Frontend Development with React.js

Introduction

Project Title: FitFlex_Personal_Fitness

Team Members: Sunilraj.k, Sanjay, Vaishak, Madesh.

Project Overview

Purpose: The purpose of this project is to create a fitness-focused web application where users can search and explore different exercises by body parts, equipment, and categories. The application provides an interactive and user-friendly interface built with React.js.

Features:
- Exercise search functionality with filtering options.
- Responsive navigation and footer.
- Dynamic UI components like Hero, HomeSearch, and Exercise details.
- Organized category-wise exercise exploration.
- Clean and modular styling with CSS.

Architecture

Component Structure: The application is structured into reusable components such as Navbar, Footer, Hero, HomeSearch, Exercise, About, and categories like BodyPartsCategory and EquipmentCategory. Pages include Home and About. Styles are separated in the styles folder.

State Management: The project primarily uses React's local state and props to manage data flow between components. Future enhancements can integrate Context API or Redux for global state management.

Routing: React Router is used for navigation between pages such as Home and About.

Setup Instructions

Prerequisites:
- Node.js (>=14.x)
- npm or yarn package manager

Installation:
1. Clone the repository.
2. Navigate to the project folder.

3. Run `npm install` to install dependencies.
4. Run `npm start` to start the development server.

Folder Structure

Client Structure:
- src/
  - assets/: Contains static images, icons, and resources.
  - components/: Reusable UI components like Navbar, Hero, Footer, etc.
  - pages/: Page-level components like Home, About.
  - styles/: CSS files for styling different components.

Utilities: Custom hooks or helper functions may be added in future for API handling or reusability.

Running the Application

To run the frontend locally:
- Run `npm start` inside the root folder to start the React development server.

Component Documentation

Key Components:
- Navbar.jsx: Navigation bar for routing across pages.
- Hero.jsx: Landing section with introductory visuals.
- HomeSearch.jsx: Search and filter component for exercises.
- Exercise.jsx: Displays exercise details.
- Footer.jsx: Footer with relevant links and credits.
- About.jsx: About page describing the application.

Reusable Components: Navbar, Footer, and categories (BodyPartsCategory, EquipmentCategory) are designed for reuse across different pages.

State Management

Global State: Not implemented yet. The project currently uses local states and props.
Local State: Components like HomeSearch handle their own state for search inputs and results.

User Interface

The UI includes a responsive navbar, hero section, category filters, and exercise display sections. CSS ensures mobile responsiveness and consistent design across pages.

Styling

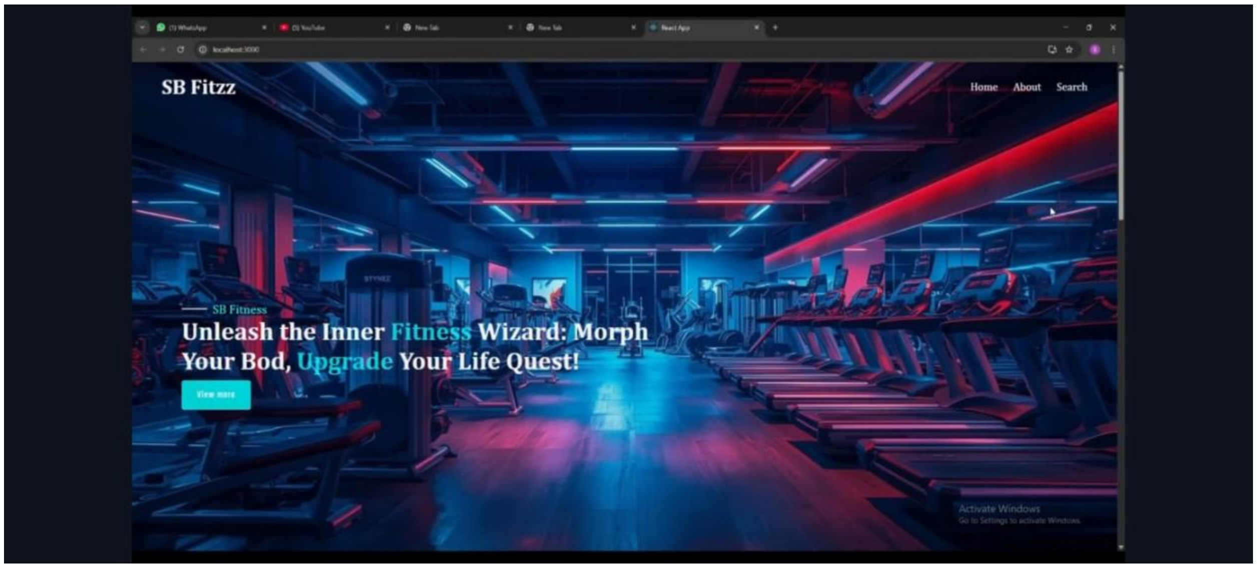
The project uses plain CSS stored in the styles folder. Each component has a dedicated CSS file for modularity. No external CSS frameworks are currently used. Future updates may include Tailwind or Material UI.

