



UE21CS352B - Object Oriented Analysis & Design using Java

Mini Project Report

“Thrift Store”

Submitted by:

Namita Patil	PES1UG21CS357
Nisha E	PES1UG22CS822
Mallidi Sirija Reddy	PES1UG21CS923
Navya Peram	PES1UG21CS924

6th Semester F Section

Prof. Course Instructor Name:

Ms. Bhargavi Mokashi

Assistant Professor

January - May 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100ft Ring Road, Bengaluru – 560 085, Karnataka, India

INTRODUCTION

1.1 Overview:

The thrift store project is a web application developed using Java with Eclipse as the Integrated Development Environment (IDE) and MySQL as the backend database. It aims to provide users with a convenient platform to browse, search for, and purchase second-hand items. The system includes essential functionalities such as user authentication, browsing items, searching for specific products, adding items to a shopping cart, processing orders, and tracking the status of placed orders.

1.2 Purpose:

The purpose of the project is to develop an online platform for a thrift store, providing customers with the ability to browse, purchase, and donate pre-owned items. The platform aims to promote sustainability by facilitating the reuse and recycling of goods while offering customers affordable shopping options.

1.3 Scope:

User Management:

- Registration process with required details.
- Login functionality using email and password.
- Access to order history for users to view past orders.

Product Management:

- Administrator capability to add, remove, and update product details: name, description, price, and quantity.
- User interface for browsing available products.
- Product details view and search functionality for users.

Order Management:

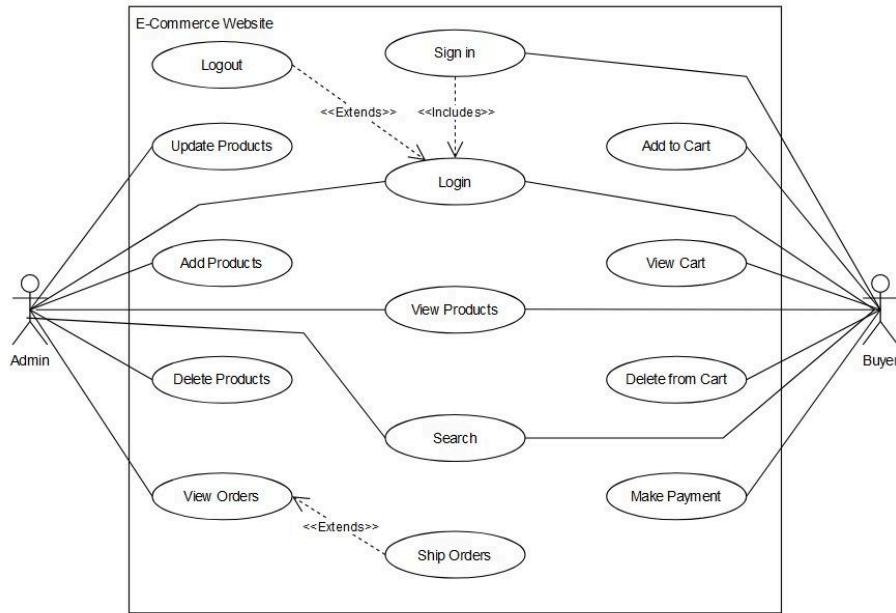
- Cart functionality for users to add products and place orders.
- Administrative access to view all orders.
- Tracking of transaction details: order IDs, transaction IDs, product IDs, quantities, and amounts.

Email Notification:

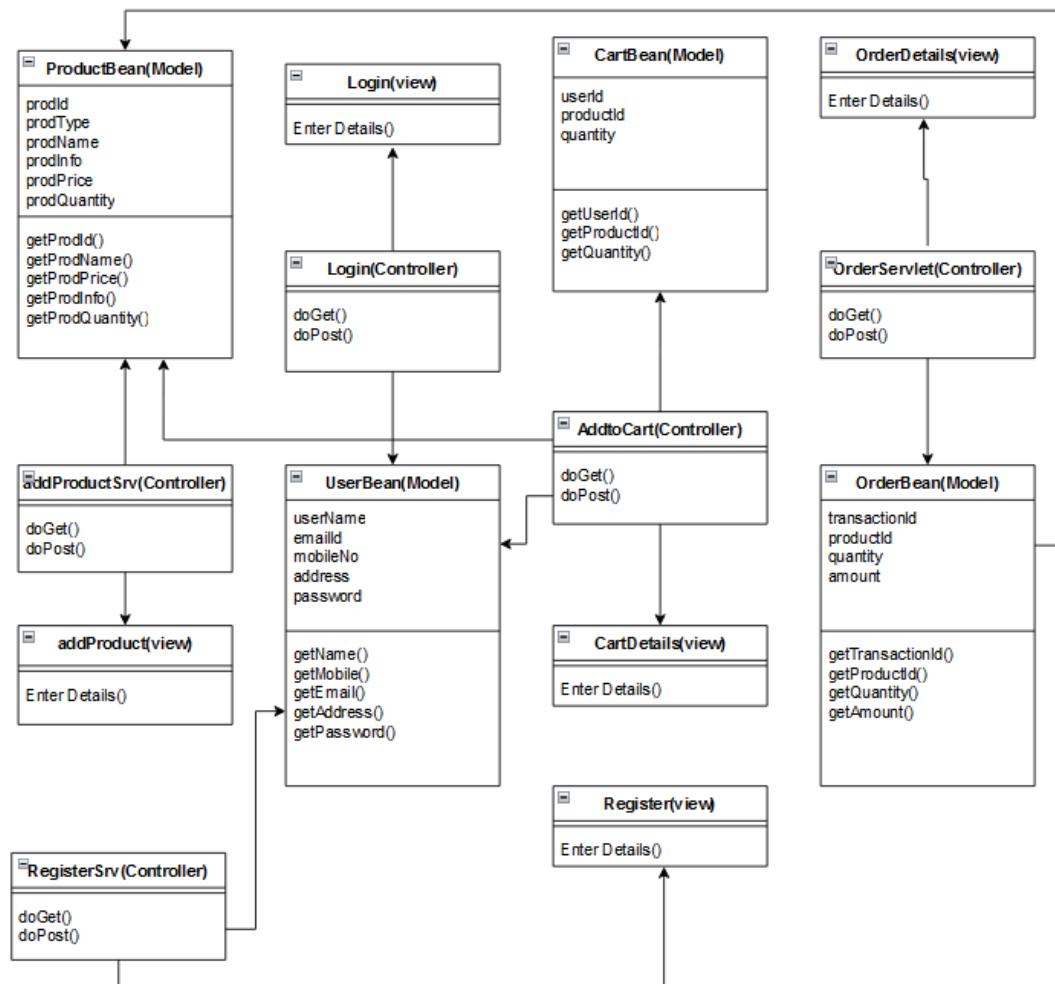
- Automated email notifications sent to users upon successful shipment of orders.

