TABLE OF CONTENTS

Introduction

This project, Store Manager: Keep Track of Inventory, is a software application developed to efficiently manage and monitor store inventory. The primary motivation for building this project was to eliminate the challenges of manual stock management, such as misplacement of records, human errors, and difficulties in updating stock levels. By providing a computerized solution, the system ensures accuracy, reliability, and easy access to inventory details whenever required.

Inventory management is an essential part of any business operation, whether small or large. Retailers, wholesalers, and warehouses depend on maintaining accurate stock information to ensure customer satisfaction and avoid financial losses. Traditionally, businesses relied on handwritten records or spreadsheets, which often became tedious, time-consuming, and prone to errors. The proposed Store

Manager system addressestheselimitationsby offering a structured and automated way to track stock levels, manage products,andgenerateusefulreports for decision-making.

The project is designedtoprovideasmoothuser experience by implementing features such as adding new products, updatingexistingstockdetails,deleting records, and searching for specific items. Store managers can easilycheckcurrentstocklevels, identify items that are running low, and restock them on time. If connectedtoadatabase,thesystemensures that all updates are stored securely and can be retrieved wheneverrequired.

From a learningperspective,thisprojecthighlights the importance of database management and CRUD operations(Create,Read,Update,Delete).Itdemonstrates how software applications can simplify realworldbusinessprocessesbyreducingworkload, minimizing errors, and ensuring efficiency. Additionally,theprojectcanbeextendedinthe future to include advanced features such as barcode scanning,salestracking,suppliermanagement, and integration with online stores.

Anotherkeyhighlightofthisprojectisitsadaptability. It can be implemented in small shops, departmentalstores,warehouses,orevenlarge-scale businesses, with minimal changes. The system is designedtobelightweightanduser-friendly,ensuring that even people with limited technical knowledge can useiteffectively.

The overallaimofStoreManager:KeepTrack of Inventory is not only to provide a practical solution for managingstockbutalsotodemonstratetheapplication of software development concepts in solving reallife problems.Bybridgingthegapbetween manual processes and digital solutions, this project shows how technologycanbringaccuracy,speed,and convenience to business operations.

In conclusion, theprojectservesasafoundation for understanding how software applications can improve day-to-daybusinessactivities.Itcombines both technical learning and practical application, offering valuableinsightsintodatabasehandling, user interface design, and efficient process management.

StoreManager – Keep Track of Inventory

Submitted by

[Swathy .V]

[Sakthi.T]

[Sofia.J]

[Teju Sree . S]

Department of Computer Applications

[Dr. M. G. R. Janaki College of Arts and Science for Women]

Academic Year: 2025 – 2026

Project Overview

Purpose

The purpose of this project was to design and implement a functional inventory management system that simplifies the process of tracking products, stock levels, and updates in a store. The project provides a platform for exploring basic database handling, CRUD (Create, Read, Update, Delete) operations, and efficient user interface design. It allows the practical application of programming concepts in a way that directly contributes to solving real-world business problems.

Features

Add Product: Allows store managers to add new items with details such as product name, category, quantity, and price.

Update Inventory: Provides functionality to update stock levels and edit product information.

Delete Product: Enables removal of items that are no longer available or discontinued.

Search Functionality: Helps quickly locate products by name or ID.

Stock Alerts: Highlights products that are running low in quantity.

Report Generation: Provides a summary of inventory status for better decision-making.

Architecture

Component Structure

The application is divided into multiple functional modules:

Login/Authentication Module: Ensures that only authorized users can access the system.

Product Management Module: Handles adding, updating, and deleting product records.

Search & Filter Module: Fetches and displays product details based on specific criteria.

Report Module: Generates simple reports or summaries of available stock.

This modular design makes the project easier to maintain and extend in the future.

State Management

The system uses basic state management for tracking and updating data across modules:

Add / Update Functions: Manage changes in product details.

Validation: Ensures correct data entry for product ID, stock, and price.

Data Storage: Can be implemented using files, arrays, or a database depending on requirements.

Since the project is small, lightweight state management was used without complex frameworks.

Routing

The application does not use complex routing, as the project scope was limited to a single main interface for product and inventory management. All functions are accessible from the same module, making the system simple and user-friendly.

Prerequisites

Setup Instructions

- Node.js (v14 or higher)
- npm or yarn package manager
- Vite (for React setup)
- JSON Server (for backendsimulation)

Installation

1. Clone the project repository.
2. Install dependencies using npm install.
3. Start JSON Server using:
   npx json-server --watchdb/db.json --port 3000
4. Run the frontend using:
   npm run dev

Folder Structure

7. Open the applicationinyour browser using the Vite local server link.

db/ → Contains db.json with all inventory item details.

public/ → Stores static assets like images, icons, or PDFs.

src/ → Main source code.

components/ → Contains InventoryList, ItemDetails, and StockControls.