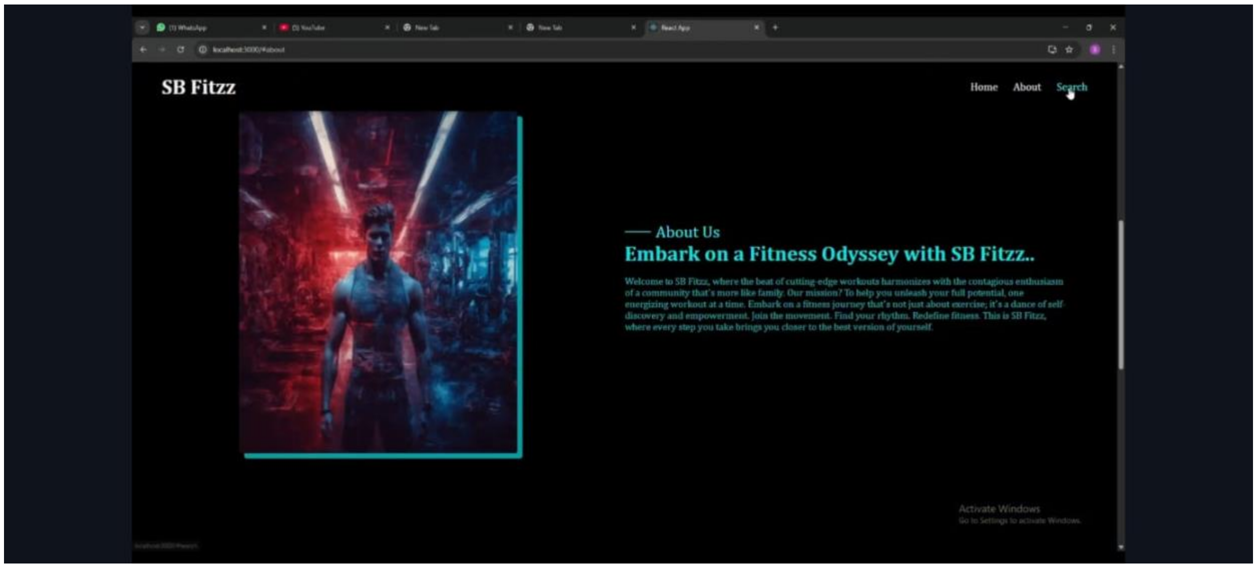
Testing

Testing Strategy: Basic unit tests are set up using Jest and React Testing Library. App.test.js contains sample test cases for ensuring components render correctly.
Code Coverage: Minimal, to be expanded in future development.

Screenshots or Demo

Screenshots of the application UI can be added to this section.

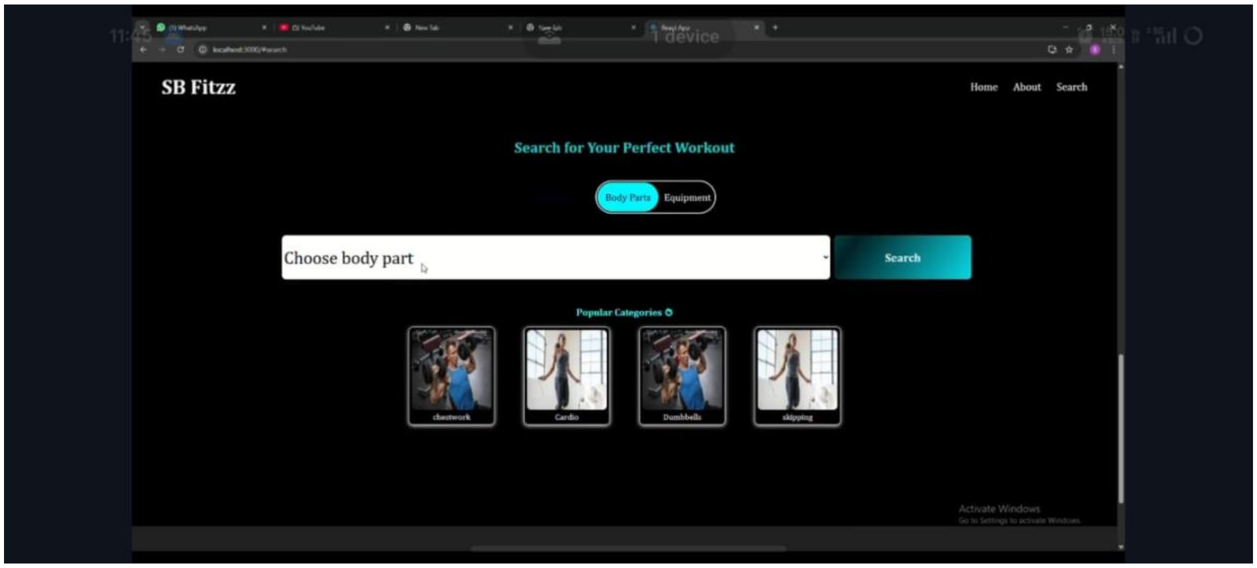

SB Fitzz
Home
About
Search
SB Fitness
Unleash the Inner Fitness Wizard: Morph Your Bod, Upgrade Your Life Quest!
View more



— About Us

Embark on a Fitness Odyssey with SB Fitzz..

Welcome to SB Fitzz, where the beat of cutting-edge workouts harmonizes with the contagious enthusiasm of a community that's more like family. Our mission? To help you unleash your full potential, one energizing workout at a time. Embark on a fitness journey that's not just about exercise; it's a dance of self-discovery and empowerment. Join the movement. Find your rhythm. Redefine fitness. This is SB Fitzz, where every step you take brings you closer to the best version of yourself.



Known Issues
- No API integration implemented yet.
- Limited state management (scalability improvements needed).
- Minimal test coverage.

Future Enhancements
- Integration with an external exercise database API.
- Improved search experience with autocomplete.
- Add animations and transitions.
- Migrate to Context API or Redux for state management.
- Improve testing coverage and CI/CD integration.