# PROJECT DOCUMENTATION

**STORE MANAGER: KEEP TRACK OF INVENTORY**

# Introduction

**Project Title:** Store Manager: Keep Track of Inventory

**Team ID :** NM2025TMID38071

**Team Leader:** Priyadharshini R

Mail id: [priyadharshini0510ravi@gmail.com](mailto:priyadharshini0510ravi@gmail.com)

**Team Members:** Rosy S

Mail id: [blackberry082004@gmail.com](mailto:blackberry082004@gmail.com)

Jeeva G

Mail id: [jeevagopi788@gmail.com](mailto:jeevagopi788@gmail.com)

Dharshini D

Mail id: [dharshu.devi.29@gmail.com](mailto:dharshu.devi.29@gmail.com)

# Project Overview

**Purpose:**The Store Manager project is designed to maintain healthy stock levels, manage product updates, track sales, and ensure smooth store operations. It simplifies inventory management with features like automatic stock updates, search, alerts, and sale records.

**Features:**  
1. Inventory Management: Maintain stock levels and update products on time.  
2. Stock Updates: Automatically update stock when products are sold or added.  
3. Cart: Add products with desired quantity for sale.  
4. Checkout at Cart: Clear cart, update inventory, and create a sale record.  
5. Adding New Products: Add new items with product name, image URL, price, stock, and tags.  
6. Alert View: Highlight depleting stocks with red background alerts.  
7. Search: Search functionality for inventory and catalog.  
8. Sale Records: Track sales with details of value, products, and datetime.

# Architecture

**Component Structure:**Inventory Component (manage stock list)  
Cart Component (manage products added for sale)  
Checkout Component (handle sale process)  
Product Form Component (add new products)  
Alert Component (show low stock alerts)  
Search Component (filter products)  
Sale Records Component (view past sales)

**Routing:**React Router can be used for navigation between pages (Inventory, Cart, Sale Records, etc.).

# Setup Instructions

**Prerequisites:**

Node.js, Github, React.js, Visual studio code

**Installation:**1. Clone repository.  
2. Run `npm install` to install dependencies.

# Folder Structure

/components → UI components (Inventory, Cart, ProductForm, etc.)  
/pages → Full pages for Inventory, Sales, Checkout  
/assets → Images, styles  
 /utils → Helper functions, custom hooks

# Running the Application

Run frontend server: `npm start`

# Component Documentation

1.Inventory Component: Displays and updates stock levels.  
2. Cart Component: Handles products added for sale with quantities.  
3. Product Form Component: Allows addition of new products.  
4. Records Component: Displays all past sales with datetime.

# State Management

**Global State:** Inventory, Cart, Sale Records.

**Local State:** Product form fields, search queries, alerts.

# User Interface

Stock list with low stock alerts.  
Cart checkout page.  
Product search functionality.  
Sale history records.

