

Fronted Development with **React.js**

FITNESS-HUB-APP

Team Members

MOHAMED HAARISH

VIGNESH S

GOWTHAM KUMAR S

JAYAPRAVEEN P

Project Overview

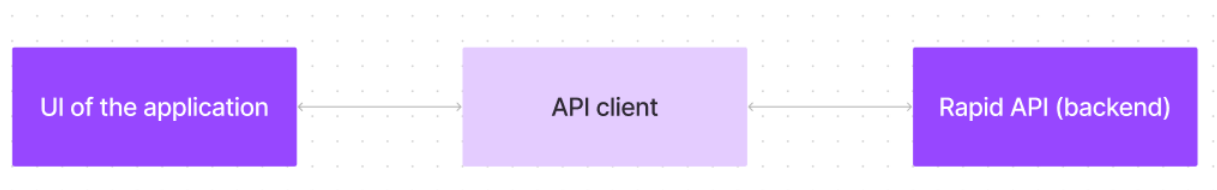
Purpose:

FITNESS-HUB-APP is a fitness tracking application designed to help users monitor their workouts, set fitness goals, and track progress over time. The app provides an intuitive interface for logging exercises, viewing analytics, and receiving personalized recommendations.

Features:

- User authentication and profile management
- Workout logging and tracking
- Exercise library with descriptions and images
- Progress analytics and reports
- Goal setting and achievement tracking
- Integration with fitness wearables (future enhancement)
- Social sharing of workouts
- Personalized workout recommendations
- Custom workout plans

Architecture



Component Structure:

- **Auth Components:** Handles user login, registration, and authentication.
- **Dashboard:** Displays user statistics and progress.

- **Workout Tracker:** Logs workouts and displays history.
- **Exercise Library:** Contains detailed exercise descriptions.
- **Profile Settings:** Allows users to manage personal information.
- **Social Feed:** Displays shared workouts from other users.
- **Goal Tracker:** Helps users set and achieve fitness goals.

State Management:

- **Redux Toolkit** is used for global state management.
- Local component states for UI interactions.

Routing:

- **React Router** is used for navigation across different pages:
 - / - Home
 - /login - User authentication
 - /dashboard - User statistics and progress tracking
 - /workouts - Workout logging
 - /profile - User profile settings
 - /social - Shared workouts from users
 - /goals - Fitness goal tracking

Setup Instructions

Setup Instructions:

Prerequisites:

Node.js:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the

local environment. It provides a scalable and efficient platform for building network applications.

Node.js & npm:

Install Node.js and npm on your development machine, as they are required to run JavaScript on the server-side.

- Download: <https://nodejs.org/en/download/>
- Installation instructions: <https://nodejs.org/en/download/package-manager/>

React.js:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create

interactive and reusable UI components, making it easier to build dynamic and responsive web

applications.

Install React.js, a JavaScript library for building user interfaces.

- Install npm in terminal:

```
npm install
```

```
npm will be installed.
```

- **Navigate to the project directory:**

```
cd code
```

This will navigate your project directory.

- **Running the React App:**

With the React app created, you can now start the development server and see your

React application in action.

- Start the development server:

```
npm start
```

This command launches the development server, and you can access your React

app at <http://localhost:5173> in your web browser.

HTML, CSS, and JavaScript: Basic knowledge of HTML for creating the structure of your app,

CSS for styling, and JavaScript for client-side interactivity is essential.

Version Control: Use Git for version control, enabling collaboration and tracking changes

throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

Git: Download and installation instructions can be found at:

<https://git-scm.com/downloads>

Development Environment: Choose a code editor or Integrated Development Environment (IDE)

that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

- **Visual Studio Code:** Download from <https://code.visualstudio.com/download>
- **Sublime Text:** Download from <https://www.sublimetext.com/download>
- **WebStorm:** Download from <https://www.jetbrains.com/webstorm/download>

Prerequisites:

- Node.js (latest LTS version)
- npm or yarn
- Git

Installation:

1. Clone the repository:

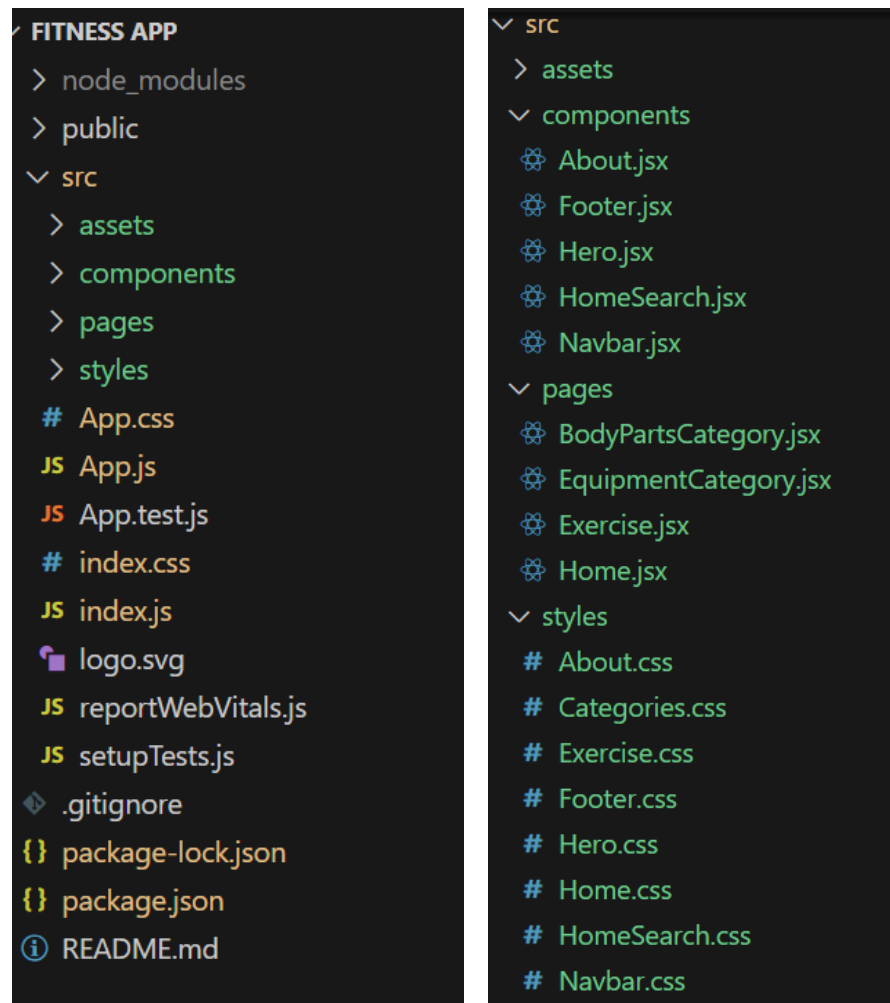
2. `git clone [repository URL]`
3. Navigate to the project directory:
4. `cd FITNESS-HUB-APP`
5. Install dependencies:
6. `npm install`
7. Start the development server:
8. `npm start`

Folder Structure

FITNESS-HUB-APP/

```
| -- src/
|   |-- components/ # Reusable React components
|   |-- pages/      # Main application pages
|   |-- assets/     # Images and static assets
|   |-- store/      # Redux store and slices
|   |-- utils/      # Helper functions and custom hooks
|   |-- App.js      # Main application component
|   |-- index.js    # Entry point of the React app
```

Project structure



Running the Application

Run the code locally:

```
npm start
```

Component Documentation

Key Components:

- **WorkoutLogger:** Handles workout tracking and entry.
- **Dashboard:** Displays user progress analytics.
- **ExerciseCard:** Shows details about each exercise.
- **SocialFeed:** Displays user-shared workouts.
- **GoalTracker:** Helps users set and monitor goals.

Reusable Components:

- **Button:** Custom button component with variant support.
- **Modal:** Reusable modal dialog for popups.
- **Card:** Generic component for displaying information.

State Management

Global State:

- Redux Toolkit is used for managing user authentication, workout data, and statistics.
- Social feed and goal tracking are managed globally.

Local State:

- Component-level states are used for UI interactions like modal toggles and input handling.

User Interface

Include screenshots or GIFs showcasing:

- Home Page
- Dashboard View
- Workout Tracking Page
- Social Feed
- Goal Tracker

Styling

CSS Frameworks/Libraries:

- **Tailwind CSS** is used for styling.
- Custom styles for theme consistency.

