

# Warehouse Stock Management System Requirements Specification

Version 1.0

May 2, 2024

## Document Signoff-Group 4

Name	Date	Signature	Role
Sri Sravani Gullapalli	05/02/2024	G Sri Sravani	Quality Assurance Specialist
Sri Laxmi Prasanna Joginipelli	05/02/2024	SRILU RAO	Software Developer
Mythri Gudibanda	05/02/2024	MYTHRI GUDIBANDA	Automation Specialist
Gayathri Kaparathi	05/02/2024	GAYATHRI KAPARTHI	Software Developer
vaibhavi rachamalla	05/02/2024	Vaibhavi .R	Database and Software Testing Specialist

## Table of Contents

<b>1. EXECUTIVE SUMMARY.....</b>	<b>3</b>
1.1 PROJECT OVERVIEW.....	3
1.2 PURPOSE AND SCOPE OF THIS SPECIFICATION.....	5
<b>2. PRODUCT/SERVICE DESCRIPTION.....</b>	<b>6</b>
2.1 PRODUCT CONTEXT.....	6
2.2 USER CHARACTERISTICS.....	6
2.3 ASSUMPTIONS.....	6
2.4 CONSTRAINTS.....	7
2.5 DEPENDENCIES.....	7
<b>3. REQUIREMENTS.....</b>	<b>9</b>
3.1 FUNCTIONAL REQUIREMENTS.....	9
3.2 USER INTERFACE REQUIREMENTS.....	12
3.3 USABILITY.....	13
3.4 PERFORMANCE.....	14
3.4.1 Capacity.....	14
3.4.2 Availability.....	15
3.4.3 Latency.....	15
3.5 MANAGEABILITY/MAINTAINABILITY.....	16
3.5.1 Monitoring.....	16
3.5.2 Maintenance.....	16
3.5.3 Operations.....	17
3.6 SYSTEM INTERFACE/INTEGRATION.....	17
3.6.1 Network and Hardware Interfaces.....	18
3.6.2 Systems Interfaces.....	18
3.7 SECURITY.....	20
3.7.1 Protection.....	20
3.7.2 Authorization and Authentication.....	21
3.8 DATA MANAGEMENT.....	21
3.9 STANDARDS COMPLIANCE.....	22
3.10 PORTABILITY.....	22
<b>4. USER SCENARIOS/USE CASES.....</b>	<b>23</b>
<b>5. DELETED OR DEFERRED REQUIREMENTS.....</b>	<b>25</b>
<b>6. REQUIREMENTS CONFIRMATION/STAKEHOLDER SIGN-OFF.....</b>	<b>25</b>
<b>7. EXECUTION.....</b>	<b>27</b>
7.1 CLONING, CONFIGURATION, AND RUNNING.....	27
7.2 USER-MANUAL.....	28
7.3 MODEL.....	35
<b>APPENDIX.....</b>	<b>39</b>
APPENDIX A. ....	36
APPENDIX B. DEFINITIONS, ACRONYMS, AND ABBREVIATIONS.....	36
APPENDIX C. REFERENCES.....	37
APPENDIX D. REQUIREMENTS TRACEABILITY MATRIX.....	37
APPENDIX E. ORGANIZING THE REQUIREMENTS.....	38

# **1.Executive Summary**

## **1. Project Overview:**

An Warehouse Stock Management System (WSMS) is a software solution for managing daily inventory and warehouse operations. It provides total inventory visibility and allows for easy inventory organization. Using an WSMS allows us visibility into warehouse processes from product entry to exit. We have 24/7 access to inventory levels and locations, as well as item history records.

### **1.1 Introduction:**

As inventory and warehouse management become more challenging, businesses need effective ways to track products, manage orders, and maintain ideal stock levels. The Warehouse Stock Management System (WSMS) integrates elements to enhance user experience, speed up procedures, and inform decision-making. In today's corporate world, proper inventory management is crucial for success. The WSMS offers a comprehensive platform to support this. Consumers may quickly create accounts, log in, and access a user-friendly dashboard. The dashboard allows users to manage inventory by viewing, adding, and removing products. The WSMS's ability to process orders is one of its most notable characteristics. Users can submit product orders and receive real-time availability checks to prevent processing orders for out-of-stock items. Providing correct and up-to-date information improves order fulfillment and customer happiness. The WSMS includes a specific customer page to enhance the customer experience. This website allows customers to order products depending on availability, with real-time updates showing "In Stock" or "Out of Stock." Ensuring clients make informed selections when placing orders promotes transparency and confidence. Administrators benefit from advanced capabilities that help manage product lists more effectively. This includes restocking the inventory with fresh products. This involves restocking inventory with new items and removing outdated ones. The technology provides order history insights, allowing organizations to monitor and analyze sales data for tactical decision-making. The WSMS is a solid alternative for streamlining warehouse and inventory management procedures. The WSMS enhances customer and administrator experiences, operational efficiency, and mistake reduction by combining user-friendly features and resolving common pain spots.

## **1.2 Problem Statement:**

Manual procedures and disconnected technologies present substantial issues, including inaccurate inventory tracking, delayed order fulfillment, and limited visibility into overall business performance. The executive team recognizes the critical need for a comprehensive warehouse management solution that can streamline operations, increase productivity, and help the firm grow. We have identified the project's key stakeholders and their specific interests:

- Warehouse managers: Managers must have real-time visibility into inventory quantities, storage locations, and general warehouse activities. This will enable them to make educated decisions to optimize operations and proactively resolve any issues that arise.
- Warehouse workers: Employees require an intuitive system that can assist them through the numerous workflows, from receiving and putting away to order selection, packaging, and shipping. This will enable them to fulfill orders more correctly and effectively while keeping precise inventory records.
- Operations executives: Operations executives demand strong reporting and business intelligence capabilities to monitor key performance measures, identify areas for improvement, and make data-driven choices to improve overall company operations.
- Logistics Providers: Our client's logistics partners require seamless interaction with the warehouse management system in order to obtain up-to-date order and shipping data. This will allow them to make more timely and precise deliveries, enhancing the client experience.
- Customers: Customers demand accurate order fulfillment, quick shipment, and transparent order tracking. Meeting these needs will be critical to sustaining a positive brand reputation and consumer loyalty.

## **1.3 Problem Solution:**

To overcome these problems and meet stakeholder expectations, we propose creating a centralized, automated, and integrated warehouse stock management system.

- Inventory Management: The system will give real-time tracking of inventory levels, locations, and statuses, allowing the firm to maintain appropriate stock levels and reduce stock-outs.
- Warehouse Operations: We will increase overall operational efficiency by streamlining essential warehouse activities such as receiving, put-away, order picking, packing, and shipping.

### ***[Group4-Warehouse Stock Management System] Requirements Specification***

- **Integration:** The solution will work smoothly with the client's existing enterprise resource planning (ERP), order management, and logistics systems, enabling end- to-end visibility and coordination throughout the supply chain.
- **Reporting and Analytics:** Comprehensive reporting and business intelligence capabilities will be built to give operations leaders with data-driven insights and aid in strategic decision-making.
- **User Interface:** Simple web-based and mobile interfaces will be created enabling warehouse employees, managers, and customers to access information and complete their jobs.

By deploying a stock management system, our client will be able to handle their present operational difficulties while also positioning themselves for future development and success in the highly competitive online retail sector.

## **1.2 Purpose and Scope of this Specification**

### **1.2.1 PROJECT SCOPE**

This project proposes a Warehouse Management System (WSMS) to address issues in product inventory tracking and management for organizations. The WSMS focuses on optimizing operational effectiveness and facilitating smooth order processing. Users can create accounts, log in, and securely manage them within the system. The dashboard allows users to effortlessly add, remove, and see products. Product details, such as name, price, and availability, are preserved and shown. Users can place product orders, and the system verifies availability before processing. The WSMS's customer page allows customers to place orders based on product availability, enhancing the user experience. The system dynamically updates product availability status to either "In Stock" or "Out of Stock." This tool helps clients make better selections while placing orders. Administrators can manage the product list by adding or removing items. It provides order history information, allowing organizations to monitor and analyze their sales data. The WSMS provides a comprehensive solution to streamline inventory and warehouse management procedures, improving productivity, reducing errors, and increasing customer and administrator satisfaction.

### **1.2.2 OBJECTIVES**

The primary goals of the WSMS project are to improve inventory management efficiency by providing firms with a comprehensive system for tracking products in real-time, from warehouse entry to leave. The initiative also intends to expedite order processing with features such as real-time product availability checks, ultimately

## ***[Group4-Warehouse Stock Management System] Requirements Specification***

enhancing order fulfillment and customer happiness. The project aims to develop a seamless platform for consumers to manage inventories, examine items, and place orders, with a focus on a user-friendly interface. Furthermore, the addition of transparent customer pages improves the entire customer experience by allowing customers to make informed selections based on real-time product availability. For administrators, advanced capabilities enable effective product list administration, including item replenishment and removal, as well as detailed purchase history for strategic decision-making. The primary goal is to increase operational efficiency by eliminating errors, shortening processing times, and enhancing workflow. In response to the problem description, the project focuses on reducing errors in inventory and warehouse management procedures, assuring correct product availability checks, and transparently communicating stock status to customers. The WSMS project seeks to improve customer and administrator satisfaction by integrating user-friendly features and tackling important inventory and warehouse management issues.

## **2. Product/Service Description:**

### **2.1. Product Context:**

The Warehouse Stock Management System is a critical component in the logistics and supply chain management process. It serves as the central hub for tracking, managing, and optimizing inventory within a warehouse or distribution center. While it operates as a standalone system, it often interfaces with various related systems such as Enterprise Resource Planning (ERP) systems, inventory management software, transportation management systems, and customer relationship management (CRM) systems. The system may also interact with external entities such as suppliers, manufacturers, and shipping carriers.

## **2.2 User Characteristics:**

### **1. Warehouse Managers/Supervisors:**

- Experience: Extensive experience in warehouse operations and inventory management.
- Technical Expertise: Moderate to high technical expertise in using computer systems and software.
- Other Characteristics: Familiarity with warehouse processes and a focus on optimizing efficiency and accuracy.

### **2. Warehouse Staff:**

- Experience: Varied, ranging from entry-level to experienced warehouse personnel.
- Technical Expertise: Basic to moderate technical skills, with training provided as needed.
- Other Characteristics: Work in a fast-paced environment, requiring user-friendly interfaces and clear instructions.

### **3. Administrators/IT Personnel:**

- Experience: Technical background with experience in system administration and IT support.
- Technical Expertise: High level of technical expertise in system configuration, troubleshooting, and maintenance.
- Other Characteristics: Responsible for system setup, configuration, and ongoing support.

## **2.3 Assumptions:**

- Availability of necessary equipment such as barcode scanners, computers, and network infrastructure.
- Users have basic training in using computer systems and are receptive to training on the Warehouse Stock Management System.
- Adequate internet connectivity for accessing cloud-based systems if applicable.

## **2.4 Constraints:**

- **Parallel Operation with Old System:** The Warehouse Stock Management System may need to operate in parallel with an existing legacy system during the transition period.
- **Audit Functions:** The system must have robust audit trail functionality to track stock movements, user activities, and system changes.
- **Access, Management, and Security:** Strict access controls and security measures are required to protect sensitive inventory data and prevent unauthorized access.
- **Criticality of the Application:** The system is critical for efficient warehouse operations, and downtime must be minimized to avoid disruptions in supply chain activities.
- **System Resource Constraints:** The system design must consider limitations on disk space, memory, and processing power to ensure optimal performance.
- **Other Design Constraints:** Compliance with industry standards, compatibility with specific hardware/software platforms, and adherence to regulatory requirements may impose additional design constraints.

## **2.5 Dependencies:**

A Warehouse Stock Management System's dependencies might vary depending on the system's individual implementation and requirements. However, there are several frequent dependencies that may need to be considered.

### **1. Hardware Infrastructure:**

- Includes servers, computers, barcode scanners, printers, and networking equipment.
- Compatibility with existing hardware infrastructure, as well as any necessary updates or extra equipment.

### **2. Software Dependencies:**

- Compatibility with server and client operating systems.
- Ability to store and retrieve data using database management systems (e.g., MySQL, SQL Server, Oracle).
- Integration with third-party software systems, such as ERP, accounting, and e-commerce platforms.

### **3. Integration Interfaces:**

- APIs or web services for connecting to external systems and services.
- Compliance with industry-standard protocols for data sharing (e.g., RESTful APIs, FTP etc..)

### **4. Data Sources:**

## ***[Group4-Warehouse Stock Management System] Requirements Specification***

- Accurate and current data sources, including product catalogs, inventory databases, and supplier information.
- Data quality assurance techniques guarantee that data inputs are consistent and reliable.

### **5. User Training and Adoption:**

- Provide training to end-users to acquaint them with system functions and procedures.
- Change management techniques to ensure easy acceptance of the new system while minimizing user opposition.

### **6. Regulatory and Compliance Requirements:**

- Follow industry norms and standards for inventory management, data privacy, and security.
- Compliance with specialized frameworks (e.g., FDA rules for pharmaceutical warehouses, ISO standards for quality management).

### **7. Infrastructure and Network Connectivity:**

- Enough network infrastructure to provide data transfer and communication among system components.
- Redundancy measures to ensure system availability and fault tolerance in the event of network outages.

### **8. Security Measures:**

- Use access controls, encryption, and other security measures to safeguard sensitive inventory data.
- Adhere to organizational security rules and procedures for data handling and access management.

### **9. Scalability and Performance Requirements:**

- Scalable design to handle increasing data volume and user demand over time.
- Performance optimization methods to provide rapid reaction times and efficient system functioning during high load periods.

**10. Dependence on suppliers for software updates, fixes, and technical assistance:** Vendor management strategies for relationship maintenance and fast issue resolution.

## **2.5.1 Dependency Architecture:**

### **(i) Database ER Diagram**

A dependency Architecture for a warehouse stock management system has four layers: application-specific, application-generic, middleware, and system-software.