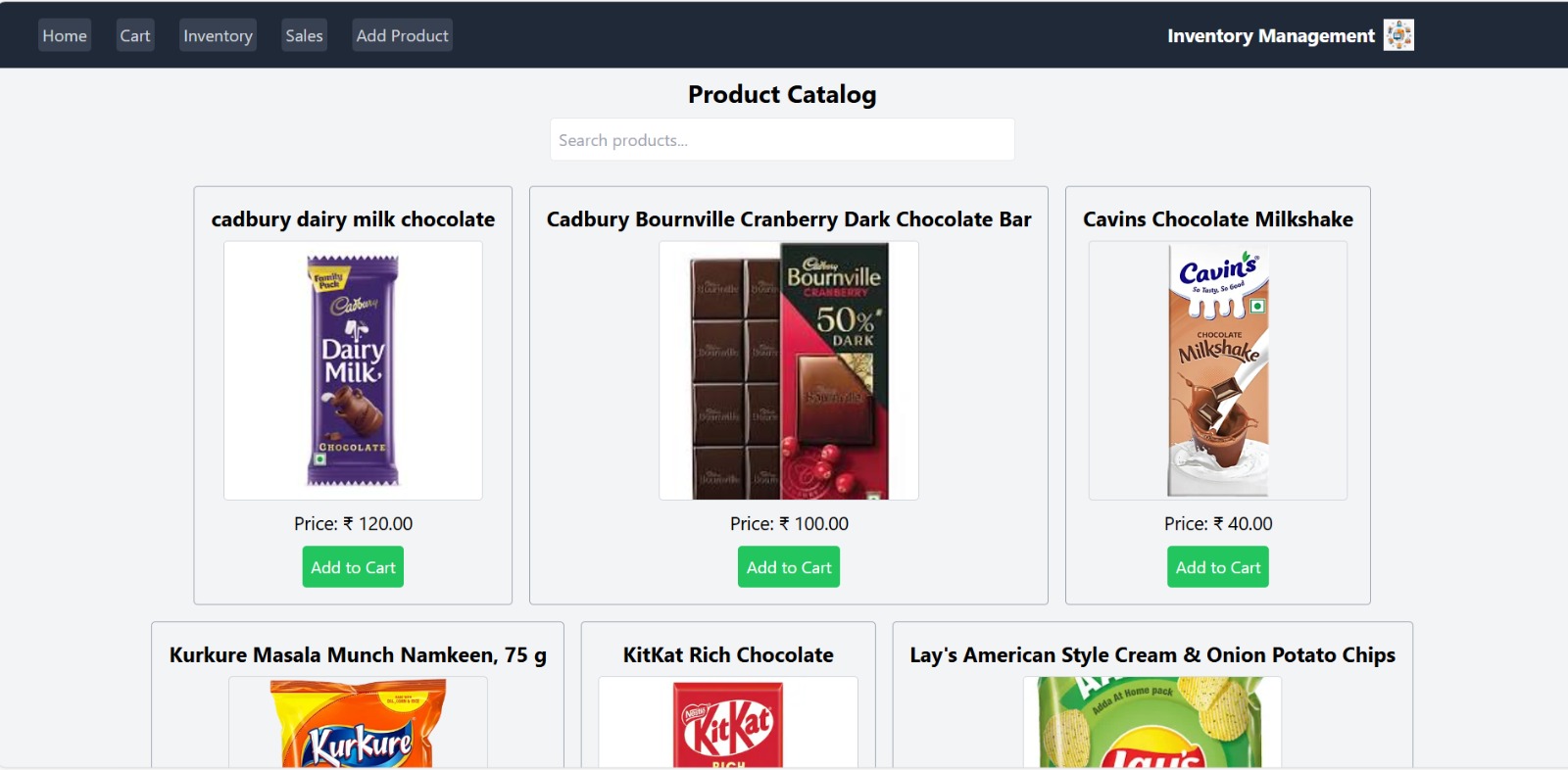
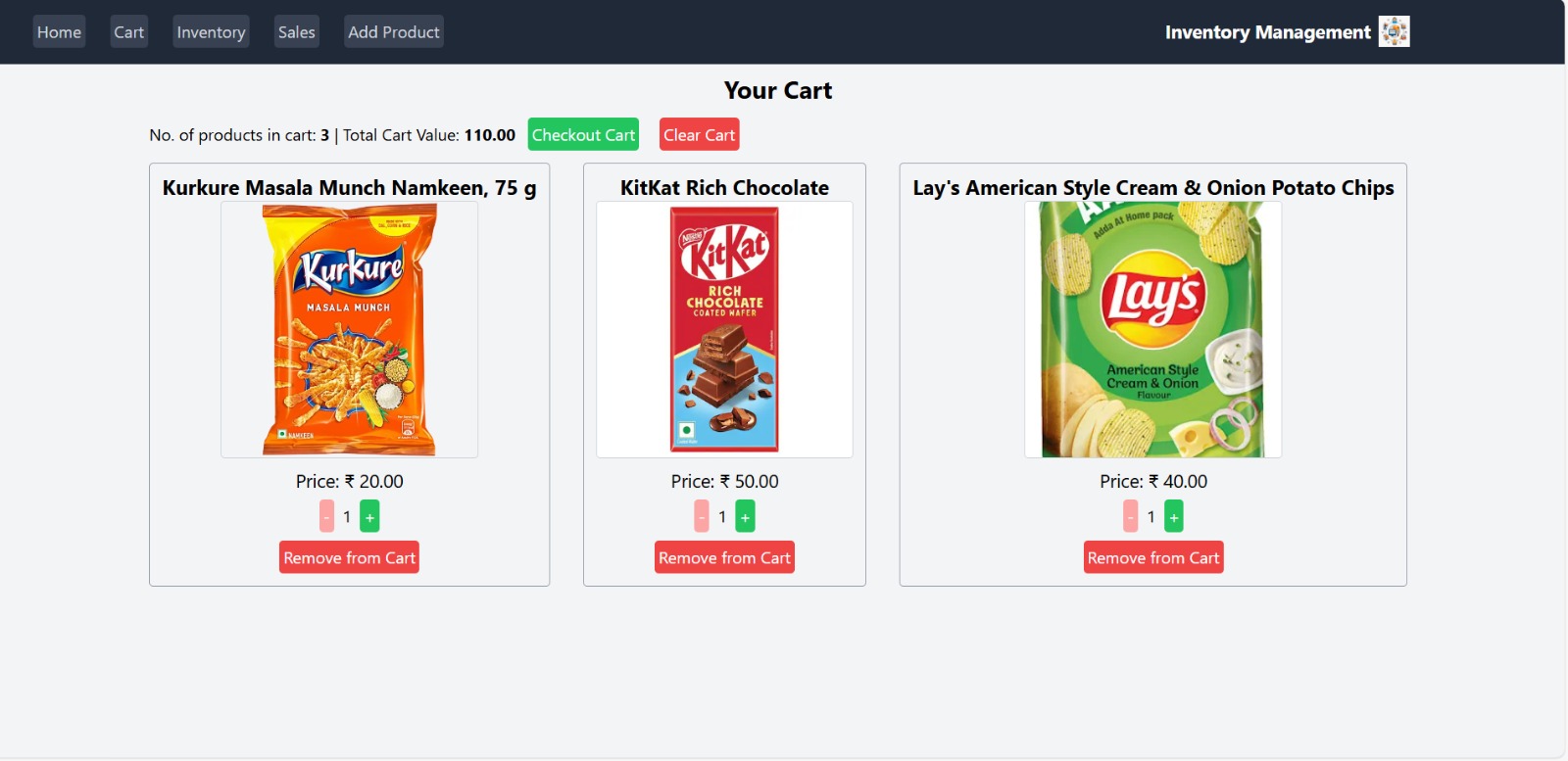
# Styling

