NEXORA
MULTI-VENDOR ONLINE MARKETPLACE

SPRING - 2025

SAIM ZAFAR	**01-135241-045**
MUHAMMAD IBRAHIM	**01-135241-035**
ZARISH FATIMA	**01-135241-052**

DEPARTMENT OF COMPUTER SCIENCE
BAHRIA UNIVERSITY, ISLAMABAD

Table of Contents

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Project Overview

Nexora is a web-based platform that allows multiple sellers to list and sell their products, while customers can browse, compare, and purchase them. Unlike traditional e-commerce stores, this system enables different vendors to register on a single platform, manage their own inventory, and interact with customers. The project focuses on simplifying online shopping by providing efficient, secure, and user-friendly system for both sellers and buyers.

This project is highly relevant in today's world, where online shopping is major industry. Many small businesses struggle to create their own e-commerce websites due to high development costs and technical difficulties. A multi-vendor system solves this problem by giving them a shared platform, reducing their operational costs, and increasing their visibility.

1.1.1 Key Features of the System

1. Vendor Registration & Management – Allows sellers to create accounts, add products, and manage their store.
2. Product Listing & Search – Customers can browse products by category, search for specific items, and view details.
3. Shopping Cart & Checkout – Buyers can add products to their cart and complete secure transactions.
4. Order Management – Vendors can track orders, update delivery statuses, and manage returns.
5. Admin Dashboard – A control panel for managing users, reviewing product listings, and monitoring platform performance.
6. Customer Reviews & Ratings – Buyers can provide feedback on products to ensure quality control.

1.1.2 Importance of the Project

- Supports Small Businesses – Provides vendors with a way to sell online without the need to build separate websites.
- Encourages Competition – Customers get better prices and more variety from different sellers.
- Enhances Shopping Experience – A well-organized system makes online shopping more efficient and convenient.
- Scalability – The system can be expanded to support additional vendors, payment options, and features like discount coupons or analytics.

1.2 Problem Statement

Online shopping has become an essential part of modern life, but many small businesses face challenges in creating their digital presence. Creating a separate e-commerce website requires significant investment in development, maintenance, and marketing, which is not sufficient for many vendors. Additionally, customers often have difficulty finding multiple sellers in one place, leading to higher costs and limited product choices.

Existing online marketplaces such as Amazon and eBay provide a solution, but they charge high commission fees and impose strict policies on sellers, making it difficult for small businesses to compete. Moreover, some marketplaces lack proper order management, transparent review systems, and flexible vendor controls, which affects the overall user experience.

This project aims to solve these issues by developing Nexora, allowing vendors to list products, manage their store, and process customer orders efficiently. The system provides a cost-effective and easy-to-use platform, ensuring a better shopping experience for customers and a clear business opportunity for vendors.

1.3 Objectives

The main objective of this project is to develop Nexora that enables multiple sellers to list and sell their products while providing customers with a seamless shopping experience, hence the system is designed to be efficient, user-friendly, and scalable, ensuring that vendors can easily manage their stores and customers can browse, compare, and purchase products with ease.

1.3.1 General Objectives

The project aims to:

- Develop a fully functional e-commerce platform that supports multiple vendors.
- Ensure a secure and reliable shopping experience for customers.
- Provide a centralized system where vendors can manage their products, track orders, and interact with buyers.
- Enable buyers to compare products and choose from multiple sellers, improving competition and pricing.
- Implement an insightful user interface for both vendors and customers.

1.3.2 Specific Objectives

To achieve these goals, the following specific objectives are defined:

For Vendors:

1. Provide a vendor registration and authentication system that allows businesses to sign up and manage their accounts securely.
2. Develop a product management system where vendors can add, edit, and delete product listings.
3. Implement an order management module that allows vendors to track orders, update status, and handle customer queries.
4. Enable sales reports and analytics to help vendors track performance and improve their business strategies.

For Customers:

1. Create a user-friendly product browsing and search system that allows customers to find relevant products quickly.
2. Implement a shopping cart and checkout system for a smooth purchasing experience.
3. Allow customers to track their orders and receive notifications regarding order status updates.
4. Develop a customer review and rating system to ensure transparency and trust in the marketplace.

For Admins:

1. Provide an admin panel for managing vendors, reviewing product listings, and ensuring agreement with marketplace policies.
2. Implement fraud detection mechanisms to prevent unethical selling practices.
3. Monitor customer feedback and complaints to improve platform reliability.
4. Generate detailed reports on marketplace activity, sales trends, and vendor performance.

1.3.3 Expected Outcomes

By achieving the above objectives, the project is expected to:

- Reduce technical and financial barriers for small businesses wanting to sell online.
- Improve customer satisfaction by offering multiple vendor choices and competitive pricing.
- Provide that can be expanded with additional features in the future.
- Enhance security and trust through a proper vendor verification and customer review system.

1.4 Scope of the Project

Nexora will offer essential features such as vendor registration, product management, order tracking, and secure payment processing. The scope of this project defines what will be covered and what is beyond the current implementation.

1.4.1 Features Covered in the Project

Vendor Features

1. Vendor Registration & Authentication – Secure sign-up and login for sellers.
2. Product Management – Vendors can add, update, and remove product listings.
3. Order Management – Vendors can process orders, update status, and manage deliveries.
4. Sales Reports – Vendors can access performance analytics and transaction history.

Customer Features

1. Product Browsing & Search – Customers can view products, and search for specific items.
2. Shopping Cart & Checkout – A system for adding items to the cart and completing secure purchases.
3. Order Tracking – Customers can track the status of their purchased items in real time.
4. Review & Rating System – Customers can provide feedback on products and sellers.

Admin Features

1. User & Vendor Management – Admins could approve or block vendors and customers.
2. Product Review & Approval – Admins will be able to monitor product listings for compliance.
3. Sales & Performance Reports – Admins can generate reports on transactions, vendors, and sales trends.
4. Security & Fraud Prevention – Admins would be able to monitor suspicious activity and enforce marketplace policies.

1.4.2 Limitations of the Project

While the project aims to provide an e-commerce platform, certain features are beyond the current implementation due to time and resource constraints. The following limitations apply:

- The system will not include AI-based product recommendations at this stage.
- Logistics and shipping integration (such as automated tracking with courier services) will not be included.
- The project will not include mobile applications, but the web platform will be mobile-responsive.
- Vendor subscription plans or commission-based pricing models will not be implemented.

1.4.3 Future Enhancements

Although these features are not included currently, the system is designed to be scalable, allowing for future improvements. Potential enhancements include:

- AI-driven product recommendations to improve customer experience.
- Integration with multiple payment gateways to support various transaction methods.
- Automated inventory restocking alerts to help vendors manage stock levels efficiently.
- Mobile application development for Android and iOS to expand accessibility.

1.5 Technologies & Tools Used

The development of Nexora requires a number of technologies for frontend, backend, database management, and deployment. The selected tech stack ensures that the system is secure, and user-friendly while maintaining efficient performance.

1.5.1 Frontend Technologies

- JS – Used for building an eye-catching and responsive user interface.
- HTML5 & CSS3 – Used for structuring and styling web pages. (used for sketching currently).

1.5.2 Backend Technologies

- Node.js– Used for handling backend logic, API requests, and managing database interactions.
- RESTful APIs – Used for data exchange between frontend and backend.

1.5.3 Database Management

- MySQL – A relational database system for storing structured data such as user accounts, products, and orders.

1.5.4 Authentication & Security (optional)

- bcrypt.js – Used for password hashing to enhance security.

1.5.5 Payment Integration (optional)

- Stripe API – Used for secure online transactions between customers and vendors.

1.5.6 Development & Deployment Tools

- Visual Studio Code (VS Code) and/or Notepad – Used as the primary code editor for development.
- Git & GitHub – Used for version control and collaboration among team members.
- Render – Used for backend hosting to deploy the Node.js API.
- Netlify / Vercel – Used for frontend deployment.

1.6 Work Distribution Among Team Members

To complete the project efficiently, the tasks are divided among the three team members: Saim, Ibrahim, and Zarish. The work is distributed so that each member focuses on specific areas, covering requirements gathering, system design, development, testing, and deployment.

1.6.1 Work Breakdown Structure

The following table shows how the tasks are divided among the team members.

Requirements & Planning

- Gathering Requirements – Zarish
- Writing Functional & Non-Functional Requirements – Ibrahim
- Use Case Writing & System Flow – Saim

System Design

- Database & Logical Design – Saim, Ibrahim
- System Architecture – Zarish, Ibrahim
- UI/UX Design – Saim

Development

- Backend (APIs, Authentication, Vendor Management) – Ibrahim, Saim(minimally)
- Frontend (Dashboard, User Interface) – Zarish, Saim

Testing & Quality Assurance

- Writing Test Cases – Zarish
- Debugging & Fixing Issues – Saim, Ibrahim
- Final Testing & Improvements – Ibrahim, Zarish

Deployment & Documentation

- Backend & Database Deployment – Ibrahim
- Frontend Deployment – Zarish
- Final Report & Documentation – Saim