**Application-particular Layer:** This layer includes software components specific to the warehouse stock management system. It includes capabilities like as inventory management, order processing, delivery tracking, and reporting.

**Application-Generic Layer:** This layer contains reusable components and libraries that are not specific to the warehouse management system but are used to build and support its functionality. It may include modules that handle user authentication, database access, logging, and error handling.

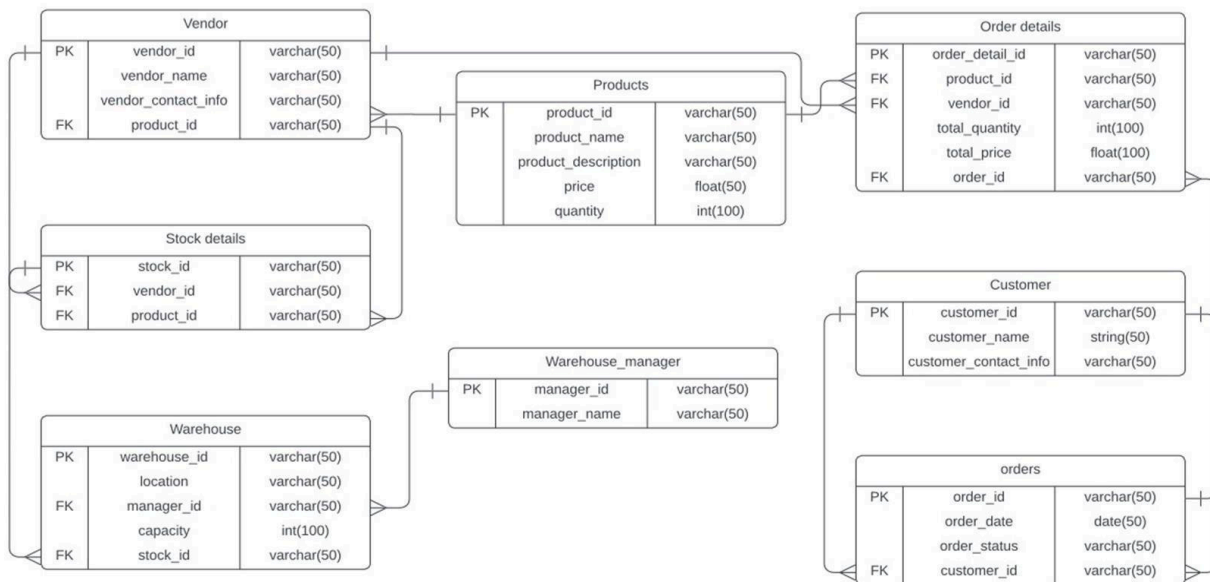
**Middleware Layer:** This layer connects the application-specific and system-software layers. Middleware components like as message brokers, API gateways, and integration frameworks facilitate communication between different parts of the system.

**System-Software Layer:** This layer includes the underlying software and



## **[Group4-Warehouse Stock Management System] Requirements Specification**

infrastructure that allows the warehouse stock management system to operate. It contains the operating system, database management system, web server, and other system-level software components required by the program to function.



## **3.Requirements**

### **3.1 Functional Requirements:**

#### **1. BR\_LR\_001: Inventory Management:**

- The system shall allow users to add new products to the inventory.
- The system shall allow users to update existing product information such as name, description, and price.
- The system shall allow users to delete products from the inventory.

#### **2. BR\_LR\_002: Stock Tracking**

- The system shall track stock levels for each product in real-time.
- The system shall update stock levels automatically upon receiving new inventory or fulfilling orders.
- The system shall provide alerts when stock levels fall below a defined threshold.

#### **3. BR\_LR\_003: Order Management**

- The system shall allow users to create new purchase orders for replenishing stock.
- The system shall allow users to update and modify existing purchase orders.
- The system shall generate order confirmations and send them to suppliers.

#### **4. BR\_LR\_004: Receiving and Warehousing**

### ***[Group4-Warehouse Stock Management System] Requirements Specification***

- The system shall facilitate the receiving process by recording incoming shipments and updating inventory accordingly.
- The system shall support barcode scanning for efficient receiving and warehousing operations.
- The system shall allocate storage locations for incoming inventory based on predefined rules.

#### **5. BR\_LR\_005: Picking and Packing**

- The system shall generate pick lists for fulfilling customer orders.
- The system shall allow users to confirm picked items and pack them for shipping.
- The system shall update stock levels and order statuses in real-time after picking and packing.

#### **6. BR\_LR\_006: Shipping and Delivery**

- The system shall generate shipping labels and packing slips for outgoing shipments.
- The system shall track the status of shipments in transit and provide tracking information to customers.
- The system shall facilitate returns and exchanges by providing return labels and processing returned items.

#### **7. BR\_LR\_007: Reporting and Analysis:**

- The system shall provide standard reports such as inventory status, sales reports, and order history.
- The system shall allow users to create custom reports based on specific criteria.
- The system shall support data analysis functionalities for identifying trends, forecasting demand, and optimizing inventory levels.

#### **8. BR\_LR\_008: User Management and Security:**

- The system shall support multiple user roles with different levels of access (e.g., administrator, warehouse manager, warehouse staff).
- The system shall enforce authentication mechanisms to ensure only authorized users can access the system.
- The system shall maintain an audit trail of user activities for accountability and security purposes.

#### **9. BR\_LR\_009: Integration with External Systems:**

- The system shall integrate with existing ERP systems for seamless data exchange.
- The system shall support APIs for integration with third-party logistics providers, payment gateways, and other external systems.
- The system shall ensure data consistency and integrity across integrated systems.

**[Group4-Warehouse Stock Management System] Requirements Specification**

Req#	Requirement	Comments	Priority
BR_LR_001	<b>Inventory Management</b> -Add new products to the inventory -Update existing product information -Delete products from the inventory	Business Process = “ product Name,product description,product prices”	High
BR_LR_002	<b>Stock Tracking</b> -Track stock levels in real-time. - Update stock levels automatically -Provide alerts for low stock levels	Business Process=”Upon receiving new inventory or fulfilling orders”	High
BR_LR_003	<b>Order Management</b> - The system shall allow users to create new purchase orders for replenishing stock. - The system shall allow users to update and modify existing purchase orders. - The system shall generate order confirmations and send them to suppliers.	Business Process = “replenishing stock,Details Modification”	High
BR_LR_004	<b>Receiving and Warehousing</b> -Record incoming shipments,Update inventory,	Business Process = “update inventory,based on predefined rules”	medium
BR_LR_005	<b>Picking and Packing</b> -Confirm pick list,Update Stock levels	Business Process = “Real Time Updates,fulfilling customer orders.	Medium
BR_LR_006	<b>shipping and delivery</b> -Track Shipment status, -Facilitate returns and exchanges	Business Process = “Provide tracking information to customers; Provide return labels, process returns”.	High
BR_LR_007	<b>Reporting and Analysis</b> - Provide standard reports -Allow users to create custom reports	Business Process = “Inventory status, sales reports, order history ”	High

### **[Group4-Warehouse Stock Management System] Requirements Specification**

Req#	Requirement	Comments	Priority
BR_LR_008	User Management and Security - Support multiple user roles - Enforce authentication mechanisms	Business Process = “ Administrator, warehouse manager, warehouse staff ”,”Ensure only authorized access” .	High
BR_LR_009	Integration with External Systems - Integrate with existing ERP systems - Support APIs for third-party integration - Ensure data consistency across systems	Business Process = “ Seamless data exchange,Logistics providers” .	Medium

## **3.2 User Interface Requirements:**

### **1. Navigation and structure:**

- The system contains an easy-to-use interface with clear menus and a well-organized structure.
- Navigation menus must be properly organized, allowing users to quickly access various modules and capabilities.
- Consistent layout and design components must be maintained across all displays to ensure familiarity and usability.

### **2. Dashboard:** - The system will offer a customized dashboard that displays critical metrics and performance indicators.

- Dashboard widgets will provide real-time data such as inventory levels, pending orders, and warehouse usage.

### **3. Product Management:** - The product management interface allows users to add, update, and delete goods from inventory.

- Product information must be displayed in a clear and structured way, including the product name, description, SKU, and price.

### **4. Order Management:** - The order management interface should help create, process, and fulfill client orders.

- Users will be able to examine order information, amend order statuses, and print packing slips or shipping labels.

### **5. Inventory Tracking:** - The inventory tracking interface should enable real-time view of stock levels and movements.

- Users will be able to search for specific goods, monitor stock numbers, and track inventory transactions (such as receipts, shipments, and adjustments).

### **6. Reports and Analytics:**

- Users will get access to conventional reports as well as customizable analytics dashboards through the reporting interface.

## **[Group4-Warehouse Stock Management System] Requirements Specification**

- Reports must cover inventory levels, sales performance, order history, and other pertinent indicators.
- Users must be able to filter and export report data in a variety of formats (e.g., PDF, CSV) for further study.

### **7. User Management:**

- Administrators may manage user accounts, responsibilities, and permissions using the user management interface.
- Administrators must be able to create, amend, or delete user accounts, assign roles, and set access levels depending on job duties.

### **8. Alerts and Notifications:**

- The system will provide alerts and notifications to users on crucial events including low stock levels, pending orders, and system issues.
- Notifications will be sent by email, in-app messaging, or SMS, based on user choices.

### **9. Error Handling and Help Resources:**

- The system should give meaningful error messages to help users resolve common difficulties or errors.

## **3.3 Usability**

1. Learnability: The user interface shall be intuitive and easy to navigate, minimizing the learning curve for new users. Onboarding tutorials or walkthroughs shall be available to guide users through the system's functionalities. The system shall provide comprehensive user documentation accessible within the interface, covering all aspects of system usage. Help resources shall be context-sensitive, offering relevant guidance based on the user's current task or location within the interface. User documentation shall include step-by-step instructions and screenshots to facilitate learning and task completion.
2. Efficiency: The system shall streamline common tasks and workflows to maximize user productivity. Frequently used features shall be easily accessible, with minimal clicks or steps required to perform actions. Keyboard shortcuts and other efficiency-enhancing features shall be available to expedite user interactions with the system.
3. Error Handling and Recovery: The system shall provide clear and informative error messages when users encounter errors or issues. Error messages shall be presented in plain language, guiding users on how to resolve the problem or recover from errors. In cases of user input errors, the system shall offer suggestions or prompts to help users correct their mistakes.
4. User Feedback and Communication: The system shall incorporate user feedback mechanisms, such as surveys or suggestion forms, to gather input from users. Feedback received from users shall be reviewed and considered for system improvements and enhancements. The system shall provide avenues for users to communicate with system administrators or support staff for assistance or inquiries.
5. Customization and Personalization: The system shall allow users to customize their interface preferences to suit their individual needs and preferences. Customization options may include configurable dashboards, color schemes, font sizes, and layout settings. Personalization features shall enable users to tailor the interface to their specific roles and workflows, improving usability and efficiency.
6. Accessibility: The system shall comply with accessibility standards (e.g., WCAG) to ensure usability for users with disabilities. Accessibility features such as screen reader compatibility, keyboard navigation, and high-contrast mode shall be available to accommodate diverse user needs.

7. Consistency and Familiarity: The system shall maintain consistency in design, layout, and terminology across all screens and modules. Familiar interface elements and interaction patterns shall be utilized to create a cohesive user experience and reduce cognitive load for users.

### **3.4 Performance**

Specify static and dynamic numerical requirements placed on the system or on human interaction with the system:

1. Static Numerical Requirements: - The system must support at least 50 terminals concurrently accessing it.

- It should be able to handle up to 100 concurrent users during peak hours.

- The system must be capable of processing at least 100,000 inventory items.

2. Dynamic Numerical Requirements:

- Under normal workload conditions, 95% of transactions should be handled in less than one second.

- The system must be able to process up to 1,000 transactions per minute during high workload situations.

3. Data Processing Capacity:

- The system should be capable of processing at least 10,000 transactions per day under typical operational conditions.

- During peak hours, the system must be capable of handling up to 500 transactions per minute.

#### **3.4.1 Capacity:**

1) Data Capacity: The system must accommodate a minimum of 100 simultaneous users. During peak hours, the maximum simultaneous user load must not exceed 500 people.

2) Per-User Memory Requirements: Each user session will have a memory footprint of no more than 100 MB. Memory use must be minimized to ensure that resources are utilized efficiently for scalability.

3. Expected Application Throughput: The system must process at least 500 transactions per hour during non-peak hours. During peak hours, the system must support a minimum of 1,000 transactions per hour.

4. Data Handling Capacity: The system should be able to manage at least 1 million inventory items. - Each inventory item must have linked data elements such as the product name, description, amount, and location.

5. Storage Capacity: The system must have enough storage space to retain transaction logs, audit trails, and historical data. The minimum storage capacity will be 1 TB, with scaling options for future development.

6. Network Bandwidth: A minimum of 100 Mbps is required for seamless data transmission between client apps and the server. The network bandwidth must be scalable to handle increasing data traffic during peak hours or system expansions.

## **[Group4-Warehouse Stock Management System] Requirements Specification**

7. Database Scalability: The database system must offer horizontal scalability to handle growing data volumes and user loads.

### **3.4.2 Availability**

1. Hours of Operation: The system is operating 24/7, except for regular maintenance periods. Maintenance must be undertaken during off-peak hours to minimize interruption to users.
2. Required Level of Availability: The system must maintain a target uptime of 99.9% over a 30-day period, excluding scheduled maintenance. Any downtime over 0.1% of total operational time will be considered non-compliant with availability criteria.
3. Geographic Coverage: The system should be available from all warehouse sites in various geographies. Users must be able to use the system from any place with internet connection, regardless of geographic limitations.
4. Impact of System Downtime on Users and Business Operations: - System downtime should have minimal impact on users' ability to perform critical tasks like order processing, inventory management, and shipping. Backup and recovery mechanisms should be in place to prevent data or transaction loss.
5. Impact of Scheduled and Unscheduled Maintenance: Users will be notified of scheduled maintenance via system alerts or email. Scheduled maintenance must not last more than two hours each week and must be conducted during off-peak hours to minimize disturbance.
6. Reliability: The system should have a mean time between failures (MTBF) of at least 1,000 hours. The maximum allowable number of failures per hour must not exceed 0.1, ensuring a reliable and stable operating environment for users.