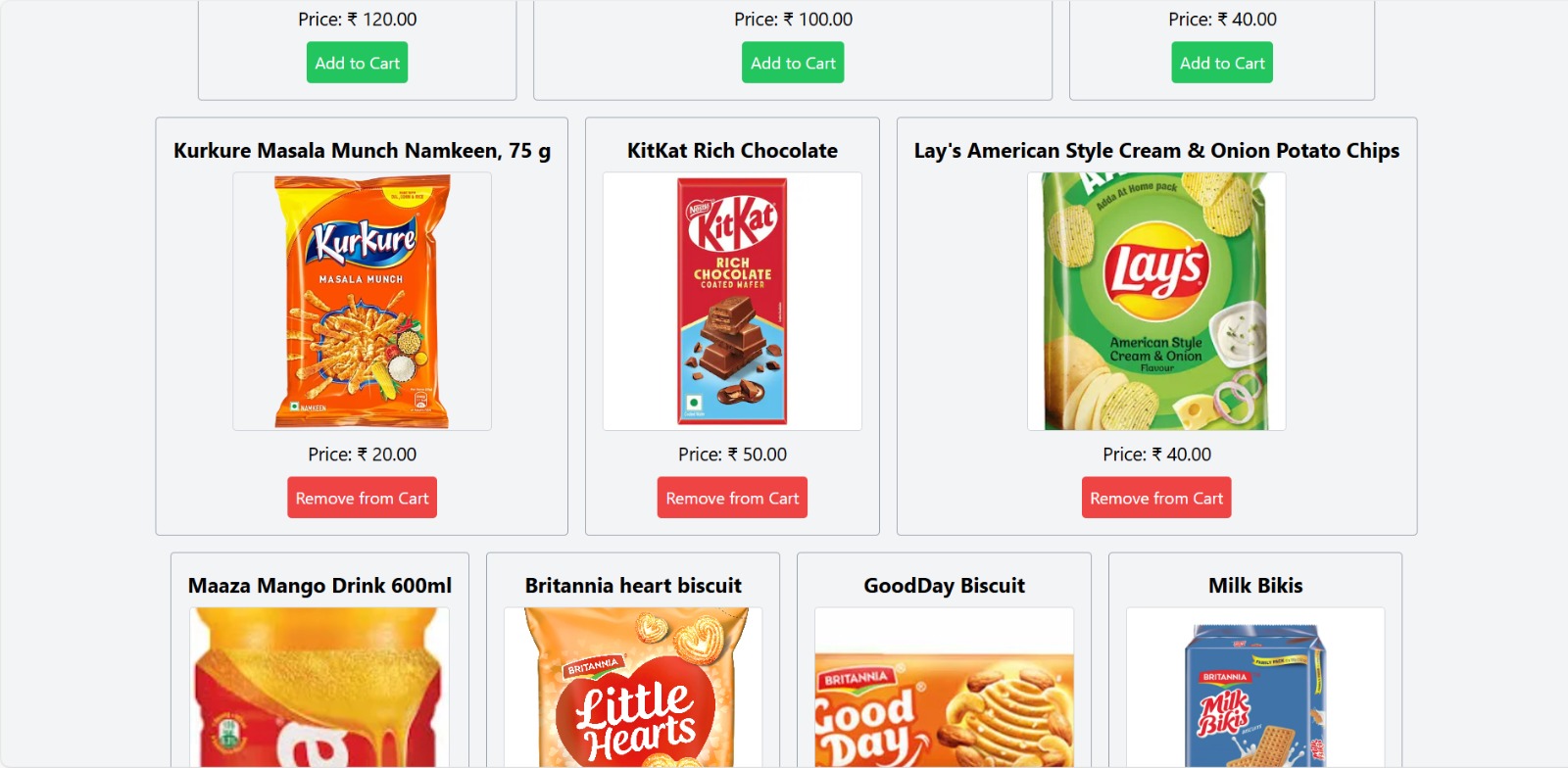
Tailwind CSS , APP.CSS.  
Custom theming for highlighting alerts.

# Testing

Unit tests using Jest & React Testing Library.  
Integration tests for cart & checkout.

# Screenshots



# Future Enhancements

Advanced reporting & analytics.  
Role-based user access (Admin, Staff).  
Barcode/QR code scanning for products.