1.6.2 Work Breakdown Structure Visualization

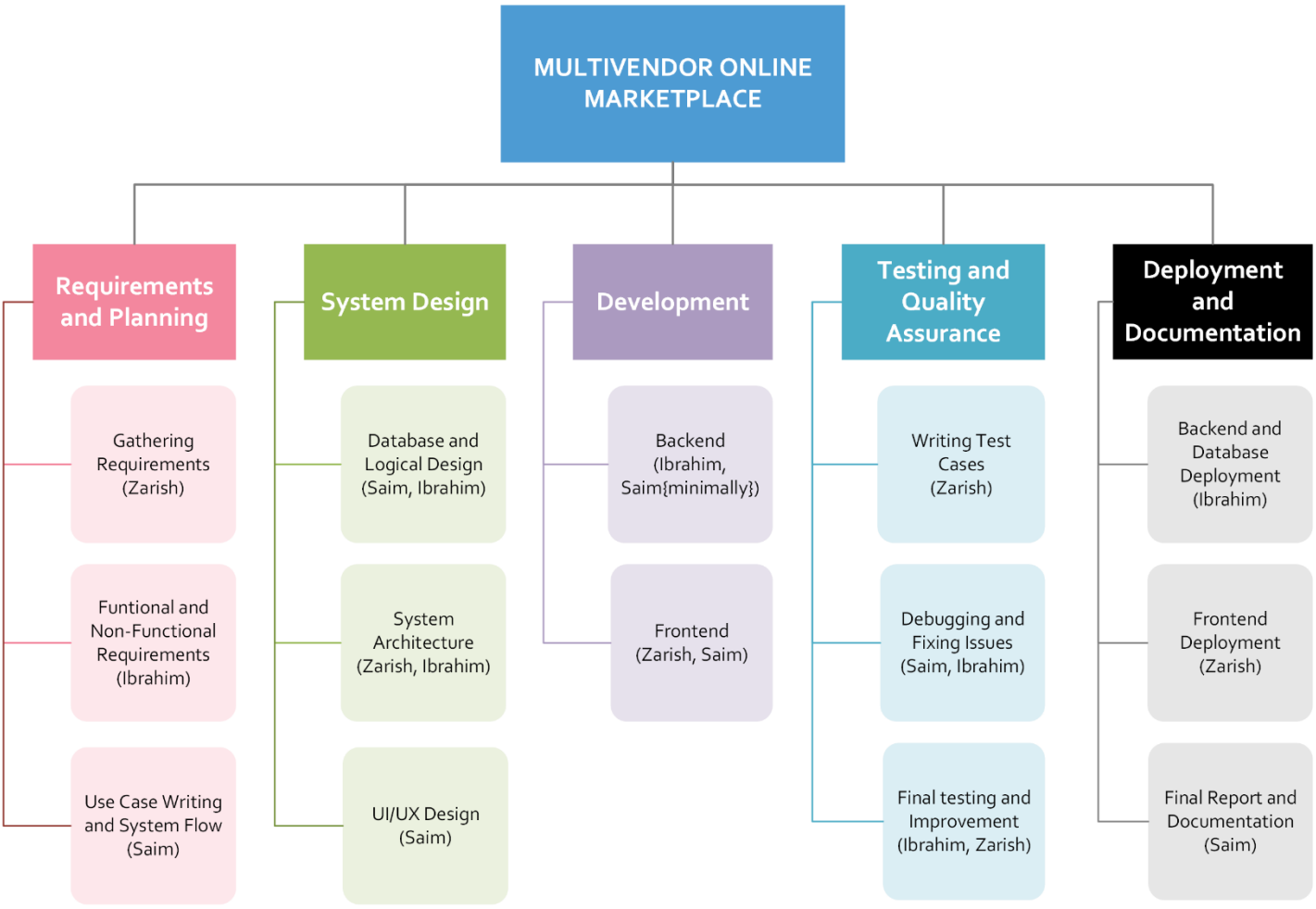

Figure 1 Work Breakdown Structure

1.7 Project Timeline & Milestones

Table 1 Project Phases and Duration

Increment	Title	Key Features	Duration
1	Core System Setup	User roles & auth, dashboard UI, basic DB setup (users, products, orders)	2 weeks
2	Product & Vendor Management	Vendor product control, product browsing/search, admin vendor monitoring	1.5 weeks
3	Shopping & Order Management	Cart & checkout, order system, vendor order handling	2 weeks
4	Reviews, Ratings & Reports	Product reviews/ratings, vendor reports, admin performance tracking	1.5 weeks
5	Testing & Deployment	Final testing, bug fixing, improvements, documentation completion	1 week

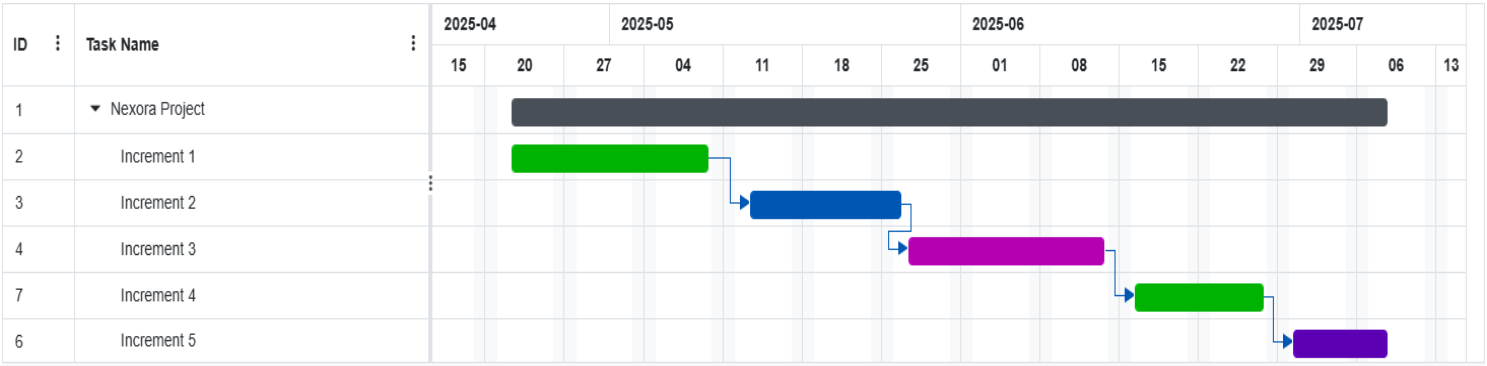

Figure 2Gantt Chart

CHAPTER 2

SOFTWARE DEVELOPMENT MODEL

2.1 Software Development Model

Choosing the right software development model is important for ensuring smooth project execution, efficient collaboration, and timely delivery. For Nexora, the Incremental Model is selected because it allows the system to be developed in phases, ensuring that core functionalities are implemented first while additional features can be added as the progress of the project continues.

2.2 Overview of the Incremental Model

The Incremental Model is a software development approach in which the system is built and delivered in small, manageable modules rather than developing the entire system at once. Each increment adds functionality until the full system is complete. This model ensures that the system is tested, reviewed, and improved at each stage, reducing risks and improving quality.

2.3 Why the Incremental Model is Used for This Project

The Incremental Model is chosen for Nexora due to the following reasons:

- Phased Development: The marketplace can be built in multiple iterations, ensuring that critical components (such as user authentication, product management, and order processing) are developed first before adding advanced features.
- Early Testing & Feedback: As the project is to be completed in two months due to limited semester time, testing each module separately helps in early detection and fixing of issues.
- Flexibility in Development: If changes are required based on testing or new requirements, they can be checked and done in future increments without affecting previous work.
- Efficient Resource Utilization: Since different team members are working on different modules, we can develop, test, and refine our assigned features independently without waiting for the entire system to be completed.

2.4 How the Incremental Model is Applied to This Project

The development of the system is divided into increments, with each phase adding new functionality:

1. Increment 1: Core System Setup
    - User authentication and access (Admin, Vendor, Customer).
    - UI structure for user dashboards.
    - Database setup with basic tables (users, products, orders).
2. Increment 2: Product & Vendor Management
    - Vendors can list, edit, and delete products.
    - Customers can browse and search for products.
    - Admin can monitor and manage vendor activities.
3. Increment 3: Shopping & Order Management
    - Customers can add items to the cart and proceed to checkout.
    - Order placement, confirmation, and tracking system.
    - Vendors receive and manage orders.
4. Increment 4: Reviews, Ratings & Reports
    - Customers can review and rate products.
    - Vendors can view sales reports and order statistics.
    - Admin can monitor marketplace performance.
5. Increment 5: Testing & Deployment
    - Final system testing, bug fixes, and improvements.
    - Documentation finalization.

2.5 Why Other Models Were Not Chosen

The Waterfall Model was not chosen because it follows a linear, sequential approach where each phase must be completed before the next begins. This rigid structure does not allow for flexibility or early testing. Since our project requires incremental improvements and testing at each stage, the Waterfall Model would not be suitable.

The Agile Model was not selected because it relies on continuous customer feedback and iterative changes, which may not be feasible within our project's timeline. Agile works best when there is constant stakeholder involvement, but our development process for the system is structured in such a way that it has only predefined core requirements, making Incremental Model a better fit.

The Spiral Model was not chosen because it is designed for high-risk projects where frequent risk assessments are required. Our project is moderate in complexity and does not involve significant uncertainties that require continuous risk analysis. The Incremental Model provides a simpler and more efficient approach for our project needs.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 Functional Requirements

Table 2 Functional Requirements

Category	Requirement ID	Functional Requirement
User Authentication & Role Management	FR-01	The system allows vendors, customers, and admins to create accounts and log in securely.
	FR-02	The system provides role-based access, restricting actions based on user type.
	FR-03	The system allows users to reset passwords through email verification.
Vendor Management	FR-04	The system allows vendors to register, update, and delete their profiles.
	FR-05	The system enables vendors to add, edit, and remove products.
	FR-06	The system allows vendors to view and manage orders.
	FR-07	The system generates sales reports for vendors based on transactions.
Product Management	FR-08	The system allows vendors to add products with images, descriptions, and prices.
	FR-09	The system enables customers to search for products using filters such as category, price, and rating.
	FR-10	The system displays product availability based on vendor stock levels.
Shopping Cart & Checkout	FR-11	The system allows customers to add, update, and remove items from the shopping cart.
	FR-12	The system displays the total price, including applicable taxes or discounts.
	FR-13	The system allows customers to proceed to check out and make payments securely.
	FR-14	The system generates a unique order ID for every purchase.

Order Processing & Tracking	FR-15	The system allows customers to track their order status (Pending, Shipped, Delivered).
	FR-16	The system notifies vendors when an order is placed and requires processing.
Customer Reviews & Ratings	FR-17	The system will allow customers to leave reviews and rate products.
	FR-18	The system shows an average rating for each product based on customer feedback.
	FR-19	The system allows admins to moderate and remove inappropriate reviews.
Admin Management & Marketplace Control	FR-20	The system allows the admin to approve or reject vendor registrations.
	FR-21	The system enables the admin to monitor and manage all users, products, and transactions.
	FR-22	The system provides sales analytics and reports to track marketplace performance.

3.2 Non-Functional Requirements

Non-functional requirements define the quality attributes of the system, ensuring it performs efficiently, securely, and is easy to use. The following table outlines the main non-functional requirements for Nexora.

Table 3 Non-Functional Requirements

Category	Requirement ID	Non-Functional Requirement
Performance	NFR-01	The system will handle at least 50 simultaneous users without significant slowdown.
	NFR-02	Page load time should not exceed 3 seconds under normal usage conditions.
Security	NFR-03	User passwords will be securely stored using encryption.
	NFR-04	All sensitive data shall be transmitted over HTTPS.
Usability	NFR-05	The system has a simple and intuitive interface for ease of navigation.
	NFR-06	Customers are able to complete a purchase in no more than 5 steps.
Scalability	NFR-07	The system will allow for future expansion, supporting additional vendors and products.
Availability	NFR-08	The system maintains an uptime of at least 99%.

