Project Documentation

1. Introduction

Project Title: Storient – Store Inventory and Sales Manager

Team ID: NM2025TMID40298

Team Leader: JAYDEN MACARIUS AMBROSE & jaydenambrose@gmail.com

Team Members:

Veron Joseph Dcouto & veronjosephalt2006@gmail.com

Mariya Akash A & mariyaakash00@gmail.com

Deepak Vishal S & deepakvishal136@gmail.com

2. Project Overview

Purpose: The Storient application helps businesses manage and track inventory and sales using a simple and intuitive interface.

It allows for efficient product listing, categorization, stock updates, and sales tracking.

Features:

- Product listing interface

- Stock management (in stock, out of stock)

- Add and edit products

- Sales records management

- Responsive design with CSS

- Simple navigation across pages

3. Architecture

Structure:

- index.html: Main dashboard

- home.html: Homepage with overview

- add.html: Add new products

- edit.html: Edit product details

- stock.html: View and manage stock

- outofstock.html: Display items that are out of stock

- sales.html: Manage and view sales records

Assets:

- assets/css/: Stylesheets for UI

- assets/js/: JavaScript for interactivity

- assets/images/: Icons and images

4. Setup Instructions

Prerequisites:

- Web Browser (Chrome, Firefox, Edge)

- Code Editor (VS Code recommended)

Installation:

git clone https://github.com/jaydenambrosecyber/Storient.git

cd Storient

To run:

Open index.html in a browser or use Live Server in VS Code.

5. Folder Structure

Storient/

|-- index.html

|-- home.html

|-- add.html

|-- edit.html

|-- stock.html

|-- outofstock.html

|-- sales.html

|-- assets/

|-- css/

|-- js/

|-- images/

6. Running the Application

Open index.html directly in a browser or serve using a simple HTTP server.

7. Component Documentation

Since Storient uses **modular HTML pages + JS functions**, "components" are structured as **pages and reusable scripts**:

- **index.html** – Main dashboard linking to all sections.
- **home.html** – Overview of inventory and navigation.
- **add.html** – Form with JS validation to add products.
- **edit.html** – Interface to modify existing product data.
- **stock.html** – Displays current stock; pulls data from storage.
- **outofstock.html** – Filters and shows only products with zero stock.
- **sales.html** – Records and manages sales history.
- **assets/js/** –
  - main.js: Handles navigation and common scripts.
  - storage.js: Wrapper for reading/writing localStorage.
  - validation.js: Ensures form inputs are correct.

Together, these act like "components" in a React app but are kept separate through HTML and JavaScript files.

8. State Management

Unlike React apps, Storient manages state using **plain JavaScript**.

- **Local Storage**: The app stores product and sales data in the browser's localStorage, ensuring data persists even after refreshing the page.
- **In-Memory Objects/Arrays**: While running, products, stock, and sales are managed using JavaScript arrays and objects.
- **Data Update Flow**:
  - When a user **adds/edits a product**, JavaScript updates the in-memory data.
  - Data is then synchronized with localStorage to preserve it.
  - Pages like stock.html and outofstock.html dynamically fetch and display data from storage.

This makes the system simple but effective without requiring backend or external libraries.

9. User Interface

UI Features:

- Minimalist UI with CSS styling

- Responsive layout (works on desktop and mobile)