### **3.4.3 Latency**

1. Service Request Latency: Maximum allowed response time for a service request is 500 milliseconds on average.
  - Service requests include tasks like querying inventories, updating product information, processing orders, and creating reports.
  - Fast response times must be prioritized in the system to guarantee that user interactions and transactions are handled on time.
2. User Interface Responsiveness: - The user interface should respond instantly to user input, including actions and navigation elements.
  - UI interactions, such as button clicks, menu selections, and data entry, must have a latency of less than 200 milliseconds to offer a consistent user experience.
3. Batch Processing Latency: - Inventory reconciliations and data imports should be finished within 5 minutes.
  - Large-scale data processing operations must be optimized for low latency and optimal use of system resources.
4. Real-time Data Updates

## **[Group4-Warehouse Stock Management System] Requirements Specification**

- Real-time data updates, such as inventory status changes and order updates, must be sent to user interfaces with a latency of less than one second.
- Users must get timely notifications and alerts on crucial events or updates affecting their workflow.

5.API Response Time: - The system's API endpoints should reply to queries within 300 milliseconds on average.

- API response times must be constant across multiple endpoints and under varied traffic situations to enable integrations with external systems and applications.

6.Data Retrieval Latency: - Data retrieval processes, such obtaining inventory records or creating reports, should have an average latency of less than one second.

- Database queries and data fetching operations must be improved for performance in order to reduce latency and ensure efficient data access.

7.Remote Access delay: - Users accessing the system remotely over the internet should experience delay of less than 500 milliseconds for page loading and data retrieval processes.

- Remote access latency will be reduced by optimizing network infrastructure and content delivery technologies.

### **3.5 Manageability/Maintainability**

#### **3.5.1 Monitoring Requirements:**

- The system shall include built-in health monitoring features to continuously monitor the performance and availability of critical components.
- Monitoring shall include real-time tracking of system metrics such as CPU utilization, memory usage, disk space, and network bandwidth.
- Automated alerts shall be generated for abnormal conditions, failures, or performance degradation, notifying administrators for prompt intervention.
- Error detection mechanisms shall be implemented to identify and log errors, exceptions, and anomalies in system behavior.
  - Comprehensive logging capabilities shall be provided to record system events, user activities, and error messages for troubleshooting and auditing purposes.
- Log files shall be stored securely and rotated regularly to prevent file bloating and ensure efficient log management.

#### **3.5.2 Maintenance Requirements:**

- The system architecture shall be designed with modularity and componentization to facilitate ease of maintenance and future enhancements.
- Codebase shall be well-documented and organized, adhering to coding standards and best practices to enhance code readability and maintainability.
- Interfaces between system components shall be well-defined and standardized to promote interoperability and ease of integration with external systems.
- Regular code reviews and peer inspections shall be conducted to identify and address code quality issues, potential bugs, and performance bottlenecks.
  - System documentation, including installation guides, user manuals, and technical specifications, shall be comprehensive and up-to-date to assist system administrators and end-users in understanding system functionalities and configurations.



### **3.5.3 Operations Requirements:**

- The system shall support both interactive operations, where users directly interact with the system interface, and unattended operations, where automated processes run without user intervention.
- Data processing support functions, such as data import/export, batch processing, and scheduled tasks, shall be provided to streamline data management operations.
- Backup and recovery mechanisms shall be implemented to ensure data integrity and minimize data loss in the event of system failures or disasters.
- Regular backups of critical system data shall be performed according to predefined schedules, with backups stored securely in offsite locations for disaster recovery purposes.
  - Safety considerations shall be integrated into system operations, including access control measures, user authentication mechanisms, and data encryption to protect sensitive information from unauthorized access and misuse.
- Disaster recovery and business resumption plans shall be developed and documented to outline procedures for restoring system functionality and operations in the event of major disruptions or disasters, ensuring business continuity and minimizing downtime.

## **3.6 System Interface/Integration**

### **1. Database Integration:**

Description: Integration with the database server for storing and retrieving inventory data.

Interface: JDBC (Java Database Connectivity)

Message Format: SQL queries and commands

Content: CRUD operations for managing inventory records, querying product information, and updating stock levels.

Configuration Characteristics: IP address, port number, database credentials

Supported Devices: PostgreSQL, MySQL, Oracle

### **2. Printer Integration:**

Description: Integration with printers for generating labels, reports, and documentation.

Interface: Printer Command Language (e.g., ZPL, EPL, PCL)

Message Format: Printable document format (e.g., PDF, ZPL commands)

Content: Product labels, shipping manifests, pick lists

Configuration Characteristics: Printer drivers installation, network settings

Supported Devices: Label printers, laser printers, thermal printers

### **3. Network Interface:**

Description: Interface with the local area network (LAN) or wide area network (WAN) for communication with external systems and services.

Interface: TCP/IP, UDP, HTTP, HTTPS

Message Format: TCP/IP packets, HTTP requests

Content: API requests, data exchange with external systems

Configuration Characteristics: IP address, subnet mask, gateway, DNS settings

Supported Protocols: Ethernet, Wi-Fi

### **[Group4-Warehouse Stock Management System] Requirements Specification**

These interfaces facilitate seamless communication and integration between the warehouse stock management system and various hardware devices, databases, and network components, enabling efficient inventory management operations within the warehouse environment.

#### **3.6.1 Network and Hardware Interfaces**

Network and Hardware Interfaces for Warehouse Stock Management System are stated as below:

1. Database Interface:

- Description: Interface between the warehouse stock management system and the database server for data storage and retrieval.
- Characteristics:
  - Number of Ports: 1
  - Protocol: JDBC (Java Database Connectivity)
  - Supported Databases: PostgreSQL, MySQL, Oracle
  - Configuration: IP address, port number, database credentials
  - Instruction Sets: SQL (Structured Query Language)

2. Printer Interface:

- Description: Interface between the warehouse stock management system and printers for generating labels, reports, and documentation.
- Characteristics:
  - Number of Ports: 1 (USB, Ethernet, or Wi-Fi)
  - Supported Devices: Label printers, laser printers, thermal printers
  - Protocol: Printer Command Language (e.g., ZPL, EPL, PCL)
  - Configuration: Printer drivers installed on the system, network settings for networked printers
  - Instruction Sets: Printer-specific command sets for formatting and printing documents

3. Network Interface:

Description: Interface between the warehouse stock management system and the local area network (LAN) or wide area network (WAN) for communication with external systems and services.

- Characteristics:
  - Number of Ports: 1 (Ethernet)
  - Supported Protocols: TCP/IP, UDP, HTTP, HTTPS
  - Configuration: IP address, subnet mask, gateway, DNS server settings
  - Security: Firewall rules, network segmentation, encryption (e.g., TLS)
  - Instruction Sets: Network protocols for data transmission and reception

These interfaces facilitate communication and interaction between the warehouse stock management system and various hardware devices, databases, and network components, enabling efficient inventory management and operations within the warehouse environment.

#### **3.6.2 Systems Interfaces**

Interface Name: Warehouse-to-Supplier Interface

Description: This interface facilitates the exchange of inventory replenishment orders between the warehouse stock management system and external suppliers.

## ***[Group4-Warehouse Stock Management System] Requirements Specification***

Interface Protocol: Email Attachment

Interface Endpoint: supplier@warehouse.com

Interface Frequency: Daily

Interface Requirements:

- The warehouse stock management system shall generate a purchase order file in CSV format containing order details.
- The purchase order file shall be sent as an email attachment to the designated email address of the supplier.
- The email subject line shall include the date and time of the order submission.
- The purchase order file shall contain the following fields for each item:
  1. Product SKU
  2. Quantity Ordered
  3. Unit Price
  4. Total Price
- The supplier shall process the purchase order and confirm receipt via email within 24 hours.

Example Purchase Order File Structure:

```
"SKU","Quantity Ordered","Unit Price","Total Price"  
"SKU123","100","10.00","1000.00"  
"SKU456","50","15.00","750.00"
```

Data Transfer Process:

1. The warehouse stock management system generates the purchase order file based on inventory levels and reorder points.
2. The purchase order file is attached to an email generated by the system.
3. The email is sent to the designated email address of the supplier (supplier@warehouse.com).
4. Upon receiving the email, the supplier processes the purchase order and updates their inventory accordingly.
5. The supplier sends a confirmation email to the warehouse, acknowledging receipt of the purchase order.

Schedule:

- Purchase orders shall be generated and sent to suppliers daily at 9:00 AM local time.
- Suppliers are required to confirm receipt of the purchase orders within 24 hours of receipt.

Note: The above interface requirements outline the communication process between the warehouse stock management system and external suppliers for inventory replenishment orders. The specifications include file format, data fields, transfer protocol, endpoint, frequency, and schedule for seamless exchange of information.

## **3.7 Security**

### **3.7.1 Protection**

#### **1. Encryption:**

- All sensitive data stored in the database, transmitted over networks, or stored in persistent storage shall be encrypted using strong encryption algorithms (e.g., AES-256).
- Encryption shall be applied to data at rest (stored data) and data in transit (communication channels) to prevent unauthorized access and interception.

#### **2. Access Control:**

- Role-based access control (RBAC) mechanisms shall be implemented to restrict access to system resources based on users' roles and permissions.
- Access to sensitive functionality and data shall be limited to authorized users only, with strict enforcement of least privilege principles.

#### **3. Authentication Mechanisms:**

- Strong authentication mechanisms, such as multi-factor authentication (MFA) or biometric authentication, shall be implemented to verify the identity of users accessing the system.
- Password-based authentication shall enforce password complexity requirements and regular password rotation policies to mitigate the risk of credential-based attacks.

#### **4. Activity Logging:**

- Comprehensive activity logging shall be implemented to record all user actions and system events occurring within the application.
- Audit logs shall capture details such as user logins, access attempts, data modifications, and administrative actions for forensic analysis and compliance auditing.

#### **5. Data Integrity Checks:**

- Data integrity checks shall be enforced at various points within the system to detect and prevent unauthorized modifications or tampering of data.
- Hash functions or digital signatures may be used to verify the integrity of data during transmission or storage, ensuring that data remains unchanged and authentic.

#### **6. Network Security:**

- Network security measures, such as firewalls, intrusion detection/prevention systems (IDS/IPS), and network segmentation, shall be implemented to protect against external threats and unauthorized access.
- Secure communication protocols (e.g., HTTPS, TLS) shall be enforced to encrypt data transmitted over networks and prevent eavesdropping or man-in-the-middle attacks.

#### **7. Regular Security Assessments:**

- Regular security assessments, including vulnerability scanning, penetration testing, and security audits, shall be conducted to identify and remediate potential security weaknesses or vulnerabilities.
- Security patches and updates shall be applied promptly to address known security vulnerabilities and mitigate the risk of exploitation by attackers.

### **3.7.2 Authorization and Authentication**

**Authentication:** The system must utilize user authentication to verify users' identities. Authentication must allow several ways, including username/password authentication, multi-factor authentication (MFA), and biometric authentication. User credentials must be securely stored and sent using encryption techniques to prevent unwanted access.

**Authorization:** Role-based access control (RBAC) will be introduced to ensure fine-grained access control based on user roles and permissions. Different user roles, such as warehouse manager, inventory clerk, sales representative, and administrator, will be specified, each with its own set of access privileges. Authorization rules must be customizable and granular, allowing administrators to specify access rights at the specific data or feature level.

## **3.8 Data Management**

### **1. Types of Information Used by Various Functions:**

- The database shall store various types of information including product details (name, description, SKU), inventory quantities, warehouse locations, supplier information, customer orders, sales transactions, and shipping records.
- Information related to stock levels, order status, shipment tracking, and pricing shall also be managed in the database.

### **2. Frequency of Use:**

- Data related to inventory quantities, product details, and order information shall be accessed frequently during daily warehouse operations.
- Transactional data such as sales orders, purchase orders, and inventory adjustments shall be updated in real-time to reflect current inventory status.

### **3. Data Access Rules:**

- Access to sensitive data such as pricing information, customer details, and financial records shall be restricted to authorized users based on their roles and permissions.
- Role-based access control (RBAC) shall be implemented to enforce data access rules and ensure data security and confidentiality.

### **4. Data Entities and Relationships:**

- The database shall model various data entities such as products, warehouses, suppliers, customers, orders, shipments, and transactions.
- Relationships between entities shall be defined, such as one-to-many relationships between products and inventory items, and many-to-many relationships between orders and products.

### **5. Integrity Constraints:**

- The database shall enforce integrity constraints to maintain data consistency and accuracy.
- Constraints such as primary key constraints, foreign key constraints, unique constraints, and check constraints shall be defined to ensure data integrity at the database level.

### **6. Data Retention:**

- The system shall define data retention policies specifying the duration for which different types of data shall be retained in the database.
- Historical data such as transaction logs and audit trails may be retained for auditing and reporting purposes, while operational data may be archived or purged after a certain period.

## **[Group4-Warehouse Stock Management System] Requirements Specification**

### **7. Valid Range, Accuracy, and Tolerance:**

- Data fields representing quantities, prices, and other numerical values shall have valid ranges defined to ensure data accuracy and consistency.
- Data validation rules shall be enforced to ensure that input data falls within the valid range and meets specified accuracy requirements.
- Tolerance levels for discrepancies in inventory counts and financial calculations shall be defined to accommodate minor variations and errors.

### **8. Units of Measure:**

- The system shall support multiple units of measure for quantities such as weight, volume, and length.
- Conversion factors and conversion rules shall be defined to facilitate seamless conversion between different units of measure.

### **9. Data Formats:**

- Data stored in the database shall adhere to standardized formats to facilitate interoperability and data exchange with external systems.
- Date and time values, currency values, and textual data shall be formatted according to industry-standard conventions.

