**Frontend** **Development** **with** **React**.**js**

# Introduction:

## Project Title:

FitFlex: Your Personal Fitness Companion

## Team Members:

Team ID: NM2025TMID29239

Team Size: 04

Team Leader: Bowjiya Begam S

Team Member: Yogeswari P

Team Member: Abinaya M

Team Member: Bhavani S

# Project Overview:

## Purpose:

FitFlex is a React.js frontend project built to help users track their fitness journey. The app provides a dashboard to monitor workouts, manage diet plans, and visualize progress through an easy-to-use interface.

## Features:

* Dashboard displaying overall fitness progress.
* Add and update workout details.
* View daily/weekly diet plan.
* Responsive design for mobile and desktop.

# Architecture:

## Component Structure:

* App.js – Root component, manages routes.
* Navbar.js – Provides navigation across pages.
* Dashboard.js – Display fitness summary and statistics.
* WorkoutForm.js – Form to add/edit workout sessions.
* DietPlan.js – Shows user diet recommendations.

## State Management:

* Local state handled using React Hooks (useState, useEffect)
* Shared/global data manages using **Context** **API.**

## Routing:

* Implemented using react-router-dom.
* “/” – Home page.
* “/dashboard” – User dashboard.
* “/workouts” – Workout management.
* “/diet” – Diet plan overview.

# Setup Instructions:

## Prerequisites:

* Node.js ( >=16.x)
* Npm ( Node Package Manager)

## Installation:

git clone <https://github.com/apsaraapsara914-ship-it/FitFlex-->

cd FitFlex--

npm install

npm start

# Folder Structure:

## Client (React Application):

/src

  /components

      Navbar.js

      WorkoutForm.js

  /pages

      Dashboard.js

      DietPlan.js

  App.js

  index.js

## Utilities:

* Custom hooks for state handling (if added).
* Helper functions for form validation (to be extended in future).

# Running the Application:

cd FitFlex--

npm start

* “This runs the app in development mode at

<http://localhost:3000> “

# Component Documentation:

## Key Components:

* **Navbar.js** -Navigation links to all pages.
* **Dashboard.js**  - Displays fitness metrics.
* **WorkoutForm.js** – Allows adding/updating workout details.
* **DietPlan.js** – Shows diet chart/plan.

## Reusable Components:

* Form elements and components can be reused across pages.

# State Management:

## Global State:

* Managed using **Context** **API**, ensures consistent fitness data across components.

## Local State:

* Controlled components (input fields) use useState.
* useEffect used for fetching/updating data on render.

# User Interface:

* **Home Page:** Motivational welcome screen.
* **Dashboard:** Stats, progress tracking, charts (future).
* **Workout Page:** Form to manage workout sessions.
* **Diet Page:** Tabular plan for daily nutrition.

# Styling:

## CSS Frameworks/Libraries:

* **TailwindCSS** used for utility-first styling.

## Theming:

* Light theme implemented.
* Dark mode planned as a future enhancement.

# Testing:

## Testing Strategy:

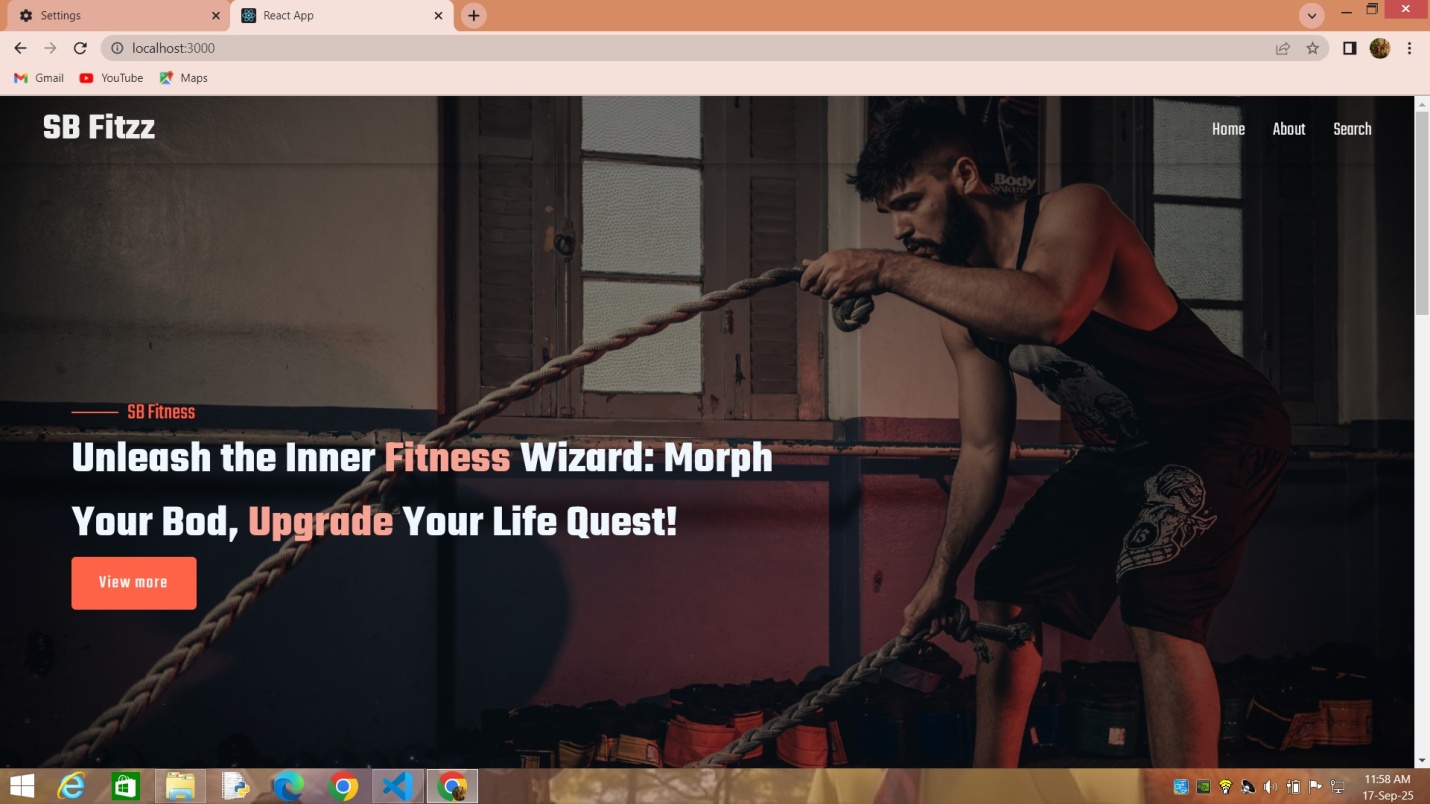
* Unit testing with **Jest**.
* Component testing with **React Testing Library**.