3.3 Architectural Diagram

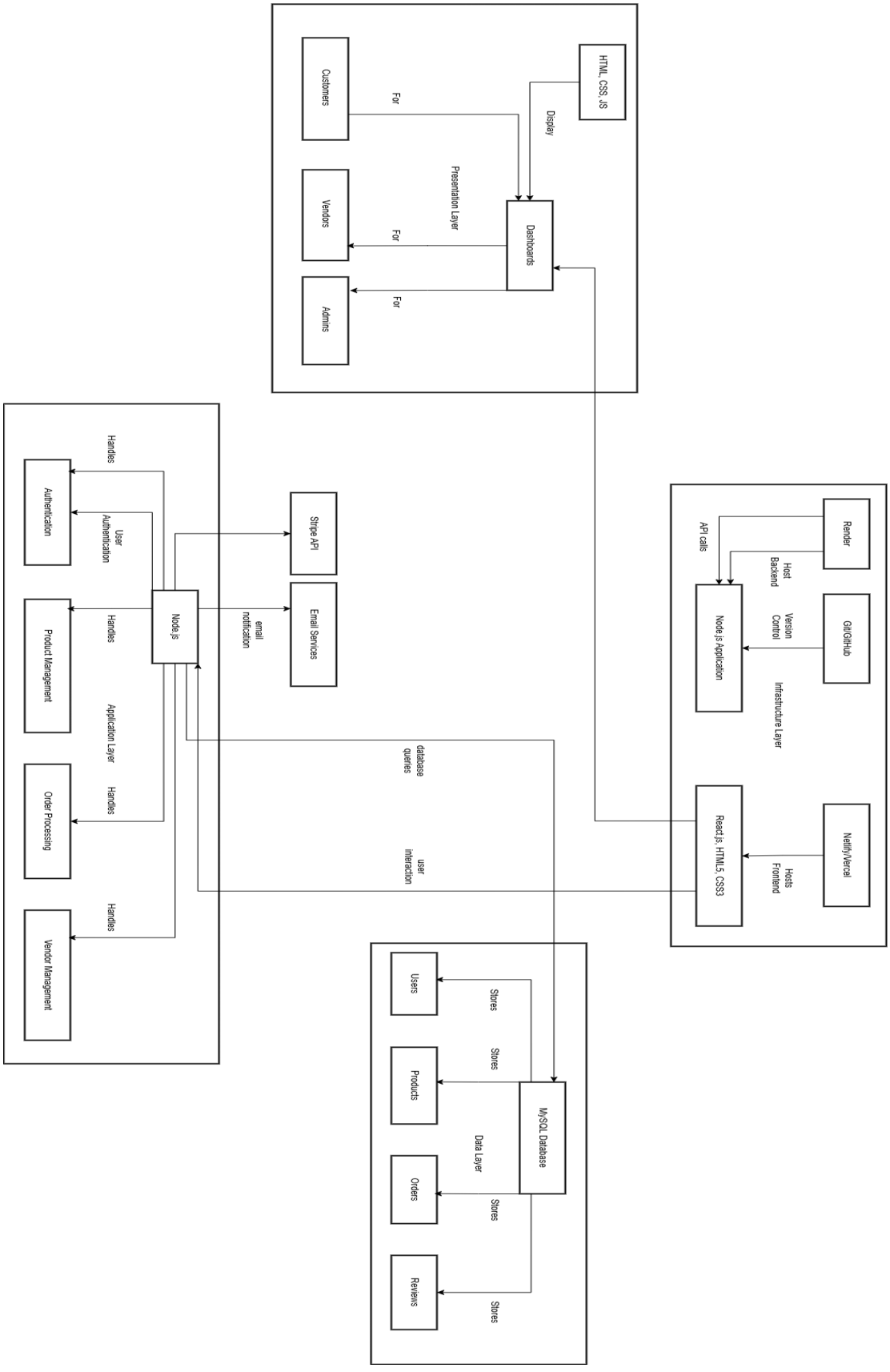

Figure 3 Architectural Diagram

3.4 Use Case Diagram

A Use Case Diagram provides a visual representation of how different users interact with the system. It helps in understanding the functional scope of the system by identifying the actors and their corresponding use cases. The Use Case Diagram for Nexora is designed to show all significant interactions, ensuring a structured and functional system workflow.

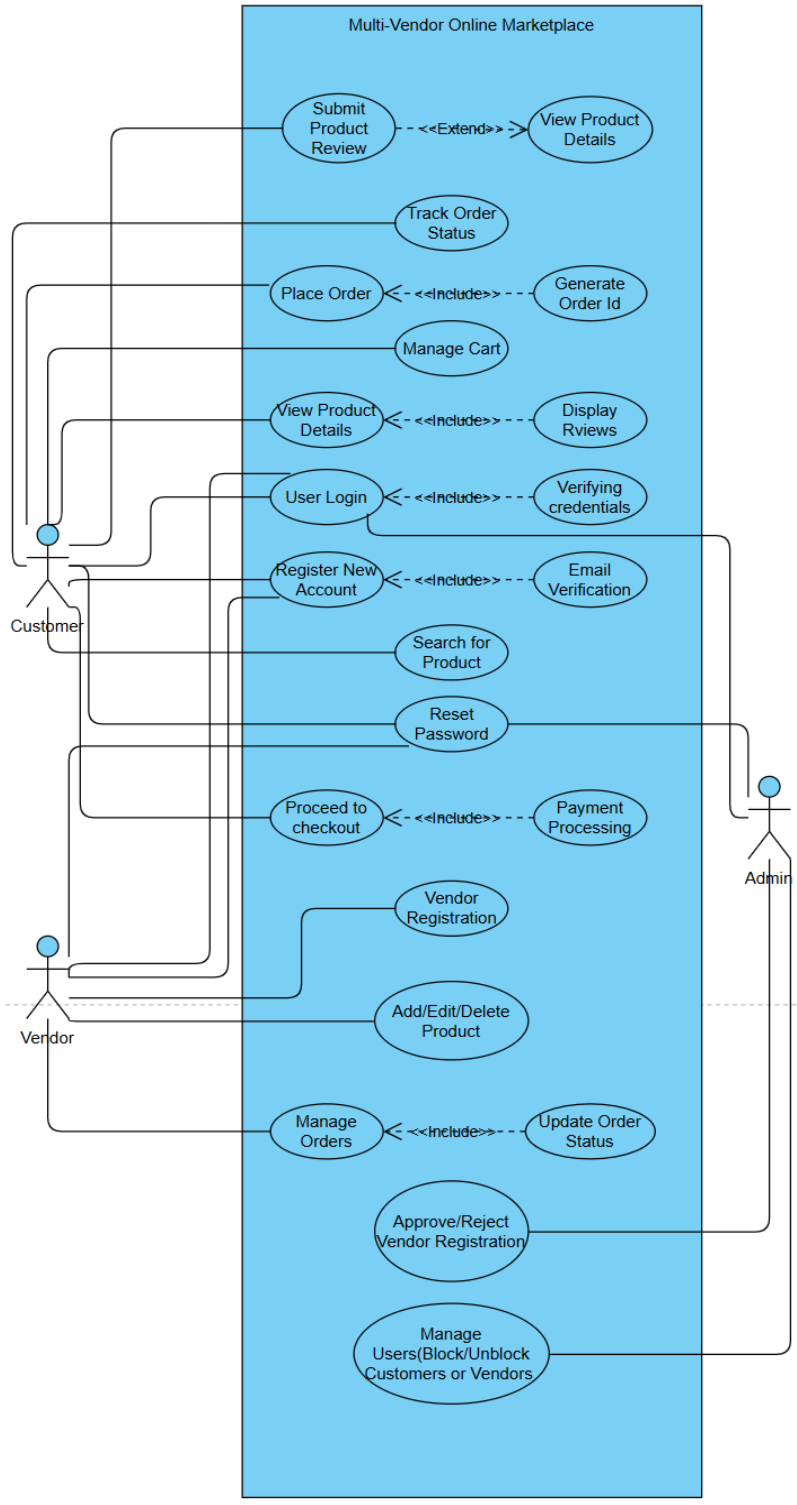

Figure 4 Use Case Diagram

3.5 Fully Dressed Use Cases

Use Case 1: User Login

Table 4 Use case: User Login

Use Case Name	**User Login**
Actors	• Customer • Vendor • Admin
Description	This use case allow users (customers, vendors, and admins) to log into the system using their registered email and password. The system verifies the provided credentials and grants them access if authentication is successful.
Preconditions	• The user must already have a registered account. • The user must provide valid login credentials.
Postconditions	• If login is successful, the user is redirected to their respective dashboard. • If login fails, an error message is displayed, and the user is asked to try again.
Main Flow	1. The user goes to the login page. 2. The user enters their email and password. 3. The system validates the credentials. 4. If the credentials are correct: • The system grants access to the user. • The user is redirected to their dashboard (Customer, Vendor, or Admin). 5. If the credentials are incorrect: • The system displays an error message ("Invalid email or password"). • The user is prompted to re-enter credentials.
Alternative Flow	**Forgot Password** 1. If the user forgets their password, they can click "Forgot Password". 2. The system redirects them to the password reset page. 3. The user enters their registered email. 4. The system sends a password reset link to the email. 5. The user follows the link to reset their password.
Exceptions	• Account Not Found → If the email is not registered, the system displays "Account does not exist." • System Error → If the system is down, an error message "Login service temporarily unavailable" is displayed.

Use Case 2: Register New Account

Table 5 Use Case 2: Register New Account

Use Case Name	Register New Account
Actors	• Customer • Vendor
Description	This use case allows new users (customers and vendors) to create an account by providing the required details. The system verifies the information and sends an email for verification before completing the registration.
Preconditions	• The user must not already have an existing account. • The user must provide a valid email address.
Postconditions	• If registration is successful, the user receives an email verification link. • If registration fails, an appropriate error message is displayed.
Main Flow	1. The user goes to the registration page. 2. The user selects their role (Customer or Vendor). 3. The user enters their full name, email, password, and other required details. 4. The system validates the entered information. 5. If all information is correct: • The system creates a new user account. • The system sends a verification email with an activation link. 6. If there are errors (e.g., missing fields, weak password): • The system displays an error message and prompts the user to correct the details.
Alternative Flow	**Email Already Exists** 1. If the email entered is already registered, the system displays "Email already in use". 2. The user is prompted to either log in or use a different email address.
Exceptions	• Invalid Email Format → If the email format is incorrect, the system prompts the user to enter a valid email. • Weak Password → If the password does not meet security criteria, an error message is shown. • Email Not Verified → If the user tries to log in without verifying their email, the system displays "Please verify your email before logging in."

Use Case 3: Search for Product

Table 6 Use Case 3: Search for Product

Use Case Name	Search for Product
Actors	• Customer
Description	This use case allows customers to search for products using keywords or filters. The system retrieves and displays matching products based on the search criteria.
Preconditions	• The system must have products available in the database.
Postconditions	• If products match the search query, they are displayed in a list. • If no products match, a message "No results found" is displayed.
Main Flow	1. The customer goes through the search bar on the website. 2. The customer enters a product name, category, or keyword. 3. The system processes the search query and retrieves relevant products.

	4. The system displays the list of matching products with images, prices, and ratings. 5. The customer can click on a product to view its details.
Alternative Flow	**Filter Search Results** 1. The customer applies filters (e.g., category, price range, rating). 2. The system refines the search results based on the applied filters.
Exceptions	• No Products Found → If no products match the search, the system displays "No results found. Try a different keyword." • System Error → If an issue occurs during search, the system displays "Unable to process search request. Please try again later."

Use Case 4: View Product Details

Table 7 Use Case 4: View Product Details

Use Case Name	**View Product Details**
Actors	• Customer
Description	This use case allows customers to view detailed information about a product, including its description, price, images, and customer reviews.
Preconditions	• The product must exist in the system and be available for viewing.
Postconditions	• The system displays the product details. • If the customer decides, they can add the product to their cart or submit a review.
Main Flow	1. The customer selects a product from the search results or product listings. 2. The system retrieves and displays the product details page, including: • Product name, description, and price. • Product images. • Availability status. • Customer ratings and reviews. 3. The customer can: • Click on images to view an enlarged version. • Read customer reviews. • Add the product to their cart.
Alternative Flow	**Product is Out of Stock** 1. If the product is out of stock, the system displays "Out of Stock" instead of the Add to Cart button. 2. The customer may choose to add the product to their Wishlist (if implemented in future versions).
Exceptions	• Product Not Found → If the selected product does not exist (e.g., deleted by the vendor), the system displays "Product not available." • System Error → If there is an issue retrieving product details, the system displays "Unable to load product information. Please try again later."