### **10. Default or Initial Values:**

- Default values shall be defined for certain data fields to ensure completeness and consistency of data entry.
- For example, default values may be provided for product attributes such as weight, dimensions, and pricing if not explicitly specified during product creation.

## **3.9 Standard Compliance**

- ISO 9001: Quality Management System: The system must comply to ISO 9001 quality management standards in order to consistently produce high-quality products and services. Quality control procedures must be adopted throughout the system development lifecycle to identify and reduce risks and problems.
- ISO 27001 - Information Security Management: The system must adhere to ISO 27001 information security management requirements in order to secure sensitive data and maintain confidentiality, integrity, and availability. Access restrictions, encryption, and intrusion detection must be introduced to reduce security risks and vulnerabilities.
- RFID Standards: When using RFID technology for inventory tracking, the system must adhere to RFID standards such as EPCglobal to ensure compatibility and interoperability with RFID hardware and systems.

## **3.10 Portability**

1. Percentage of Host-Dependent Code: - The system should reduce host-dependent code to enable portability across host machines and OS.

- Goal: Less than 20% of the system's codebase should be host-dependent.

2. Use of Portable Language: - Develop the system with a cross-platform compatible programming language.

- Target: 100% of the system's codebase will be built in a language recognized for portability, such as Java or Python.

3. Compiler and Language Subset: - The system must use a widely accepted compiler or language subset to provide interoperability with different operating systems and hardware architectures.  
- Target: The system must be compilable and executable with standard compiler tools available on major operating systems (e.g., GCC for Unix-based systems, Microsoft visual studio for json).

4. Operating System Independence:- The system should work effortlessly with Windows, macOS, and Linux distributions.

- The system's operation must be consistent independent of the underlying operating system environment.

- Target: The system will be tested and validated on many operating systems to ensure compatibility and portability.

5. Environment Independence:- The system should operate consistently across many deployment settings, such as development, testing, staging, and production.

- Configuration settings and dependencies must be managed dynamically to meet environmental variances without needing code changes.

- Target: The system must be deployable to several environments with little configuration modifications, providing ease of deployment and maintenance.

## **4. User Scenarios/Use Cases**

### **4.1. User Stories:**

Here are some user stories identified:

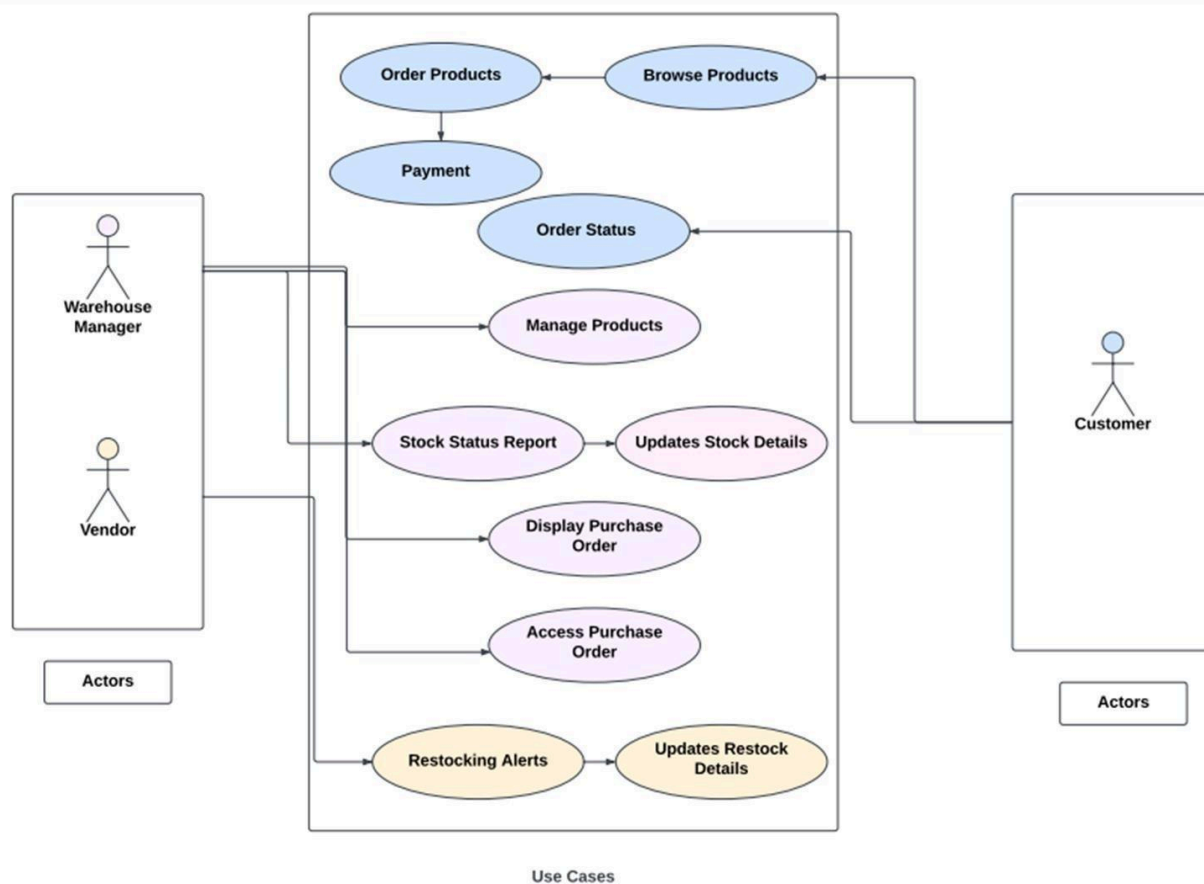
- As a Customer, I would like to be able to browse and order products, so that I can conveniently purchase items online.
- As a Manager, I would like to be able to manage product details, so that the inventory reflects accurate and current product information.
- As a Vendor, I would like to be able to manage purchase orders, so that I can efficiently process and fulfill orders from the warehouse.
- As a Customer, I would like to be able to track my orders, so that I am informed about the status of my purchases.
- As a Warehouse Manager, I would like to be able to create and access stock status reports, so that I can oversee inventory levels effectively.
- As a Warehouse Manager, I would like to be able to update stock details, so that the inventory is kept accurate and up-to-date.
- As a Vendor, I would like to be able to manage restocking details, so that the warehouse maintains sufficient inventory levels.
- As a Vendor, I would like to be able to view purchase orders in a user-friendly manner, so that I can easily understand and process them.
- As a Warehouse Manager, I would like to be able to receive automated alerts for low-stock products, so that I can prevent stock-outs and manage inventory proactively.
- As a Developer, I would like to optimize the system for mobile use and conduct usability testing, so that users can access the system efficiently on mobile devices.

## **[Group4-Warehouse Stock Management System] Requirements Specification**

- As a Customer, I would like to be able to use a payment system, so that I can complete transactions securely and conveniently.

### **4.2 Use case Diagram:**

The use case diagram depicts the Warehouse Stock Management System's primary features and interactions. It illustrates the most common use cases, including Order Products, Browse Products, Manage Products, Display Purchase Order, Access Purchase Order, Stock Status Report, Updates Stock Details, and Restocking Alerts.. These use examples describe the system's main features as well as the interactions between the Warehouse Manager, Vendor, and Customer actors. This graphic gives a brief overview of the system's scope and functioning.





## 5.Deleted or Deferred Requirements

Req#	Business Requirement	Status	Comments	Pr i	Date Rvwd	SME Reviewed /Approved
BR_LR_01	The system should track inventory items using GPS for real-time location updates.	May 2024: Deleted. This requirement has been deleted due to complexity in implementation.	Business Process = "Inventory Logistics and Tracking".	1	04/10/2024	mohseen
BR_LR_02	The system should automatically place restocking orders based on AI-driven predictions of stock depletion.	May 2024: Deleted. This requirement has been deleted due to AI integration complexity.	Business Process = "Automated Inventory Replenishment".	2	03/15/2024	mohseen
BR_LR_03	The system should integrate with external e-commerce platforms to synchronize inventory and order data.	May 2024: Deferred This requirement has been deferred because that requires additional research on API integrations.	Business Process = "Inventory Synchronization and Order Processing".	1	04/15/2024	mohseen

## 6.Requirements Confirmation/Stakeholder sign-off

Include documentation of the approval or confirmation of the requirements here. For example:

Meeting Date	Attendees (name and role)	Comments
--------------	---------------------------	----------

**[Group4-Warehouse Stock Management System] Requirements Specification**

03/20/2024	<p>Sri Sravani Gullapalli, Quality Assurance Specialist</p> <p>Sri Laxmi Prasanna Joginipelli, Software Developer</p> <p>Gayathri Kaparathi, Software Developer</p> <p>Vaibhavi Rachamalla, Database and Software Testing Specialist</p> <p>Mythri Gudibanda, Automation Specialist</p> <p>Sai Tharun Reddy Rupireddy and Karthik Reddy Pandi, Project Advisors.</p>	<p>All planned functionalities including basic inventory management interface, and initial database setup were confirmed and approved for development.</p>
04/10/2024	<p>Sri Sravani Gullapalli, Quality Assurance Specialist</p> <p>Sri Laxmi Prasanna Joginipelli, Software Developer</p> <p>Gayathri Kaparathi, Software Developer</p> <p>Vaibhavi Rachamalla, Database and Software Testing Specialist</p> <p>Mythri Gudibanda, Automation Specialist</p>	<p>After reviewing the initial development progress and time constraints, the team decided to defer advanced reporting features and third-party ecommerce integrations to future phases of the project. Features requiring complex AI-driven algorithms for inventory forecasting were deleted from the scope due to integration complexity and resource limitations.</p>

## 7. EXECUTION

### 7.1 Cloning, Configuration and Running of the project

#### Github Execution:

#### Downloading from Github and required Installation:

To download the project from github and set it up

**Clone the repository :** Use git clone

[https://github.com/vibzvaibhavi/ITCS6112\\_MDSP24\\_Group\\_4.git](https://github.com/vibzvaibhavi/ITCS6112_MDSP24_Group_4.git) to clone the repository.

#### Install Required Software:

follow these steps:

1. **\*\*Install Node.js and NPM\*\***: If you haven't already, download and install Node.js and NPM from [Node.js website] (<https://nodejs.org/>) .
2. **\*\*Install MongoDB\*\***: Install MongoDB on your system. You can download MongoDB from [here] (<https://www.mongodb.com/try/download/community>) .
3. **\*\*Install Dependency Packages\*\***: Navigate to the project directory in your terminal or command prompt and run the following command:

This will install all the required dependencies for the project.

**Development Tools:** Such as IDE(we used VS code) and Git for version control.

**Runtime Environment:** Installation of node.js, javascript and other suitable runtime.

**Environment setup:** Configure environment variables as required as API keys, database connections etc.

#### Frontend:

1. Navigate to the `frontend` directory in the terminal or command prompt.
2. Run `npm run dev` or `yarn dev`.

#### Backend:

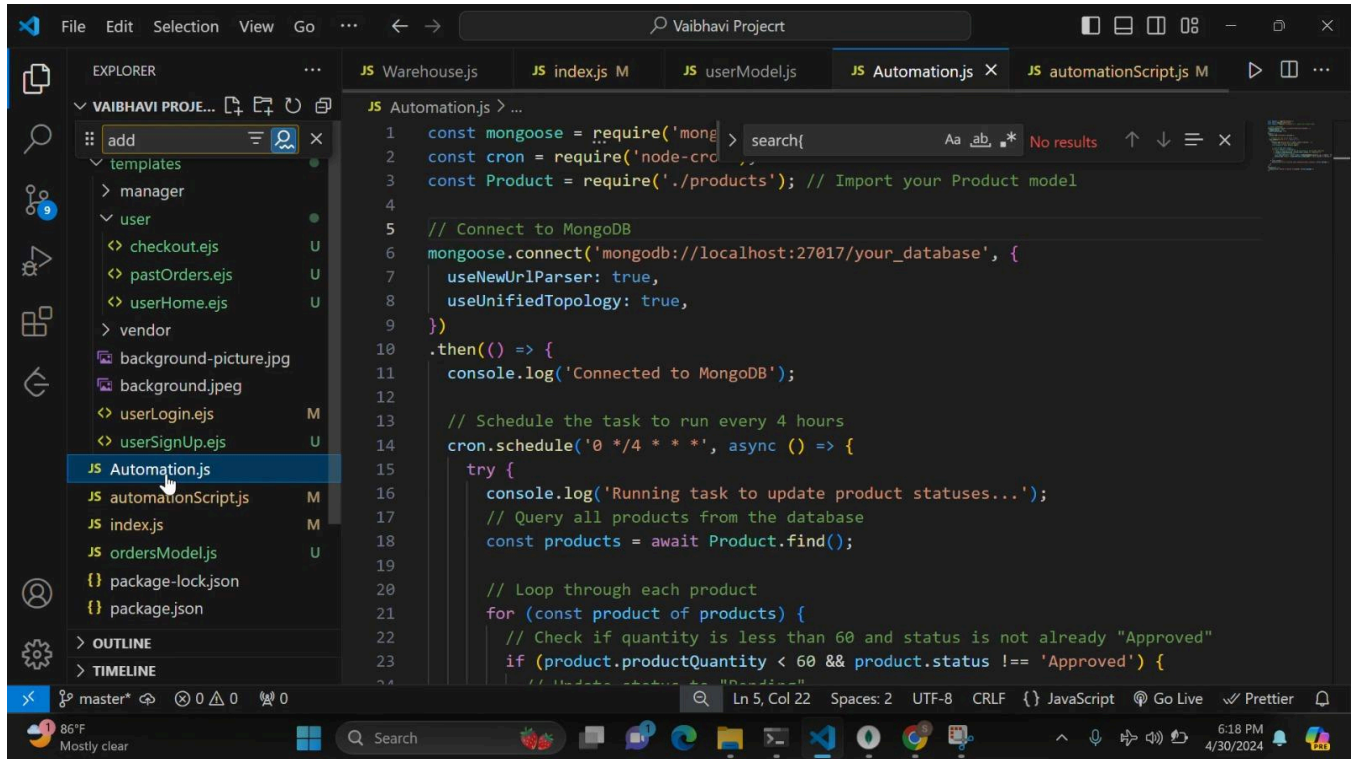
1. Navigate to the `backend` directory in the terminal or command prompt.