App.jsx → Root component.

index.css → Styles. main.jsx

→ Entry point.

Running the Application

1. Start JSON Server with the database.
2. Run the frontend development server using npm run dev.
3. Access the app in the browser at http://localhost:5173/.

Component Documentation

Key Components

Product Manager: Core component that handles adding, updating, deleting, and viewing product details.

Inventory Controller: Manages stock levels and checks for low inventory.

Search Module: Allows users to search products by ID, name, or category.

Report Generator: Summarizes stock status and generates inventory reports.

Reusable Components

Input Forms: Reusedfor product entry and updates.

Validation Functions:Shared across modules to ensure correct data input.

UI Elements (buttons,menus, tables): Used consistently for navigation and display.

State Management

Global State

Not implemented, as the project size is small. Data is managed through local storage (database or file handling).

Local State

Local state was sufficient to handle:

Current product details (ID, name, quantity, price)

Stock updates (increment/decrement)

Search results for quick product lookup

Temporary data during add/update/delete operations

User Interface

The user interface (UI) of the Inventory Management System has been designed with a focus on simplicity, clarity, and ease of use. Inventory management is a task that store managers perform frequently, and therefore, the application must have an intuitive design that minimizes complexity while maximizing functionality. To achieve this, the interface has been kept minimalistic, allowing users to navigate through the inventory and perform stock operations without unnecessary distractions.

The layout is divided into two major sections: the inventory list area and the stock management area. The inventory list area displays all items currently in stock. Each entry in the list includes details such as the item name, category, quantity available, unit price, and supplier information. This not only provides essential information at a glance but also improves usability by incorporating a structured, easy-to-read format.

The stock management area is where the selected item is displayed in detail. Here, the user can view all relevant information about the item, including current stock levels, reorder thresholds, and recent stock movements. Below this section, actionable controls are placed in an accessible position. These include buttons for adding stock, deducting stock, and updating item details, all styled consistently to maintain uniformity. The buttons use clear labels and icons, ensuring that even new users can understand their function without additional guidance.

Responsiveness has also been prioritized while designing the interface. The use of Flexbox and Grid layouts ensures that the application adapts automatically to different device sizes. On larger screens such as desktops or laptops, the inventory list and stock management sections appear side by side for better visibility, while on smaller devices like tablets or smartphones, the layout stacks vertically to facilitate easy navigation. This adaptability ensures a smooth user experience across a wide range of devices.

Color choices in the UI were made to balance readability and visual appeal. A light background was selected to provide clarity, while key elements like action buttons and alerts are highlighted using distinct colors to draw attention. Text is simple, legible, and spaced appropriately to prevent visual clutter. Icons and subtle design elements enhance the overall aesthetic without overcomplicating the interface.

Usability testing was conducted informally by interacting with the application multiple times. The focus was on whether users could quickly identify how to view stock levels, add or remove inventory, and update item details. Feedback from these tests was incorporated into the final layout to reduce unnecessary steps and streamline the workflow.

In summary, the user interface of the Inventory Management System effectively combines functionality with a clean, intuitive design. By maintaining a minimal yet engaging layout, the application ensures that store managers can manage inventory efficiently without being overwhelmed by complexity. The responsive design principles make the system adaptable to multiple devices, ensuring usability in diverse working environments. Future enhancements, such as automated alerts, barcode integration, and theme customization, can further elevate the usability and interactivity of the system.

Styling

- Plain CSS was used for styling.
- Layout handled using Flexbox and Grid.
- A light theme with highlighted buttons was chosen for simplicity and clarity.

Testing

Testing of the Inventory Management System was conducted manually to ensure that all core functionalities work correctly and that the application is stable under normal usage conditions.

Manual testing included:

Verifying that inventory items can be added, updated, and deleted correctly.

Checking that stock levels are accurately updated when items are added or removed.

Ensuring that reorder alerts trigger correctly when stock falls below the minimum threshold.

