Store Manager .js

# Introduction

Project Title: Store Manager

Team Leader:  
 N.Kirubakaran

Team Members:  
-R.Sanjay

-V.Deepak

-A.Naveen

-M.Manimaran

# Project Overview

**Store Manager** is a web-based application designed to manage inventory, sales, and employee tasks in a retail environment. The application enables store managers to add products, update stock, track sales, view analytics, and manage staff permissions.

**Tech Stack (Example):**

* **Frontend**: React.js
* **Backend**: Node.js + Express (or Firebase)
* **Database**: MongoDB (or Firestore)
* **State Management**: Redux Toolkit
* **Styling**: Tailwind CSS / SCSS
* **Authentication**: JWT / Firebase Auth
* **Testing**: Jest + React Testing Library

# Architecture

This component handles user login, registration, and role-based access control. It ensures that only authorized personnel (such as store managers, cashiers, or inventory staff) can access specific features of the system. Technologies like JWT (JSON Web Tokens) or OAuth2 are commonly used for secure user sessions.

# Setup Instructions

Prerequisites

* Node.js ≥ 18
* npm or yarn
* MongoDB or Firebase account

# Folder Structure

Client (React)

├── Components

├── Pages

├── Redux Store

├── Services (API calls)

├── Routes (Protected/Public)

Server (Node.js / Express)

├── Routes (Product, Sales, Auth)

├── Controllers

├── Models (MongoDB)

├── Middleware (Auth, Error Handling)Running the Application

# Component Documentation

Key Components:  
- Navbar: Navigation across the app  
- Dashboard: Displays summary of user fitness stats  
- Workout Tracker: Allows adding and viewing workouts  
- Diet Planner: Tracks daily meals and nutrition  
- Profile Manager: User information and preferences

Reusable Components: Buttons, Modals, Cards, and Input Fields.

# State Management

 authSlice: Handles login, logout, user roles

 productSlice: Stores product data

 salesSlice: Handles order and transaction data

 uiSlice: Controls loading states, modals, etc.

# User Interface

* Fully responsive design using **Tailwind CSS**
* Accessible form components
* Dark mode support
* Icons via Heroicons / Font Awesome

# Styling

 Utility-first styling with Tailwind (or use SCSS for component-based styling)

 Theming available through global config (tailwind.config.js)

# Testing

Product CRUD functionality

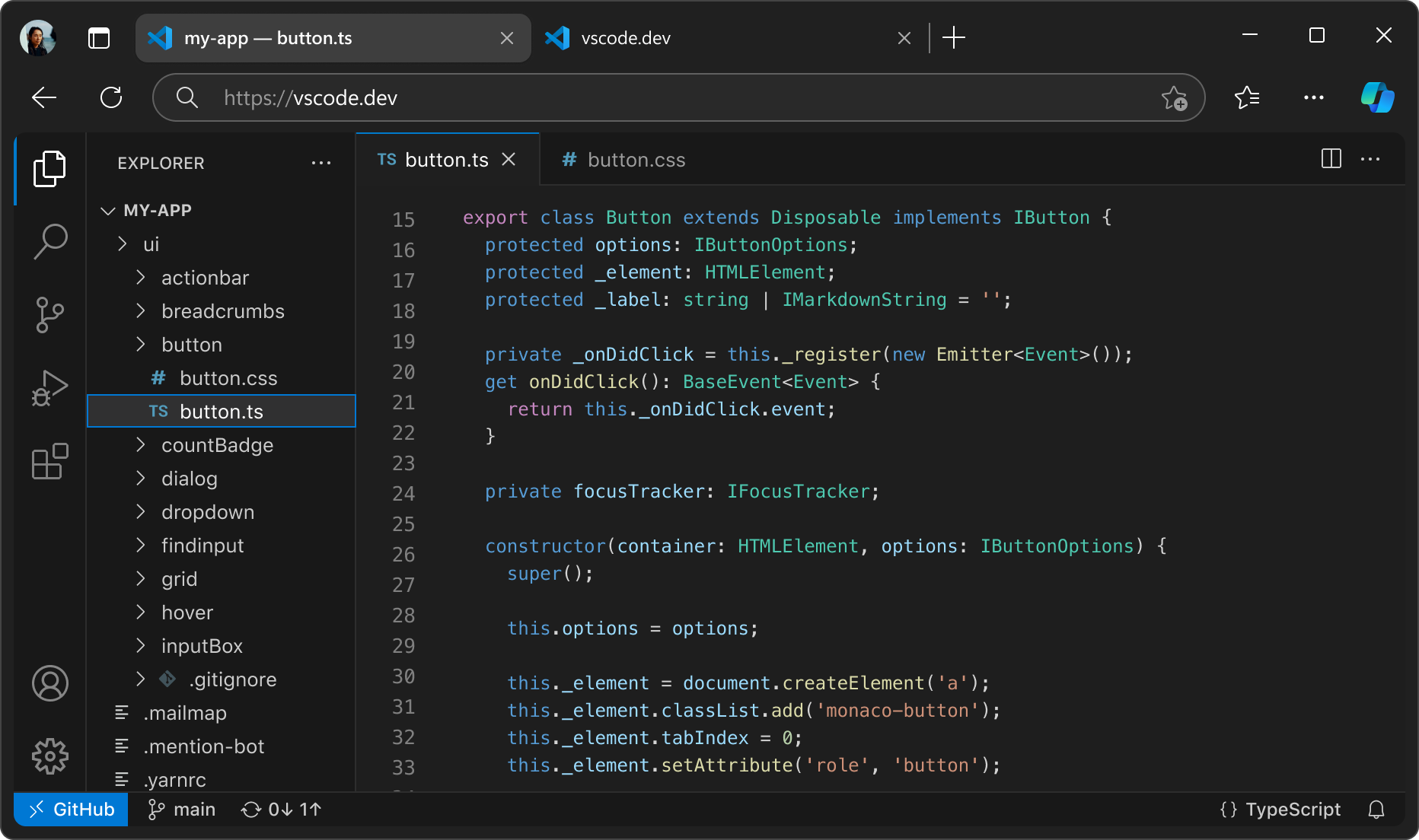
Auth flow

API integration

Component rendering

# Screenshots or Demo

Screenshots and demo links will be provided showcasing the application features

.

# Known Issues

Currently, the app may face performance issues with large datasets.

# Future Enhancements

Planned features include AI-powered workout recommendations, social features, and wearable device integration.