## [Group4-Warehouse Stock Management System] Requirements Specification

2. Run nodemon app.

## 7.2 User-manual

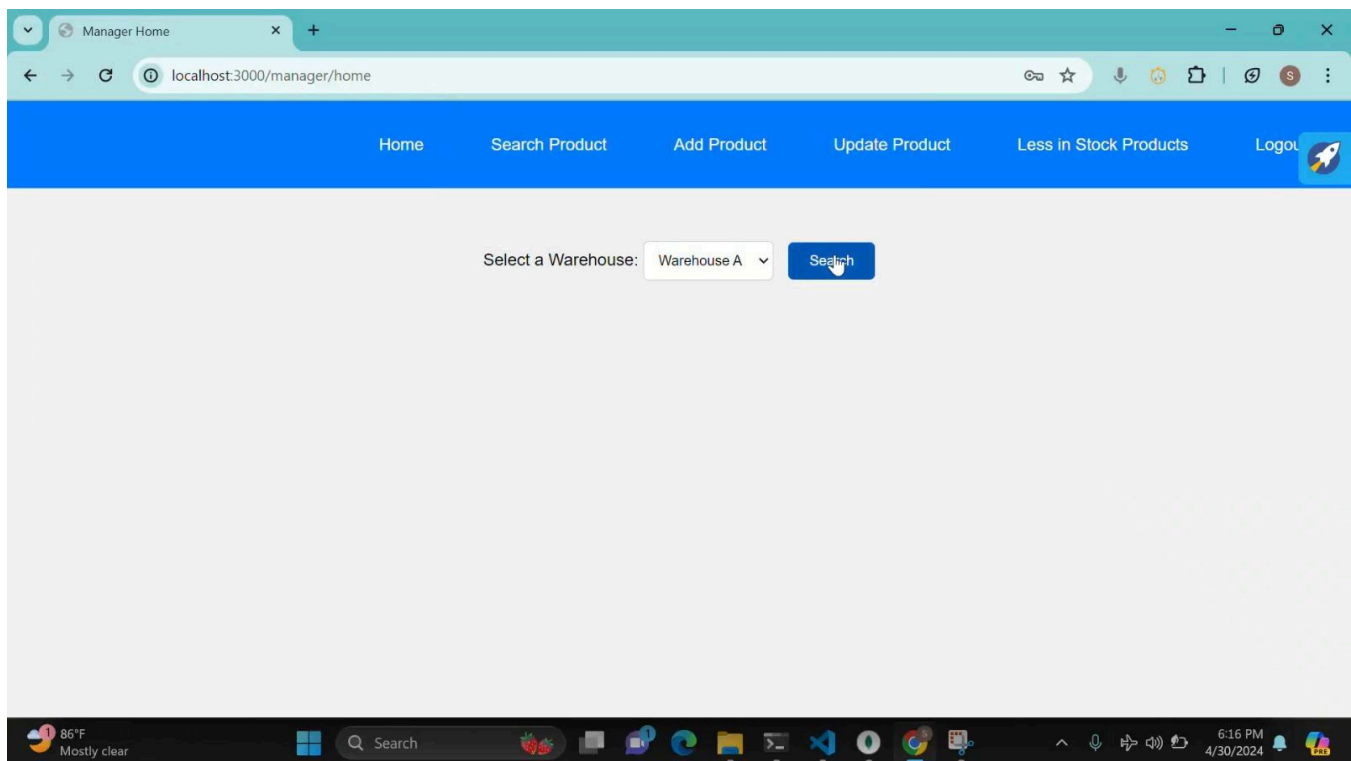
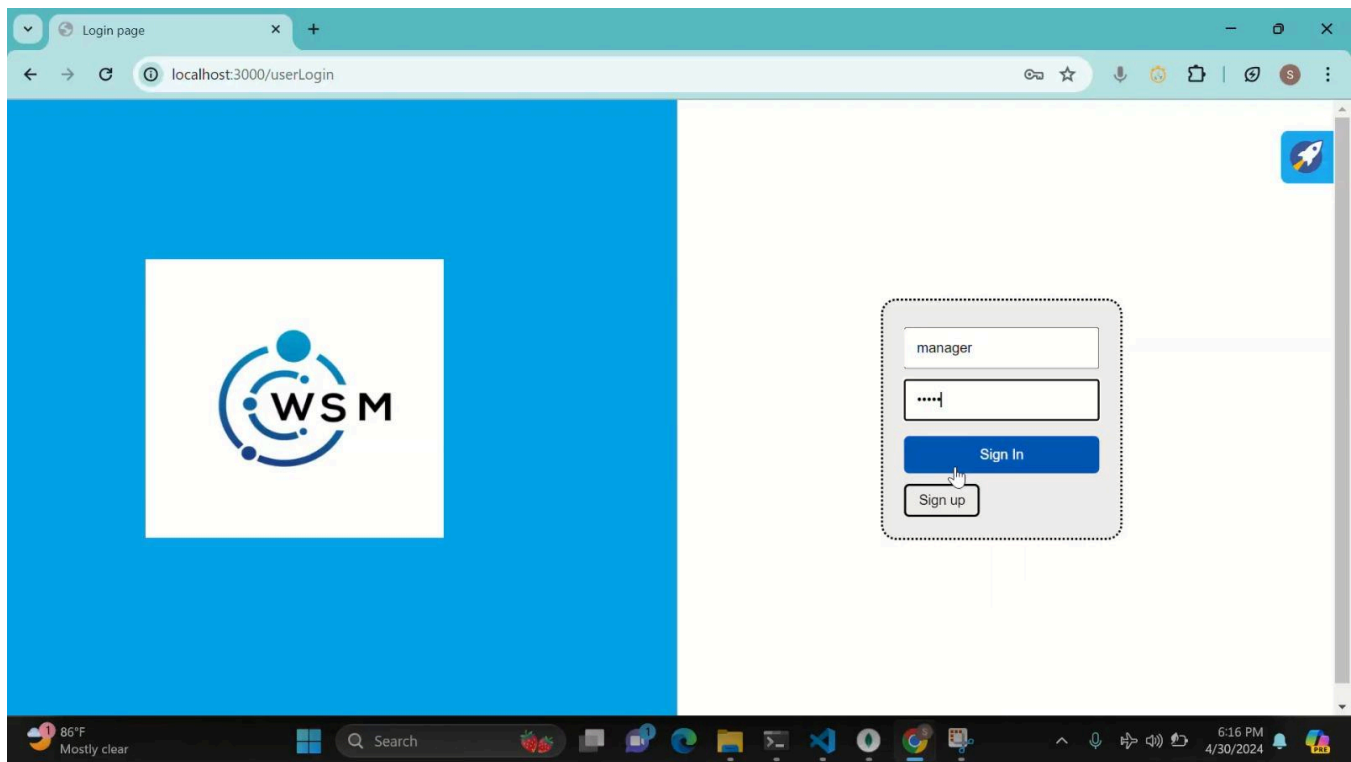
**Automation:** automation file which will run after every 4 hours to check for products less in stock



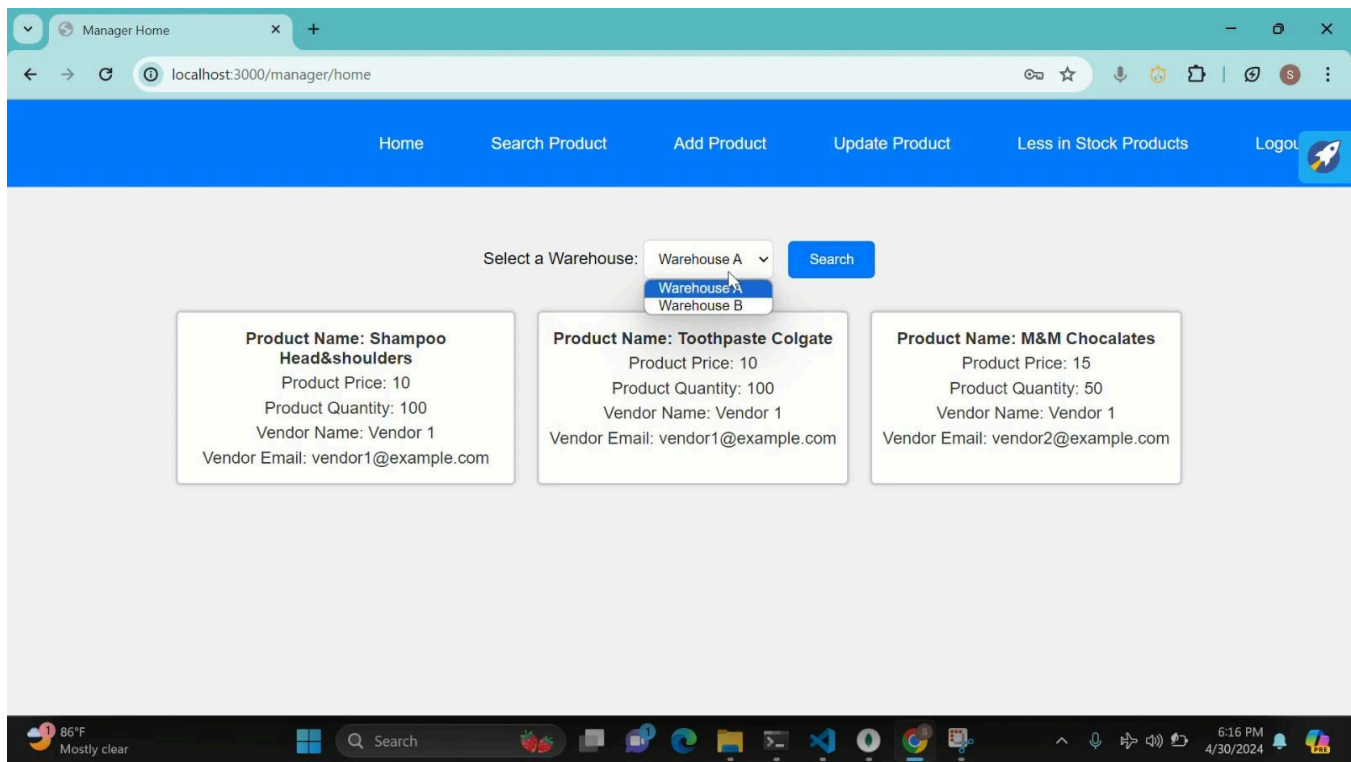
```
1 const mongoose = require('mongoose');
2 const cron = require('node-cron');
3 const Product = require('./products'); // Import your Product model
4
5 // Connect to MongoDB
6 mongoose.connect('mongodb://localhost:27017/your_database', {
7   useNewUrlParser: true,
8   useUnifiedTopology: true,
9 });
10
11 .then(() => {
12   console.log('Connected to MongoDB');
13
14   // Schedule the task to run every 4 hours
15   cron.schedule('0 */4 * * *', async () => {
16     try {
17       console.log('Running task to update product statuses...');
18       // Query all products from the database
19       const products = await Product.find();
20
21       // Loop through each product
22       for (const product of products) {
23         // Check if quantity is less than 60 and status is not already "Approved"
24         if (product.productQuantity < 60 && product.status !== 'Approved') {
```

**Manager Login page :**

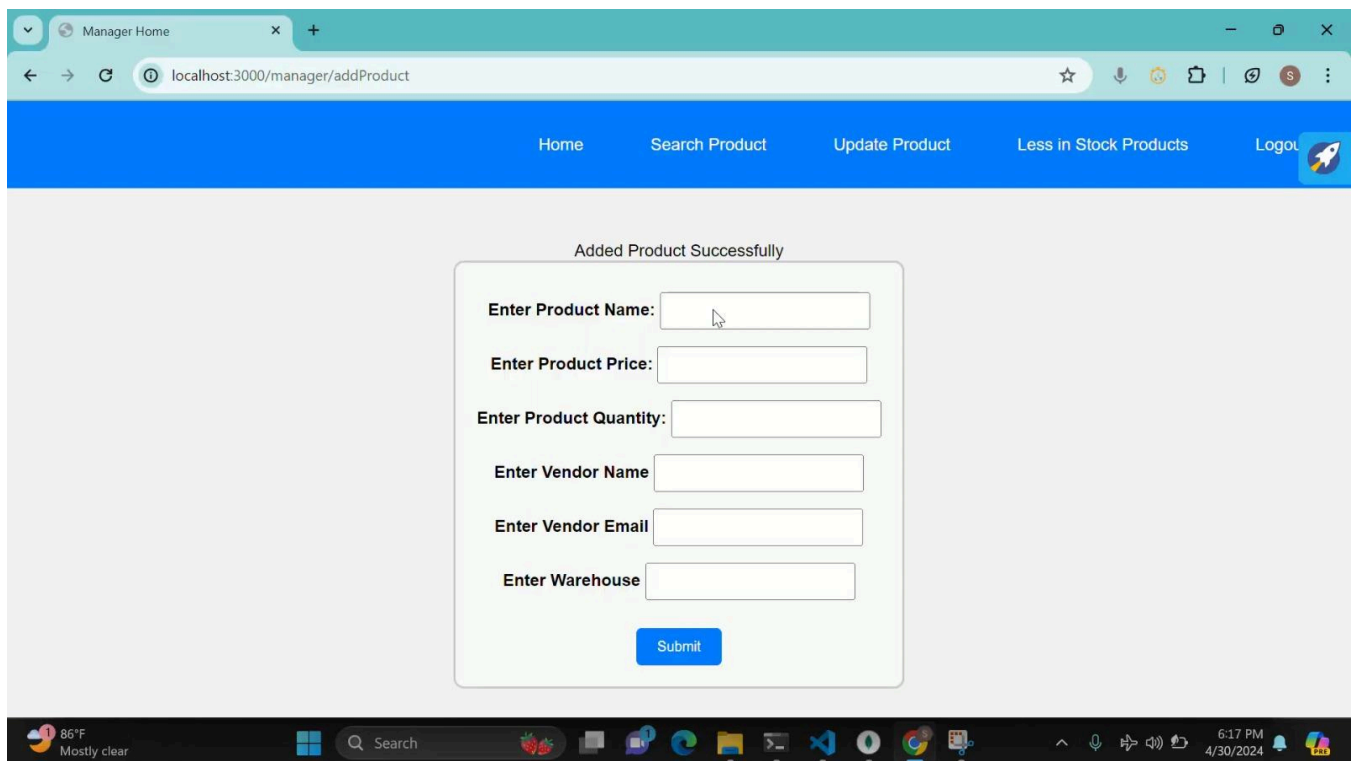
## [Group4-Warehouse Stock Management System] Requirements Specification