Use Case 5: Manage Cart (Add/Remove Products)

Table 8 Use Case 5: Manage Cart (Add/Remove Products)

Use Case Name	**Manage Cart (Add/Remove Products)**
Actors	• Customer
Description	This use case allows customers to manage their shopping cart by adding or removing products before proceeding to checkout.
Preconditions	• The customer must be logged in (if required for cart functionality). • The selected product must be available in stock.
Postconditions	• The cart is updated with the selected product(s). • The total price is recalculated.
Main Flow	**Adding a Product to Cart:** 1. The customer selects a product and clicks "Add to Cart." 2. The system checks if the product is available in stock. 3. If available, the system: • Adds the product to the cart. • Updates the total price. 4. The system displays a confirmation message "Product added to cart successfully." Removing a Product from Cart: 1. The customer navigates to the Cart page. 2. The customer selects a product and clicks "Remove." 3. The system: • Removes the product from the cart. • Updates the total price. 4. The system displays a confirmation message "Product removed from cart."
Alternative Flow	**Adjusting Product Quantity** 1. The customer selects a product and updates the quantity (e.g., increases or decreases the number of items). 2. The system recalculates the total price and updates the cart.
Exceptions	• Product Out of Stock → If the product becomes unavailable after adding it to the cart, the system removes it and notifies the customer. • System Error → If an issue occurs while updating the cart, the system displays "Unable to update cart. Please try again later."

Use Case 6: Proceed to Checkout

Table 9 Use Case 6: Proceed to Checkout

Use Case Name	Proceed to Checkout
Actors	• Customer
Description	This use case allows customers to proceed to checkout after selecting products in their cart. The system collects shipping and payment details before finalizing the purchase.
Preconditions	• The customer must have at least one product in their cart. • The selected products must be available in stock.
Postconditions	• The order is successfully placed if payment is completed. • The system generates an order ID and updates the order status.
Main Flow	1. The customer routes to the Cart page and clicks Proceed to Checkout. 2. The system displays the order summary, including product details, prices, and total amount. 3. The customer enters the shipping address and selects a payment method. 4. The customer clicks Confirm & Pay. 5. The system processes the payment (Use Case 8: Make a Payment is included here). 6. If payment is successful: • The system generates a unique order ID. • The system updates the order status to Processing. • The system sends an order confirmation email to the customer. 7. The system redirects the customer to the Order Confirmation Page.
Alternative Flow	**Applying Discount Code** 1. The customer enters a discount code before making the payment. 2. The system verifies and applies the discount if valid. 3. The total amount is updated.
Exceptions	• Cart is Empty → If the cart is empty, the system disables the checkout button and displays Your cart is empty. Add products before proceeding. • Payment Failure → If the payment fails, the system displays Payment unsuccessful. Please try again or use a different method. • Product Out of Stock → If an item goes out of stock before checkout, the system removes it from the cart and notifies the customer.

Use Case 7: Track Order Status

Table 10 Use Case 7: Track Order Status

Use Case Name	Track Order Status
Actors	• Customer
Description	This use case allows customers to track the status of their orders, providing updates on processing, shipping, and delivery.
Preconditions	• The customer must have placed at least one order.
Postconditions	• The system displays the current order status.
Main Flow	1. The customer directs to the Order History page. 2. The customer selects an order to view its details. 3. The system retrieves and displays the order details, including: • Order ID • Order date and total price • Status (Processing, Shipped, Delivered) • Estimated delivery date 4. The customer can exit the order tracking page and then return to the main dashboard.
Alternative Flow	**Order Status Updated** 1. If the order status changes (e.g., from Processing to Shipped), the system automatically updates it in real-time. 2. The customer receives an email notification with the updated status.
Exceptions	• No Orders Found → If the customer has no orders, the system displays You have no orders yet. • System Error → If the system fails to retrieve order details, an error message Unable to fetch order status Please try again later is displayed.

Use Case 8: Submit Product Review

Table 11 Use Case 8: Submit Product Review

Use Case Name	Submit Product Review
Actors	• Customer
Description	This use case allows customers to submit reviews and ratings for purchased products. The system verifies that the customer has bought the product before allowing them to submit a review.
Preconditions	• The customer must have purchased the product.
Postconditions	• The review is successfully submitted and displayed on the product page.
Main Flow	1. The customer navigate to the Order History page. 2. The customer selects a purchased product and clicks Write a Review. 3. The system checks whether the customer has previously purchased the product. 4. If eligible, the system displays the review form with fields for rating (1–5 stars) and text

	review. 5. The customer enters a rating and review and clicks Submit. 6. The system saves the review and updates the product page with the new rating and feedback. 7. A confirmation message Your review has been submitted successfully. is displayed.
Alternative Flow	**Edit or Delete Review** 1. If the customers want to edit or delete a submitted review, they navigate to the Reviews Section. 2. The system allows them to either modify the review or remove it entirely. 3. The product page updates accordingly.
Exceptions	• Customer Did Not Purchase Product → If the customer tries to review a product they have not bought, the system displays You can only review products you have purchased. • System Error → If a system issue occurs while saving the review, an error message "Unable to submit review. Please try again later." is displayed.

Use Case 9: Vendor Registration

Table 12 Use Case 9: Vendor Registration

Use Case Name	**Vendor Registration**
Actors	• Vendor
Description	This use case allows a vendor to register an account to sell products on the marketplace. The system verifies the provided details and requires admin approval before activation.
Preconditions	• The vendor must provide valid registration details.
Postconditions	• The vendor account is created and sent for admin approval. • The vendor can log in only after approval.
Main Flow	1. The vendor will go through the Vendor Registration page. 2. The vendor enters the required details, including: • Business name • Email address • Contact information • Business verification documents (if required) 3. The system validates the entered details. 4. If all information is correct, the system creates the vendor account 5. The system notifies the admin for approval. 6. The vendor receives a confirmation message: Your registration request has been submitted for approval.
Alternative Flow	**Admin Approves Registration** 1. The admin reviews the vendor's application. 2. If approved, the system activates the account, and the vendor is notified. 3. The vendor can now log in and manage their store.
Exceptions	• Email Already Registered → If the vendor enters an email already in use, the system displays This email is already registered. Try logging in.

	• Missing Required Information → If required fields are empty, the system prompts the vendor to fill them before proceeding. • Admin Rejects Registration → If the admin rejects the registration, the system notifies the vendor with the reason for rejection.

Use Case 10: Add/Edit/Delete Product

Table 13 Use Case 10: Add/Edit/Delete Product

Use Case Name	Add/Edit/Delete Product
Actors	• Vendor
Description	This use case allows vendors to add new products, edit existing product details, or remove products from their listings.
Preconditions	• The vendor must be logged into their account.
Postconditions	• The product listing is updated in the system.
Main Flow	**Adding a Product:** 1. The vendor directs through the Product Management section. 2. The vendor clicks "Add New Product." 3. The vendor enters product details, including: • Product name • Description • Price • Stock quantity • Product images 4. The system validates the details and saves the product. 5. The product is added to the vendor's store and becomes visible to customers. **Editing a Product:** 1. The vendor selects a product from their store. 2. The vendor clicks "Edit." 3. The vendor modifies the necessary details. 4. The system updates the product information. 5. The updated product details are reflected on the marketplace. **Deleting a Product:** 1. The vendor selects a product to delete. 2. The vendor clicks Delete. 3. The system asks for confirmation. 4. If confirmed, the system removes the product from the vendor's store.
Exceptions	• Missing Required Fields → If the vendor does not fill in all required fields when adding/editing a product, the system prompts them to complete the form. • Product Not Found → If a vendor tries to edit or delete a product that has already been removed, the system displays Product no longer exists. • System Error → If an issue occurs while saving changes, the system displays Unable to update product. Please try again later.

Use Case 11: Manage Orders

Table 14 Use Case 11: Manage Orders

Use Case Name	Manage Orders
Actors	• Vendor
Description	This use case allows vendors to view, process, and update the status of customer orders.
Preconditions	• The vendor must be logged into their account. • At least one order must exist in the system.
Postconditions	• The order status is updated in the system. • Customers are notified of any changes to their order status.
Main Flow	1. The vendor goes to the Orders Dashboard. 2. The system displays a list of pending, processing, and completed orders. 3. The vendor selects an order to manage. 4. The vendor can perform the following actions: • View Order Details – Displays product information, customer details, and payment status. • Update Order Status – Changes order status (e.g., Processing → Shipped → Delivered). • Cancel Order – Cancels an order if necessary, notifying the customer. 5. The system saves any changes and updates the customer accordingly.
Alternative Flow	**Order Partially Processed** 1. If an order contains multiple items and only some are ready for shipment, the vendor can mark certain items as Shipped, while others remain Processing. 2. The system updates the order details accordingly.
Exceptions	• Order Not Found → If the vendor tries to manage an order that has already been removed, the system displays Order not available. • System Error → If the system fails to update order status, an error message "Unable to update order. Please try again later." is displayed.

Use Case 12: Approve/Reject Vendor Registration

Table 15 Use Case 12: Approve/Reject Vendor Registration

Use Case Name	Approve/Reject Vendor Registration
Actors	• Admin
Description	This use case allows the admin to review vendor registration requests and either approve or reject them based on verification criteria.
Preconditions	• The vendor must have submitted a registration request.
Postconditions	• If approved, the vendor account is activated, and they can log in. • If rejected, the vendor is notified with the reason for rejection.
Main Flow	1. The admin logs into the system and goes through the Vendor Approval Dashboard. 2. The system displays a list of pending vendor applications. 3. The admin selects a vendor application to review. 4. The admin checks the provided details, including: • Business name and contact information • Business verification documents 5. The admin chooses one of the following options: • Approve Registration → The system activates the vendor account and sends a

	confirmation email. • Reject Registration → The system asks for a rejection reason, notifies the vendor, and disables the request.
Alternative Flow	**Request More Information** 1. If the submitted details are incomplete, the admin can request additional information from the vendor. 2. The vendor is notified and prompted to provide the missing details. 3. The system keeps the registration in a Pending state until all required details are submitted.
Exceptions	• Vendor Already Approved → If the admin attempts to approve an already approved vendor, the system displays "Vendor account is already active." • System Error → If the system fails to process the approval or rejection, an error message "Unable to process request. Please try again later." is displayed.

Use Case 13: Manage Users (Block/Unblock Vendors & Customers)

Table 16 Use Case 13: Manage Users (Block/Unblock Vendors & Customers)

Use Case Name	Manage Users (Block/Unblock Vendors & Customers)
Actors	• Admin
Description	This use case allows the admin to manage user accounts by blocking or unblocking vendors and customers based on violations or inactivity.
Preconditions	• The user must have an active account in the system.
Postconditions	• If blocked, the user cannot log in or access the system. • If unblocked, the user regains access to their account.
Main Flow	1. The admin logs into the system and navigates to the User Management Panel. 2. The system displays a list of registered users, including vendors and customers. 3. The admin selects a user account to manage. 4. The admin chooses one of the following actions: • Block User → The system disables the account, preventing access. • Unblock User → The system reactivates the account, allowing access. 5. The system updates the user status and notifies them via email.
Alternative Flow	**Temporary Suspension** 1. Instead of permanently blocking a user, the admin can suspend the account for a specific period. 2. The system sets an expiration date for the suspension. 3. The user will automatically regain access after the suspension period ends.
Exceptions	• User Already Blocked → If the admin tries to block an already blocked user, the system displays This account is already blocked. • User Not Found → If the selected user does not exist, the system displays User does not exist. • System Error → If a technical issue prevents account updates, the system displays "Unable to update user status. Please try again later."

