

An-Najah National University

Faculty of Engineering and Information Technology

Department of Computer Science

****

**Prepared by**: Anas Habash, Yasmeen Ghazi

**Supervised by**: Dr. Adnan Salman

*A project document submitted in partial fulfillment of the requirements for the Bachelor of Science degree in Computer Science.*

An-Najah National University

September, 2024

**Table of Contents**

[Acknowledgment 2](#_Toc174116640)

[Disclaimer 2](#_Toc174116641)

[Abstract 2](#_Toc174116642)

[Introduction 2](#_Toc174116643)

[Overview 2](#_Toc174116644)

[Key Features 2](#_Toc174116645)

[Purpose of Forked Spider Web Company 2](#_Toc174116646)

[Product Scope 2](#_Toc174116647)

[Definitions, Acronyms, and Abbreviations 2](#_Toc174116648)

[Overall Description 2](#_Toc174116649)

[Product perspective 2](#_Toc174116650)

[Product Functions 2](#_Toc174116651)

[User Classes and Characteristics 2](#_Toc174116652)

[Operating Environment 2](#_Toc174116653)

[Technology stack 2](#_Toc174116654)

[Assumptions and dependencies 2](#_Toc174116655)

[Specific Requirements 2](#_Toc174116656)

[User Interfaces 2](#_Toc174116657)

[Software interfaces 2](#_Toc174116658)

[Communication interfaces 2](#_Toc174116659)

[Functional requirement 2](#_Toc174116660)

[Nonfunctional Requirements 2](#_Toc174116661)

**List of Figure**

[Figure 1-Customer Interface 19](#_Toc174174210)

[Figure 2- Warehouse Manager Interface 21](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174211)

[Figure 3-Admin Interface 22](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174212)

[Figure 4-Customer Registration 25](#_Toc174174213)

[Figure 5- Login Page 25](#_Toc174174214)

[Figure 6- Password Recovery 26](#_Toc174174215)

[Figure 7- Change Password 26](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174216)

[Figure 8- Edit Profile For Customer 27](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174217)

[Figure 9- Return Product For Customer 27](#_Toc174174218)

[Figure 10- Customer Rating Product 28](#_Toc174174219)

[Figure 11- Add To Cart 28](#_Toc174174220)

[Figure 12- Search box 29](#_Toc174174221)

[Figure 13- The element being researched 29](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174222)

[Figure 14- User List 29](#_Toc174174223)

[Figure 15- Users And Roles 30](#_Toc174174224)

[Figure 16- Change User Role 30](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174225)

[Figure 17- Sales Order 31](#_Toc174174226)

[Figure 18- Sales Order Form 31](#_Toc174174227)

[Figure 19- Order Items 32](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174228)

[Figure 20- Sales Order Pdf 32](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174229)

[Figure 21- Convert Al Sales Orders To Excel File 33](#_Toc174174230)

[Figure 22- Delivery Order 33](#_Toc174174231)

[Figure 23- Delivery Order Added To The List 34](#_Toc174174232)

[Figure 24- Sales Return 34](#_Toc174174233)

[Figure 25-Transfer Out 35](#_Toc174174234)

[Figure 26- Transfer In 35](#_Toc174174235)

[Figure 27- Scrapping 36](#_Toc174174236)

[Figure 28- Log Sessions 36](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174237)

[Figure 29- Stock Counts 37](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174238)

[Figure 30- Log Analytics 37](#_Toc174174239)

[Figure 31- Log Errors 38](#_Toc174174240)

[Figure 32- Manufacturing Order 38](#_Toc174174241)

[Figure 33- Manufacturing Order PDF 39](#_Toc174174242)

[Figure 34- Convert All Manufacturing Orders To Excel File 39](#_Toc174174243)

[Figure 35- Receive Products From Factory 40](#_Toc174174244)

[Figure 36- Manufacturing Return Products 41](#_Toc174174245)

[Figure 37- Warehouse Manager – Company Chat System 41](#_Toc174174246)

[Figure 38- Confirm Email 42](#_Toc174174247)

[Figure 39- Reset Password 42](#_Toc174174248)

[Figure 40- Order Confirmation 42](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174249)

[Figure 41- New Order 43](file:///D:\c%23\GraduationProject\GraduationProject\Documentation\GRADUATIONDOC%20(1).docx#_Toc174174250)

[Figure 42- Email Products Return 43](#_Toc174174251)

[Figure 43- Email New Order For Delivery Company 44](#_Toc174174252)

[Figure 44- Warning Email 44](#_Toc174174253)

# Acknowledgment

First and foremost, we want to thank Dr. Adnan Salman for his help and support with our project. We also thank all of the academics in the Computer Science Department who helped us along the way. Moreover, we are immensely thankful to our families and friends for their unwavering support, and belief in our potential. Their contributions have been instrumental in the successful completion of this work.

## Disclaimer

This report was written by Anas Habash and Yasmeen Ghazi at the Computer Science Department, Faculty of Engineering and Information Technology, An-Najah National University. It has not been altered or corrected, other than editorial corrections, as a result of assessment and it may contain language as well as content errors. The views expressed in it together with any outcomes and recommendations are solely those of the student(s). An-Najah National University accepts no responsibility or liability for the consequences of this report being used for a purpose other than the purpose for which it was commissioned.

## Abstract

In a global supply chain, efficient distribution is crucial for meeting customer demands promptly. This document outlines a comprehensive workflow for a distribution company that bridges the gap between customers and manufacturers. The workflow involves receiving customer orders, creating orders for manufacturers, managing inventory in warehouses, and distributing goods to sub-distributors and end customers. The proposed system ensures high efficiency in managing orders, and inventory, enabling the company to deliver products swiftly and accurately.

## Introduction

Efficient distribution is a cornerstone of successful supply chain management, especially in a global marketplace where products are sourced from various countries. This document presents a detailed plan for a distribution company that handles orders from customers and coordinates with manufacturers to fulfill these orders. By creating a seamless process from order reception to final delivery, the company can enhance customer satisfaction and maintain a competitive edge. The proposed system emphasizes accurate inventory management, timely order fulfillment, and effective communication between all parties involved.

## Overview

The workflow for the distribution company involves several interconnected stages that ensure efficient handling of orders from customers to final delivery. Initially, the process begins when customers place orders for various products. These orders are meticulously documented and processed to initiate the subsequent steps. The company then creates separate purchase orders for the relevant manufacturers based on these customer orders, sending them to the factories that produce the requested products.

Upon receiving the products from the manufacturers, the company stores them in designated warehouses, with each shipment carefully documented and inventory levels updated accordingly. To maintain accurate stock levels and ensure product availability, the company employs a robust inventory management system. This system tracks stock levels, manages storage, and sets up notifications to alert managers when stock levels are low, thus preventing shortages.

Subsequently, sub-distributors place orders to purchase specific quantities of the stored products. The company processes these orders based on inventory availability and prepares the shipments. Finally, the goods are shipped to the end customers, with each shipment tracked to ensure timely delivery and maintain the condition of the products. This systematic approach not only streamlines operations but also enhances the overall efficiency of the distribution process, ensuring that customer needs are met swiftly and effectively.

## Key Features

1. **Order Management System**: Efficiently handles customer orders, ensuring accurate documentation and processing for timely fulfillment.
2. **Manufacturer Coordination**: Generates and manages purchase orders to various manufacturers based on customer demands, ensuring the correct products are sourced from the appropriate factories.
3. **Inventory Management**: Utilizes a robust inventory management system to track stock levels, manage storage, and ensure product availability. Alerts are set up to notify managers when stock levels are low.
4. **Warehouse Management:** Real-time inventory tracking and automated restocking ensure that stock levels are always optimal, reducing the risk of shortages and overstocking.
5. **User Reviews and Ratings:** Products are ranked based on user reviews, offering valuable insights from other businesses to help make informed purchasing decisions.
6. **User-Friendly Interface:** The platform is designed with a simple and easy-to-use interface, ensuring that users can navigate and utilize its features without any hassle.
7. **Proximity-Based Warehouse Selection:** The platform uses an algorithm to determine the user's location and identifies the nearest warehouse, ensuring that products are shipped from the closest location to reduce delivery times and costs.
8. **Warehouse Manager Communication:** A built-in chat feature facilitates communication between warehouse managers and the company, streamlining coordination and addressing any issues promptly.
9. **Sales Forecasting**: The platform leverages historical demand data from warehouses to predict future sales. By analyzing past order quantities and demand trends, it estimates the sales percentages for the upcoming month, helping businesses plan and optimize their inventory and sales strategies accordingly.

## Purpose of Forked Spider Web Company

Forked Spider Web Company is designed to achieve the following goals:

* Is to streamline and optimize the entire supply chain process, from receiving customer orders to delivering products to end customers. By efficiently managing orders, coordinating with manufacturers, and maintaining accurate inventory levels, the system ensures that customer demands are met promptly and accurately. This enhances customer satisfaction and loyalty, reduces operational inefficiencies, and minimizes the risk of stockouts or overstocking.
* Additionally, the system aims to improve communication and coordination between all parties involved, including customers, manufacturers, sub-distributors, and internal teams. Ultimately, this workflow system seeks to create a seamless and efficient distribution process that supports the company's growth and competitiveness in the market.

## Product Scope

Forked Spider Web Company is designed to offer a comprehensive suite of functionalities aimed at optimizing user experience, enhancing operational efficiency, and ensuring seamless interactions between customers, administrators, and warehouse managers. The product scope includes:

**User Interface:**

* A clean, intuitive, and user-friendly interface designed to facilitate seamless navigation and efficient use of the platform’s features.

**Sales Forecasting:**

* The platform leverages historical demand data from warehouses to predict future sales. By analyzing past order quantities and demand trends, it estimates the sales percentages for the upcoming month, helping businesses plan and optimize their inventory and sales strategies accordingly.

**AI-Powered Chat Support:**

* An advanced AI-driven chat system that provides instant and accurate responses to user and manager queries, enhancing customer service and support.

**Warehouse Management:**

* Support for managing multiple warehouses with dedicated user interfaces. Features include real-time inventory tracking, automated restocking alerts, and efficient communication between warehouse managers.

**Inventory Tracking:**

* Real-time monitoring of stock levels across various warehouses to prevent shortages and overstock situations.

**Analytics and Reporting:**

* Comprehensive tools for generating detailed reports and analyzing data on sales trends, inventory levels, and other key metrics to support strategic decision-making.

**User, Warehouse Manager, and Admin Interfaces:**

* Specialized interfaces for different user roles, including:
  + **Admin Interface:**
    - Dashboard: Overview of system activities and shortcuts to key functionalities.
    - Sales: View customer types, sales orders, and sales order items.
    - Log: Access log errors, log sessions, and error tracking.
    - Settings: Manage company information, add delivery companies, view registered users, manage roles, and taxes.
    - Manufacturing: Manage factories, create manufacturing orders and items.
    - Inventory: Add warehouses, and manage warehouse manager activities.
  + **Warehouse Manager Interface:**
    - Dashboard: Sales and Returns Overview, Goods Receive and Transfer Overview, and Latest Activities.
    - Real-time inventory tracking and management.
  + **Customer Interface:**
    - Browse and order products, manage personal account details, and receive order confirmation emails.

**Review and Rating System:**

* A system for ranking products based on user reviews, providing valuable insights from other businesses to aid in purchasing decisions.

**Shortest Path Algorithm for Sales Order:**

* An algorithm designed to optimize delivery orders for customers by identifying the nearest warehouse to the customer, ensuring efficient and timely delivery. Additionally, it determines the closest delivery company to the customer to further enhance the delivery process and minimize transportation time and costs.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Forked Spider Web Company | Forked Spider Web Company is a comprehensive platform designed for bulk purchasing, warehouse management, and efficient delivery coordination, offering user-friendly interfaces, AI-powered features, and real-time inventory tracking to optimize business operations. |
| |  | | --- | | EOQ | | |  | | --- | | Economic Order Quantity; a formula used to determine the most cost-effective quantity to order to minimize inventory costs. |  |  | | --- | |  | |
| ERP | |  | | --- | | Enterprise Resource Planning; software used to manage and automate core business processes. |  |  | | --- | |  | |
| User Account Management | Involves creating, modifying, and managing accounts for customers on the Forked Spider Web Company platforms |
| |  | | --- | |  |   SMTP | |  | | --- | | Simple Mail Transfer Protocol; an internet standard for email transmission. |  |  | | --- | |  | |
| |  | | --- | | DTO | | |  | | --- | | Data Transfer Object; an object that carries data between processes to reduce the number of method calls. |  |  | | --- | |  | |
| Warehouse Management | The process of overseeing the operations of a warehouse, including inventory tracking, goods receipt, and stock management. |
| Inventory Tracking | The monitoring of stock levels, locations, and movements of inventory items. |
| B2B | Business-to-Business; refers to transactions conducted directly between two businesses, such as the bulk purchasing platform offered by Forked Spider Web Company. |

## Overall Description

## Product perspective

Forked Spider Web Company stands as a web-based platform within the technology ecosystem, aiming to provide a seamless interface to desktop and mobile users, supported by standard hardware components and adherence to security and data privacy standards. Forked Spider Web Company emphasizes user interaction, external system integration, and security within its technology framework.

## Product Functions

Forked Spider Web Company is designed to offer a comprehensive set of functionalities to optimize user experience and operational efficiency.

* **Product Browsing and Ordering:** Customers can explore a wide range of products, place orders, and manage their personal account details. The platform is tailored for businesses or distributors purchasing products in bulk for resale or operational needs.
* **Account Management:** Users, including customers and admin staff, can manage and update their profiles. This feature ensures that all user information is current and accurate.
* **Admin Functionalities**: Admins have comprehensive control over platform operations, including managing delivery companies, handling roles and permissions, monitoring sales data, and overseeing inventory and orders.
* **Warehouse Management:** Warehouse managers can request, receive, and manage goods, handle stock and inventory, and create manufacturing orders, ensuring efficient logistics and inventory control.
* **Quick Delivery:** The platform is designed to ensure rapid delivery to customers, enhancing customer satisfaction and operational efficiency.

## User Classes and Characteristics

Forked Spider Web Company serves a diverse user base, each with distinct roles and characteristics:

1. **Customer:**

* **Roles**: Browse products, Place Order, manage personal account details, Provide feedback or reviews on products
* **Characteristics**: Customers are primarily businesses or distributors purchasing large quantities of products for resale or operational use.

1. **Admin:**

* **Roles**: Add delivery companies, Edit company information, Role management, Assign roles, See log sessions, Log Analytics, Error tracking, Add new factories, View sales data for each day to monitor performance and trends, Create and manage delivery orders, Add new products to the inventory, Add and manage taxes, Create orders and add items by reading files and storing the information in the database
* **Characteristics**: Platform administrators are responsible for managing overall system functionality.

1. **Manager Warehouse:**

* **Roles**: Request Goods from Factories, Receive Goods, Return Goods to Factory, Warehouse Management, View Warehouses, Assign Products to Warehouses, Stock and Inventory Management, Transfer Products, Scrapping, Stock Counts, View User Returns, Manufacturing Orders, Create Manufacturing Orders, Chat with Admin
* **Characteristics**: Highly organized to manage inventory and logistics effectively.