## [Group4-Warehouse Stock Management System] Requirements Specification

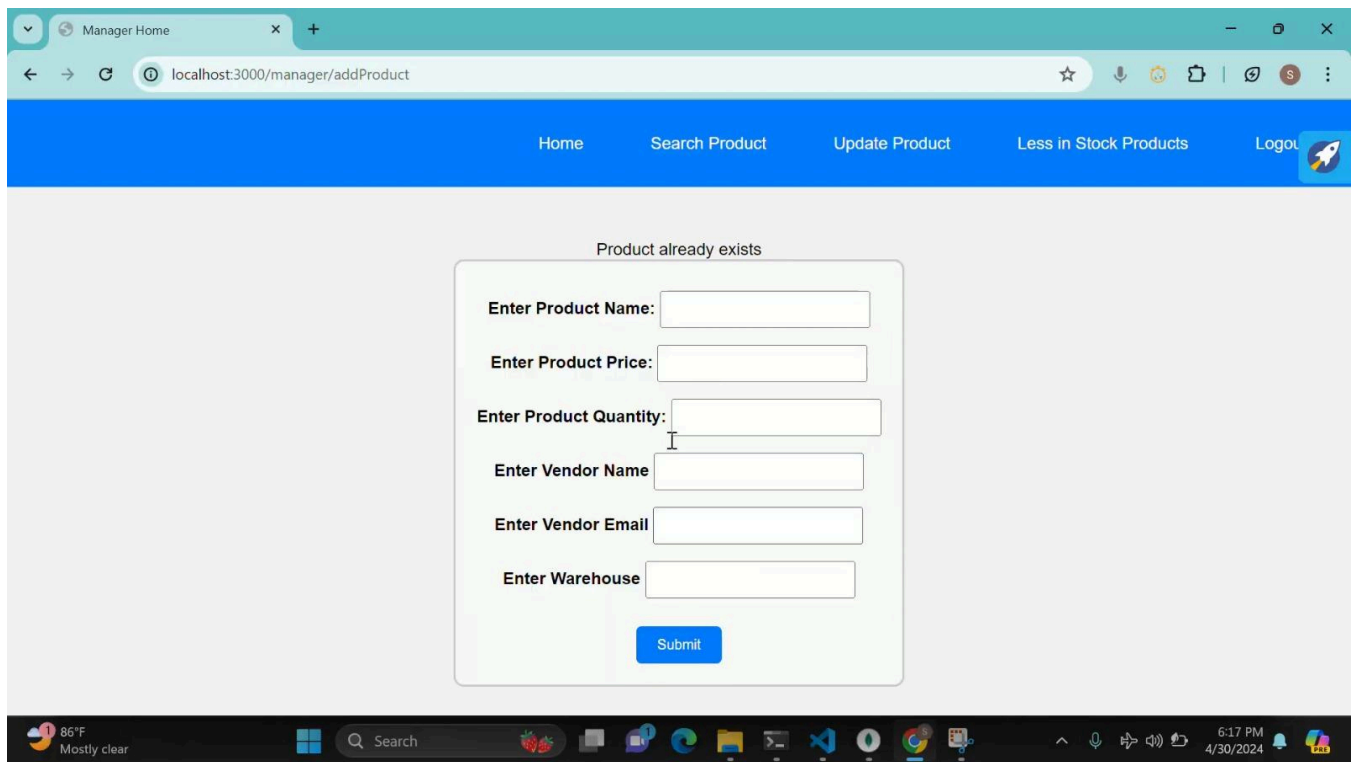


### Adding a product:



If the same product exists ,the page displays as “product already existed”

## [Group4-Warehouse Stock Management System] Requirements Specification



Manager Home

localhost:3000/manager/addProduct

Home Search Product Update Product Less in Stock Products Logout

Product already exists

Enter Product Name:

Enter Product Price:

Enter Product Quantity:

Enter Vendor Name

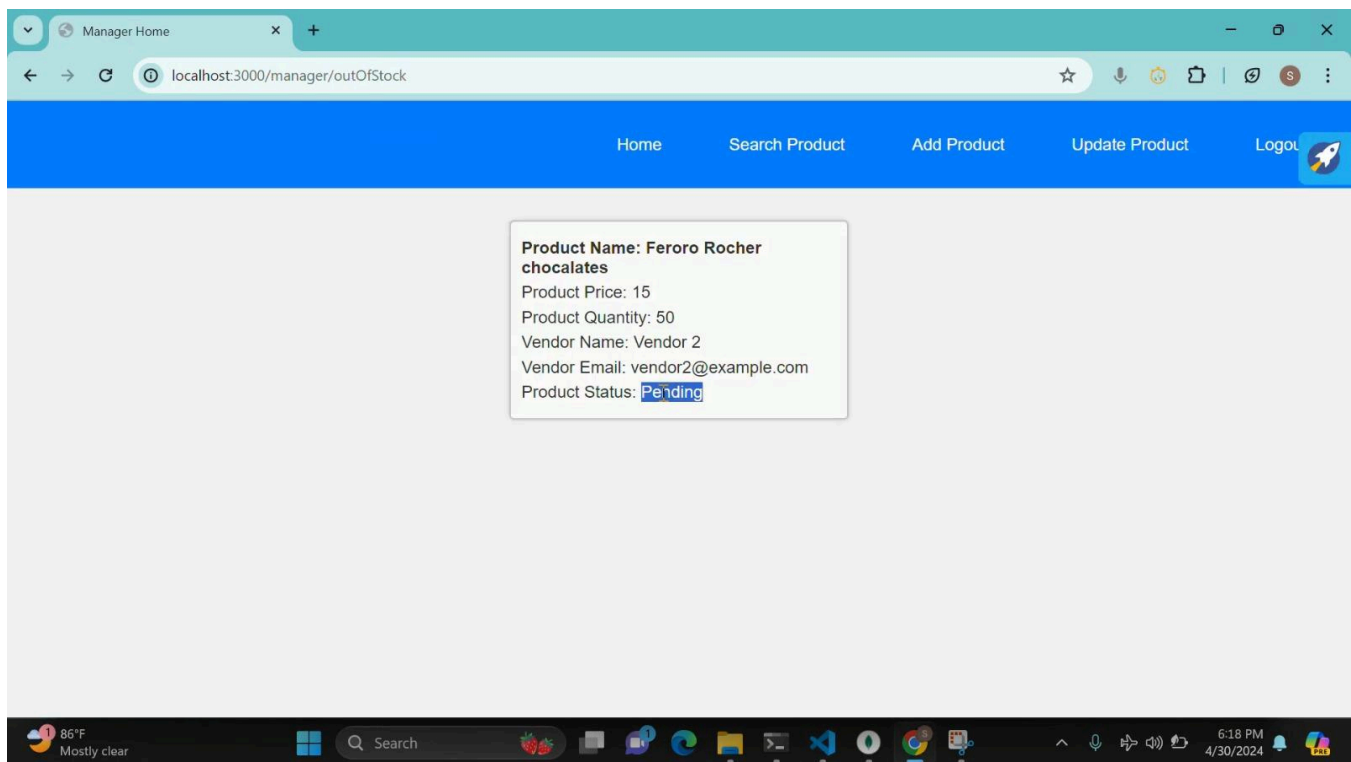
Enter Vendor Email

Enter Warehouse

Submit

86°F Mostly clear 6:17 PM 4/30/2024

### Items less in product



Manager Home

localhost:3000/manager/outOfStock

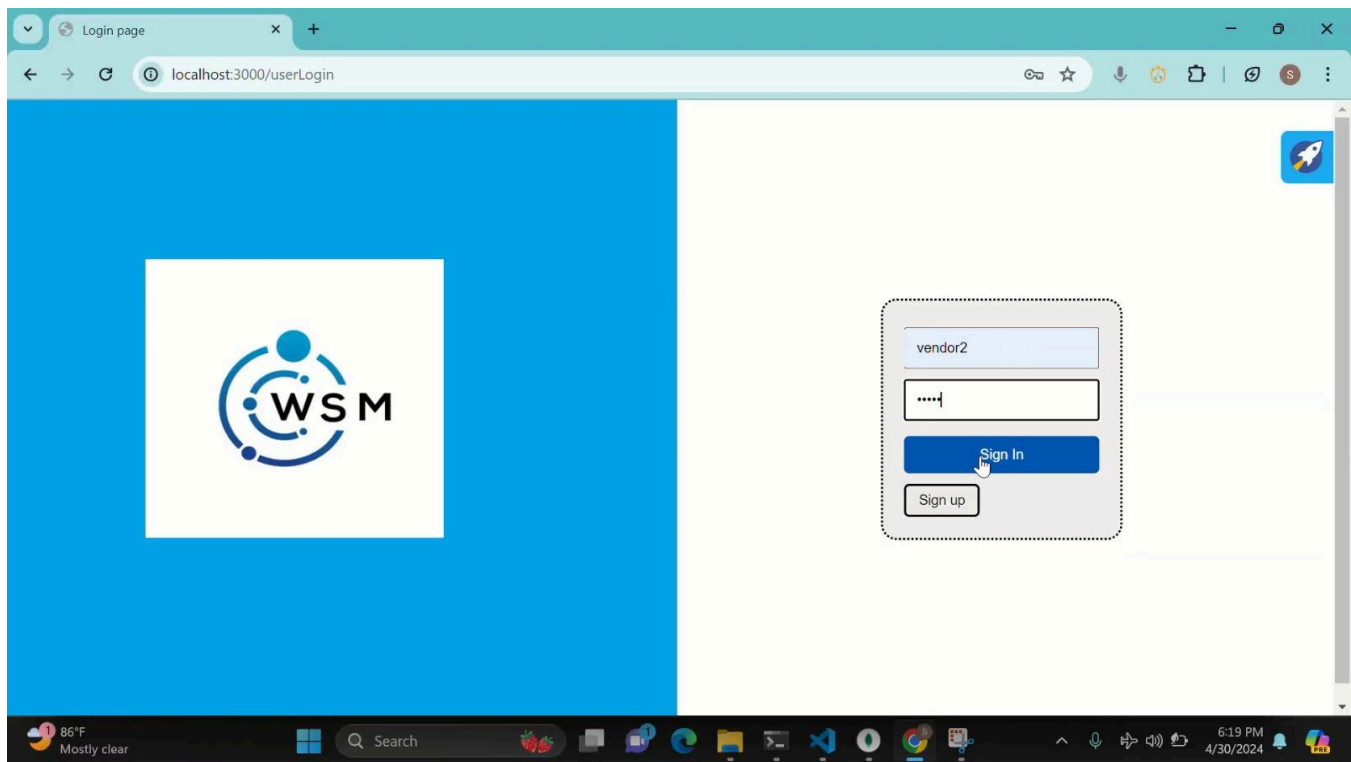
Home Search Product Add Product Update Product Logout

Product Name: Feroro Rocher chocolates  
Product Price: 15  
Product Quantity: 50  
Vendor Name: Vendor 2  
Vendor Email: vendor2@example.com  
Product Status: Pending