CHAPTER 4

FRONTEND DESIGN (SKETCHING)

4.1 Introduction

The frontend UI design serves as a visual representation of how users will interact with the system. This section presents the sketches of key screens created using HTML, CSS, and JavaScript without functionalities, ensuring a clear structure before implementation. The design focuses on different user roles: customers, vendors, and administrators, ensuring usability, clarity, and better user experience.

The following screens have been created as part of the sketching phase:

1. Login Page – The entry point for users, allowing authentication for customers, vendors, and admins.
2. Dashboard – A central hub for users to navigate their respective features based on their roles.
3. Product Browsing (Customer) – A display of all available products with filtering and sorting options.
4. Cart Page (Customer) – A view where customers can manage selected products before checkout.
5. Order Confirmation & Payment – The final step where customers confirm their order and proceed with payment.
6. Order Management (Admin) – A panel for administrators to track and manage customer orders.
7. Product Management (Admin) – Allows administrators to add, edit, and remove products.
8. User Profile & Settings – A section where users can update their personal details and preferences.
9. Vendor Earnings & Payout – A dashboard for vendors to track their earnings and request payouts.

4.2 UI Sketches for Major Parts

1. Login

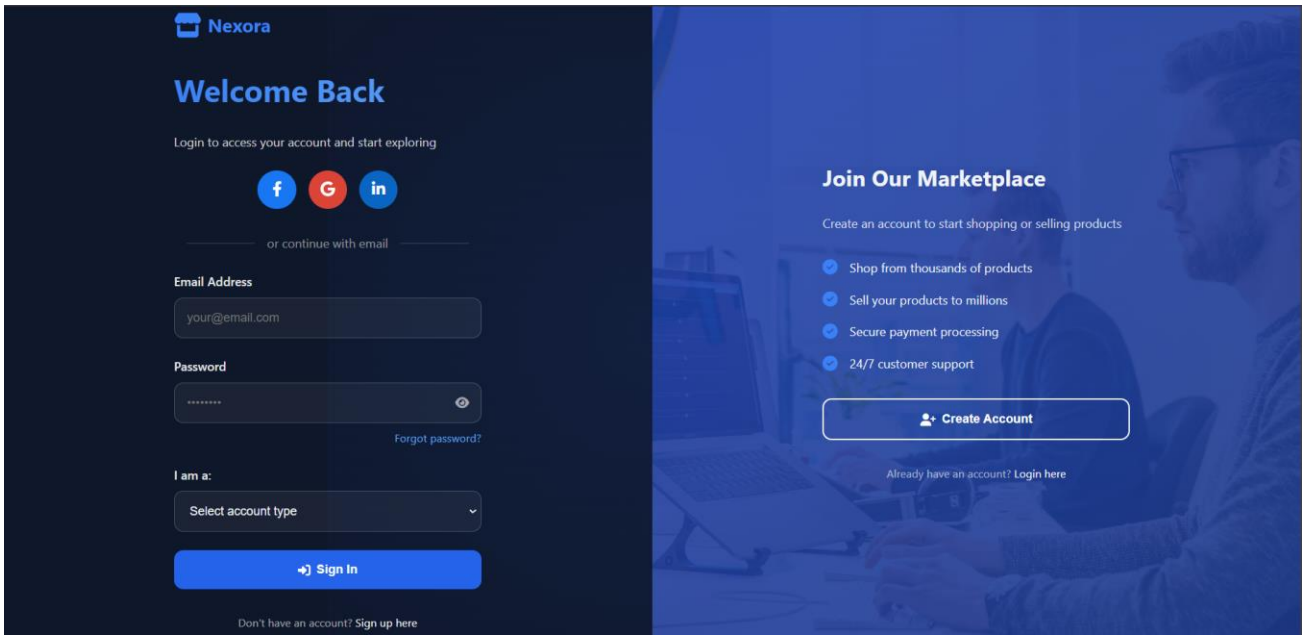