## Operating Environment

Forked Spider Web Company operates within a specified environment, defining the conditions under which the platform functions optimally:

* **Device Compatibility:**

Forked Spider Web Company is designed to be accessible on a variety of devices, including desktops, laptops, tablets, and mobile phones.

* **Web Browsers:**

Forked Spider Web Company is compatible with popular web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

* **Database Management:**

Forked Spider Web Company integrates with a database system, ensuring efficient data retrieval and storage of application records.

* **Server Infrastructure:**

The platform relies on a reliable server infrastructure to handle user interactions, data storage, and seamless operation.

## Technology stack

Forked Spider Web Company development primarily uses a combination of front-end and back-end technologies chosen for their efficiency, scalability, and seamless integration. This standardization simplifies development processes, enhances performance, and facilitates maintenance.

**HTML, CSS, and JavaScript for Front-End Development:**

* **HTML**: Used for structuring the content on the web pages.
* **CSS**: Employed to style and layout the web pages, ensuring a visually appealing and responsive design.
* **JavaScript**: Utilized to create dynamic and interactive elements on the web pages.

**ASP.NET Core 8.0 Razor Pages (C#) and MVC for Back-End Development:**

* **ASP.NET Core 8.0 Razor Pages**: A framework for building dynamic, data-driven web applications with a clean separation of concerns.
* **ASP.NET MVC**: A design pattern for implementing user interfaces, allowing for a clear separation between the model, view, and controller, which helps in organizing and managing the code efficiently.

**Why ASP.NET?**

I chose ASP.NET Core for its high performance, cross-platform capabilities, and modern architecture. It supports development across Windows, macOS, and Linux, offering flexibility in deployment. ASP.NET Core’s unified framework for web UI and APIs, along with its robust tooling and community support, enhances development efficiency. Its lightweight, modular design ensures optimal performance and scalability for handling complex applications. Additionally, the integration with modern client-side frameworks and tools aligns perfectly with current web development practices.

**SQL Server Management Studio:**

* **SQL Server**: A relational database management system used for storing and retrieving data. It provides high performance, reliability, and advanced data querying capabilities.

**Why SQL Server Management Studio?**

* **Performance: Provides high-speed data retrieval and storage capabilities.**
* **Reliability**: Ensures data integrity and supports complex transactions.
* **Advanced Querying**: Offers powerful querying features for managing and analyzing data.

**Python for Machine Learning:**

* **Python**: A versatile programming language known for its ease of use and extensive libraries. In this project, Python is used for machine learning to predict inventory levels and sales for each product. The language's powerful libraries and frameworks enable sophisticated data analysis and the development of predictive models.

**Why Python?**

* **Ease of Use**: Python's clear syntax and rich libraries make it an ideal choice for developing machine learning models.
* **Predictive Analytics**: Enables the prediction of inventory levels and sales for each product, helping to optimize stock management and forecast demand more accurately.

**Other technologies & tools**

* **apiOData** **:** Supports efficient data operations on the server side and handles large datasets with advanced filtering, searching, and sorting capabilities
* **WkHtmlToPdf:** Maintains the exact layout and styling of web pages in PDF format, Supports a wide range of HTML and CSS features for generating high-quality PDFs
* **SignalR:** is a free and open-source library for ASP.NET that enables real-time, bi-directional communication between server and client. It facilitates the delivery of asynchronous notifications from the server to client-side web applications, making it ideal for scenarios where real-time updates are crucial. SignalR supports WebSockets and other fallback mechanisms, ensuring reliable communication across various network conditions. It's commonly used for applications requiring live chat, notifications, real-time data updates, and collaborative features
* **OpenStreetMap**: This tool helps you visualize and determine the coordinates (latitude and longitude) of any location on the map. It's valuable for geographic data visualization and mapping tasks.
* **Haversine Algorithm**: This algorithm calculates the shortest distance between two points on the Earth's surface, given their latitude and longitude. It’s useful for determining the nearest delivery company or warehouse to a customer, ensuring efficient logistics and route planning.
* **Python with Scikit-Learn**: Scikit-Learn is a powerful library for machine learning in Python. It can be used to build predictive models for forecasting inventory needs, sales, or demand. This enables data-driven decision-making and improves operational efficiency.
* **MailKit**: A cross-platform .NET library for email processing. It allows you to send, receive, and manage emails programmatically. MailKit is useful for integrating email functionality into your application, such as sending notifications, alerts, or reports
* **Postman**: A tool for testing and interacting with APIs. It is especially useful for testing the apiOData endpoints, ensuring they work correctly and efficiently.
* **Authorization for Roles and Email SMTP**: Ensures secure access control by authorizing users based on roles and managing email communications using SMTP.
* **AutoMapper**: A popular open-source library in C# that simplifies mapping data between different classes or objects. It helps eliminate repetitive and error-prone code when copying data from one object to another. AutoMapper is especially useful in scenarios like mapping database entities to DTOs (Data Transfer Objects) or ViewModel objects. AutoMapper maps the properties of two different objects by transforming the input object of one type to the output object of another.
* **Repository Pattern**: A design pattern that promotes a clean separation of concerns and improves testability by abstracting data access logic. Common methods include:
* GetAll()
* GetByIdAsync()
* GetByRowGuidAsync()
* AddAsync()
* UpdateAsync()
* DeleteByRowGuidAsync()
* **Globalization**: Supports multiple cultures and formats, crucial for applications that serve users from different regions. This includes services like CurrencyService and CountryService to handle various currencies and country-specific data.
* **Economic Order Quantity (EOQ) and Reorder Point (ROP):** These inventory management tools help optimize inventory levels. EOQ determines the optimal order quantity that minimizes the total cost of inventory management, including ordering and holding costs. ROP indicates the level of inventory at which a new order should be placed to avoid stockouts. These metrics are essential for maintaining efficient inventory control and ensuring timely replenishment of stock.

## Assumptions and dependencies

The development and operation of our project are based on several key assumptions and dependencies:

* **Required Audit**: The platform assumes compliance with data privacy and security regulations, relying on adherence to these standards to protect user information. Regular audits are conducted to ensure these standards are maintained.
* **Server and Database Stability**: Stability and reliability of the server infrastructure and SQL Server database systems are assumed for seamless user experiences and effective data management.
* **User Efficiency**: It is assumed that users have basic proficiency in using web-based platforms, ensuring effective interaction with the features of the application.
* **User Data Accuracy**: The accuracy and reliability of user data are critical for providing personalized and relevant services. It is assumed that users provide accurate and up-to-date information.
* **Scalability Planning**: The platform is designed with scalability in mind to accommodate future growth and increased user demand. This includes scalable server infrastructure, database management, and application architecture to ensure consistent performance as usage expands.

## Specific Requirements

## User Interfaces

**Customer Interfaces:**

* **Home Page**: Users can easily place orders directly from the home page.
* **Product Categories**: Navigate through different categories of products available on the platform.
* **Products**: View a comprehensive list of all products offered by Forked Spider Web Company.
* **Product Search**: Allows users to search for specific products quickly and efficiently.
* **Product Return**: Allows customers to fill out a form and return products seamlessly.
* **My Account**: Users can update their profiles, including adding a profile picture and personal information.
* **Cart**: Users can view items added to their cart and proceed to checkout.
* **Add to Cart**: Easily add products to the cart for a streamlined shopping experience.
* **Product Rating**: Users can rate products based on their experience and satisfaction.
* **Top Selling Orders**: Displays the most popular orders based on sales.
* **Top Rated Products**: Highlights the highest-rated products as per customer reviews.
* **Lowest Priced Products**: Showcases the most affordable products available on the platform.

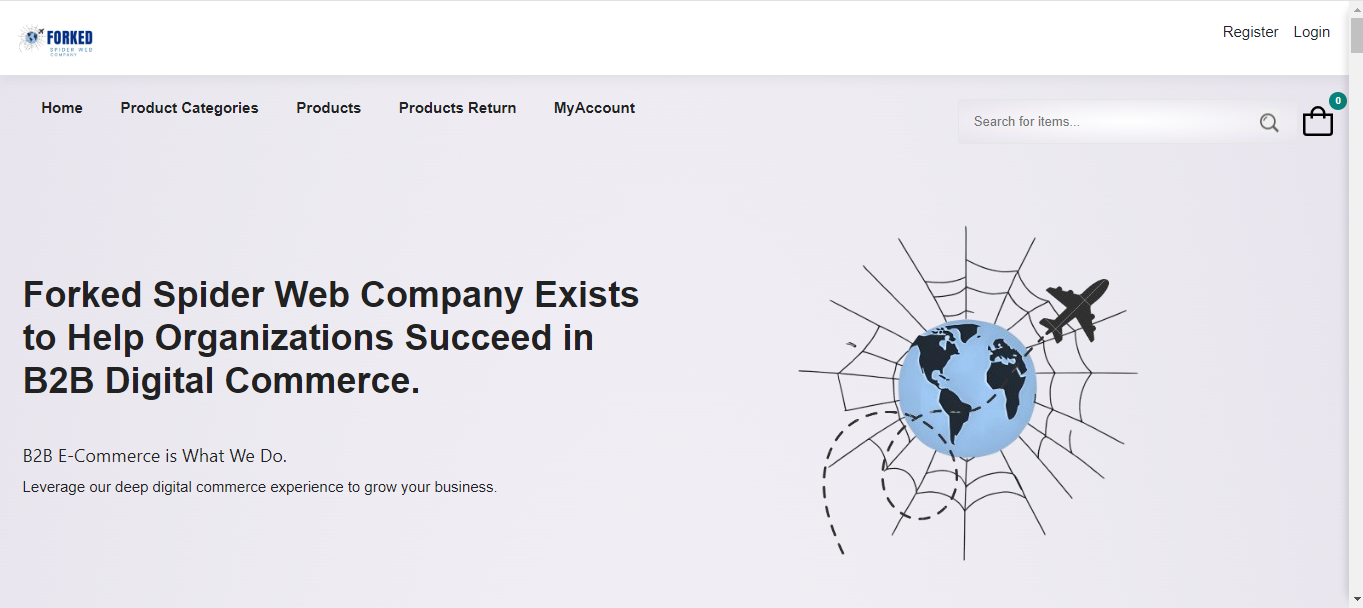
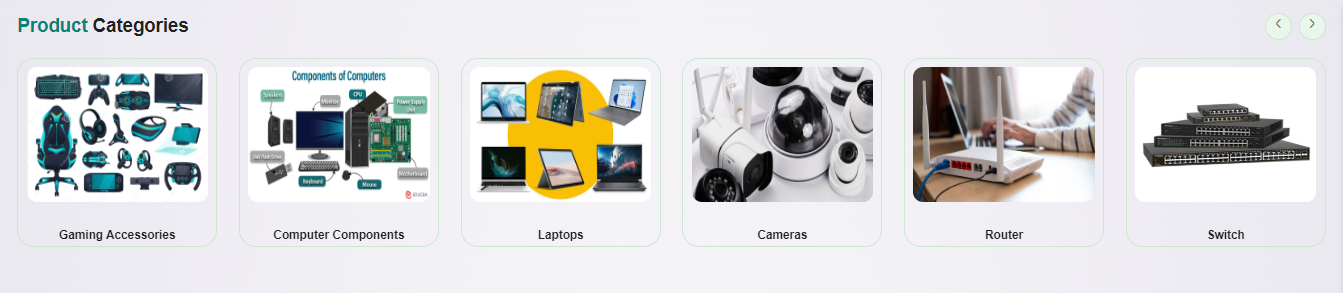
****

Figure 1-Customer Interface

****

**Warehouse Manager Interface:**

* **Dashboard:**

The Dashboard provides a comprehensive overview of key warehouse operations and activities, including:

* **Sales and Returns Overview:** View summary counts for Sales, Sales Returns, Manufacturing Orders, Manufacturing Returns, and Delivery Orders, offering a snapshot of the warehouse's operational status and performance.
* **Goods Receive and Transfer Overview:** Monitor activities related to Goods Receiving, Transfer Out, and Transfer In, ensuring efficient tracking and management of inventory movements.
* **Latest Activities:** Access the most recent updates on Sales, Manufacturing Orders, Inventory Stock, and Inventory Transactions, providing real-time insights into warehouse operations and inventory changes.
* **Request Goods from Factories**: Place orders for goods from factories.
* **Manage Inventory**: Request Goods from Factories, Receive Goods, Return Goods to Factory, View Warehouses, Assign Products to Warehouses, Transfer Products, Scrapping, Stock Counts, and View User Returns.
* **Chat with Admin**: Communicate directly with the admin via a chat feature to resolve any issues or coordinate activities.

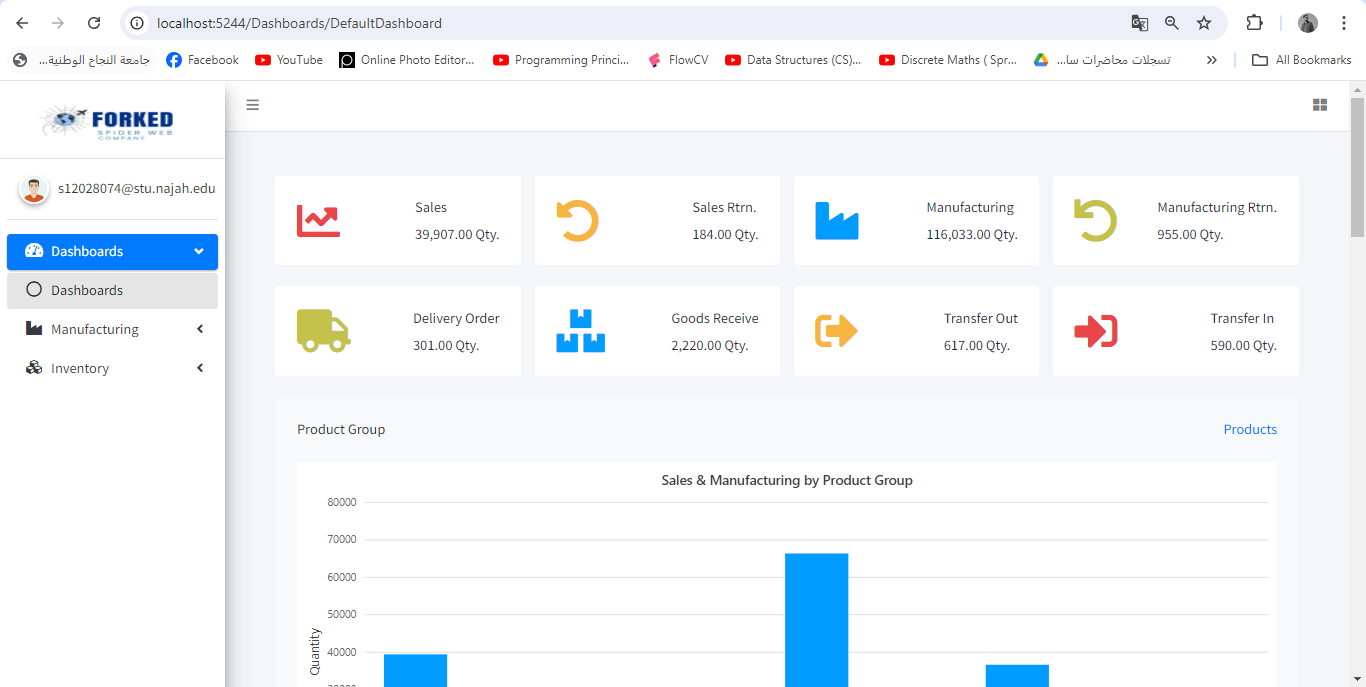


Figure 2- Warehouse Manager Interface

**Admin Interface:**

* **Dashboard:** The dashboard provides a high-level overview of all recent activities and key statistics within the system. It highlights important updates and provides shortcuts to various sections, allowing for efficient monitoring and management.
* **Sales:** In this section, admins can view detailed sales data, including customer types, received sales orders, and sales order items. This enables a clear understanding of sales trends and customer purchasing patterns.
* **Manufacturing:** This section focuses on production management, including:
* **Factories:** View and manage information about available factories.
* **Manufacturing Orders:** Create and manage manufacturing orders and items, facilitating efficient production planning and execution.
* **Inventory:** The Inventory section enables admins to:
* **Add Warehouses:** Create and manage new warehouse locations.
* **Monitor Warehouse Activities:** Review actions taken by warehouse managers, including stock management and inventory updates.
* **Settings:** This area allows admins to manage various settings, including:
  + **Company Information:** Update company details and contact information.
  + **Delivery Companies:** Add and manage delivery companies.
  + **User Management:** View registered users and manage roles and permissions.
  + **Tax Management:** Configure and manage tax settings.
* **Log:** The Log section is crucial for system maintenance and troubleshooting. It includes:
* **Log Errors:** Displays records of system errors for analysis and resolution.
* **Log Sessions:** Tracks user session activity, providing insights into usage patterns.
* **Error Tracking:** Monitors and reports errors in real-time, helping maintain system reliability.

