed information about each workout and allows users to start guided sessions.
   * 7. Local Storage: Used to store favorite workouts and premium workout access status.
   * **REUSABLE COMPONENETS**: These components can be reused throughout the application to:
   * - Reduce code duplication
   * - Improve maintainability
   * - Enhance consistency in design and functionality
2. **STATE MANAGEMENTS:**

1. Search Query: The current search query. This state is updated when the user types in the search bar.

2. Filter Criteria: The current filter criteria (category and level). This state is updated when the user selects a filter option.

3. Favorite Workouts: The list of workouts saved as favorites by the user. This state is stored in local storage.

4. Premium Workout Access: A boolean indicating whether the user has access to premium workouts. This state is also stored in local storage.

5. Current Workout: The workout being displayed in the modal window or guided session.

The state is managed through a combination of:

1. Global Variables: The state object stores the current search query, filter criteria, and other application state.

2. Local Storage: Used to store favorite workouts and premium workout access status.

The UI is a card-based fitness dashboard with search, filters, a workout list, a wellness sidebar, favorites, and interactive modals — designed for a clean, responsive, and user-friendly experience.

1. **STYLING:**

**1. Color Scheme: A calming color scheme with blues and whites, which is fitting for a fitness application.**

**2. Typography: The Inter font is used throughout the application, providing a clean and modern look.**

**3. Layout: A responsive layout that adapts to different screen sizes, ensuring a good user experience on various devices.**

**4. Card-based Design: Workouts are displayed in cards, making it easy to scan and compare different options.**

**5. Button Styles: Buttons are styled with a consistent design language, using colors and padding to make them visually appealing.**

**6. Modal Window: The modal window is styled to provide a clear and concise view of workout details.**

**The styling is achieved through the use of:**

**1. CSS: The application's styles are defined in the style.css file.**

**2. CSS Variables: Used to define and reuse colors, fonts, and other styles throughout the application.**

1. **THEMING**: The theme is “Classic Fitness & Wellness” — a single-page fitness companion app that blends exercise routines, lifestyle tips, and personalization into an accessible digital trainer.**Testing**

* **TESTING STATERGY**: Unit Testing – test individual functions (search, filter, favorites, unlock).
* Integration Testing – check component interactions (search + filter, favorites + modal).
* Functional Testing – validate end-to-end flows (search → open → guided mode).
* UI/UX Testing – ensure responsive design, readability, and smooth navigation.
* Performance Testing – check fast load, smooth filtering, modal responsiveness.
* Security/Data Testing – validate localStorage, prevent XSS, handle input safely.
* User Acceptance Testing (UAT) – real user feedback on usability and flow.

1. **SCREENSHORT AND DEMO:**

**https://drive.google.com/file/d/1RRcd0iA6uJotmobjdz8b0-00AKdHuqSi/view?usp=drivesdk**

**13.FUTURE ENHANCEMENT:**

1. User Authentication: Implement user authentication to allow users to save their progress and access their data across devices.

2. Workout Customization: Allow users to create and save custom workouts based on their specific needs and goals.

3. Progress Tracking: Add features to track user progress, such as completed workouts, weight lifted, or distance run.

4. Integration with Wearables: Integrate with popular wearable devices or fitness trackers to track user activity and provide more accurate data.

5. Social Sharing: Allow users to share their progress and workouts on social media platforms.

6. More Workout Content: Add more workout content, including videos, images, and tutorials, to provide users with a wider range of options.

7. Personalized Recommendations: Use machine learning algorithms to provide personalized workout recommendations based on user preferences and goals.

8. Notifications: Send notifications to remind users to work out, track their progress, or provide motivational messages.

9. Multi-Language Support: Add support for multiple languages to make the application more accessible to a wider audience.

10. Analytics and Insights: Provide users with detailed analytics and insights into their workout habits and progress.

These enhancements could improve the user experience, increase engagement, and provide more value to users.