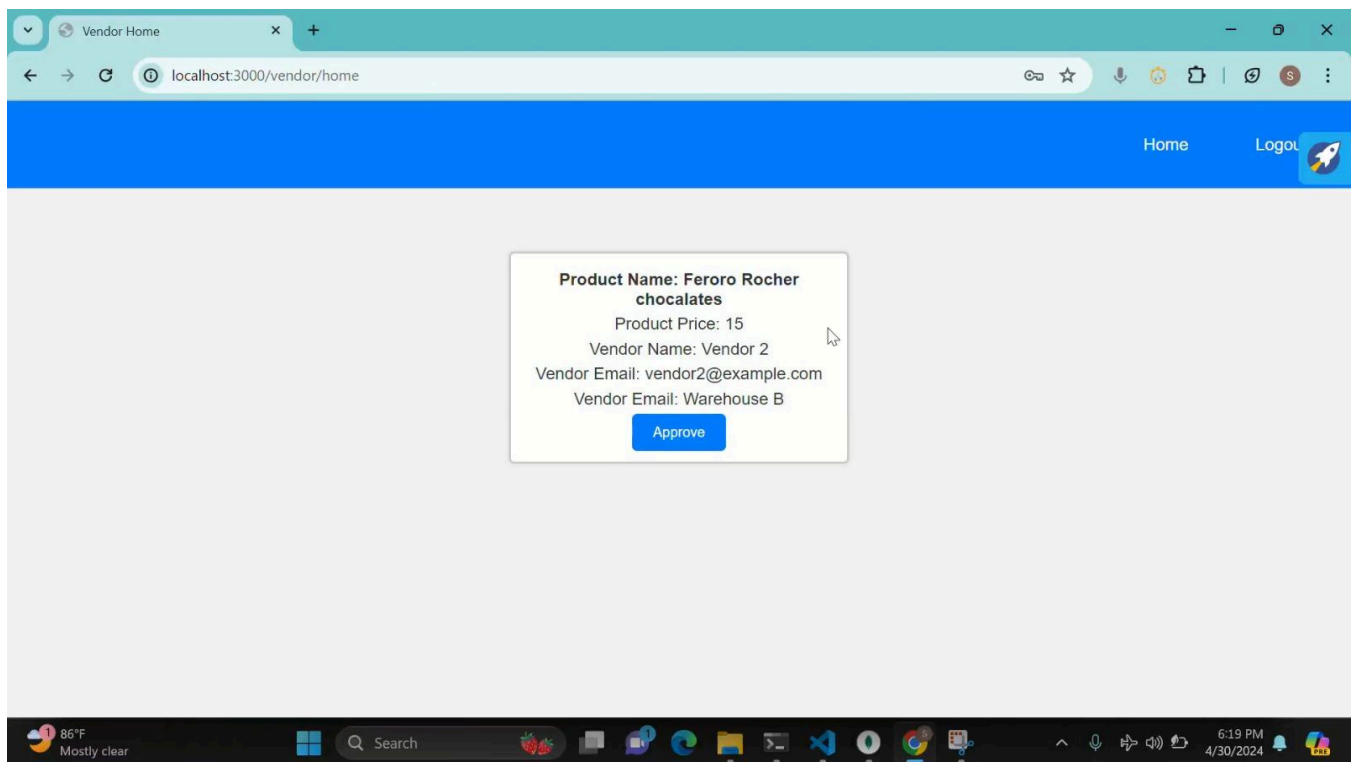
86°F Mostly clear 6:18 PM 4/30/2024

### Vendor Login page:

## [Group4-Warehouse Stock Management System] Requirements Specification



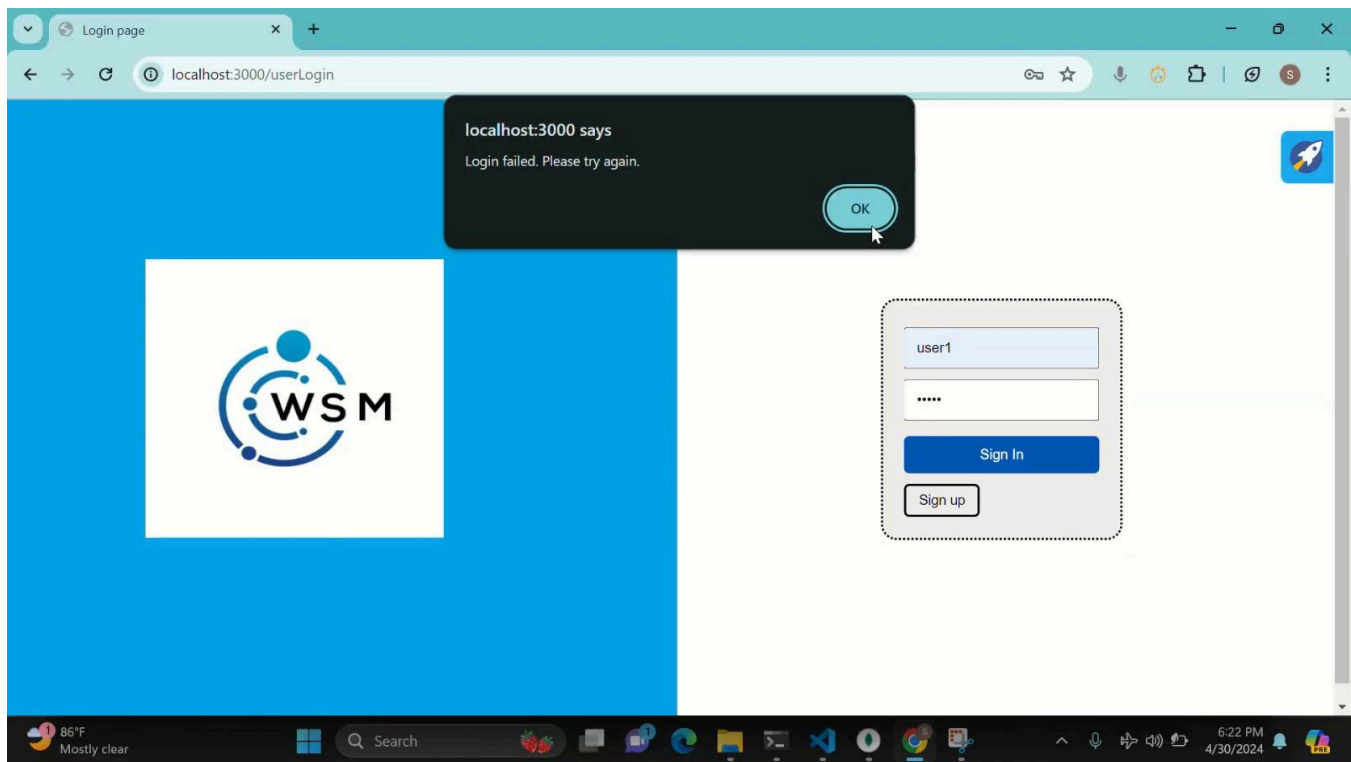
### Vendor gets notified about less in stock



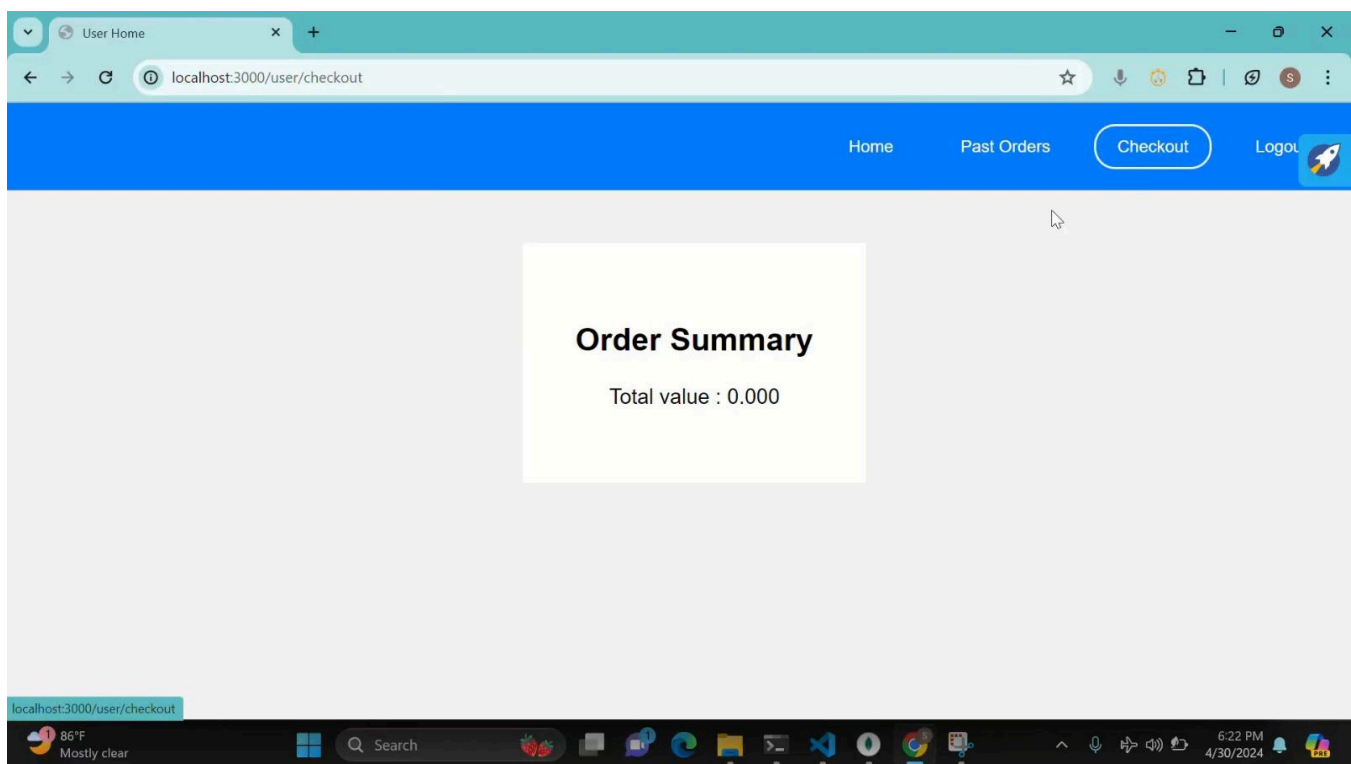
### User login page:



## [Group4-Warehouse Stock Management System] Requirements Specification



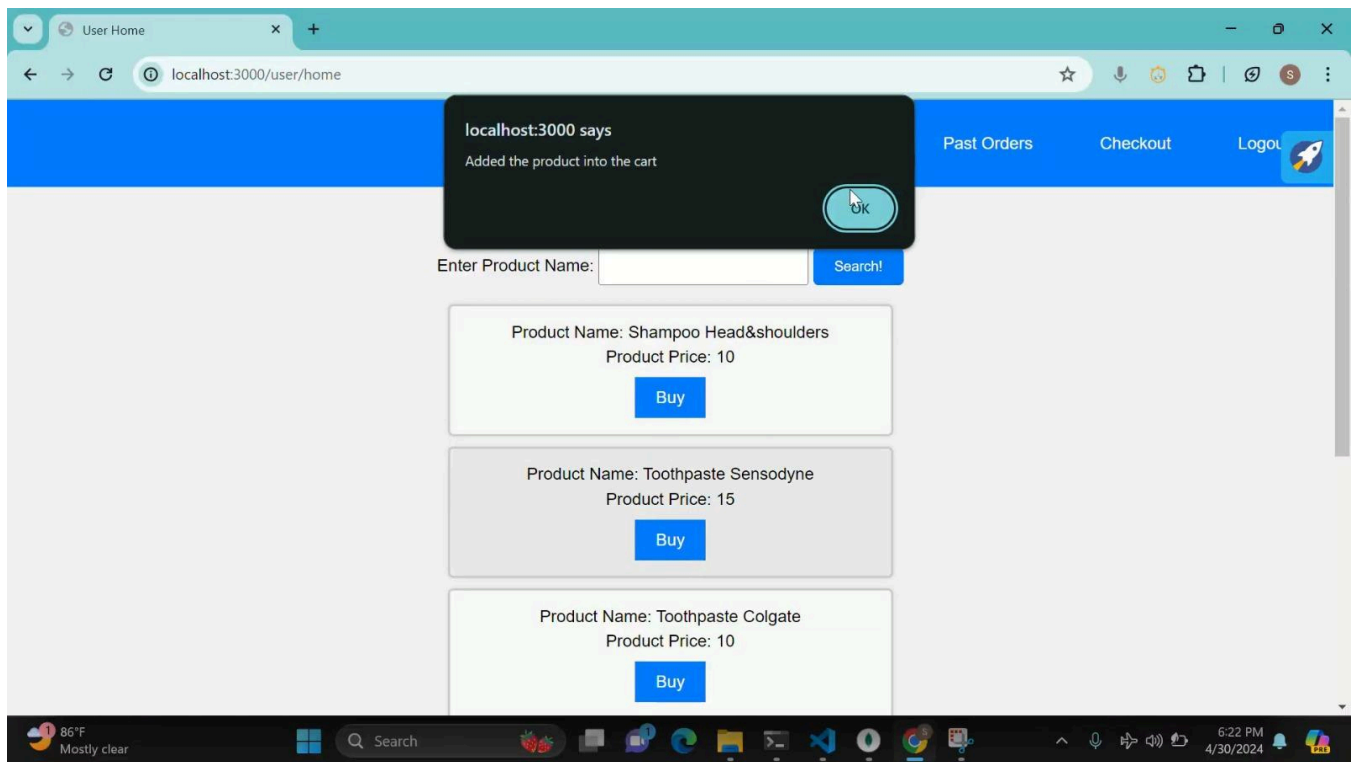
user login page if the password given is wrong.



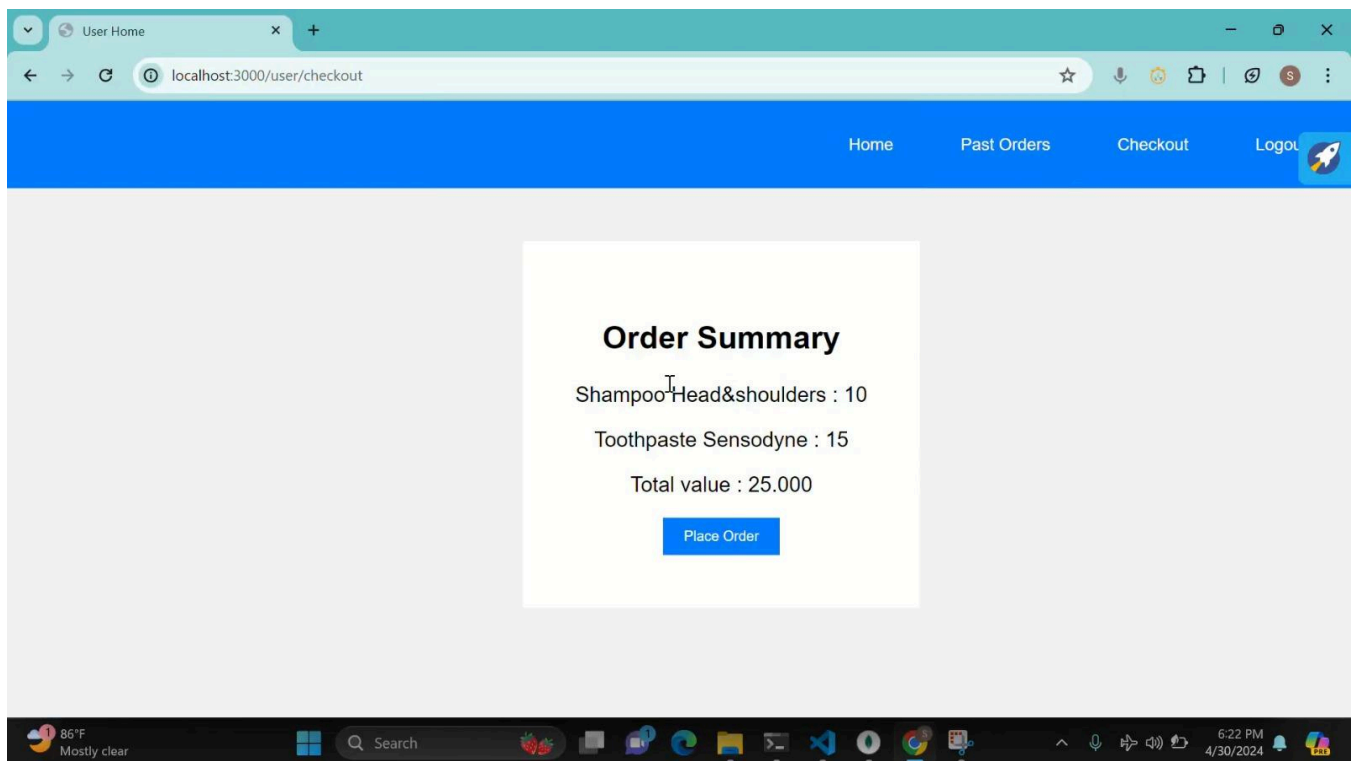
Ordering summary before getting to start the order products