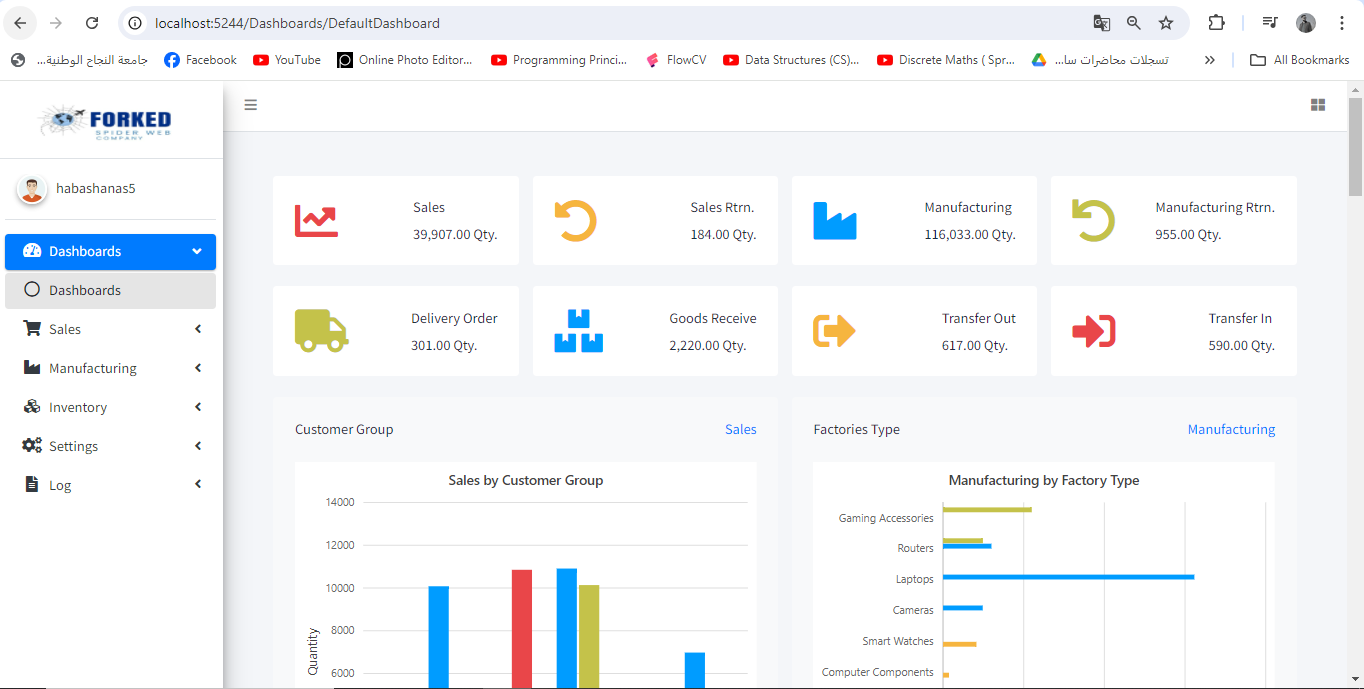


Figure 3-Admin Interface

## Software interfaces

* **Internet browsers:**

Forked Spider web company is compatible with standard web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. Users can access the platform seamlessly through these browsers.

* **Security software:**

Forked Spider web company uses security software to ensure the confidentiality and integrity of user data. Encryption and secure communications standards are applied to protect user information

* **API between the backend and the database:**

**Establishing Connection:**

The backend uses an application programming interface (ApiOData) to interact with the database, allowing it to send queries to retrieve or update data in the database and allowing clients to perform operations such as pagination, filtering, searching, and sorting efficiently.

**Data updates:**

The backend can update data in the database through an application programming interface (API), allowing it to perform operations such as adding new warehouses, creating new orders, adding new delivery companies, and adding new taxes, among other functionalities.

## Communication interfaces

Forked Spider web company communication interfaces play a crucial role in enabling seamless interactions between different components within the system:

**Backend-to-Database Communication:**

The backend communicates with the database through a custom application programming interface (API), making it easier to retrieve and manipulate data. This interaction involves sending queries to the database to perform operations such as reading or updating records.

**Internal Chat System**

To enhance communication between the company and warehouse managers, an internal chat system is implemented. This system allows for real-time messaging, image sharing, and emojis, ensuring that any issues or updates regarding inventory and orders can be promptly addressed. The chat system is accessible through the company's web platform and is secured with end-to-end encryption to protect sensitive information.

**Email Notification**

Automated email notifications are sent to relevant stakeholders for critical updates such as new order placements, stock level alerts, shipment dispatches, and delivery confirmations. This ensures that everyone stays informed and can take timely actions.

## Functional requirement

**Customer Registration**

Create an account for the customer by entering his basic information.

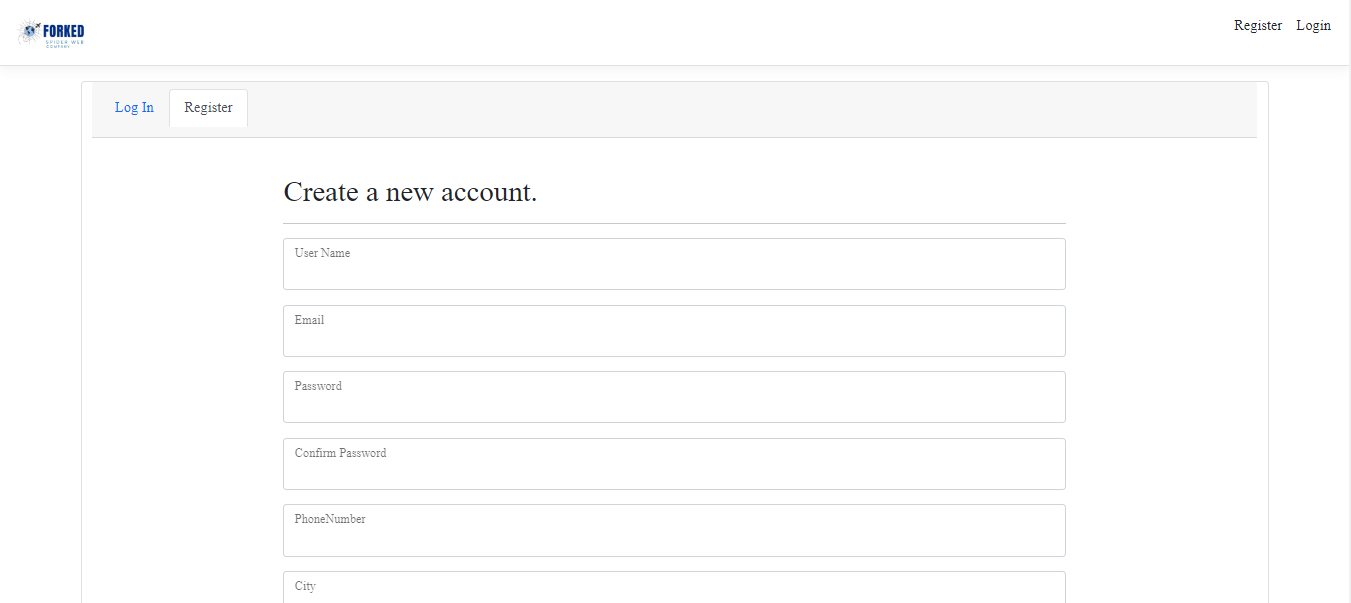


Figure 4-Customer Registration

**Login**

Log in to the account for the customer, warehouse manager, and admin

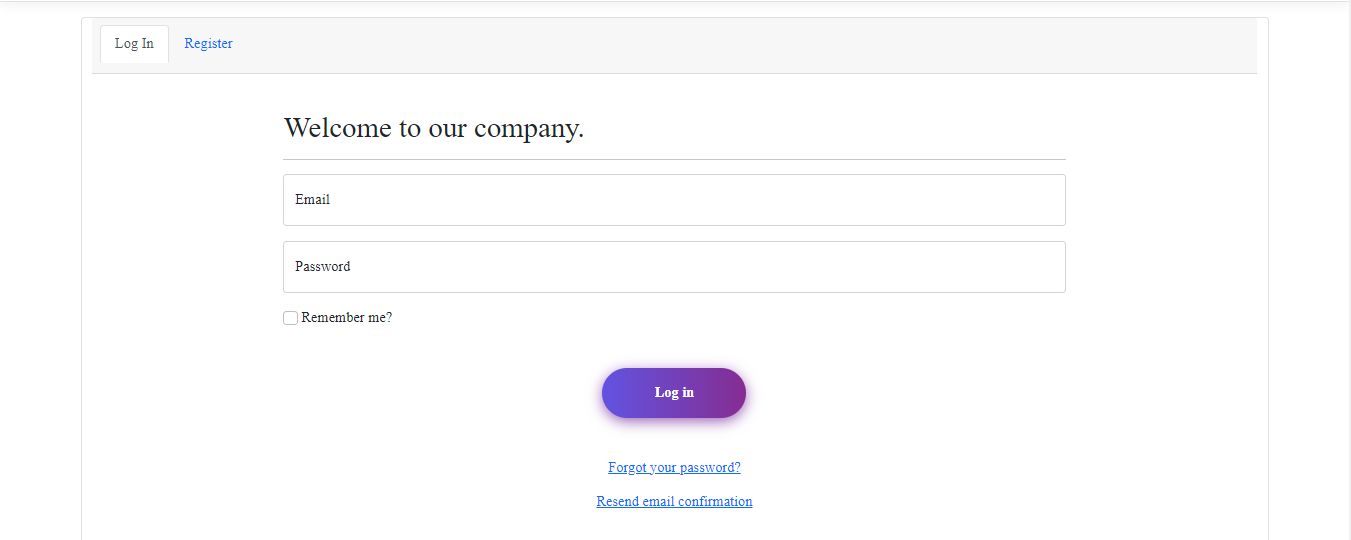


Figure 5- Login Page

**Password Recovery**

Restore the account via email through the Forget Password option

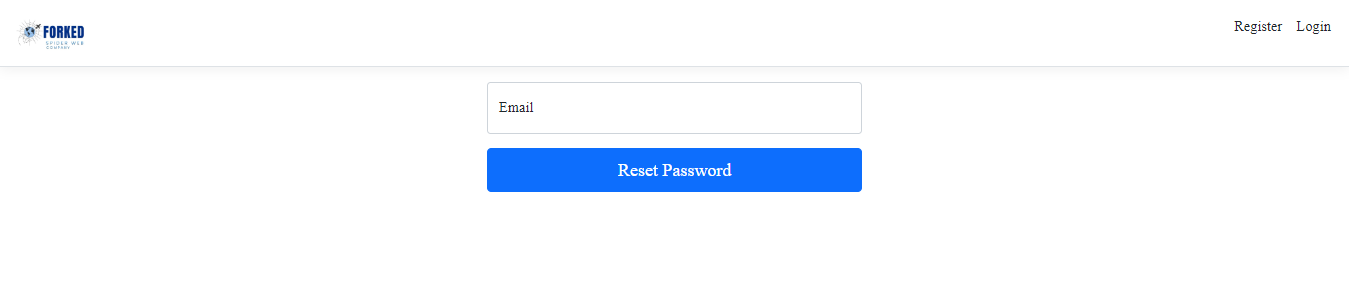


Figure 6- Password Recovery

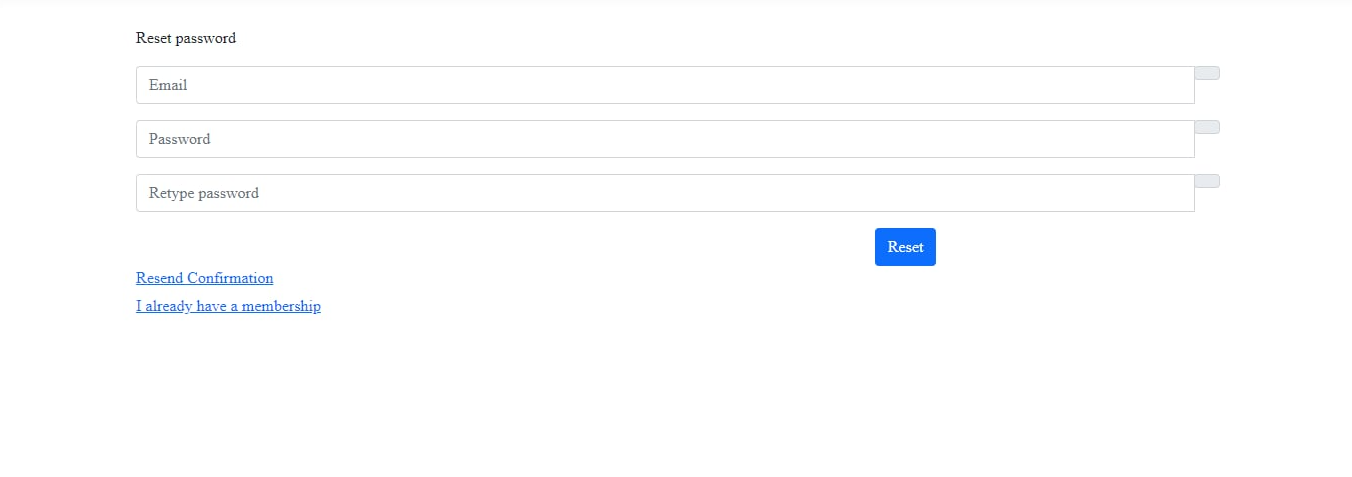
****Change password after recovery

Figure 7- Change Password

**Customer Profile Management**

Customer can edit, and manage their profiles, including personal information, Two Factor Authentication, and Change Password.