REQUIREMENT ANALYSIS

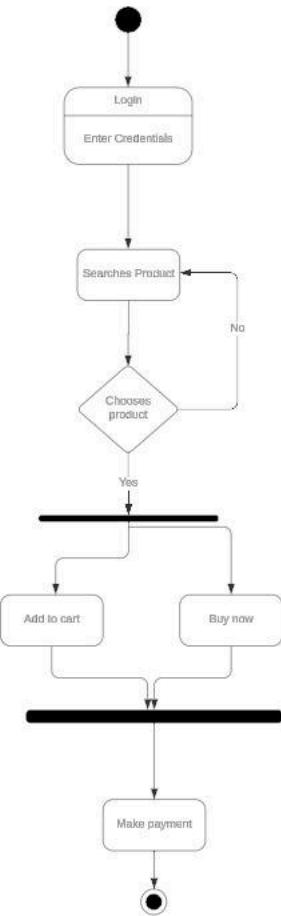
Use Case Diagram



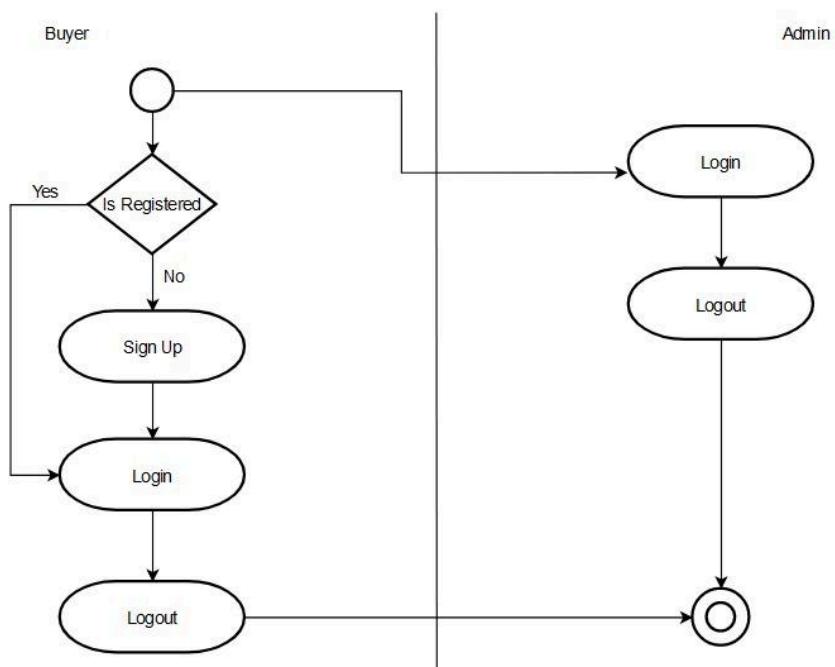
Class Diagram

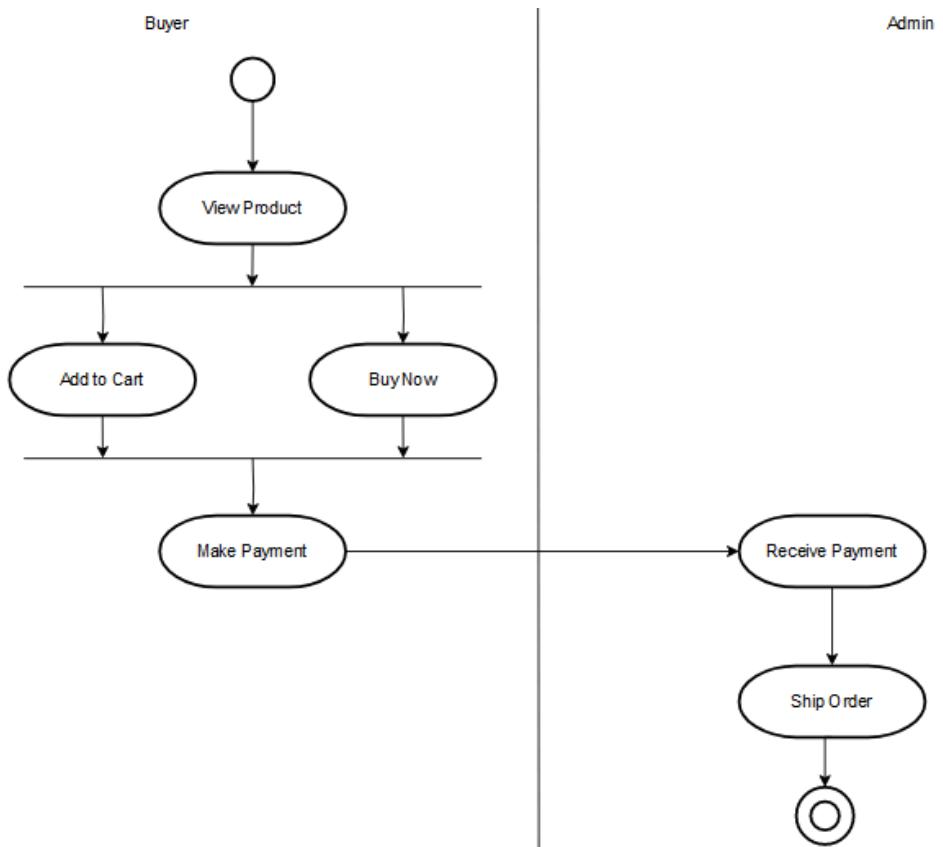


State Diagram

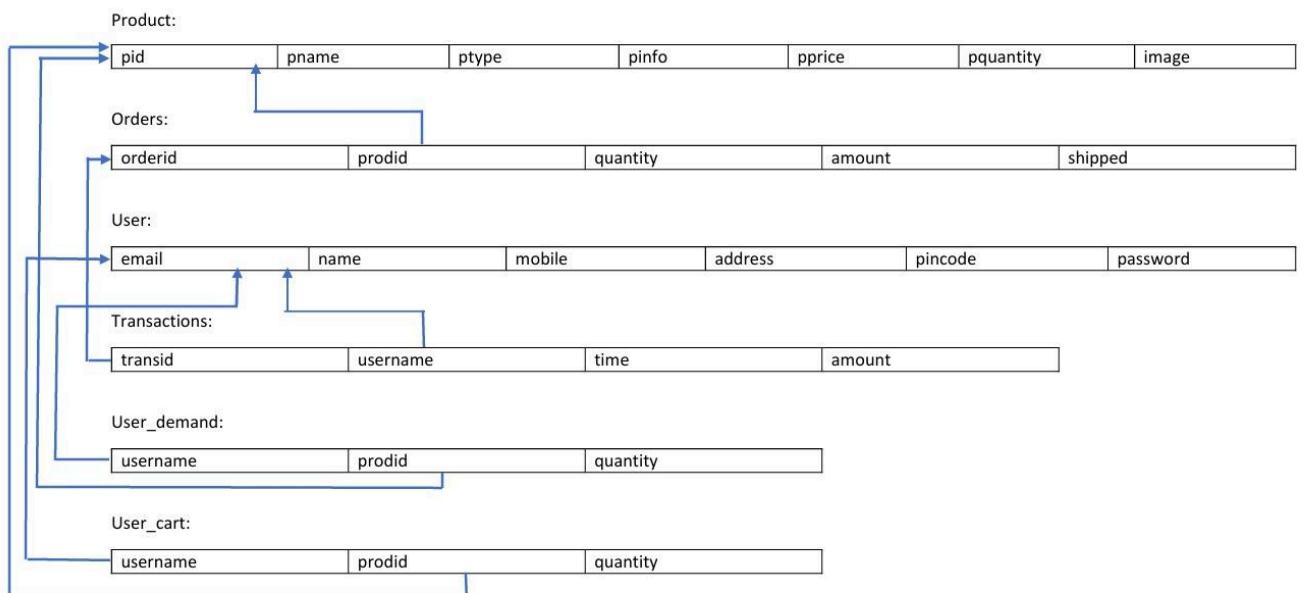


Activity Diagrams

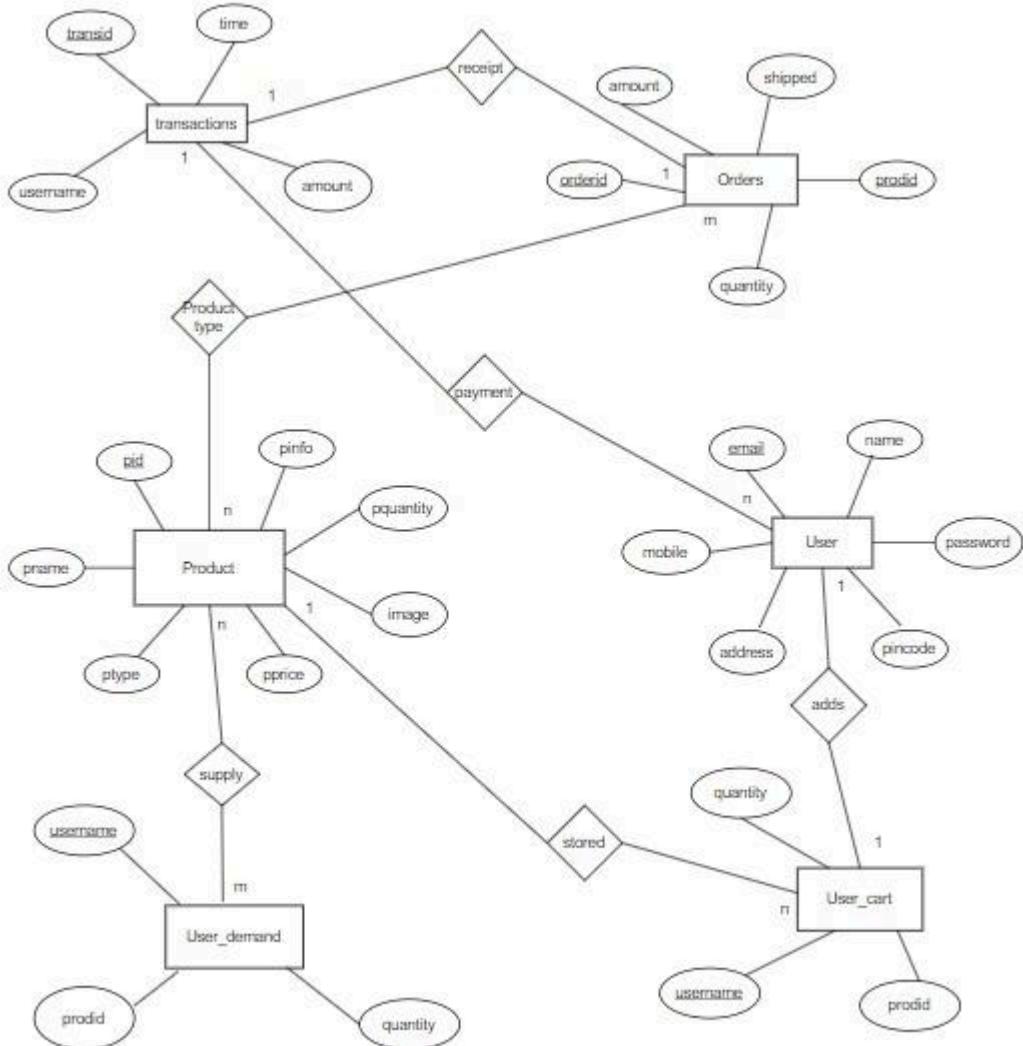




Schema Diagram



E R Diagram



USAGE OF DESIGN PATTERNS

Factory Pattern:

- The ProductService interface serves as a factory method abstraction, allowing the creation of instances of concrete classes like ProductServiceImpl.
- Each implementation of ProductService can act as a factory for creating objects related to product management.
- ProductServiceImpl provides concrete implementations for methods like addProduct, removeProduct, updateProduct, etc., handling product-related operations and object creation.
- In the TransactionBean class, constructors serve as factory methods for creating transaction objects, encapsulating the logic for generating unique transaction IDs and timestamps.