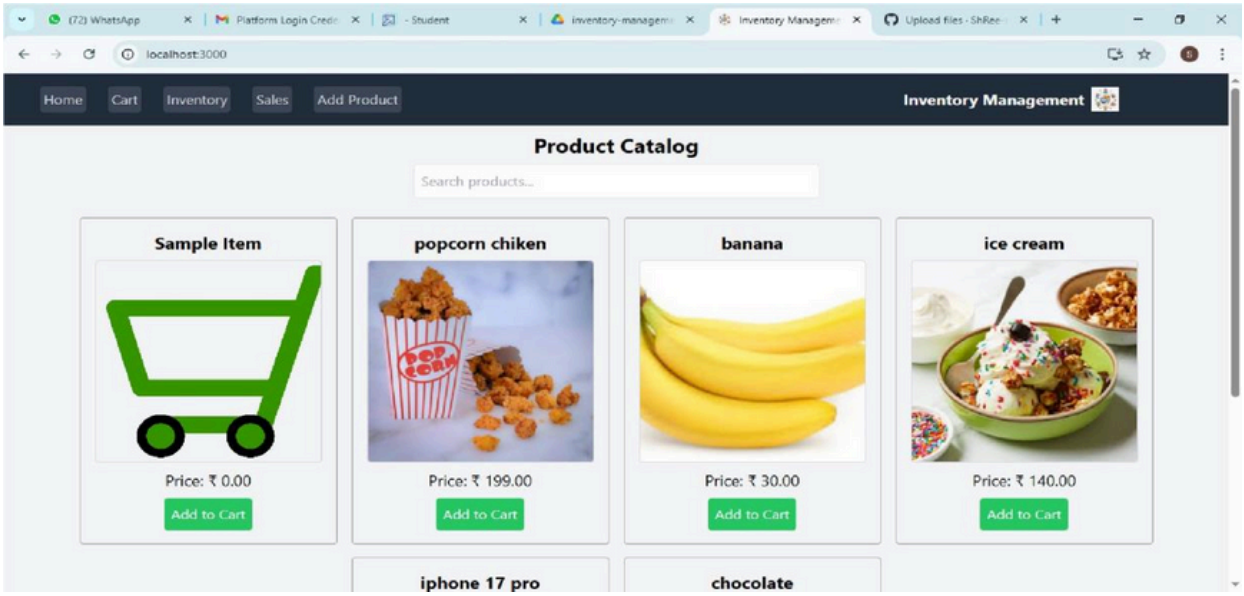
Testing the inventory list display, ensuring all item details such as name, quantity, category, and supplier are shown accurately.
Evaluating the user interface across multiple screen sizes to confirm that the layout remains responsive and easy to navigate on desktops, tablets, and smartphones.

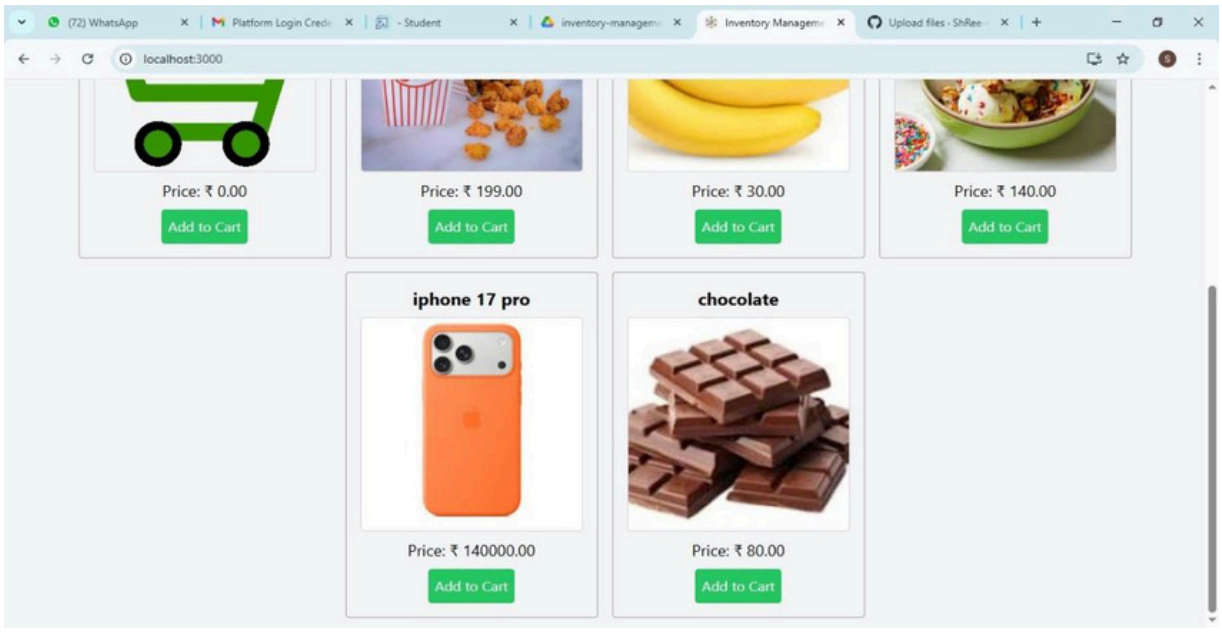
In addition to these manual tests, the system was also verified for error handling and stability. For example, scenarios such as attempting to update an item that does not exist, entering invalid stock quantities, or leaving required fields empty were tested to ensure that the application handled these cases gracefully without crashing. Console logs were monitored during testing to identify potential warnings or errors, and fixes were applied where necessary.