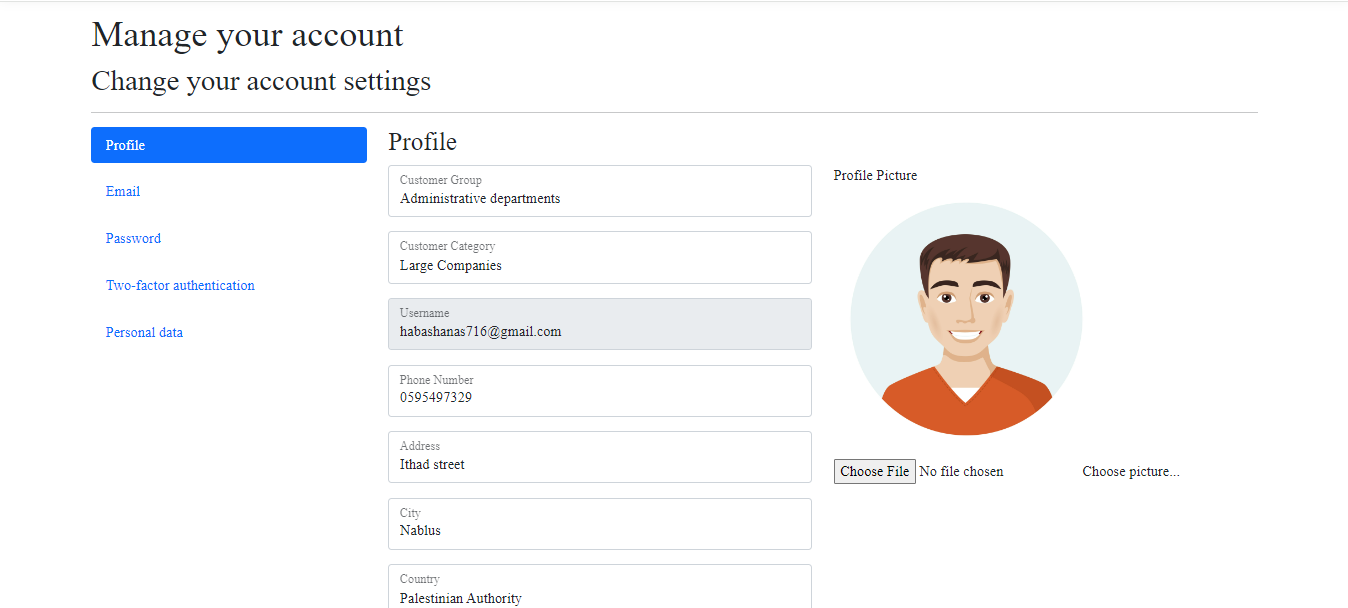


Figure 8- Edit Profile For Customer

**Customer Product Return Form**

The customer can return the goods after receiving them and provide a reason for the return.

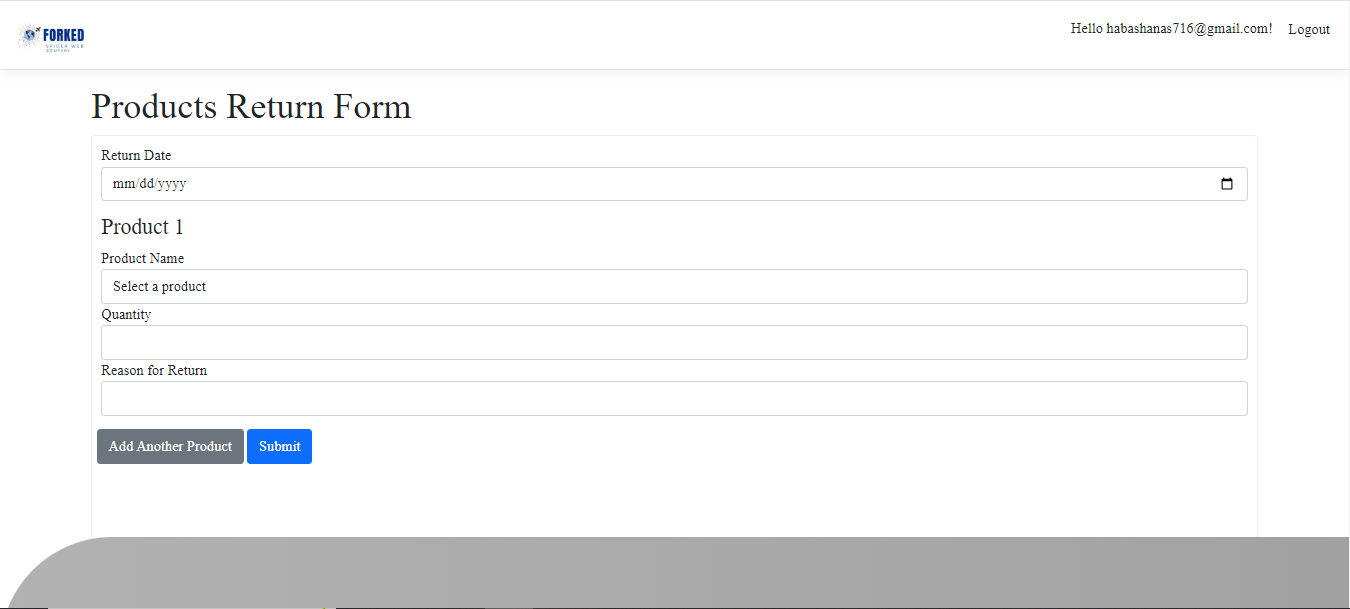
****

Figure 9- Return Product For Customer

**Customer Rating**

The customer can rate the product by clicking the rating icon. The rating is calculated by incrementing the total number of ratings and updating the average rating based on the new input

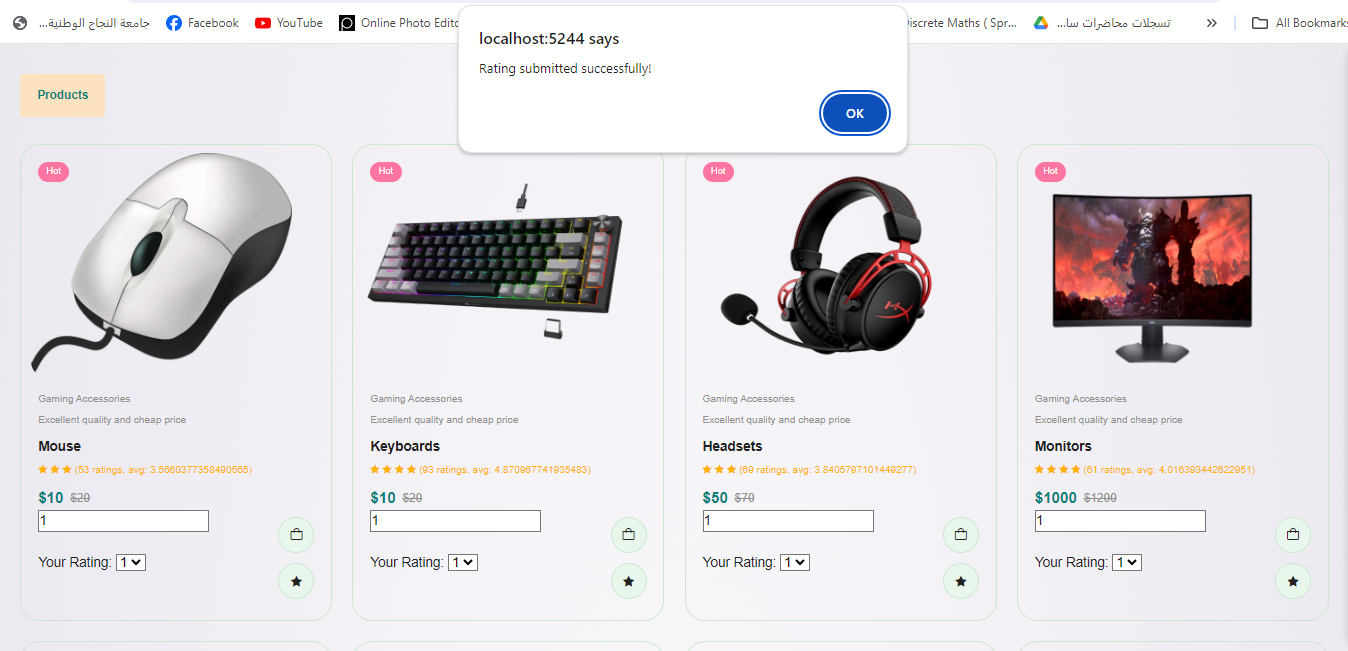
****

Figure 10- Customer Rating Product

**Add to Cart and Order Confirmation**

The customer adds a product to the cart by clicking the “Add to Cart “ icon. After confirming the order, the customer will receive an email stating that their order will be delivered as soon as possible.

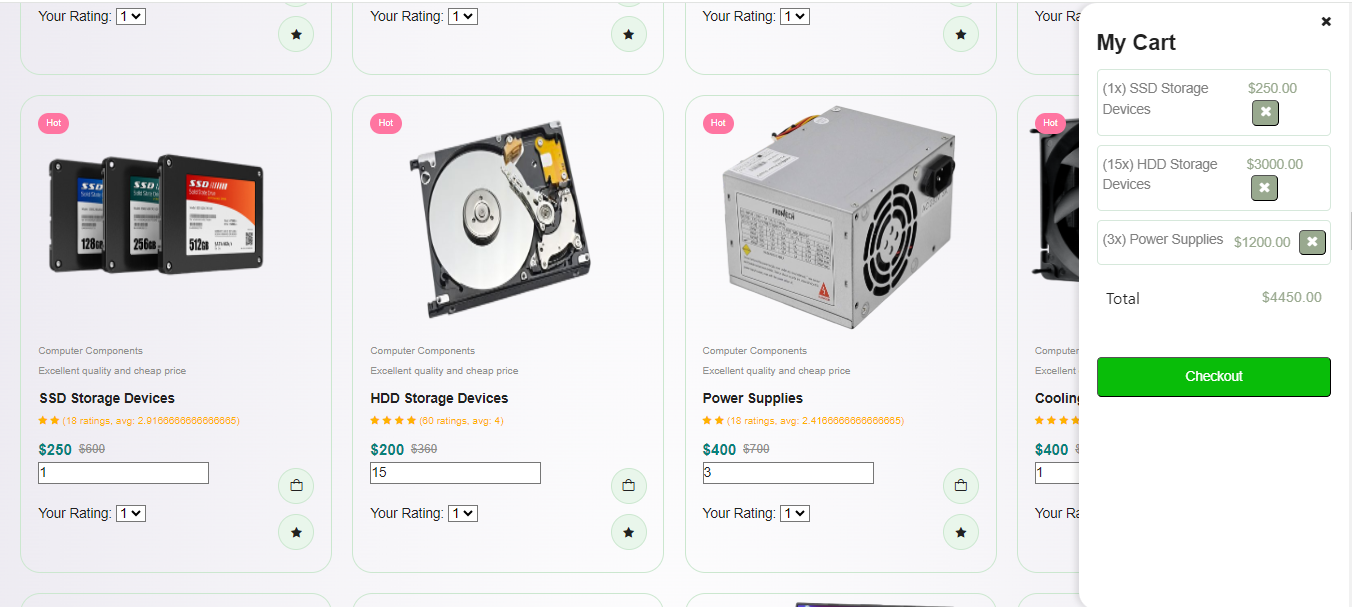


Figure 11- Add To Cart

**Product Search Feature**

The customer can search for a specific product and the system will display the matching product result.