## Code Coverage:

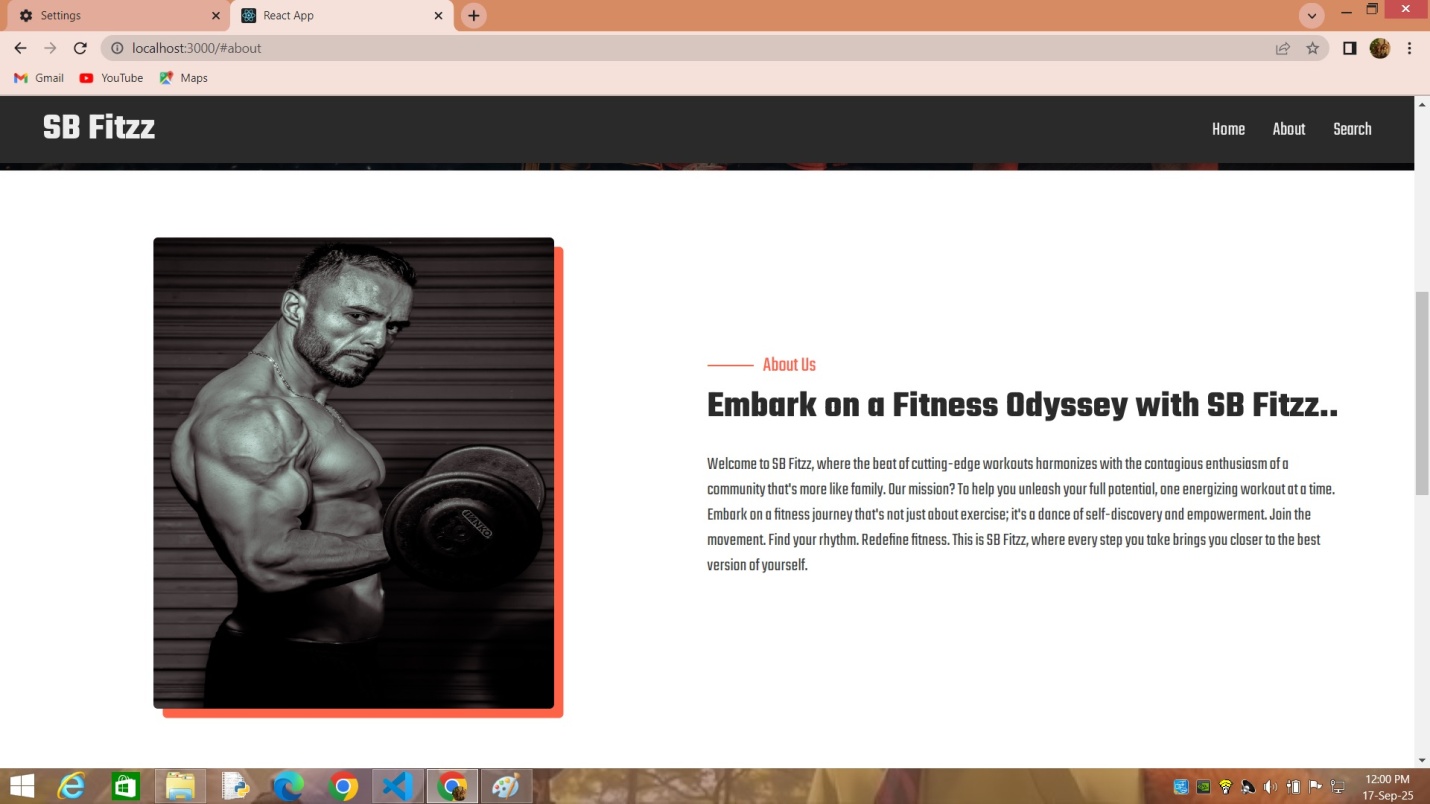
* Focus on critical components (Dashboard, WorkoutForm).
* More integration testing required for complete coverage.