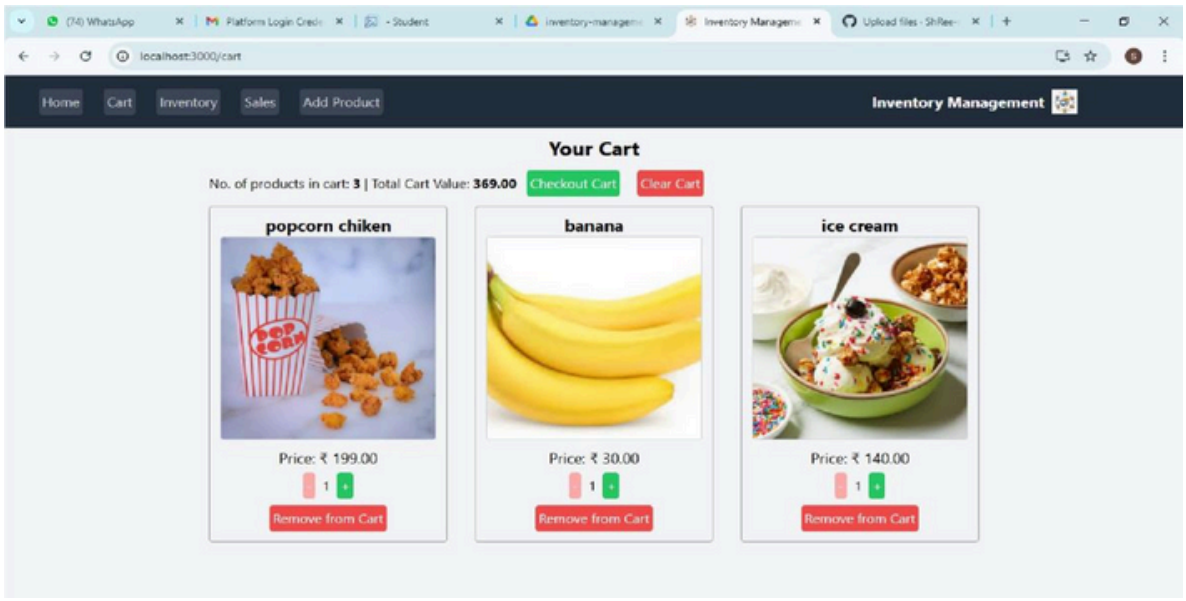
These tests were essential in improving the reliability and robustness of the system, ensuring that store managers can manage inventory smoothly and efficiently without interruptions or unexpected behavior.