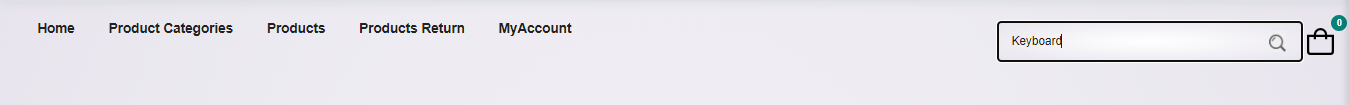


Figure 12- Search box

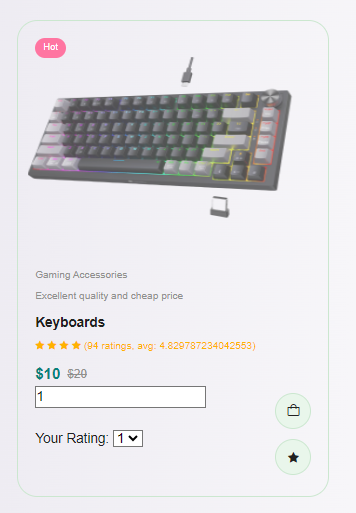


Figure 13- The element being researched

**Accounts management**

The admin's ability to see all accounts, roles and delete any of them

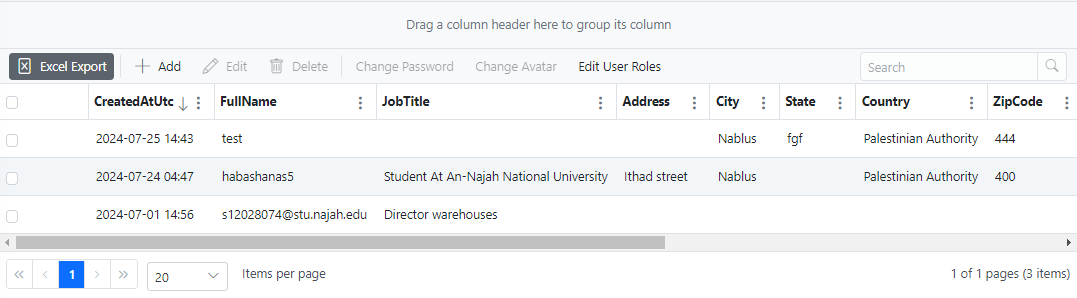


Figure 14- User List

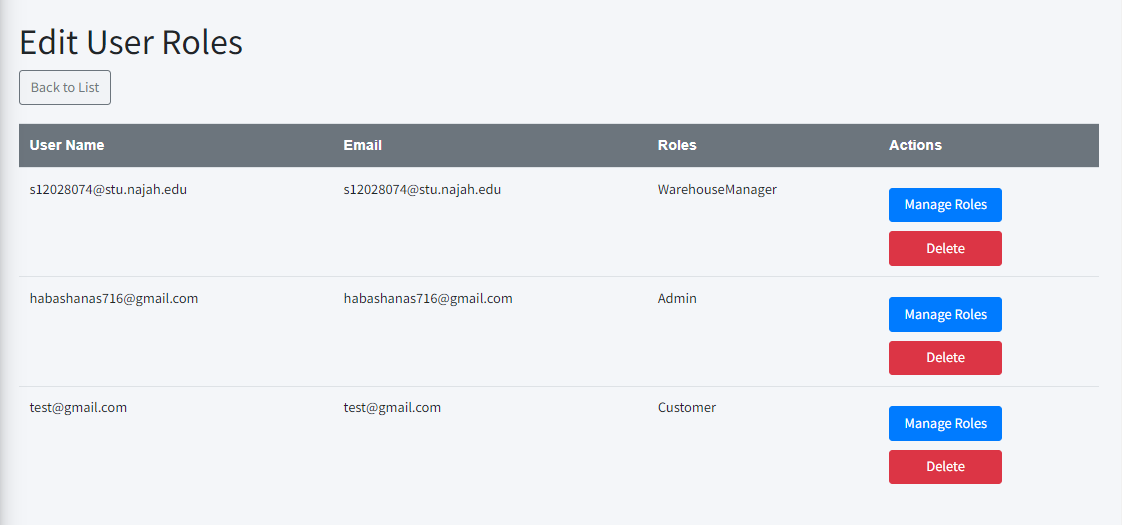


Figure 15- Users And Roles

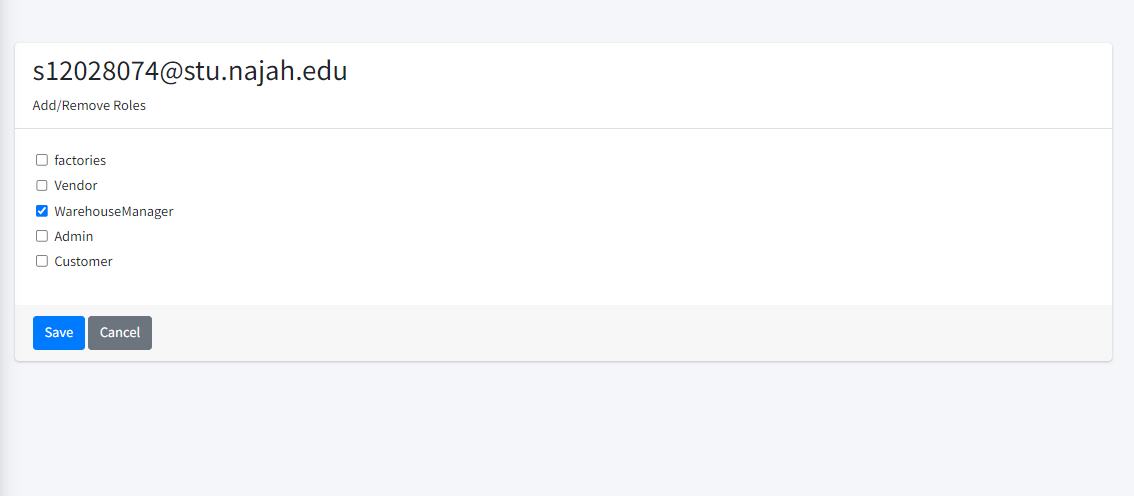


Figure 16- Change User Role

**Sales Order**

The Company will receive the sales order, which includes their customer information and order details.

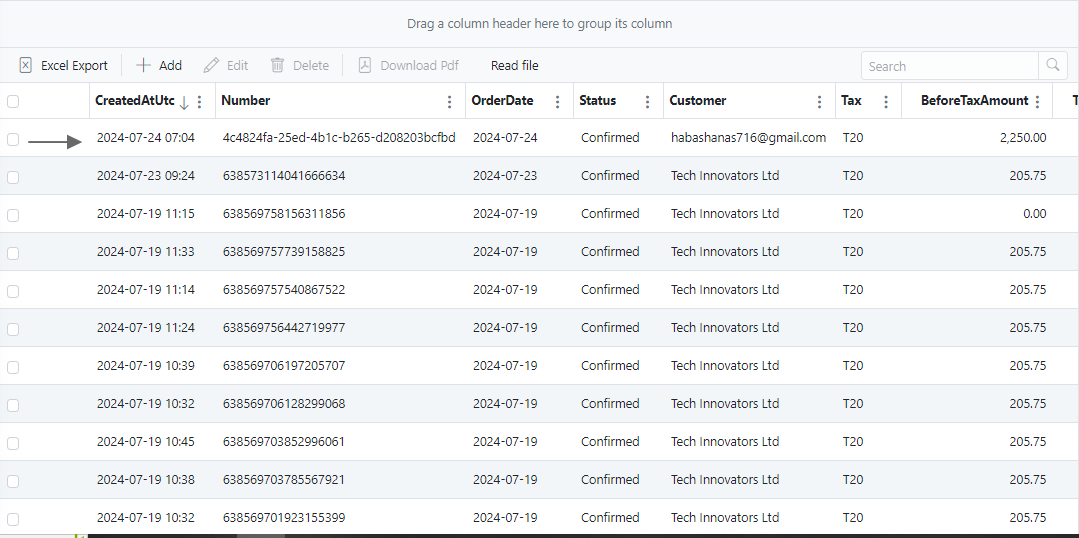


Figure 17- Sales Order

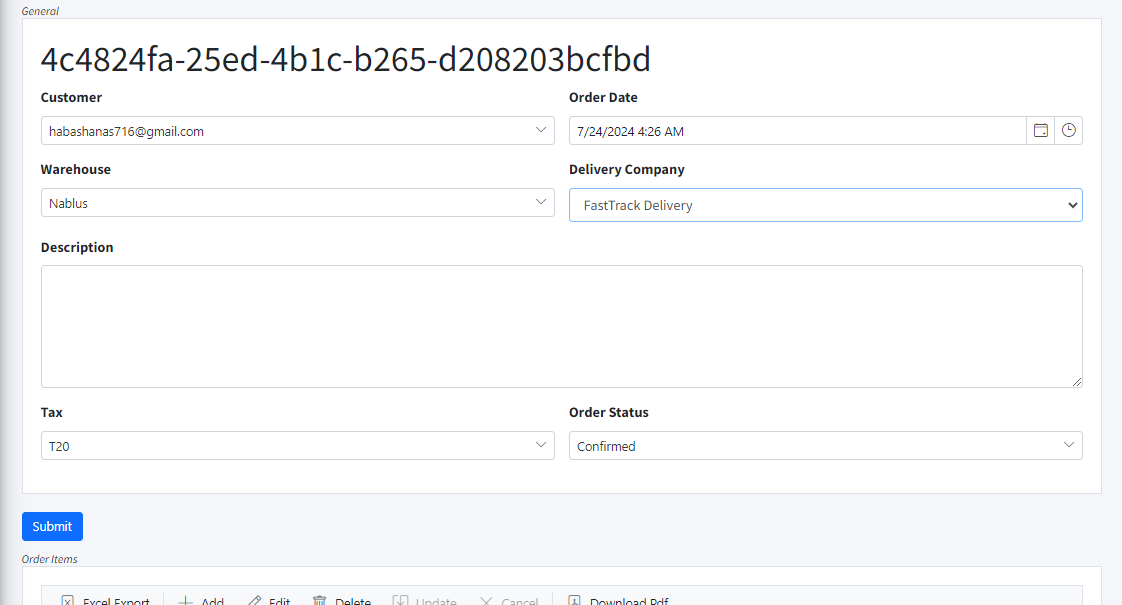


Figure 18- Sales Order Form

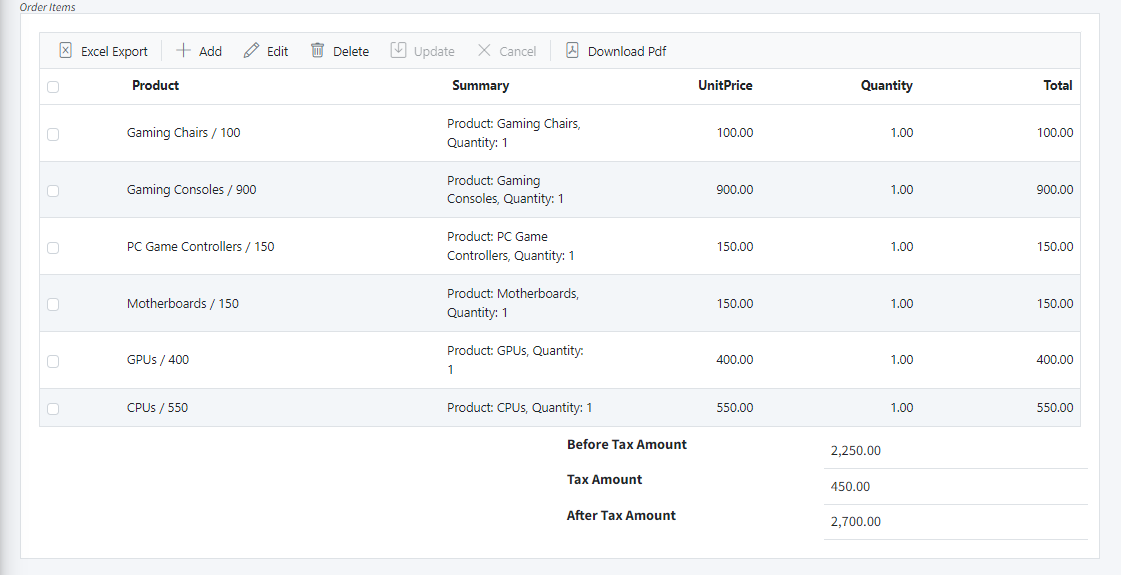
**Sales Order PDF View**

Figure 19- Order Items

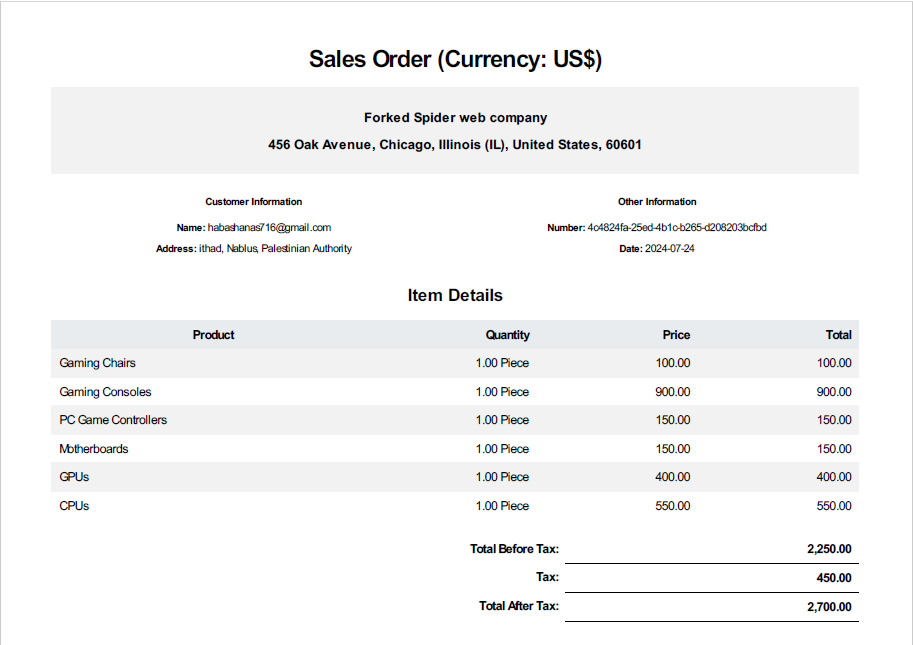
The company can open their sales order and view the ordered products through a PDF file

Figure 20- Sales Order Pdf

**Export Orders to Excel**

The company can download all their orders into an Excel file.

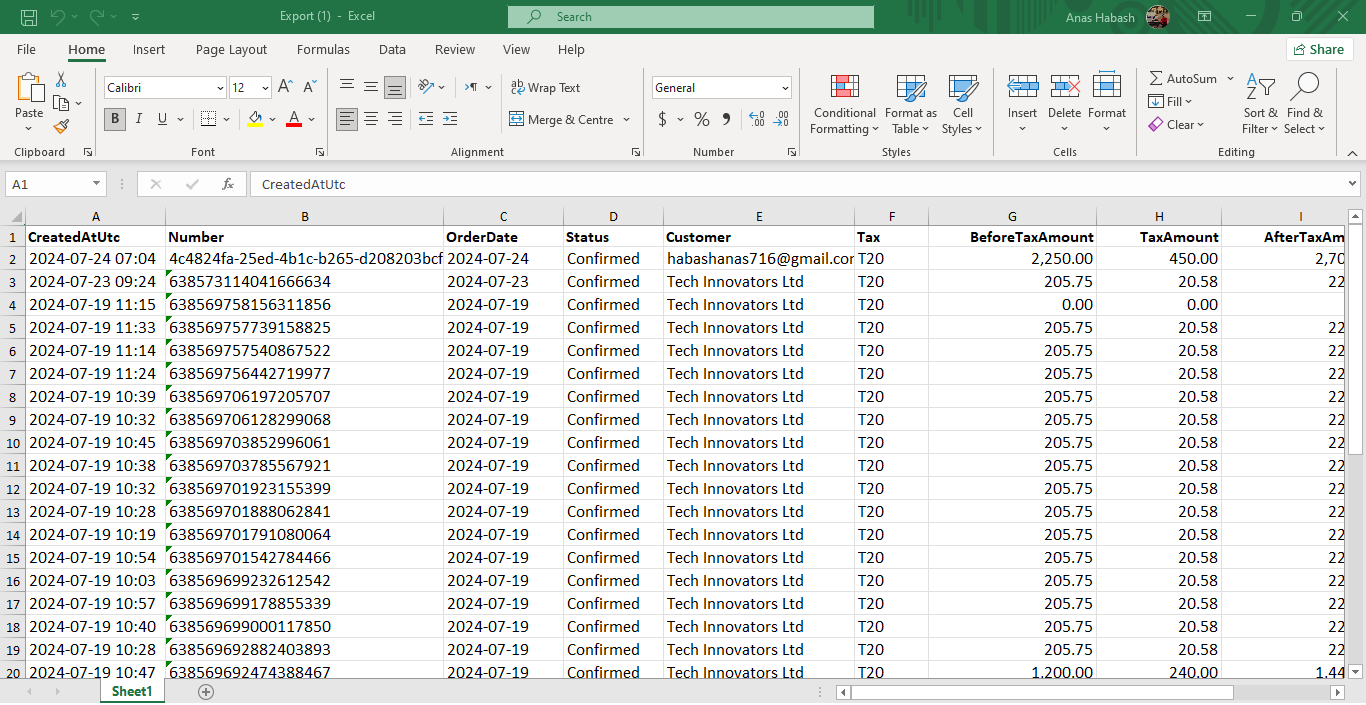


Figure 21- Convert Al Sales Orders To Excel File

**Create Delivery Order Request**

Fill out the following details to create a delivery order: **sales order, delivery date, order status**, **and** **delivery company.**