Testing

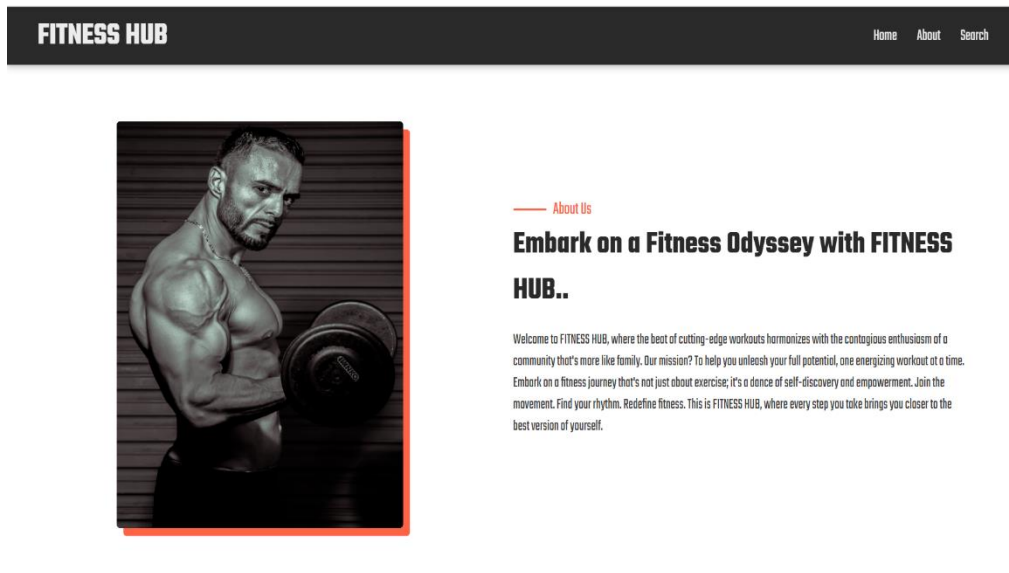
Testing Strategy:

- **Jest** and **React Testing Library** for unit testing.
- **Cypress** for end-to-end testing.
- **Manual UI Testing** for UX and interaction testing.

Code Coverage:

- Ensuring key functionalities have test coverage above 80%.

Screenshots or Demo



Search for Your Perfect Workout

Search by:

Body Parts

Equipment

Choose body part

Search

Popular Categories



Back



Cardio

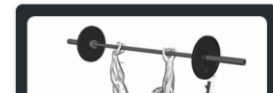
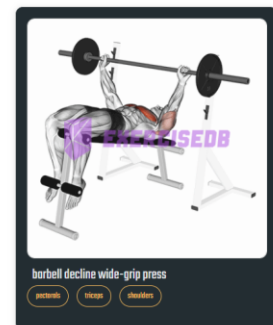
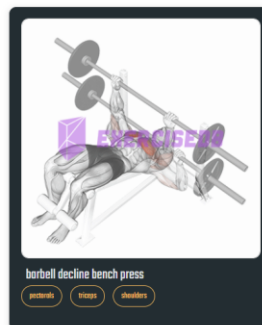
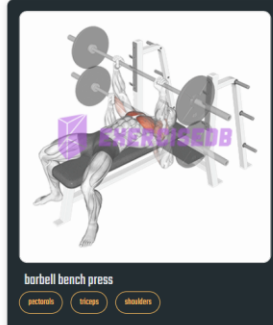
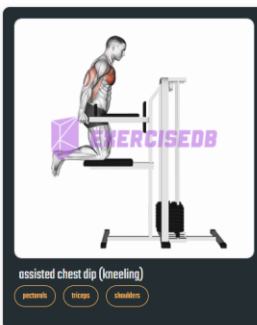


Dumbbells



Chest

category: chest



assisted chest dip (kneeling)

Target:

pectorals

Equipment:

leverage machine

Secondary Muscles:

triceps

shoulders

Instructions

- Adjust the machine to your desired height and secure your knees on the pad.
- Grasp the handles with your palms facing down and your arms fully extended.
- Lower your body by bending your elbows until your upper arms are parallel to the floor.
- Pause for a moment, then push yourself back up to the starting position.
- Repeat for the desired number of repetitions.

Known Issues

- **React-Scripts Deprecation:** The project uses react-scripts@5.0.1, which is deprecated. Consider migrating to Vite or Webpack.
- **Unused Files:** Files like reportWebVitals.js and setupTests.js are present but may not be used.
- **No .env Handling:** If external APIs are used, sensitive data should be stored in a .env file.
- **Missing Type Checking:** No TypeScript or PropTypes are implemented, which may lead to runtime errors.
- **Unstructured Folder Organization:** src/ folder lacks proper structure; files should be organized into separate folders.

Future Enhancements

- Integration with fitness wearables.
- AI-based workout recommendations.
- Social features like workout sharing.
- Community challenges and leaderboards.
- Voice command support for workout logging.

Demo link:

<https://drive.google.com/file/d/1jyFH7vF1geZqLnGmhsWpKwD3jfLmQ8MO/view?usp=sharing>

GitHub Code link:

<https://github.com/VIGNESH787-S/FITNESS-APP.git>
