**GYM IT**

**DOCUMENTATION**

# 1.Introduction

**Project Title: GYM IT**

**Team Members:**

* **Rishi** – Team Leader
* **Monesh** – Team member
* **Cheran -** Team member
* **Gnana Prakash -** Team member
* **Dinesh Kumar -** Team member

# 2. Project Overview

This project appears to be a React-based application. It includes essential frontend files like App.js, index.js, and App.css. The presence of package. Json indicates that the project relies on Node.js and npm for dependency management.

3. Architecture

* **Frontend:** React (JavaScript-based UI framework).
* **State Management:** Likely using React hooks or Context API (can be confirmed in the code).
* **Styling:** Managed using App.css
* **Testing:** Setup available with setupTests.js.

# 4. Setup Instructions

**Prerequisites**

* Install **Node.js** (latest LTS version recommended).
* Install **npm** (comes with Node.js).

**Installation**

# Clone the repository

git clone <repository-URL>

# Navigate to the project directory

cd extracted project/code

# Install dependencies

npm install

# 5.Folder Structure

/code

│── /src

│ ├── App.js # Main React component

│ ├── index.js # Entry point of the application

│ ├── App.css # Styling file

│ ├── logo.svg # Logo asset

│ ├── setupTests.js # Testing setup

│ ├── reportWebVitals.js# Performance measuring script

│── .gitignore

│── package.json

│── package-lock.json

│── README.md

# 6. Running the Application

# Start the development server

npm start

### Build for production

npm run build

# 7.Component Documentation

* **App.js** – Main component that controls the application’s rendering.
* **index.js** – Renders App.js into the root DOM node.
* **logo.svg** – Static asset used in the UI.

# 8. State Management

The project might be using React’s built-in state management (useState and use Effect) or Context API. This can be confirmed by checking App.js.

# 9.User Interface

The UI is styled using App.css, and it includes a logo image (logo.svg).

# 10. Styling

The project primarily uses CSS for styling. Any additional libraries (e.g., Bootstrap, Tailwind) should be checked in package.json

# 11.Testing

* The presence of setupTests.js suggests that Jest is used for testing.
* To run tests:

npm test

# 12. Known Issues

* **UI Bugs** – Misalignment, responsiveness issues, or broken layouts.
* **Performance Issues** – Slow loading, high memory usage, or lagging interactions.

# 13. Future Enhancements

* **UI/UX Improvements** – Enhancing the design, adding animations, or improving responsiveness.
* **New Features** – Adding functionalities like authentication, search filters, or user profiles.
* **Performance Optimization** – Reducing load times, improving API calls, or optimizing database queries.