Adding products to cart:

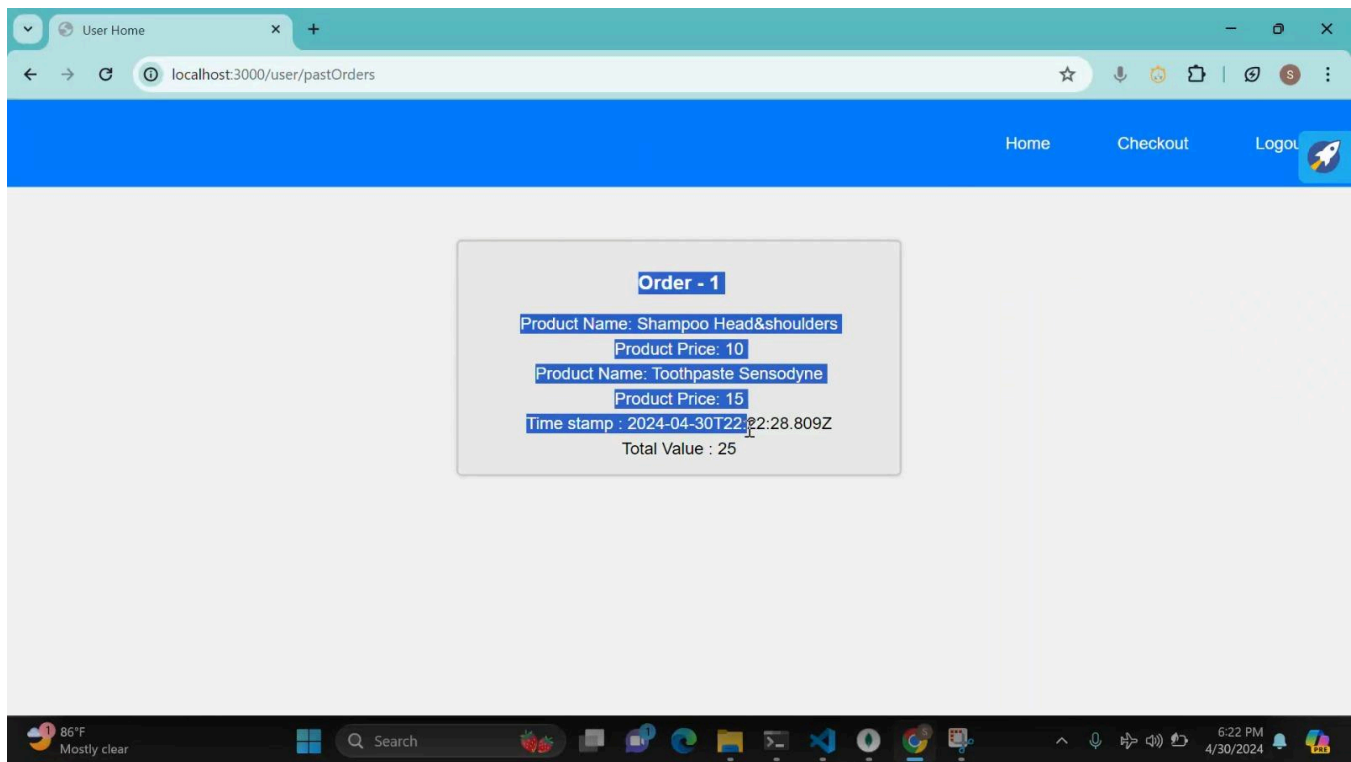
## [Group4-Warehouse Stock Management System] Requirements Specification



Order summary after adding products to cart:



Past order page:



## 7.3.2. Model View Controller Architecture:

### 1. Warehouse Stock Management View:

This view enables warehouse personnel to interact with and manage inventory. This includes:

#### **Display of current stock:**

Lists the available products, including their name, number in stock, and location inside the warehouse.

#### **Search and Filtering Functionality:**

Customers can search for specific products or filter inventory using parameters like category or location.

#### **Add or Remove Products:**

Allows to add new products to the inventory or remove current ones.

#### **Update the Product Information:**

Allows to update product characteristics such as number, location, and other pertinent attributes.

#### **Referring to GP\_Data\_Model:**

The View accepts data from the Controller and displays it.

The Model (GP\_Data\_Model) provides data on goods, quantities, and other inventory details.

### 2. Order Processing View:

This view handles customer orders and fulfillment. This includes:

#### **Order details:**

### **[Group4-Warehouse Stock Management System] Requirements Specification**

Displays extensive information about a specific order, including products, quantities, and customer information.

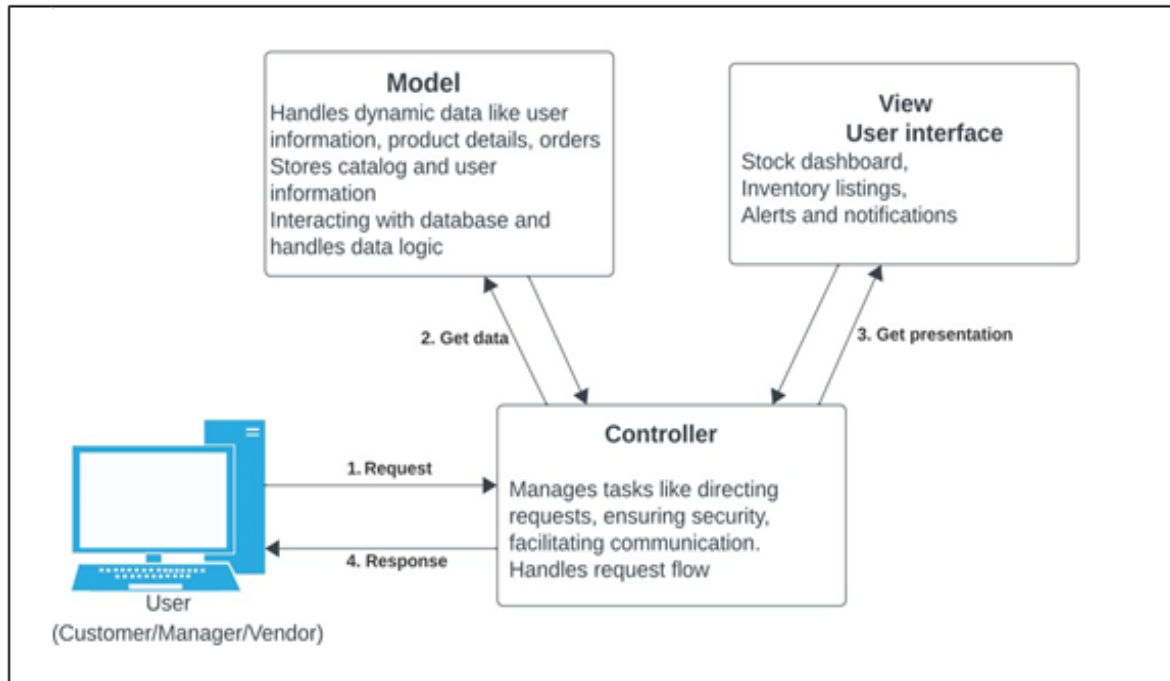
#### **Order Processing Actions:**

Allows warehouse staff to update an order's status (processed, shipped) and manage inventory accordingly.

#### **Referring to GP\_Data\_Model:**

The View communicates with the Controller to handle customer actions related to order fulfillment.

The Model (GP\_Data\_Model) returns data on orders, products, and customer information.



## **APPENDIX**

### **Appendix A.**

- **Sample Input/Output Formats:** Examples of user interfaces for inventory management, order tracking, and authentication screens. Also, examples of system-generated reports such as inventory status and order history.
- **Description of Cost Analysis Studies:** A hypothetical analysis includes estimates of time and resources required by the team, hosting costs for project development.
- **Results of User Surveys:** Hypothetical feedback collected from potential end-users within a simulated environment or from peer reviews within the academic setting.

### **Appendix B. Definitions, Acronyms, and Abbreviations**

- WSMS: Warehouse Stock Management System
- TA: Teaching Assistant
- SME: Subject Matter Expert

## **[Group4-Warehouse Stock Management System] Requirements Specification**

- QA: Quality Assurance
- UI/UX: User Interface/User Experience

### **Appendix C. References**

- <https://github.com/UplandsDynamic/simple-stock-management>
- <https://github.com/AsjadIqbal/InventoryManagementSystem>

### **Appendix D. Requirements Traceability Matrix**

<b>Business Requirement</b>	<b>Area</b>	<b>Deliverables</b>	<b>Status</b>
WSMS_01 Develop and implement a comprehensive database schema tailored to manage and store data about inventory items efficiently. This includes attributes such as item ID, description, quantity, location, and status. The design should support scalability, performance, and quick querying capabilities.	BUA	BUA-CD-01 Inventory DB Design	Accepted
WSMS_02 Establish a secure authentication system to manage user access to the WSMS. This includes processes for user registration, login, and session management. The system should ensure data security and privacy, allowing access only to authorized users based on their roles and permissions.	BUA	BUA-PF-01 Auth Process Flow	Accepted
WSMS_03 Design a user-friendly interface that allows warehouse managers and staff to easily manage inventory levels, add new items, update existing items, and track item movement. The interface should be intuitive, responsive, and accessible from various devices.	BUA	BUA-CD-02 Inventory Mgmt Design	Accepted
WSMS_04 Design and implement a process flow that facilitates efficient order processing within the WSMS. This includes order placement, processing, fulfillment, and tracking. The system should integrate seamlessly with the inventory management system to update stock levels automatically as orders are processed.	BUA	BUA-PF-02 Order Process Flow	Accepted

### **[Group4-Warehouse Stock Management System] Requirements Specification**

<b>Business Requirement</b>	<b>Area</b>	<b>Deliverables</b>	<b>Status</b>
WSMS_05 Develop a reporting module that generates timely and accurate reports about inventory status, sales, order history, and other analytics to support decision-making. The system should offer customizable report options and the ability to export data in various formats.	BUA	BUA-CD-03 Reporting System Design	Accepted

## **Appendix E. Organizing the Requirements**

For the WSMS, the organization of requirements could be particularly focused on:

### **By System Mode**

- Description: Systems may operate differently under various modes such as training, normal operations, or emergency scenarios.
- Application to WSMS: Defining different operational modes for system, for example:
  - Normal Mode: Regular daily operations.
  - Maintenance Mode: Occurs during updates or system checks.
  - Emergency Mode: Activated in case of critical system failures or external emergencies affecting warehouse operations.

### **By User Class**

- Description: Organizing requirements based on different user roles who will interact with the system, each having distinct permissions and capabilities.
- Application to WSMS:
  - Administrators: Manage system settings, user roles, and high-level operations.
  - Warehouse Managers: Oversee inventory management, order fulfillment, and reporting.
  - Vendors: Access portions of the system relevant to supply management and order tracking.

### **By Objects**

- Description: Focuses on real-world entities and their digital counterparts within the system, where each object has attributes and functions.
- Application to WSMS:
  - Inventory Items: Attributes include stock numbers, locations, and status. Functions might include updating stock levels, reordering, etc.
  - Orders: Attributes cover order ID, status, and tracking details. Functions involve order processing, status updates, etc.

### **By Feature**

- Description: Organize by externally visible services that the system provides, which can be described as sequences of stimulus-response pairs.
- Application to WSMS:
  - Inventory Tracking: Includes features such as real-time inventory levels, low stock alerts, and automated reordering.
  - Order Processing: Encompasses order placement, modification, cancellation, and tracking.

### **By Stimulus**

### ***[Group4-Warehouse Stock Management System] Requirements Specification***

- Description: Systems functions are described in terms of stimuli they respond to, useful in reactive systems.
- Application to WSMS:
  - New Order Notification: The system processes new orders as they come in.
  - Stock Level Adjustment: Responds to changes in inventory such as sales, returns, or receipt of new shipments.

#### **By Response**

- Description: Focuses on the functions needed to generate specific system responses.
- Application to WSMS:
  - Generate Reports: Responses include the generation of sales reports, inventory depletion reports, etc.
  - Alerts for Stock Replenishment: Automatic notifications when items reach reorder levels.

#### **By Functional Hierarchy**

- Description: Organizes the system into a hierarchy of functions based on common inputs, outputs, or internal data access.
- Application to WSMS:
  - Top Level: User Management, Inventory Control, Order Management.
  - Second Level: Under User Management — User Registration, Authentication; under Inventory Control — Stock Entry, Stock Edit; under Order Management — Order Entry, Order Tracking.

#### **Additional Comments**

- Multiple Organizational Techniques: Employ a combination of organizational methods such as functional hierarchy for system operations and user classes for role-based access, to ensure comprehensive coverage of all system interactions.
- Utilization of Modeling Tools: Implement state charts for dynamic behaviors, object-oriented analysis for entity modeling, stimulus-response sequences for interactive functionalities, and data flow diagrams to visualize data processes. These tools will enhance clarity and facilitate a better understanding of the system.
- Automated Support Tools: Use automated tools like UML software and project management platforms to maintain consistency, improve efficiency, and reduce manual errors in documentation and development processes.