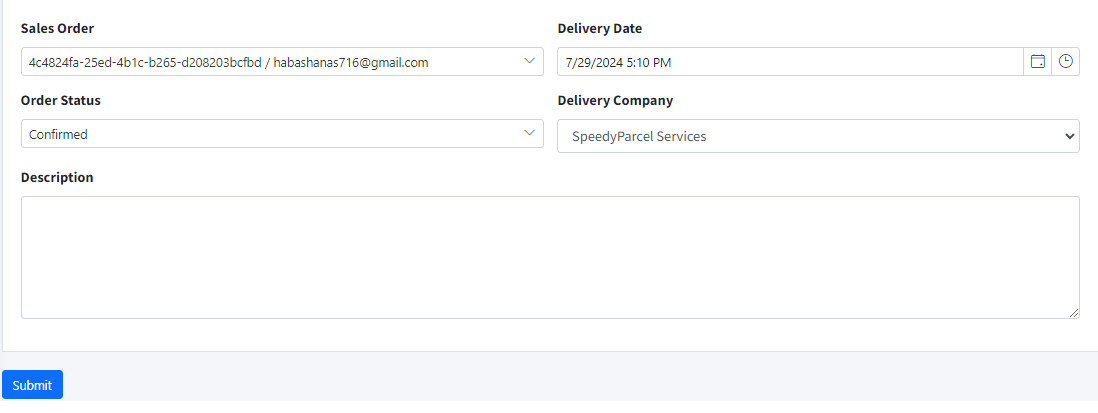


Figure 22- Delivery Order

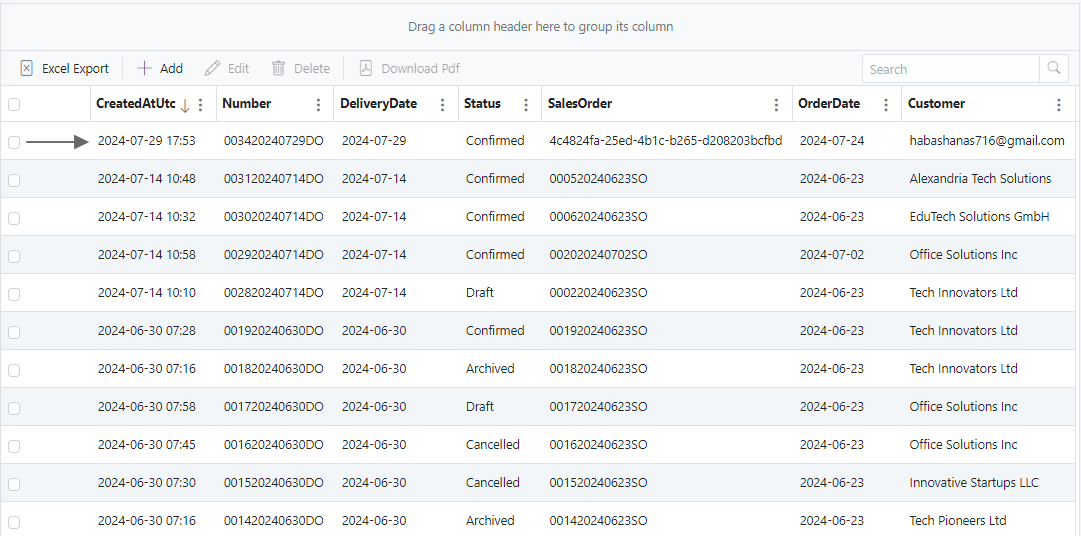


Figure 23- Delivery Order Added To The List

**Sales Return**

The customer can initiate a sales return after receiving orders

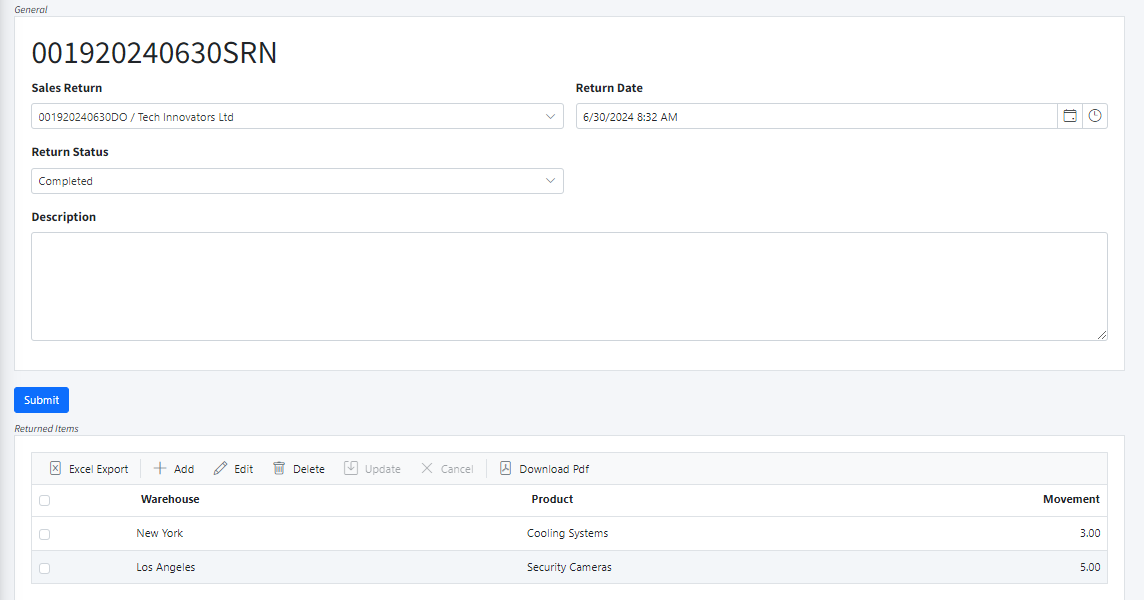


Figure 24- Sales Return

**Transfer Products Between Warehouses**

Specify the following details to transfer products between warehouses: **Source Warehouse, Destination Warehouse, Product Name, and Quantity.**

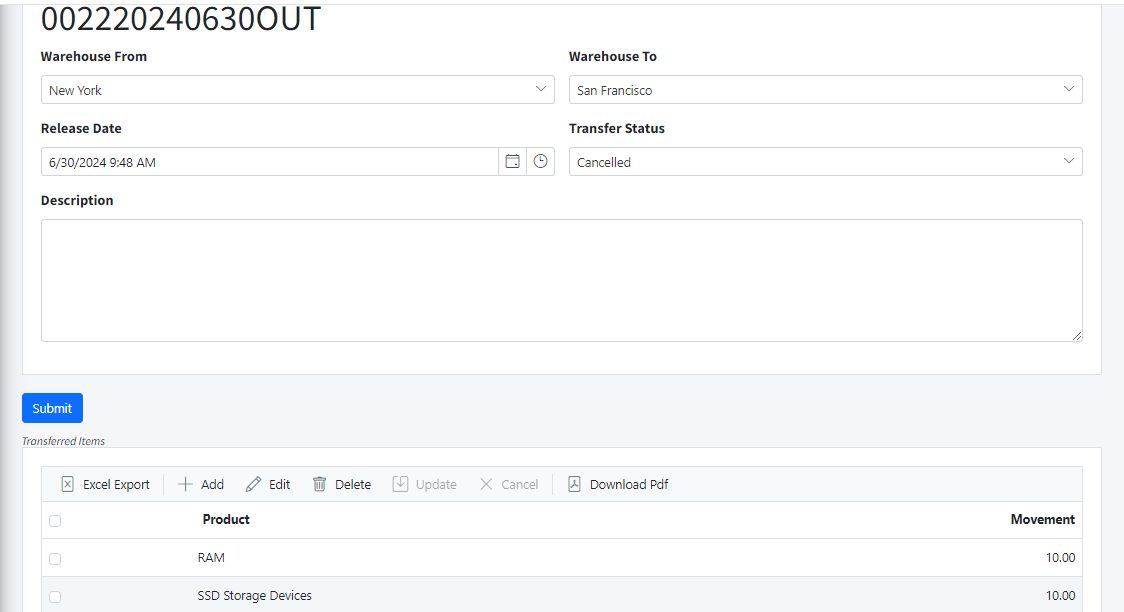
****

Figure 25-Transfer Out

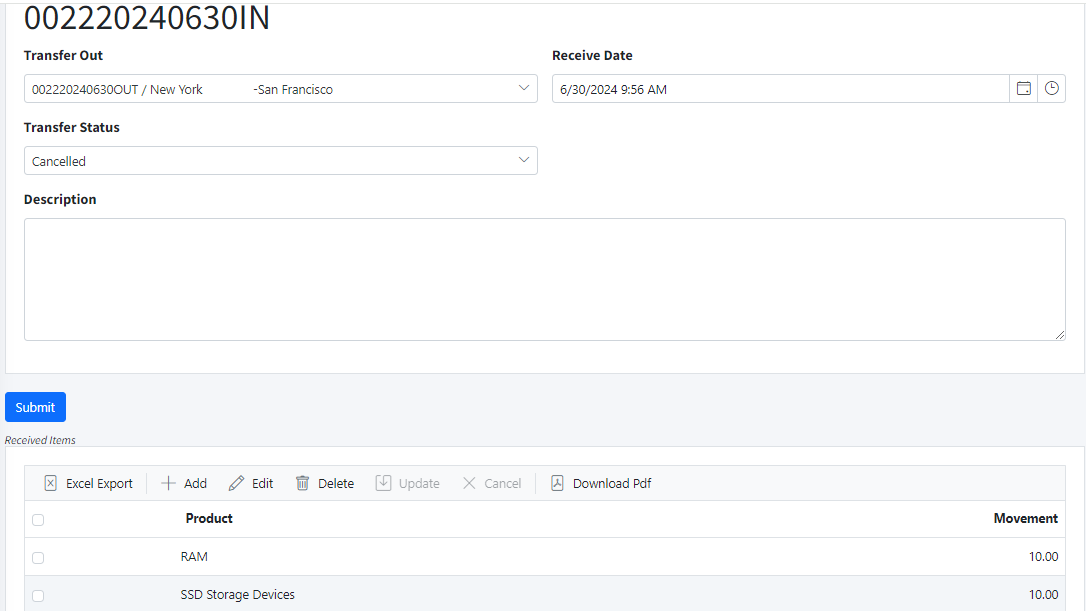


Figure 26- Transfer In

**Scrapping**

It is the warehouse manager’s responsibility to inspect the warehouses and dispose of any damaged goods.

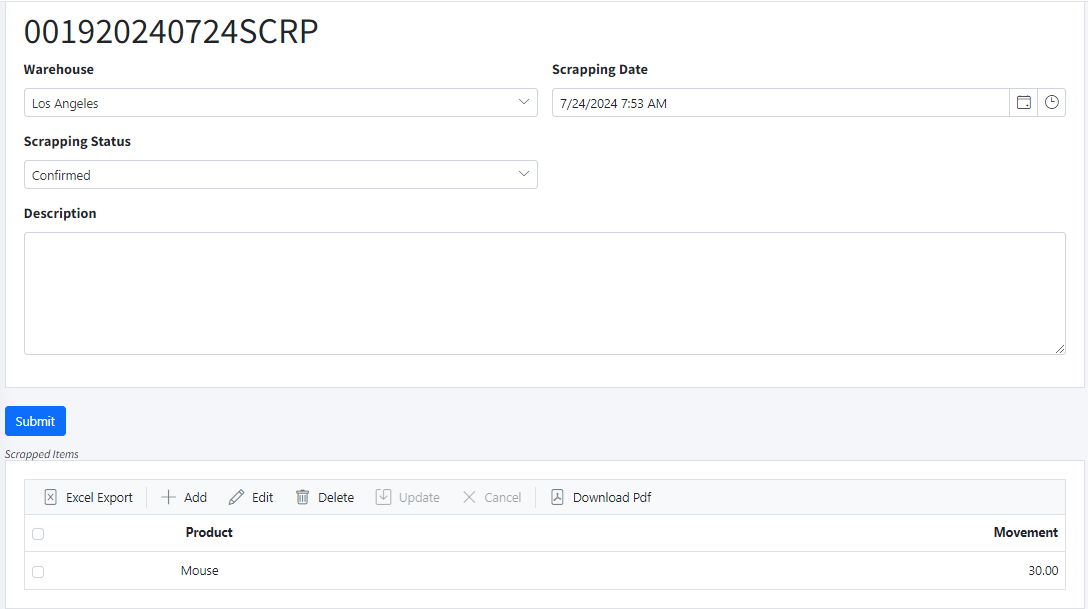
****

Figure 27- Scrapping

**Log Sessions**

Track user sessions by capturing the **username**, **IP address**, and the **action** performed.

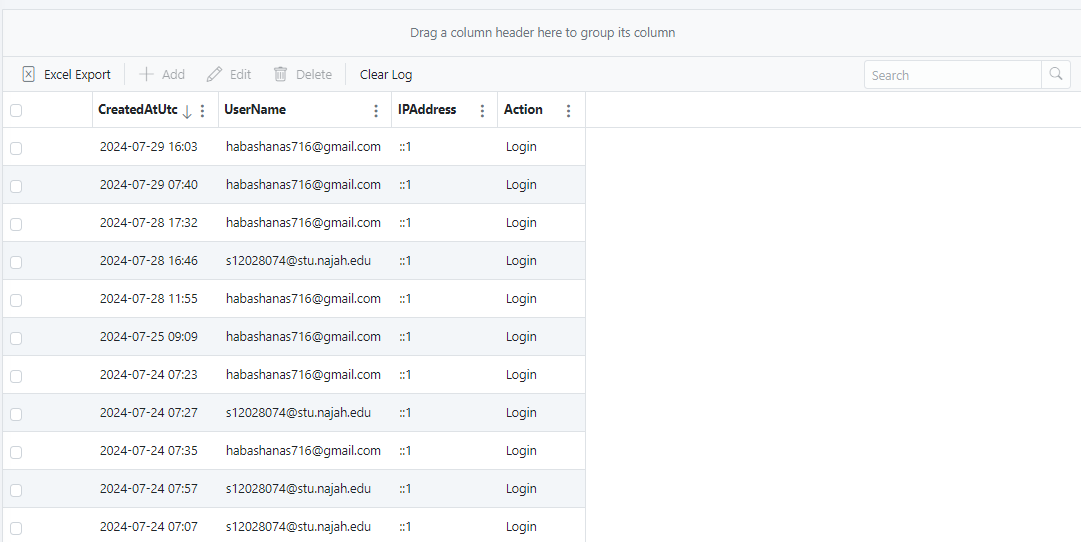


Figure 28- Log Sessions

**Stock Counts**

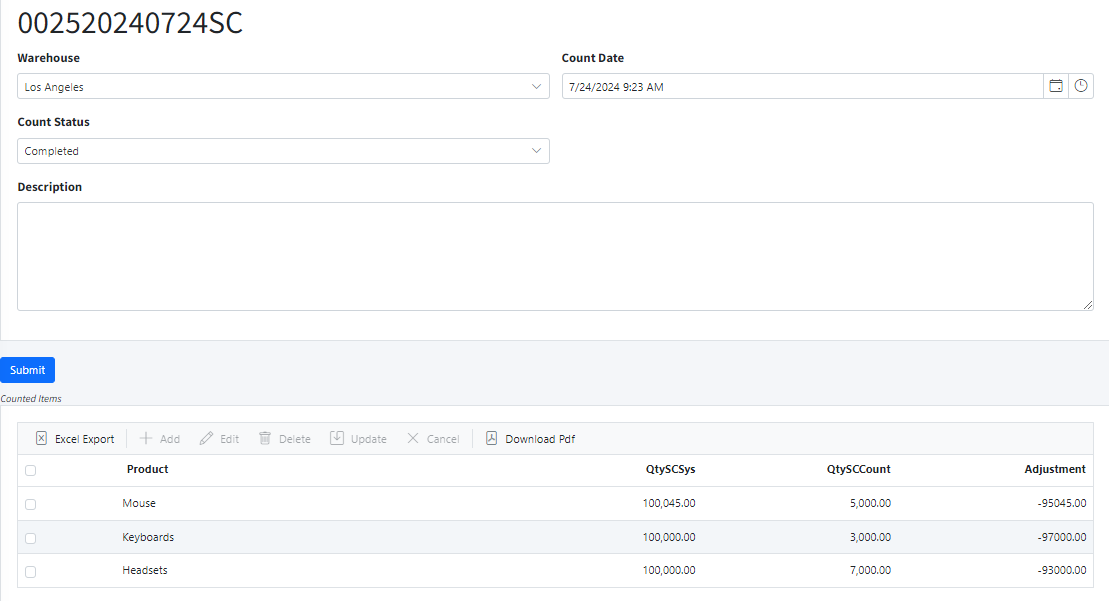
It is the responsibility of the warehouse manager to conduct stock counts. This involves verifying the quantity of goods in the warehouses by comparing the system records with the actual stock on hand, and identifying any discrepancies such as excess or shortage.