Figure 5 Login

2. Admin Dashboard

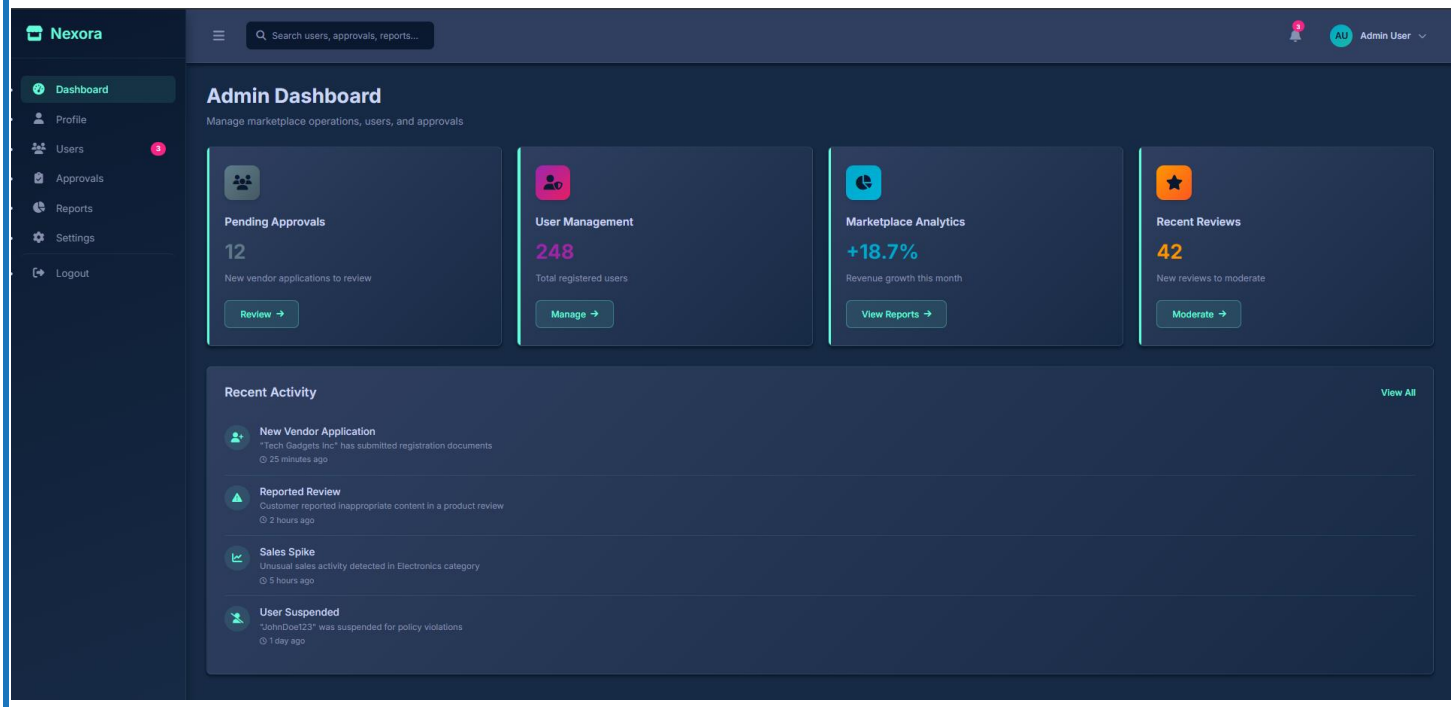

Figure 6 Admin Dashboard

3. Vendor Dashboard

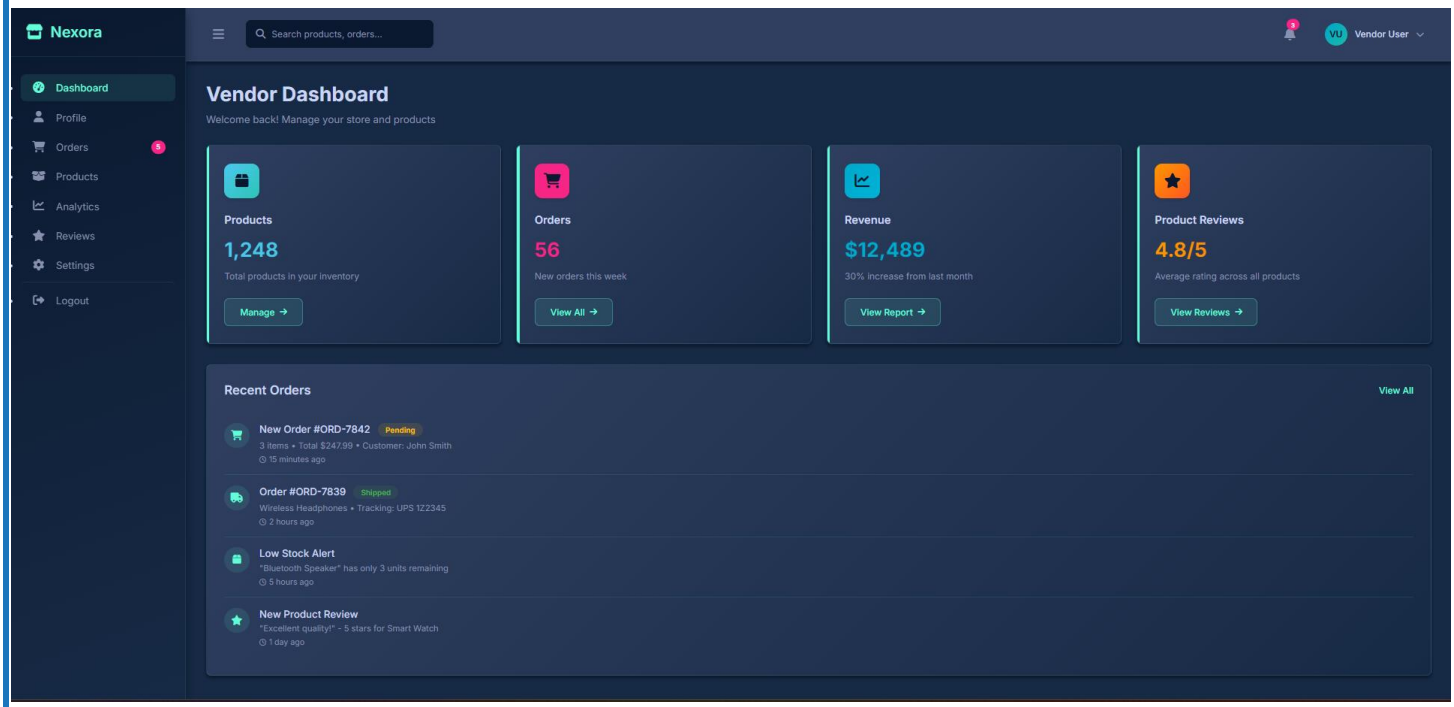

Figure 7 Vendor Dashboard

4. Customer Dashboard

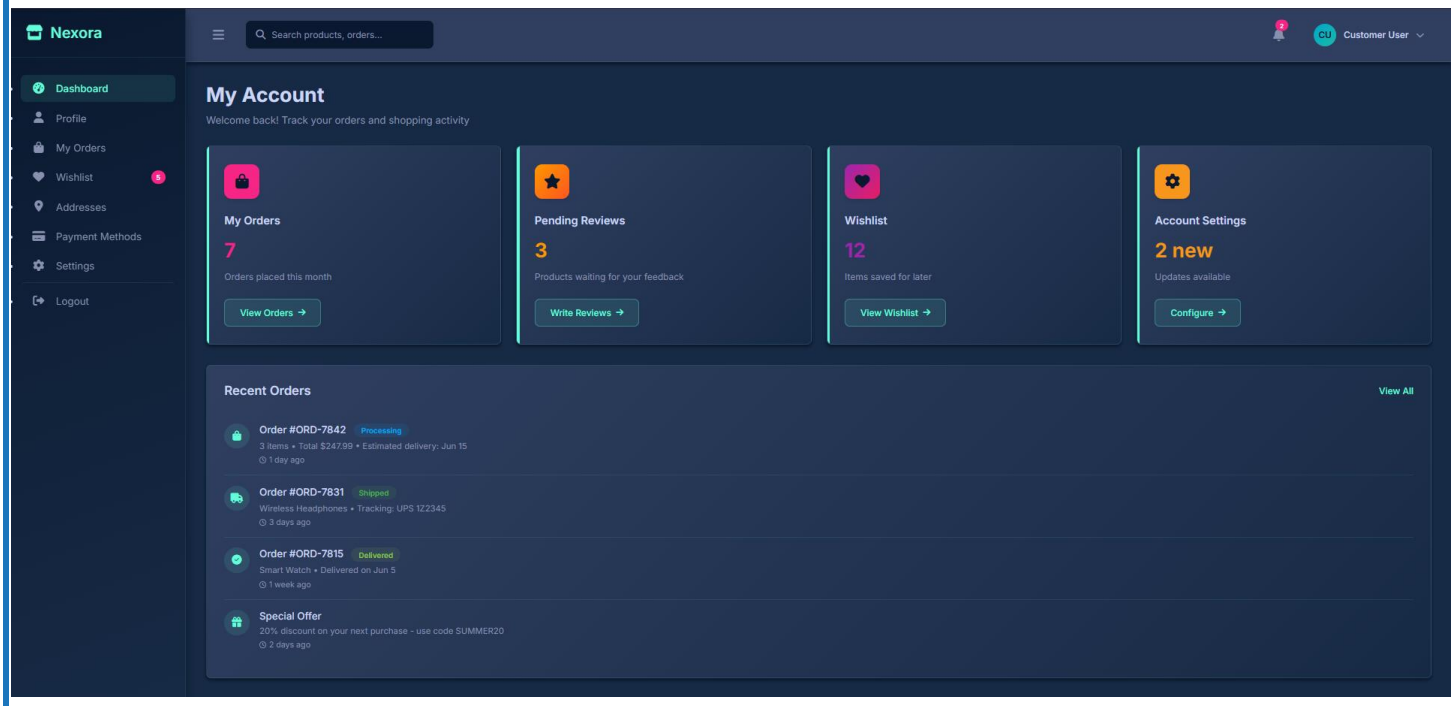

Figure 8 Customer Dashboard

5. Product Browsing

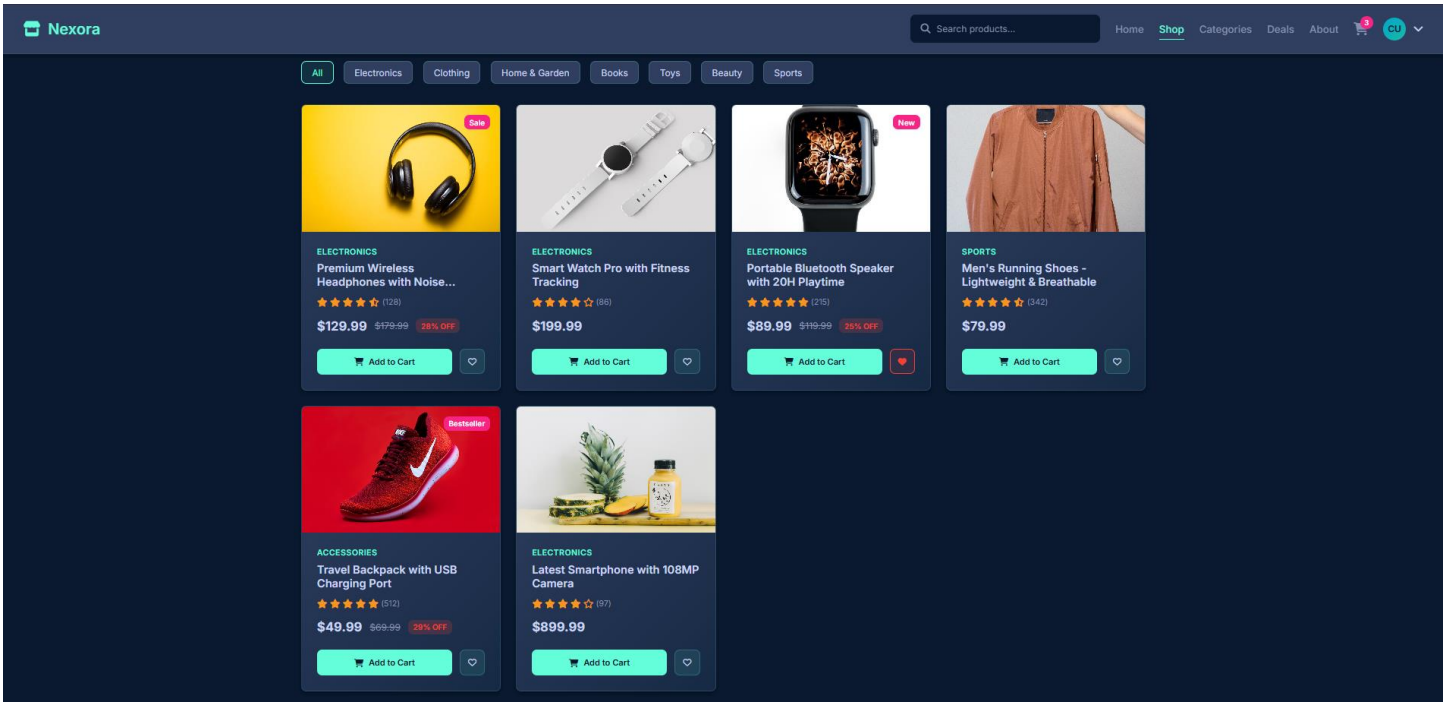

Figure 9 Product Browsing

6. Cart Page

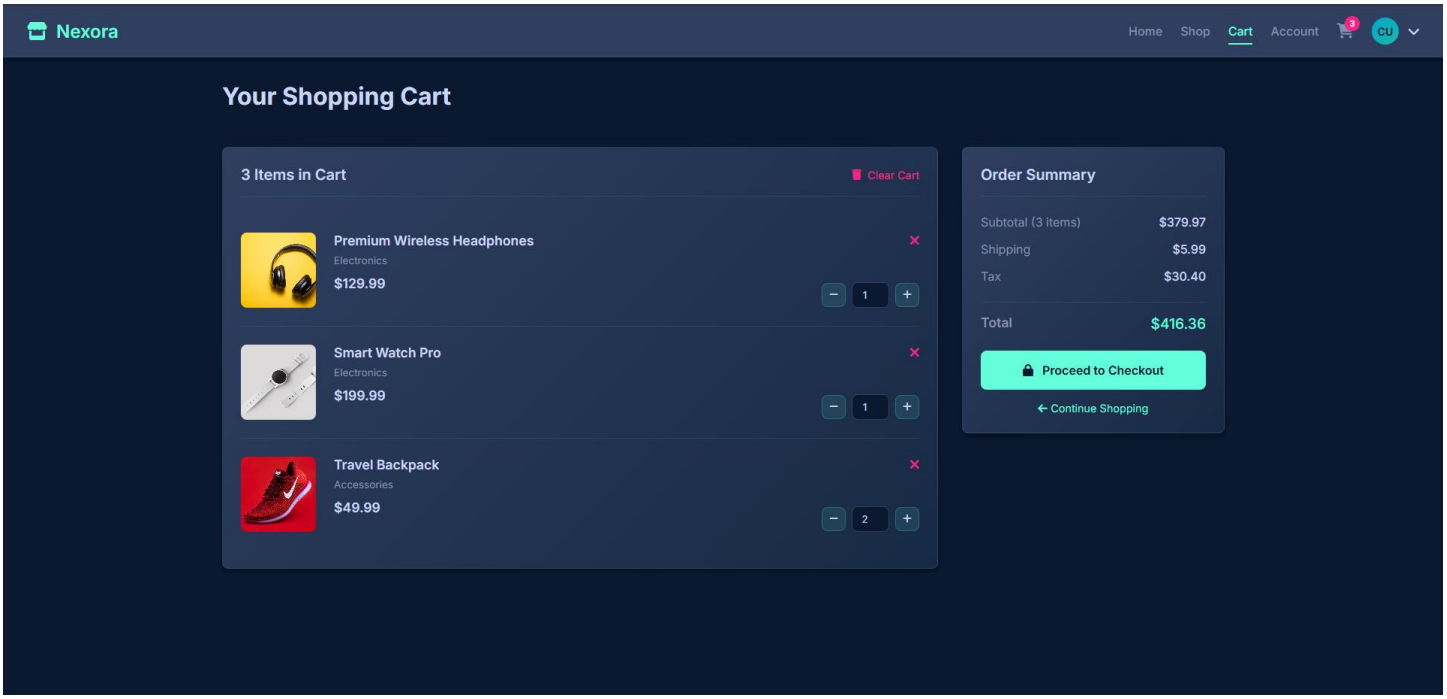

Figure 10 Cart Page

7. Order Confirmation and Payment

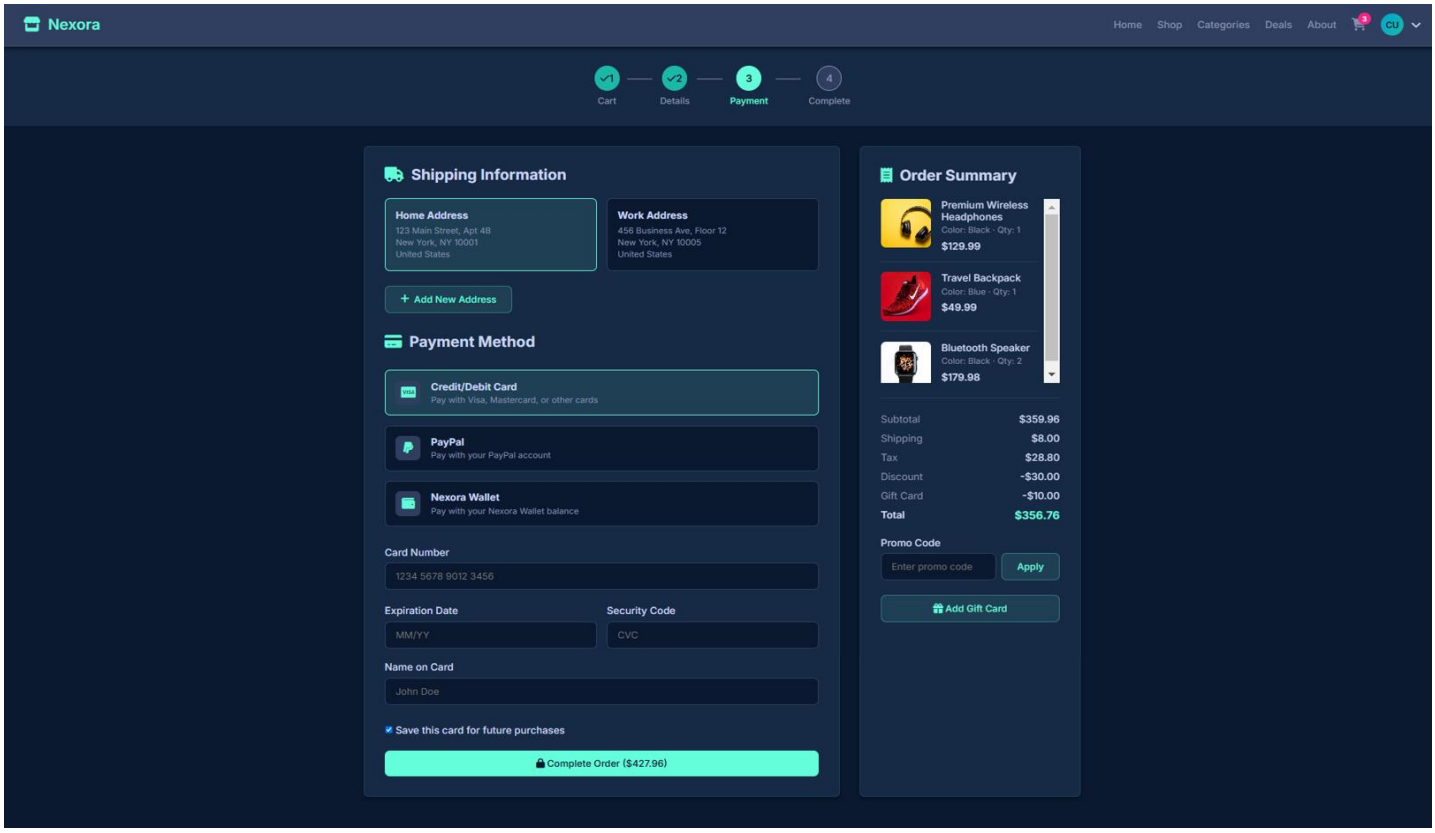

Figure 11 Order Confirmation and Payment

8. Order Management

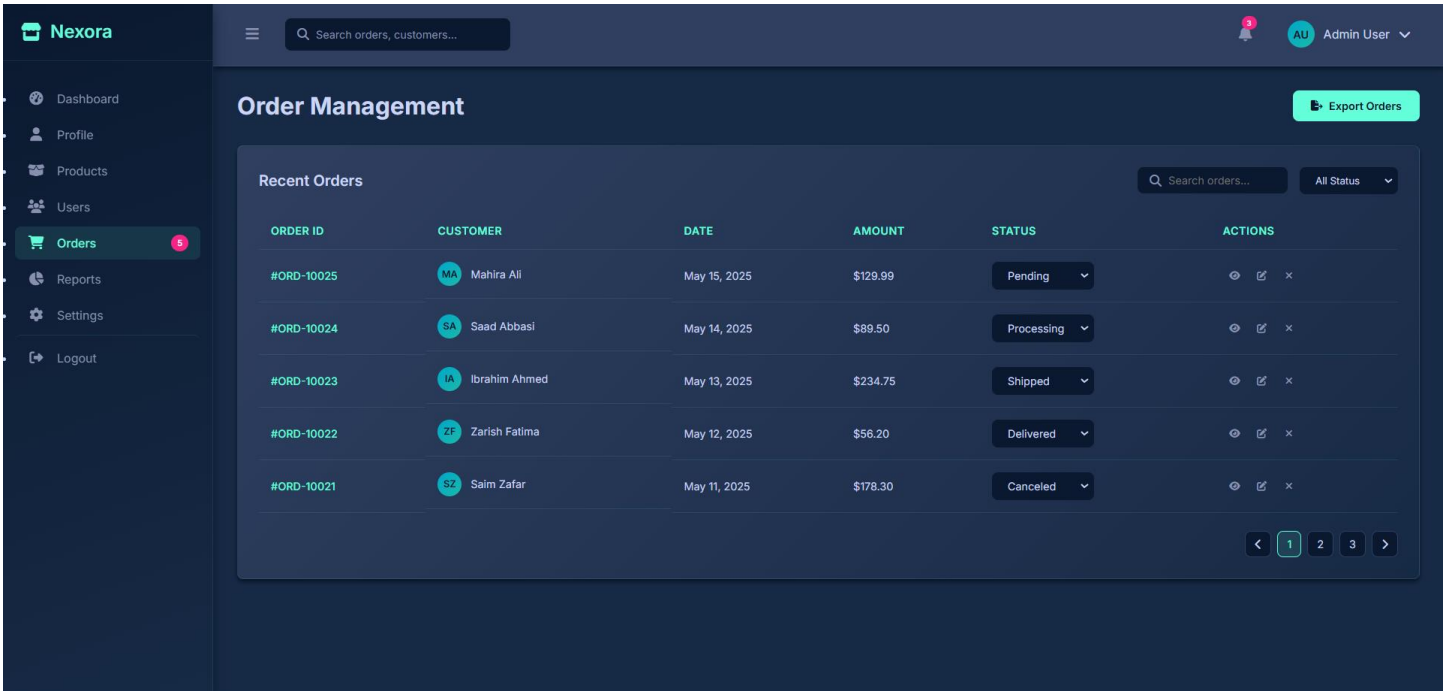

Figure 12 Order Management

9. Product Management

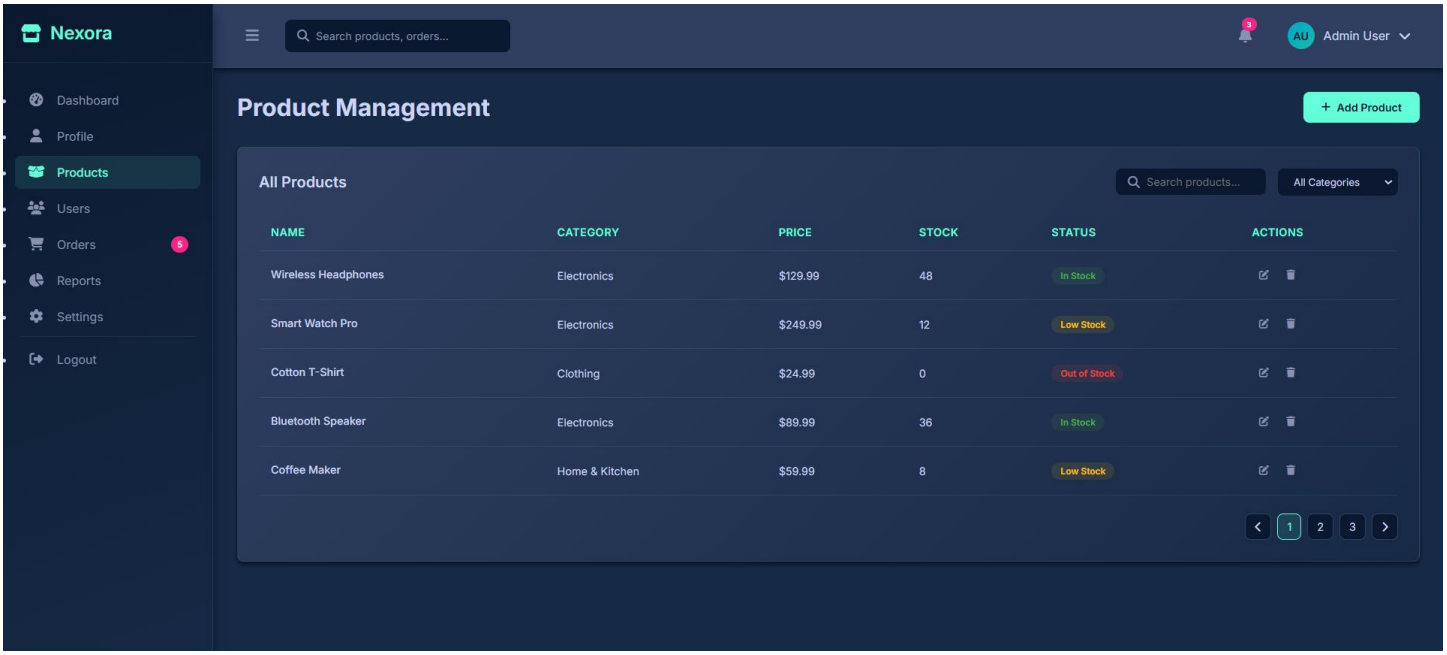

