FRONTEND DEVELOPMENT

Project Documentation

1. Introduction

- Project Title:

  FITFLEX: Your Personal Fitness Companion

- Team ID: NM2025TMID38059

- Team Leader:

Name: THULASI

Mail id: thulasiumesh24@gmail.com

- Team Members:

Name: KUMUTHA

Mail id: kumutharamesh19@gmail.com

Name: FASIHA FARIN

Mail id: fasihafarink@gmail.com

Name: JOYCE BEULAH

Mail id: joycebeulah01@gmail.com

2. Project Overview

Purpose:

The output page is designed to show the results after a user submits input or performs an action. It gives clear and structured feedback to the user.

Features:

- Displays processed results clearly.
- Provides navigation back to the home or input page.
- Responsive design that works on desktop and mobile.

3. Architecture

- Component Structure:

OutputPage – Main wrapper for the page.

ResultCard – Shows individual result data.

Header – Contains the title and navigation.

Footer – Basic project footer.

- State Management:

Used React Context to share result data across components.

- Routing**:**

React Router is used to navigate between input and output pages.

4. Setup Instructions

- Prerequisites:

Node.js (v16 or above)

npm or yarn

- Installation:

Copy code

Bash

```
git clone [repository-url]

cd client

npm install
npm start
```

5. Folder Structure

- Client:

src/components → Contains reusable components like Header, Footer, ResultCard.

src/pages → Contains OutputPage.jsx.

src/assets → Images, icons, and styles.

- Utilities:

src/utils/formatData.js → Helper functions for formatting output.

src/hooks/useResults.js → Custom hook for fetching or storing results.

6. Running the Application

Copy code

Bash

```
cd client

npm start
```

This starts the frontend server on http://localhost:3000.

7. Component Documentation

- OutputPage:

Displays the main results section and arranges components.

- ResultCard:

Shows each result item with a title, description, and optional icon. Props: title, value.

- Header/Footer:

Navigation and basic info.

8. State Management

- Global State:

Results are stored in a global context so any page can access them.

- Local State:

Components like ResultCard manage their display (expanded/collapsed view).

9. User Interface

- Results shown in cards with clean spacing.
- Navigation bar at the top.
- Simple buttons for "Go Back" or "New Submission".

10. Styling

- CSS Frameworks/Libraries: Tailwind CSS for styling.
- Theming: Light theme with consistent primary colors for buttons and highlights.

11. Testing

Testing Strategy:

- Unit testing with Jest for ResultCard and utilities.
- Integration testing with React Testing Library for OutputPage.
- Code Coverage: Jest coverage reports are enabled.

12. Known Issues

- Page reload may clear results if not persisted in localStorage.
- Styling may appear inconsistent on very small screens.

13. Future Enhancements

Add animation to result cards. Provide export options (PDF).