Figure 29- Stock Counts

**Log Analytics**

Analyze user interactions by logging the **username**, **URL**, **browser**, and **device**.

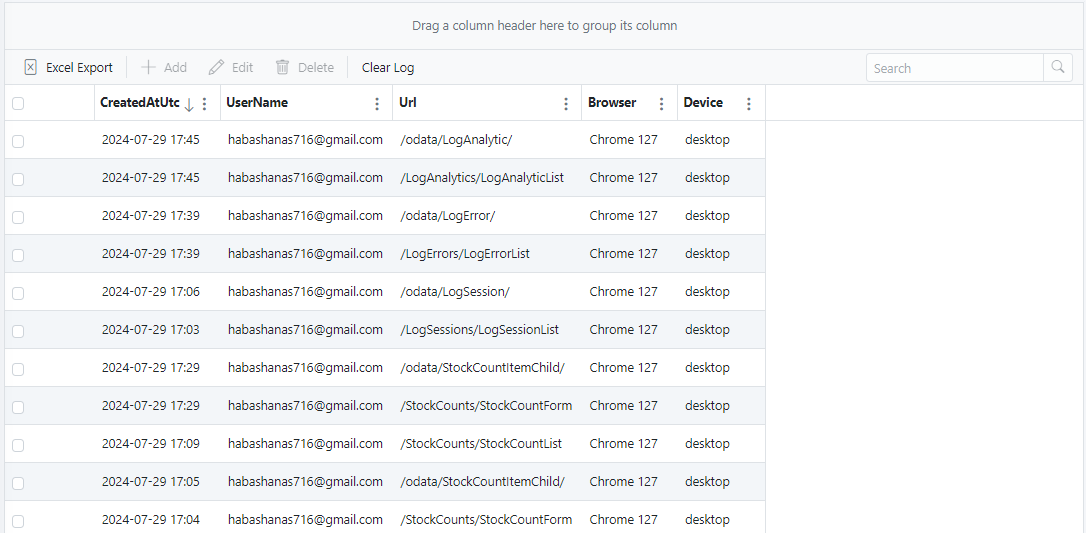
****

Figure 30- Log Analytics

**Log Errors**

Monitor system errors using the **exception message** and **stack trace**. This helps identify and resolve issues that users may encounter while using the system.

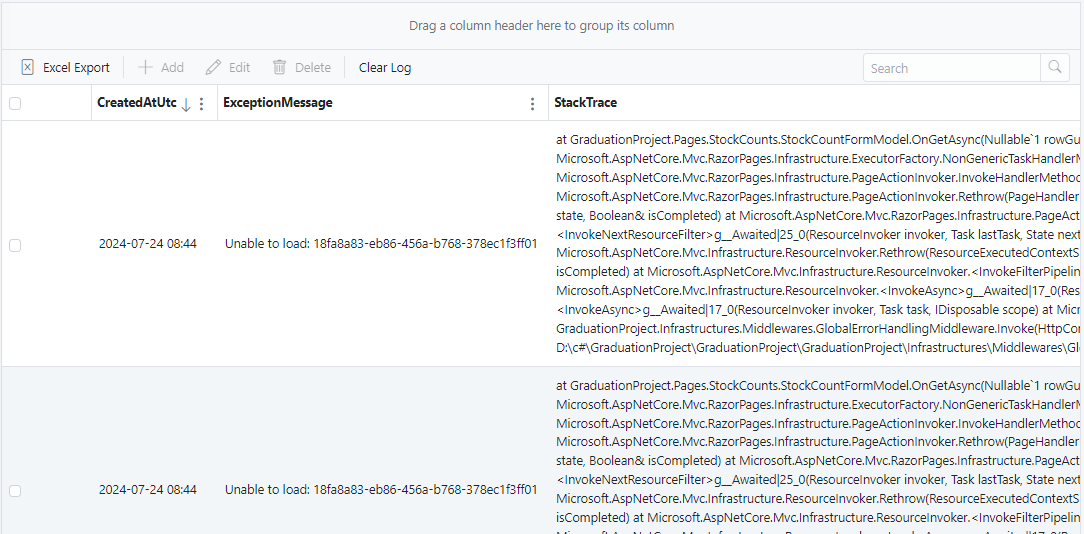


Figure 31- Log Errors

**Manufacturing Order**

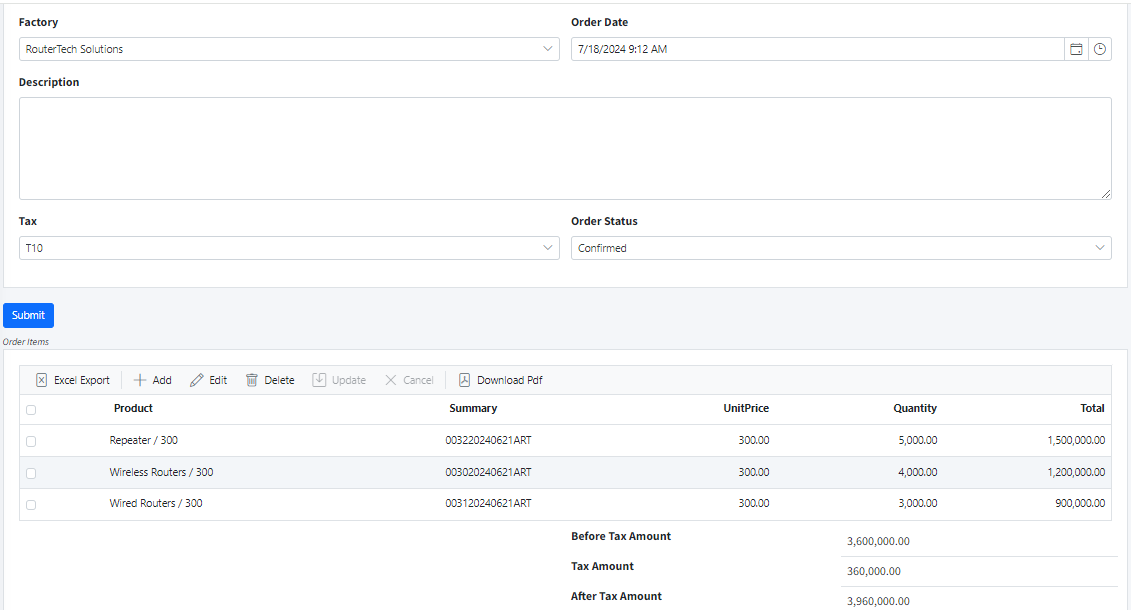
It is the responsibility of the warehouse manager to place orders with the manufacturer for the required products. 

Figure 32- Manufacturing Order

**Manufacturing Order PDF View**



Figure 33- Manufacturing Order PDF

**Export Manufacturing Orders to Excel**

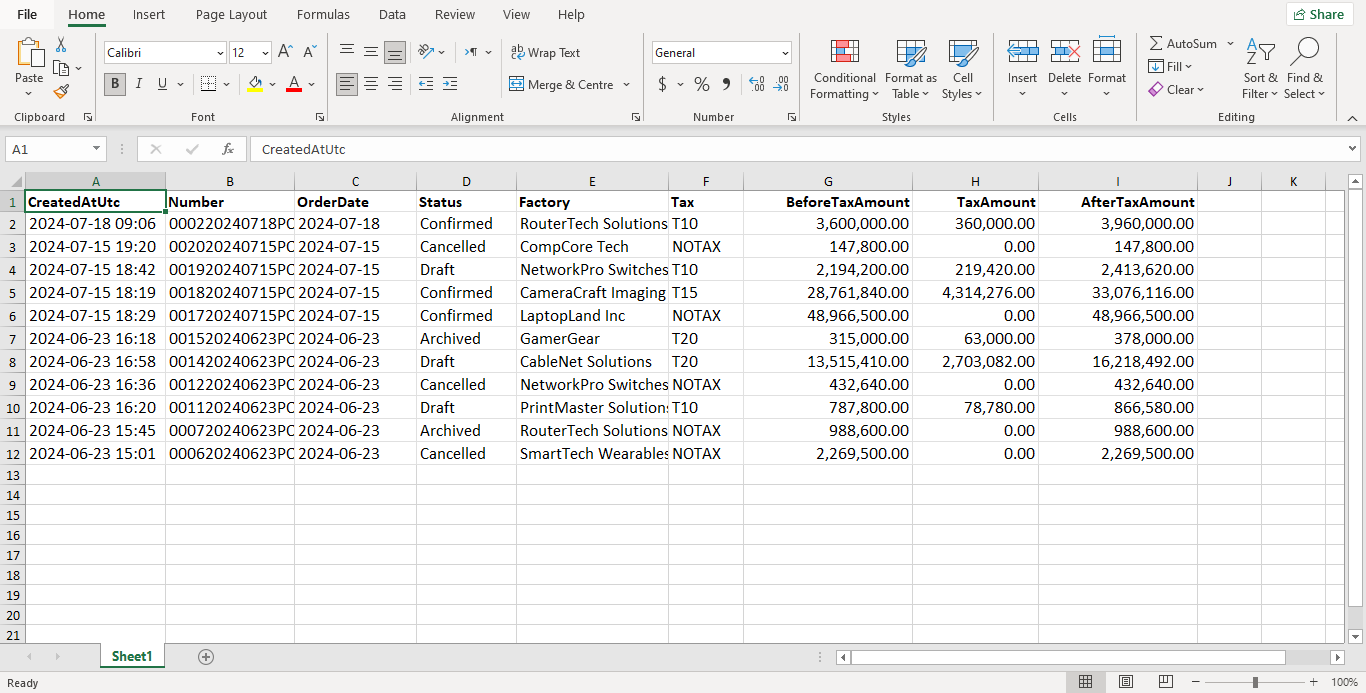


Figure 34- Convert All Manufacturing Orders To Excel File

**Receive Products From Factory**

When receiving products from the factory, the warehouse manager is responsible for **Receiving**, and **Storage**:

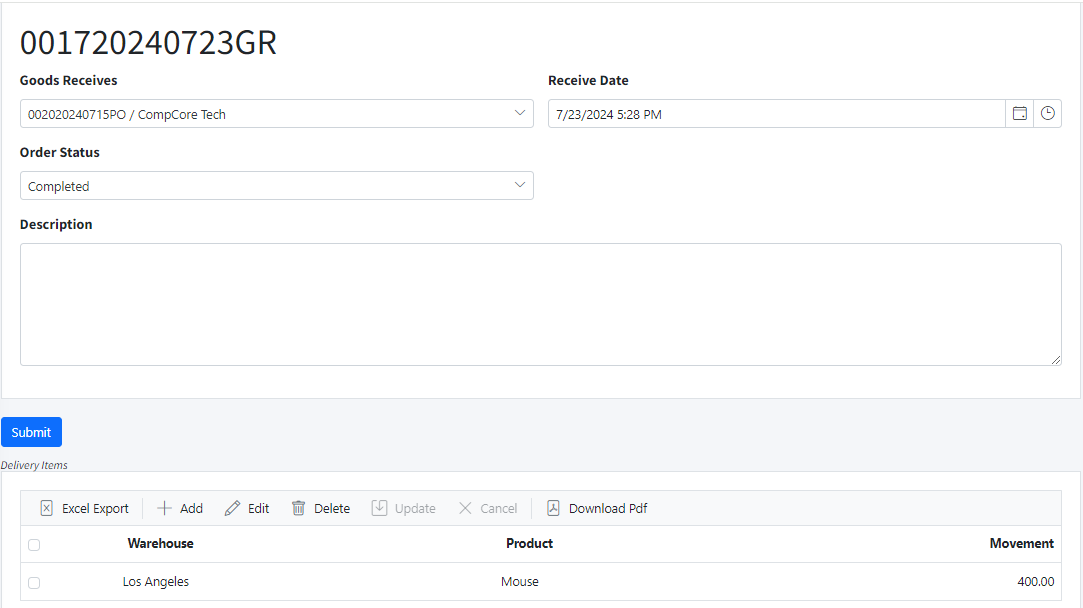
****

Figure 35- Receive Products From Factory

**Manufacturing Return Products**

the warehouse manager will:

* **Return Products**: Send the products back to the factory for reasons such as defects or incorrect items.
* **Reason for Return**: Provide a clear explanation for the return.

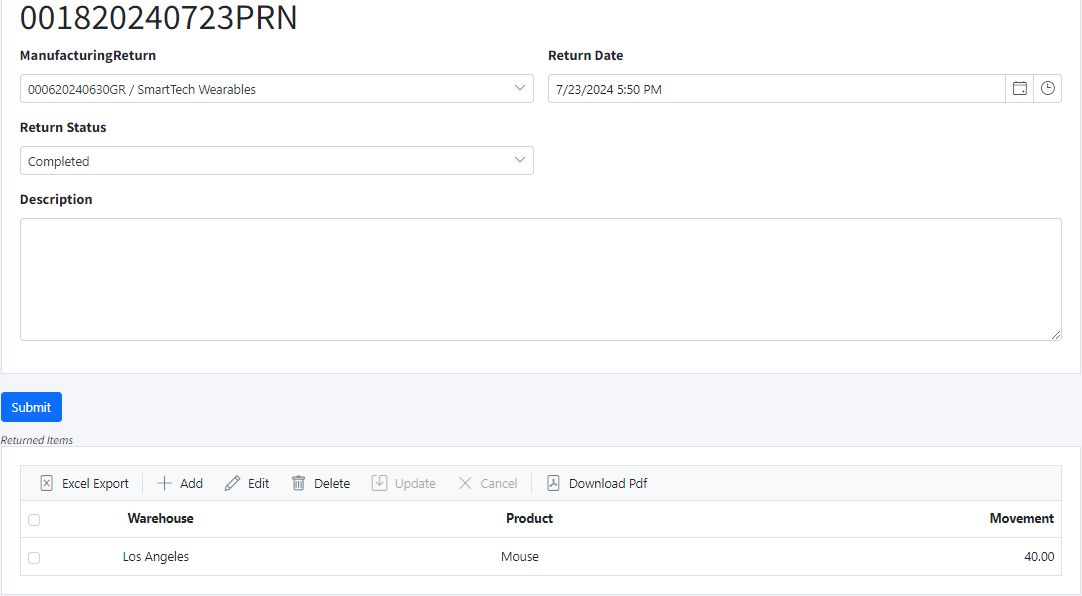
****

Figure 36- Manufacturing Return Products

**Warehouse Manager – Company Chat System**

A chat system will be implemented to facilitate communication between the warehouse manager and the company. This system will include:

* **Real-Time Messaging**: Enable instant communication for inquiries, updates, and coordination.
* **Message History**: Keep a record of past conversations for reference.
* **Attachments**: Allow sharing of documents and images related to warehouse operations and orders

