FITFLEX-PERSONAL FITNESS COMPANION

# Project Documentation

1. **Introduction**

Project Title: FitFlex – Personal Fitness Team Leader:

KEERTHIKA [S - 202400328@sigc.edu](mailto:S%20-%20202400328@sigc.edu)

Team Members:

ABINAYA K – [202400993@sigc.edu](mailto:202400993@sigc.edu) KAVIYA SRI B - [202400884@sigc.edu](mailto:202400884@sigc.edu) MALARVIZHI A – [202400395@sigc.edu](mailto:202400395@sigc.edu)

HARINI S – [202400400@sigc.edu](mailto:202400400@sigc.edu)

# Project Overview

Purpose: FitFlex is a personal fitness web application designed to help users track workouts, monitor progress, and manage their fitness journey.

Features:

* + Log daily workouts (type, duration, intensity)
  + Add custom exercises and routines
  + Pre-built workout templates (e.g., strength, cardio, HIIT)
  + Goal setting and tracking (e.g., lose weight, build muscle)
  + Share progress with friends or community
  + Join fitness challenges

# Architecture

Component Structure:

Outline the structure of major React components such as Dashboard, Workout Planner, Nutrition Tracker, Progress Tracker, and Profile.

State Management:

Global state management handled with Redux (for workouts, user data, and progress), Context API for theme and authentication.

Routing:

React Router is used for navigation across pages like Dashboard, Workouts, Nutrition, Progress, and Profile.

# Setup Instructions

Prerequisites: Node.js, npm/yarn installed.

Installation:

git clone <repository-url> cd fitflex

npm install

**npm start**

# Folder Structure

Client:

/components: Reusable UI components (buttons, cards, charts)

/pages: Main pages like Dashboard, Workouts, Nutrition, Profile

/assets: Images, icons, and styles

/utils: Helper functions (e.g., calorie calculator, date formatter)

# Running the Application

Frontend:

npm start

Runs the development server locally.

# Component Documentation

Key Components:

Dashboard: Overview of workouts, nutrition, and progress.

Workout Planner: Create and manage personalized workout routines.

Nutrition Tracker: Log daily meals and calories.

Progress Tracker: Display charts for weight, workouts, and progress.

Reusable Components:

Buttons, Input fields, Cards, Progress Bars, Charts, Modals.

# State Management

Global State:

Managed with Redux for workouts, nutrition logs, user authentication.

Local State:

Used for handling form inputs, modal visibility, and UI interactions.

# User Interface

Provide screenshots or GIFs showing:

Dashboard overview

Workout planner form

Nutrition tracker logs

Progress charts and statistics

# Styling

CSS Frameworks/Libraries: Tailwind CSS for styling; Styled-components for custom themes.

Theming: Supports light/dark theme toggle for better user experience.

# Testing

Testing Strategy:

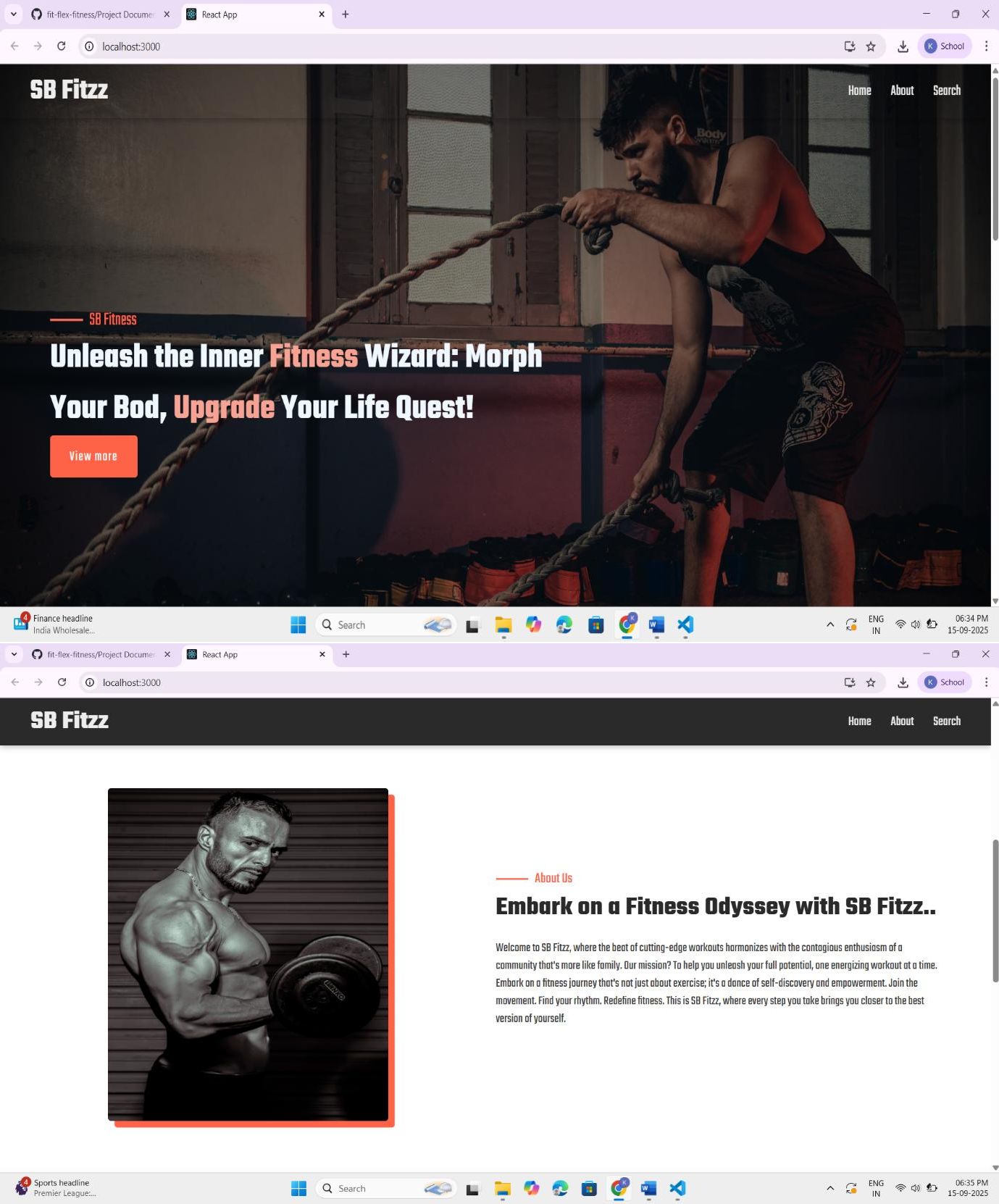
Unit testing with Jest, Component testing with React Testing Library, and end-to-end testing with Cypress.

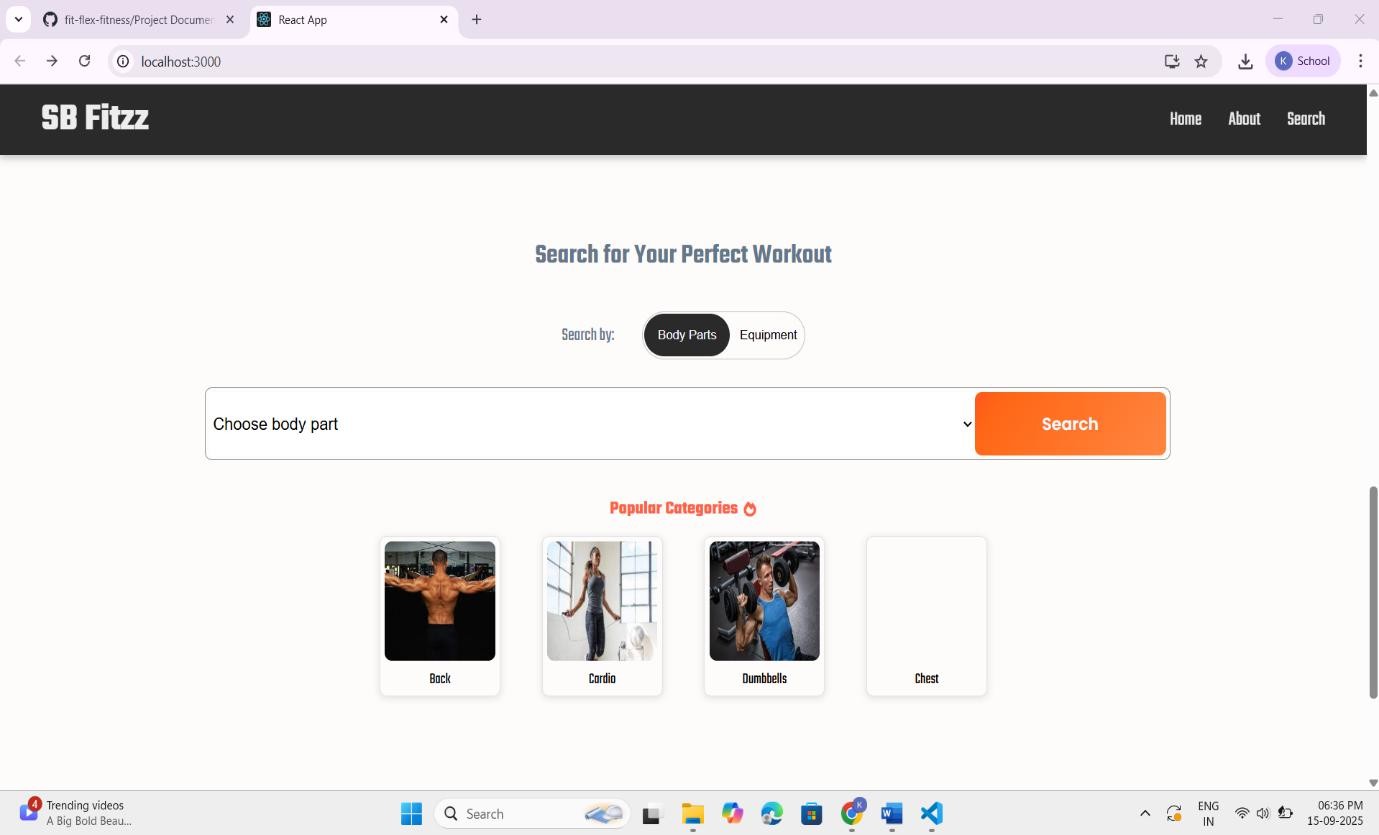
Code Coverage:

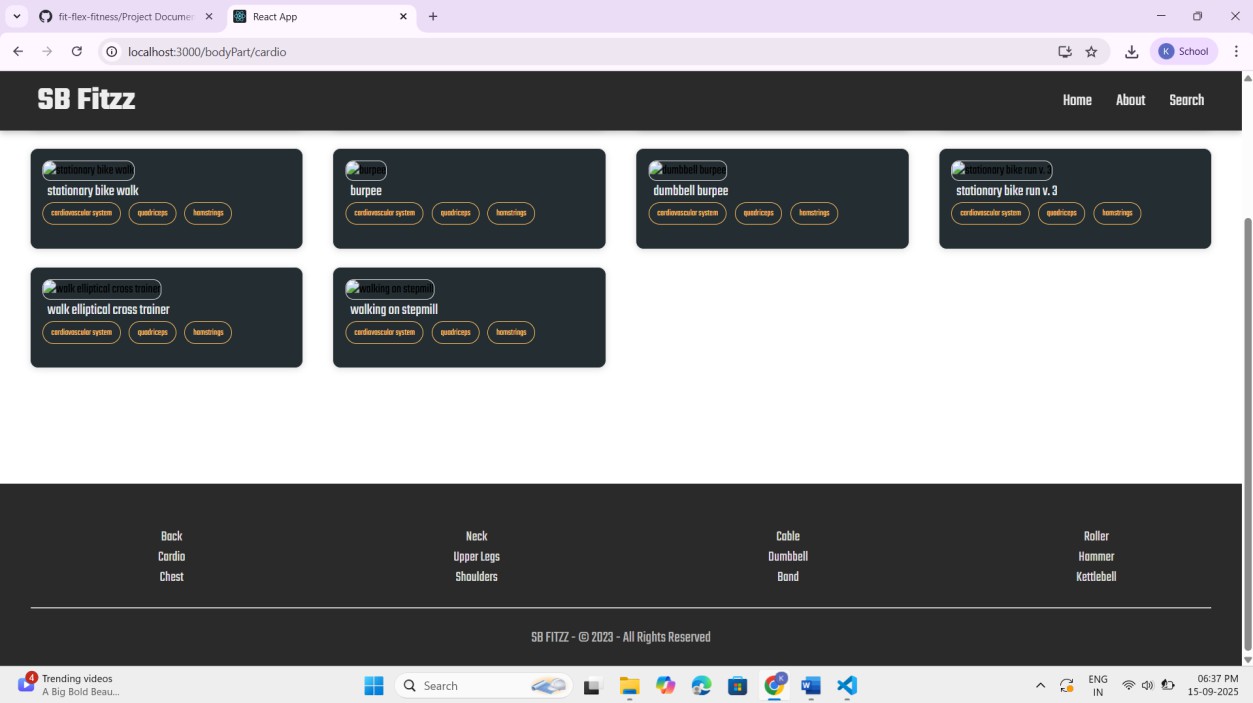
Ensuring adequate test coverage with Jest reports.

# Screenshots or Demo: <https://drive.google.com/file/d/12WHM9HXo8LGyYRu61cZ0Kh5ociOW1VTU/view?usp=drive_link>

Provide screenshots:







Dashboard

Workout Planner

Nutrition Tracker

Progress Tracking

# Known Issues

Integration with wearables (Fitbit, Apple Watch) may not sync in real-time.

Large datasets in progress charts may slow rendering.

# Future Enhancements

AI-based workout recommendations

Social community features (friends, challenges)

Enhanced analytics and performance insights

Mobile app version for iOS/Android