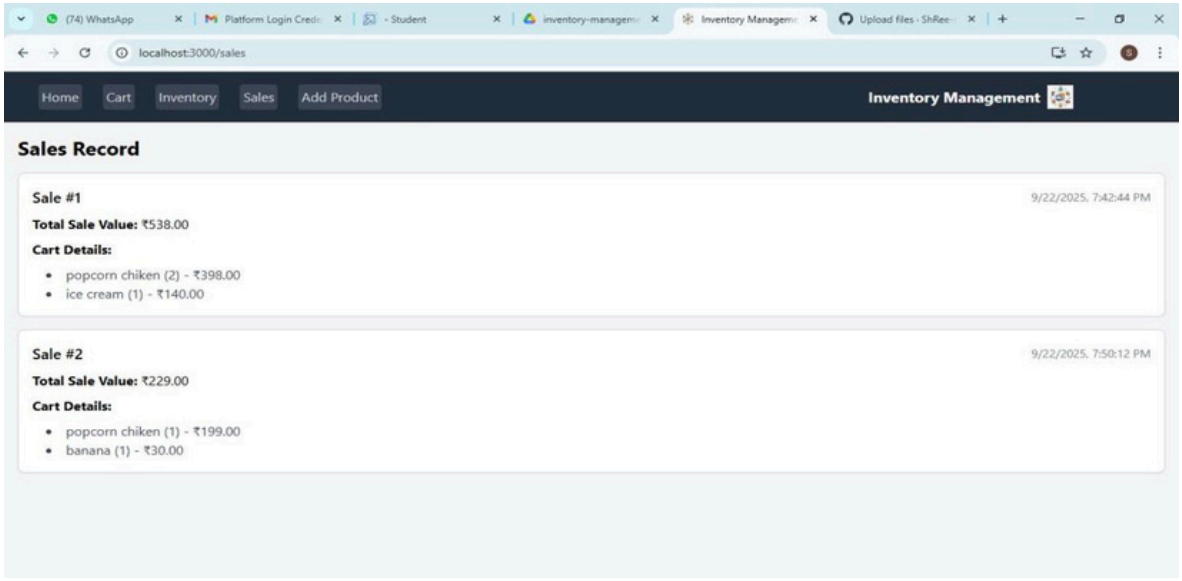
Screenshots or Demo

Screenshot


localhost:3000
Home
Cart
Inventory
Sales
Add Product
Inventory Management
Product Catalog
Search products...
Sample Item
Price: ₹ 0.00
Add to Cart
popcorn chiken
Price: ₹ 199.00
Add to Cart
banana
Price: ₹ 30.00
Add to Cart
ice cream
Price: ₹ 140.00
Add to Cart
iphone 17 pro
chocolate


localhost:3000
Price: ₹ 0.00
Add to Cart
Price: ₹ 199.00
Add to Cart
Price: ₹ 30.00
Add to Cart
Price: ₹ 140.00
Add to Cart
iphone 17 pro
Price: ₹ 140000.00
Add to Cart
chocolate
Price: ₹ 80.00
Add to Cart


localhost:3000/cart
Home
Cart
Inventory
Sales
Add Product
Inventory Management
Your Cart
No. of products in cart: 3 | Total Cart Value: 369.00
Checkout Cart
Clear Cart
popcorn chiken
Price: ₹ 199.00
1
Remove from Cart
banana
Price: ₹ 30.00
1
Remove from Cart
ice cream
Price: ₹ 140.00
1
Remove from Cart


localhost:3000/sales
Home
Cart
Inventory
Sales
Add Product
Inventory Management
Sales Record
Sale #1
9/22/2025, 7:42:44 PM
Total Sale Value: ₹538.00
Cart Details:
popcorn chiken (2) - ₹398.00
ice cream (1) - ₹140.00
Sale #2
9/22/2025, 7:50:12 PM
Total Sale Value: ₹229.00
Cart Details:
popcorn chiken (1) - ₹199.00
banana (1) - ₹30.00

Demo Video

To demonstrate the functionality of the Inventory Management System, a short demo video has been recorded. The video highlights the main features of the application, including adding new inventory items, updating existing stock, and deleting items when necessary. It also showcases the reorder alert system, the display of current stock levels, and the detailed view of each item with relevant information such as category, supplier, and unit price.

The video further illustrates the responsive design of the interface, showing how the application adapts seamlessly across different devices, including desktops, tablets, and smartphones...
[12:51 pm, 22/09/2025] Shree: Here's a professional adaptation of the Known Issues section for your Inventory Management System documentation:

link to the demo video
https://drive.google.com/file/d/1loGfjdBnRrOZ0FH0rEZ2jHK9sdUcvx6T/view?usp=drivesdk

Known Issues

Although the Inventory Management System functions smoothly in most scenarios, a few limitations and issues were identified during testing.

Currently, the system relies on manually entered data or a local database, which means that without proper setup, the inventory list may not load correctly. Some advanced features, such as automated stock forecasting, barcode scanning, and multi-user access, are not yet implemented. Additionally, the application has not been extensively tested across all devices and operating systems, so minor display or compatibility issues may exist on certain platforms.

Future Enhancements

While the current version of the Inventory Management System successfully achieves its primary goal of managing inventory efficiently, there are several features and improvements that can be added to enhance both functionality and user experience.

One immediate improvement would be the integration of automated stock forecasting and reorder suggestions based on historical usage. This would allow store managers to plan purchases more effectively and reduce the risk of stockouts. Similarly, implementing barcode scanning for item addition and updates would speed up data entry and improve accuracy.

Another important enhancement would be the inclusion of theme options, such as light mode and dark mode, to improve usability under different lighting conditions. The user interface could also benefit from modern design elements like smooth animations, hover effects, and improved layouts, making the application visually appealing while maintaining clarity. Accessibility improvements, such as keyboard shortcuts or screen reader compatibility, could make the system inclusive for all users.

In the long term, the project could be expanded to support cloud-based databases, enabling multiple store locations or users to access inventory data simultaneously. Deployment on platforms such as AWS or Firebase would make the system accessible from anywhere, eliminating reliance on local setups. Advanced features like user accounts with role-based access, search and filter options, and customized reporting dashboards could further transform the system into a complete, scalable inventory management solution.