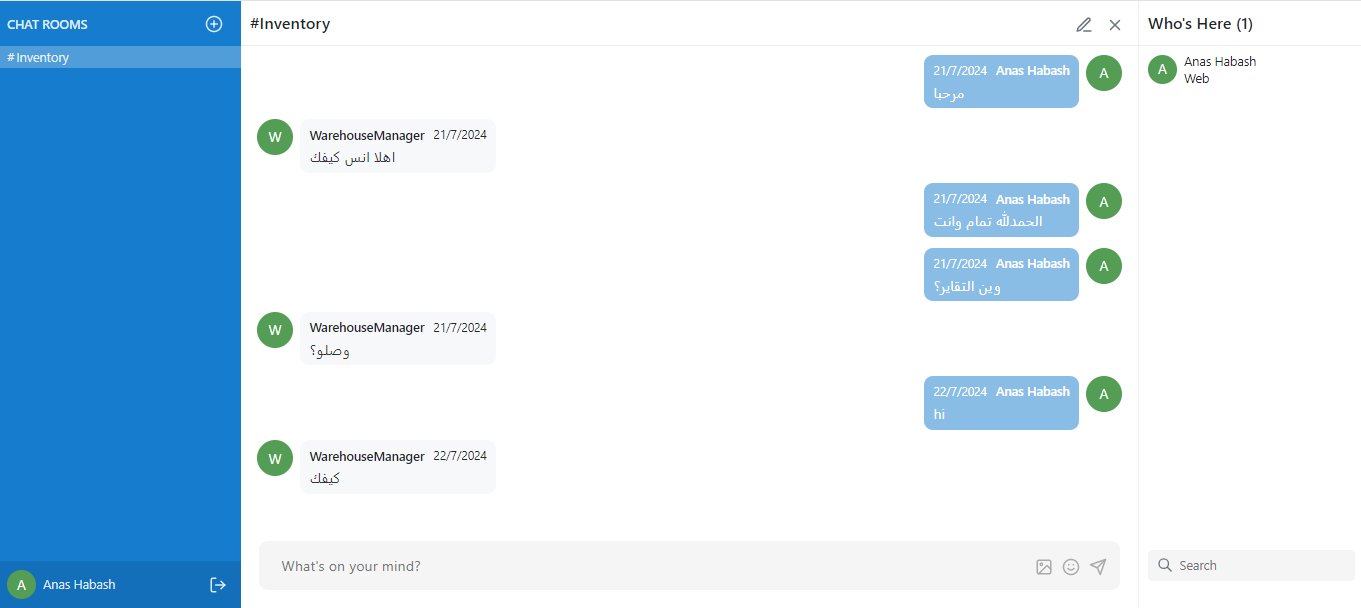


Figure 37- Warehouse Manager – Company Chat System

When you register an account, you will receive an email to confirm your email address.

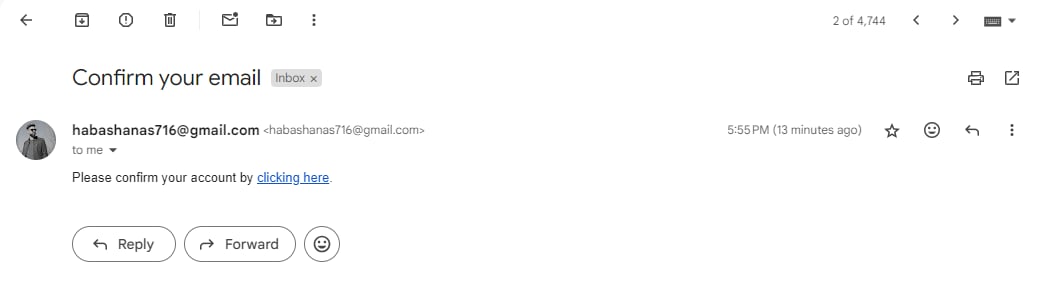
****

Figure 38- Confirm Email

When you forget your password, you will receive an email to reset your password.

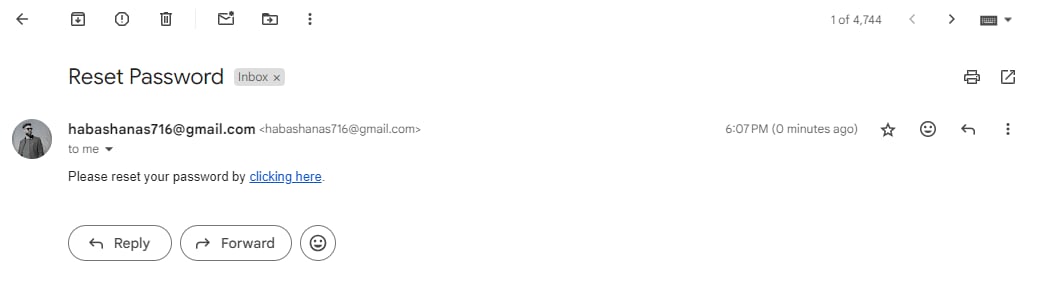
****

Figure 39- Reset Password

When a customer places an order, they will receive an email with a "New Order Confirmation."

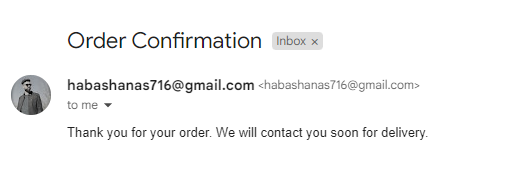
****

Figure 40- Order Confirmation

Additionally, the admin will be notified via email that an order has been placed.

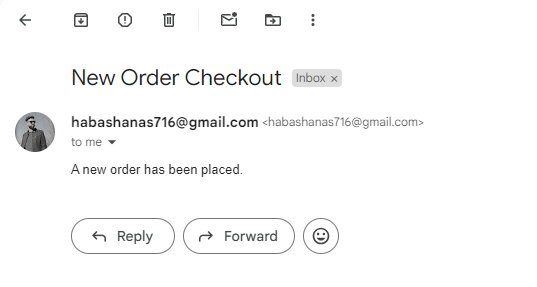
****

Figure 41- New Order

When a return is made, both the admin and the warehouse manager will receive an email notification that goods have been returned.

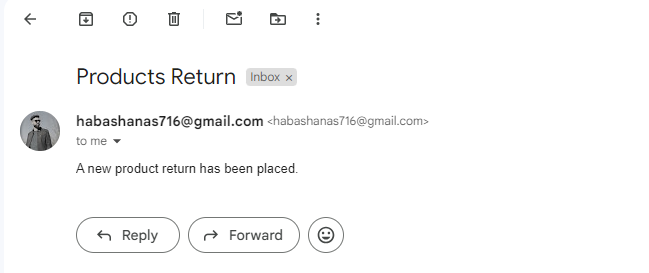
****

Figure 42- Email Products Return

When a new order is placed, the delivery company will receive an email with the subject "New Order".

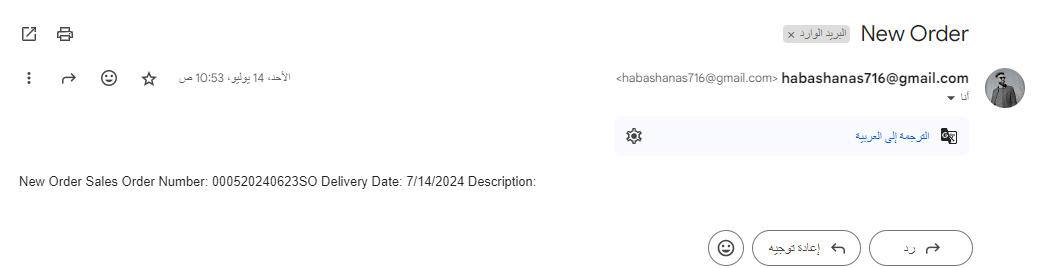
****

Figure 43- Email New Order For Delivery Company

When the difference between the number of sales and the number of manufactured units reaches 2000, an automated email will be sent to the warehouse manager and the company admin. This email will contain a warning about the significant discrepancy between sales orders and available stock, highlighting the need for immediate action to ensure smooth shipping and manufacturing operations.

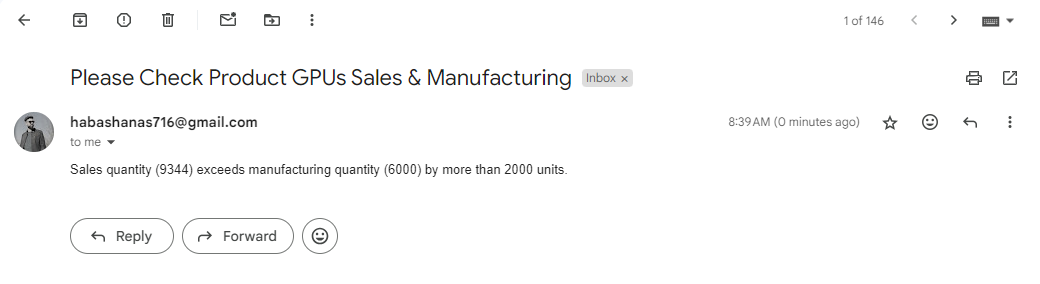


Figure - Warning Email

## Nonfunctional Requirements

**Performance Requirements**

* **Response Time:**

The system responds to user actions, such as loading pages, processing orders, and updating account details, within 2 seconds under normal operating conditions.

* **Scalability:**

Forked Spider Web Company is designed to scale seamlessly as the user base grows. The platform must handle increased data traffic and user interactions without significant degradation in performance.

* **Throughput:**

The platform supports up to 500 concurrent users without performance degradation, ensuring smooth and efficient operation during peak times.

* **Data Consistency:**

All transactions and data updates are processed reliably, ensuring data integrity across the system

* **Usability:**

The user interface is intuitive and easy to navigate, ensuring a positive user experience for customers, administrators, and warehouse managers.

* **Maintainability:**

The platform is designed with modular architecture and follows coding best practices to facilitate easy maintenance, updates, and enhancements.

**Security Requirements**

* **Account Registration Email Confirmation:**  
  When a user creates a new account, a confirmation email is sent to their registered email address to verify and activate their account.
* **Reset Password Email Verification:**

When a user forgets their password, a secure verification code is sent to their registered email address to confirm their identity.

* **Immediate Database Update:**

Once the user confirms the new password, the change is applied directly to the database, ensuring that the updated password is immediately effective.

**Safety Requirements**

Forked Spider Web Company safety requirements focus on ensuring user data protection, system integrity, and a secure user experience. Key safety considerations include:

* **Data Encryption:**

All sensitive user data, including login credentials and personal information, must be encrypted during transmission to and from the platform to prevent unauthorized access.

* **User Authentication:**

There is a strong user authentication mechanism to verify users' identity during login and account-related activities, preventing unauthorized access.

**Software Quality Attributes**

* **Reliability:**

Forked Spider Web Company operates reliably, ensuring consistent performance and minimal downtime. The system is handling a large number of concurrent users.

* **Maintainability:**

The platform's codebase well-organized and documented, facilitating ease of maintenance and future updates. Modular and scalable architecture contributes to maintainability.

* **Scalability:**

Forked Spider Web Company scale efficiently to accommodate an increasing user base and growing data volumes. Scalability is crucial to maintaining optimal performance as the platform expands

* **Performance:**

Our platform exhibits high performance, responding promptly to user interactions. Efficient database queries, optimized code, and effective use of resources contribute to overall performance

* **Security:**

Forked Spider Web Company uphold robust security measures to protect user data, prevent unauthorized access, and ensure the confidentiality and integrity of sensitive information.

* **Flexibility:**

The system flexible to accommodate future updates, feature additions, and changes in user requirements without significant disruption to the existing functionality.

**Logical Database Requirements**

Forked Spider Web Company logical database requirements include efficiently structuring and managing data to support platform functionality:

* **Chat**:
* **Room:** Store information about chat rooms, including Room ID, Room Name, and associated Users.
* **Message:** Store messages with details like Message ID, Room ID, Sender ID, Timestamp, and Message Content.
* **Cart:**
* **Cart Item:** Maintain details about items added to the cart, including CartItem ID, Cart ID, Product ID, Quantity, User ID, and Total Price.
* **Company**
* Store company profiles with details such as Company ID, Name, Currency, Email Address, Description, City, Phone Number, Street, Country, Website, Fax Number, and State.
* **Customer:**
* Maintain customer profiles with information like Customer ID, Name, Street, City, State, Zip Code, Country, Phone Number, Email Address, Website, Customer group, and Customer Category.
* **Delivery Order:**
* Store delivery orders with details like Delivery Order ID, Delivery Date, Status, Description, Sales Order ID, and Delivery Company ID.
* **Delivery Company:**
  + Store delivery company information with details like Delivery Company ID, Name, Contact Information, Email, Address, City, Country, Lat, and Lng.
* **Number Sequence:**
* Maintain sequences for various entities with details like Sequence ID, Entity Name, Current Number, Suffix, and Prefix.
* **Product:**
* Store product information with details like Product ID, Name, Description, Unit Price, Physical, Unit Measure, Product Group, Meta Keyword, rating, old Price, Rating Average, and Image.
* **Manufacturing Order:**- Track manufacturing orders with details like Manufacturing Order ID, Description, Factory ID, Status, Order Date, Tax, Before Tax Amount, Tax Amount, and After-Tax Amount.
* **Sales Order:**
* Store sales orders with details such as Sales Order ID, Customer ID Status, Order Date, Tax, Before Tax Amount, Tax Amount, and After-Tax Amount.
* **Sales Summary by Day:** Maintain daily sales summaries with details like Summary ID, Date, Total Sales Amount, Product Name, and Total Items Sold.
* **Sales Return:**
* Maintain records of sales returns with details like Sales Return ID, Sales Order ID, Customer ID, Return Date, Return Reason, and associated Product IDs.
* **Unit Measure:**
* Store unit measures with details like Unit Measure ID, Name, and Description.
* **Warehouse:**
* Maintain warehouse details with information like Warehouse ID, Name, Description, Phone Number, Address, City, Postal Code, State , Lat ,and Lng.
* **Factories:**
* Maintain details about factories with information such as Factory ID, Name, Description, Street, City, Sate, Zip code, Country, Phone Number, Fax Number, Email Address, Website, Manufacturing type, Manufacturing Classification, and Rating.
* **Goods Receive:**
* Maintain records of goods received, including Goods Receive ID, Received Date, Status, Description, and Manufacturing Order ID.
* **Manufacturing Return:**
* Maintain records of manufacturing returns with details like Manufacturing Return ID, Manufacturing Order ID, Product ID, Quantity, Return Reason, and Return Date.
* **Tax:**
* Manage tax information with details like Tax ID, Tax Name, Description, and Percentage.
* **Log:**
* **Log Error:** Store error logs with details like Log ID, Error Message, Stack Trace, and Additional info.
* **Log Analytic:** Track analytic logs with details like Log ID, User ID, User Name, IP Address, URL, Device, and Browser,
* **Log Session:** Record session logs with details like Session ID, User ID, IP Address, and Action.
* **Inventory Transaction:**
* Log inventory transactions with details like Transaction ID, Module ID, Movement Date, Status, Warehouse ID, Product ID, Movement, Stock, QtySCSys, QtySCCount, and Transaction Type (e.g., In, Out, Transfer, Scrapping, Stock Count).

**System Models**