Figure 13 Product Management

10. Vendor Earning and Payout

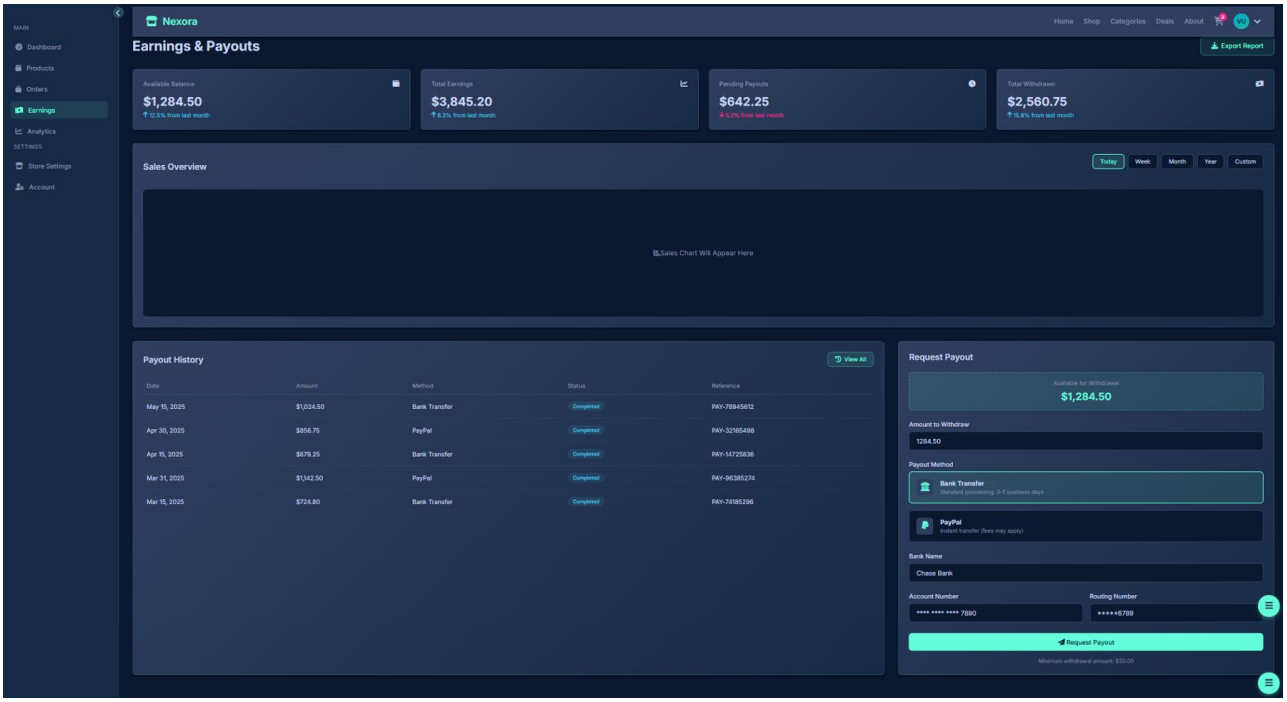

Figure 14 Vendor Earning and Payout

11. Admin Profile and Setting

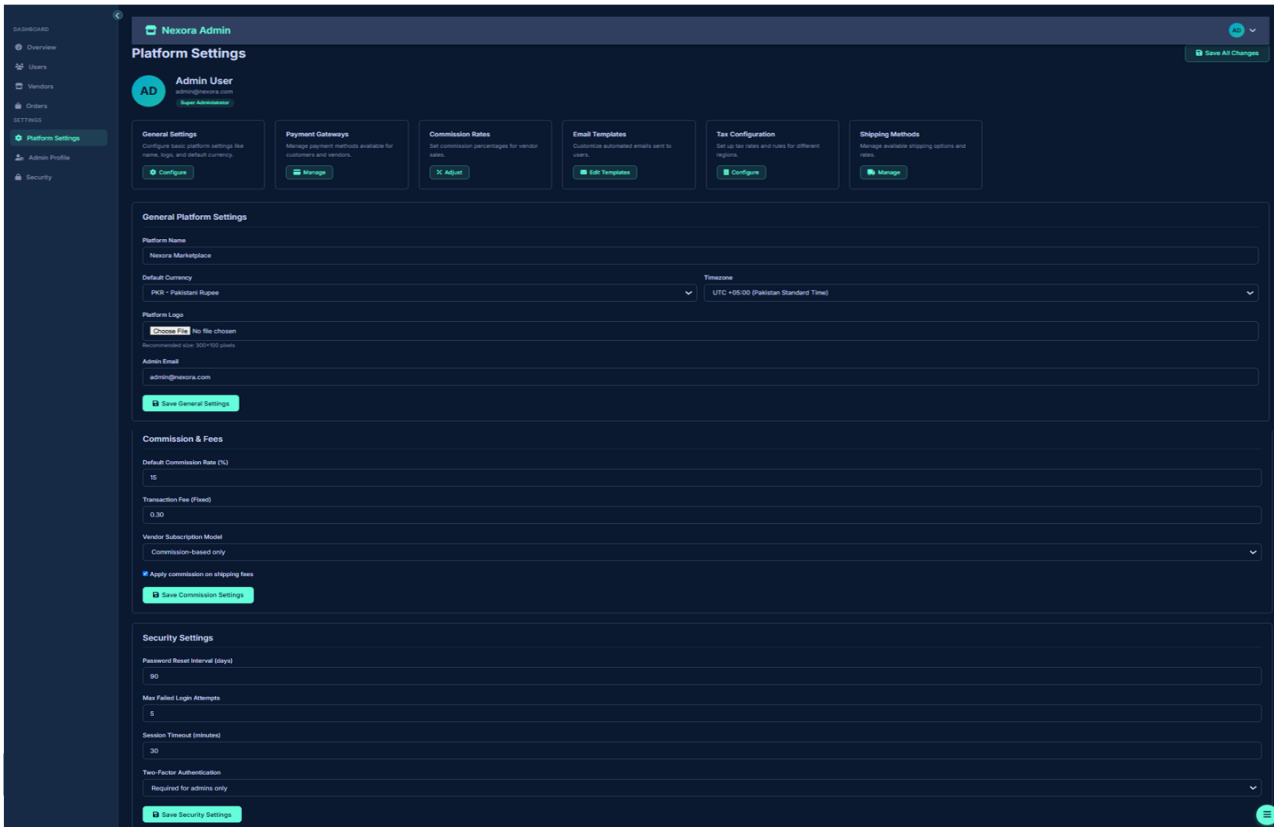

Figure 15 Admin profile and Setting

12. Vendor Profile and Setting

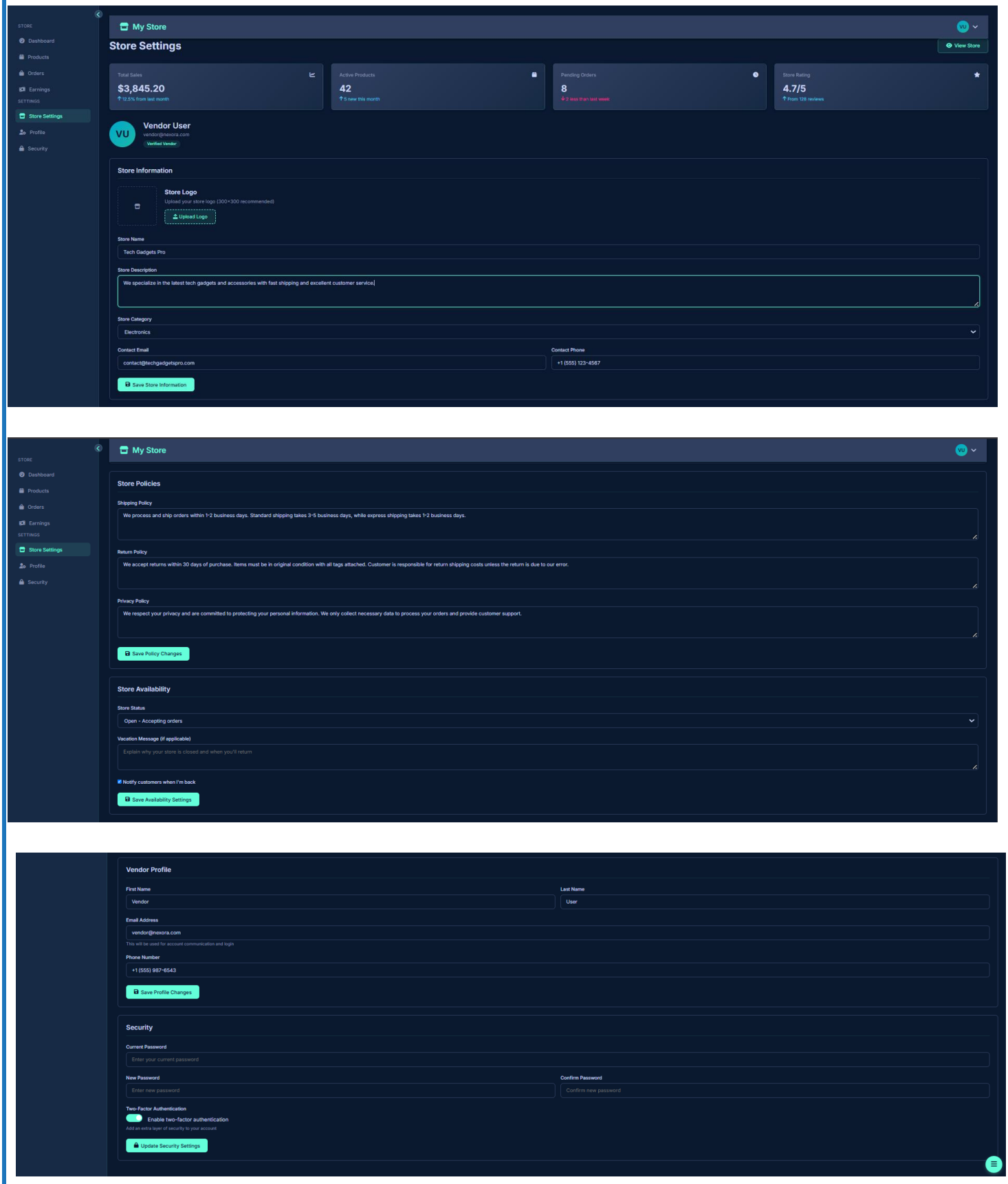

Figure 16 Vendor Profile and Setting

13. Customer Profile and Setting

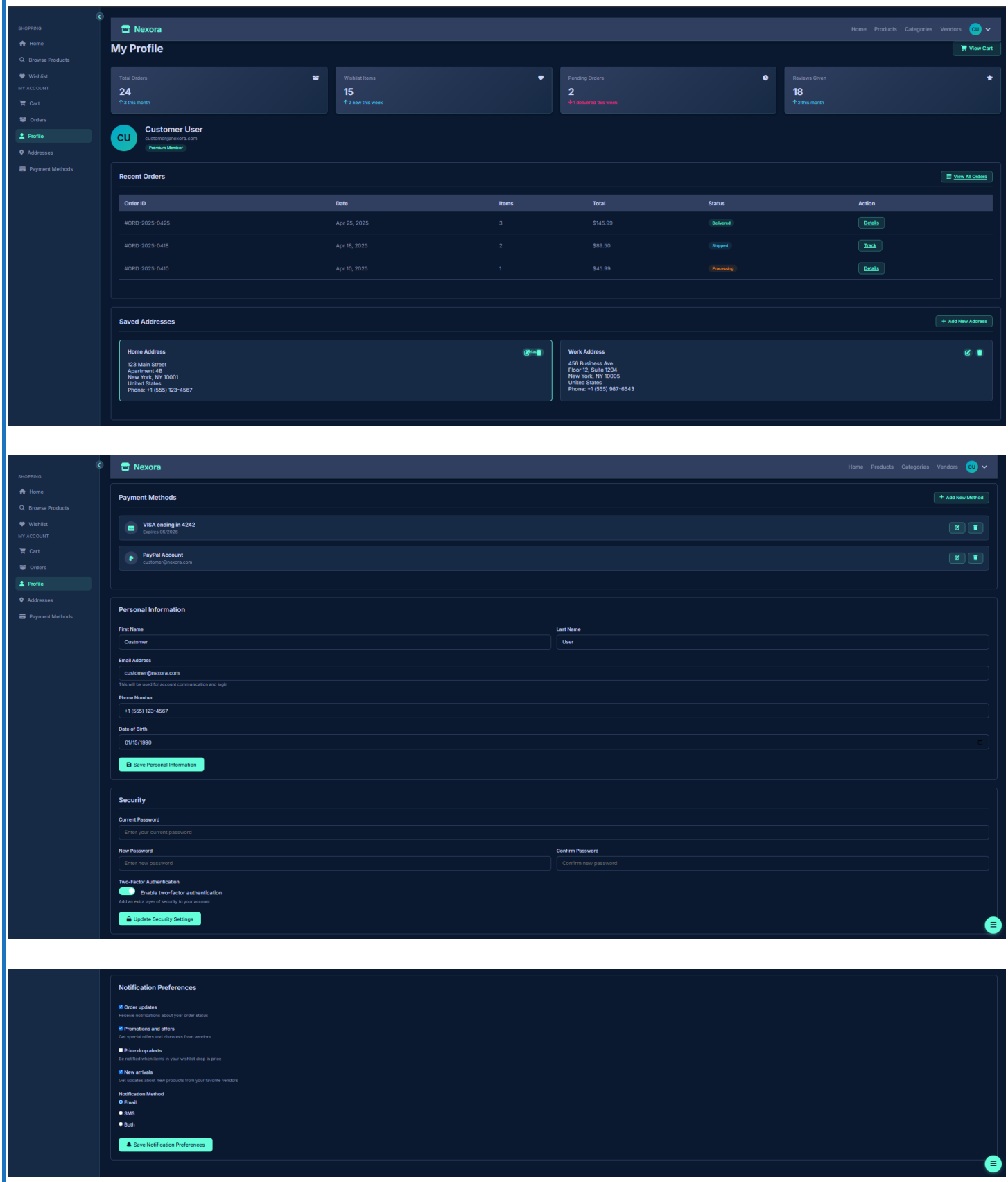

Figure 17 Customer Profile and Setting

References

- Sommerville, I. (2015). *Software Engineering (10th Edition)*. Pearson.
- W3Schools. *HTML, CSS, and JavaScript*. https://www.w3schools.com
- Diprella. https://diprella.com/sign-in
- CSS Tricks. https://css-tricks.com
- Stack overflow. https://stackoverflow.com