# Screenshots or Demo:

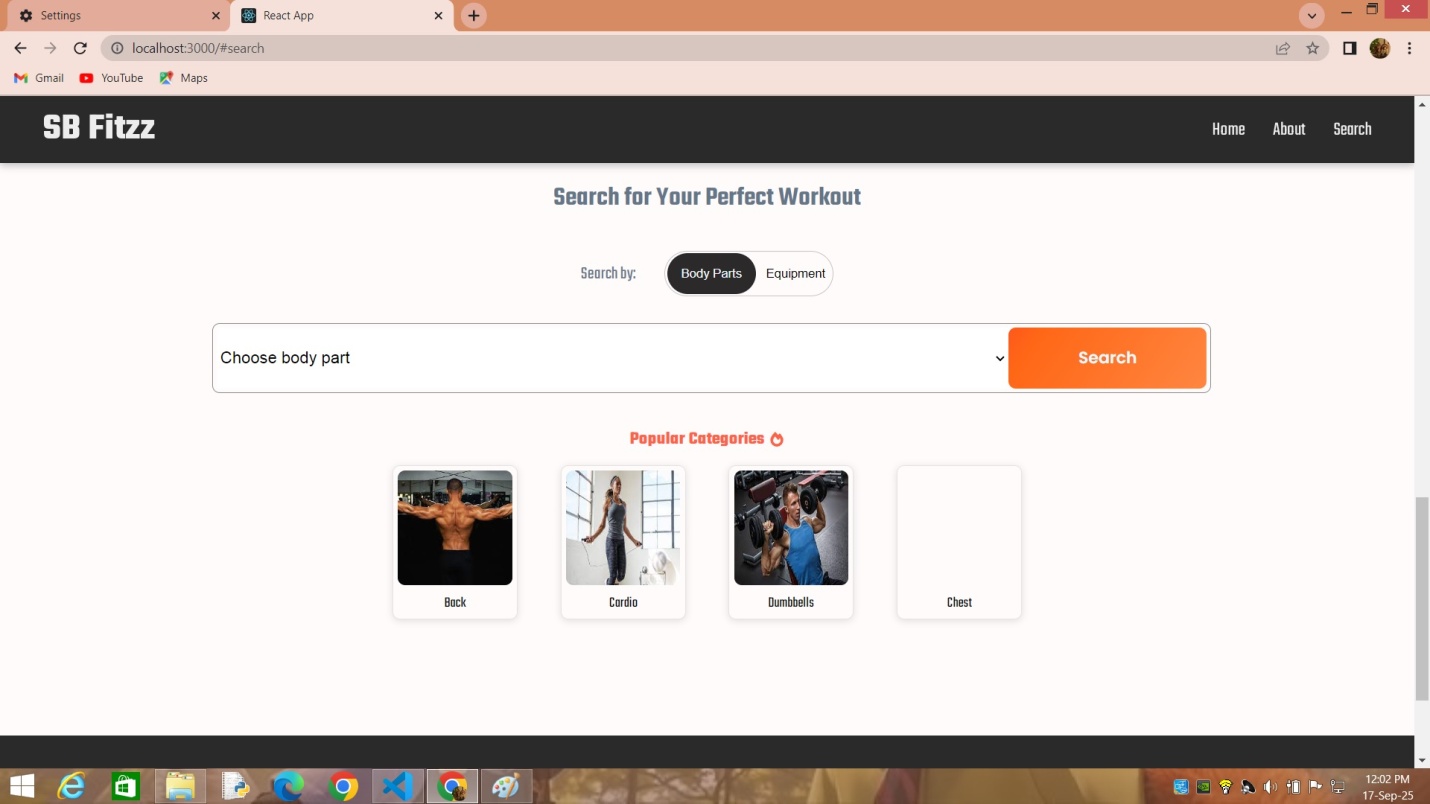
## Home Page:



## About Page:



## Search Page:



# Known Issues:

* No backend integration (data lost on refresh).
* Limited form validation.
* No charts/visualization implemented yet.

# Future Enhancements:

* Add authentication & user profiles.
* Save data using local Storage or backend API.
* Charts and graphs for fitness progress.
* Calorie tracker and BMI calculator.
* Support for dark mode and themes.
* Export fitness reports (Excel/PDF).

https://ssl.gstatic.com/ui/v1/icons/mail/images/cleardot.gif