- Easy navigation across sections

10. Styling

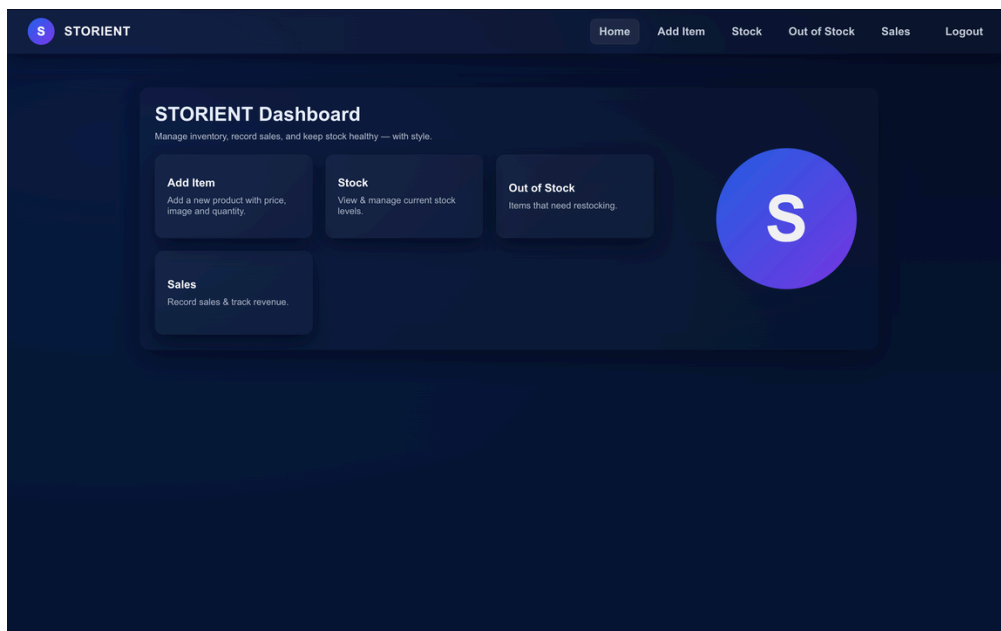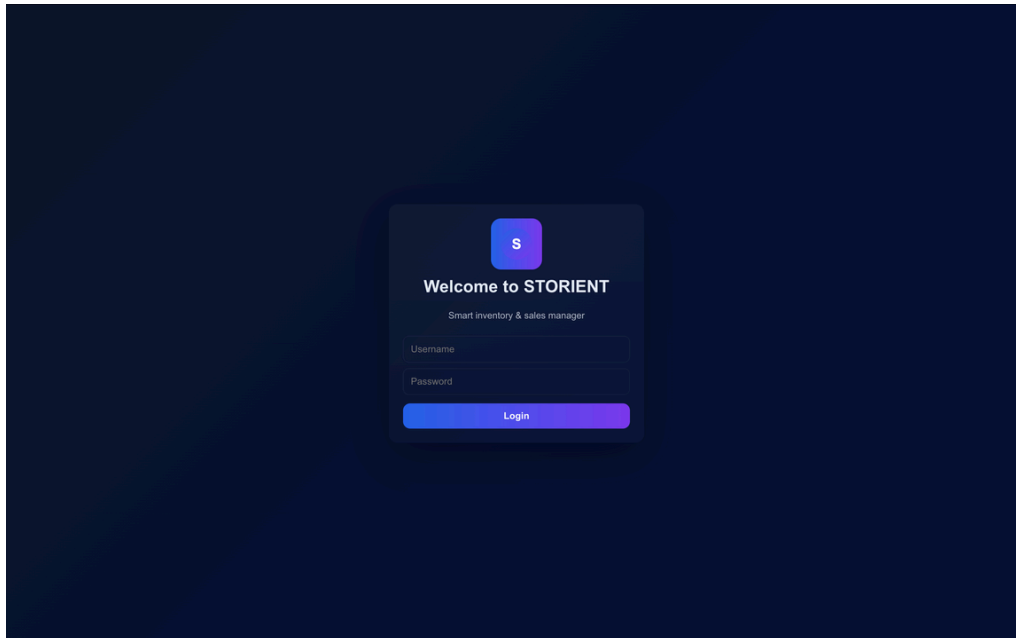CSS Framework:

- Plain CSS (no external frameworks used)

11. Testing

Testing Strategy:

- Manual testing across different browsers

- Validate form inputs during add/edit operations

12. Screenshots or Demo

Home Add Item Stock Out of Stock Sales Logout

Add New Item

Rice

60

90

data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAAD/2wCEA

Preview

Tip: you can paste an image URL or leave blank for default logo.

Add Item

Home Add Item Stock Out of Stock Sales Logout

Total Revenue
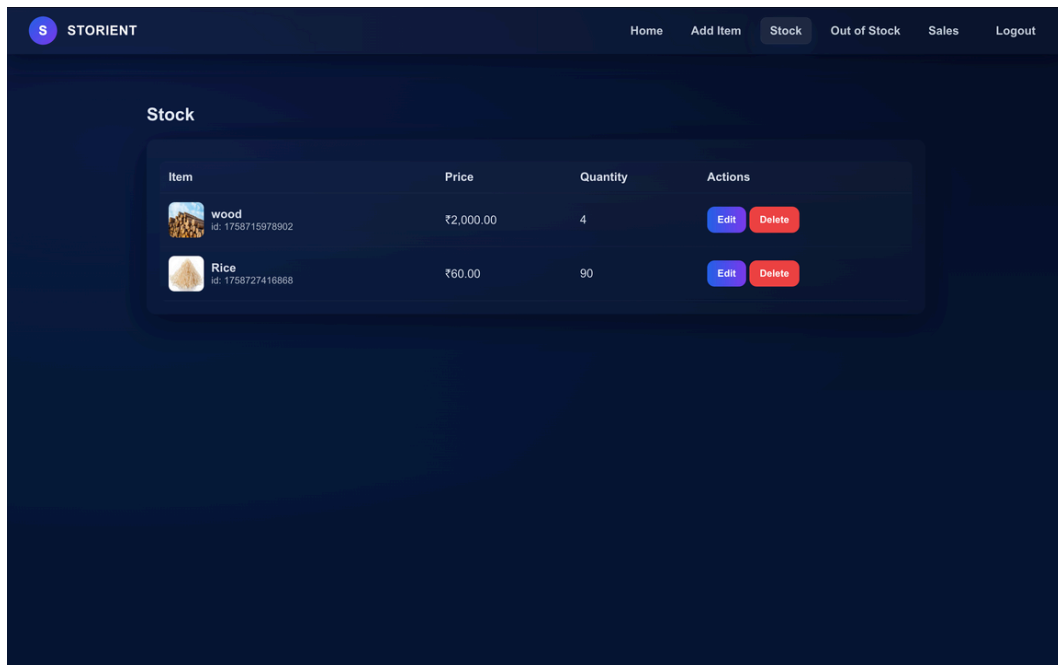**₹6,590.00**

Record Sale

wood - 4 in stock

1

Record Sale

Sales Log

Item	Amount
Books	₹250.00
wood	₹2,000.00
wood	₹4,000.00
wood	₹40.00
Rice	₹300.00

13. Known Issues

- No backend integration (data not persistent)

- Limited form validation

- No authentication system

14. Future Enhancements

- Add login/authentication flow

- Connect to backend/database for persistent data

- Add advanced filtering and sorting in stock and sales

- Deploy online for accessibility