Builder Pattern:

- ProductBean utilises the Builder pattern, although it doesn't use a separate builder class.
- The constructor of ProductBean allows constructing instances with all necessary attributes, while not all attributes are required for every instance.
- Users can create ProductBean objects with partial information by passing only the required attributes to the constructor.
- UserBean class also follows a similar pattern, enabling construction of complex objects with optional parameters without a dedicated builder class.

Façade Pattern:

- The Facade pattern simplifies the usage of a subsystem by providing a unified interface.
- ShowImage servlet acts as a Facade, offering a simple interface for clients (web browsers) to access and display product images.
- Behind the scenes, ShowImage servlet encapsulates the complexity of retrieving and formatting images for display.

Command Pattern:

- The Command pattern encapsulates requests as objects, supporting parameterization, queuing, and undoable operations.
- ShipmentServlet servlet exemplifies the Command pattern by encapsulating the request to ship an order.

- ShipmentServlet acts as the Invoker, while OrderServiceImpl (not shown) likely serves as the Receiver, handling order shipment operations.
- This design allows requests to be queued and executed independently and supports undoing operations if necessary.

USAGE OF MVC ARCHITECTURE

Model:

- Classes such as UserBean, ProductBean, OrderBean, and TransactionBean serve as the core components of the model.
- These classes encapsulate data related to users, products, orders, and transactions, respectively.
- Service interfaces like UserService, ProductService, OrderService, and TransService define operations pertinent to users, products, orders, and transactions.
- Implementations of these interfaces (e.g., UserServiceImpl, ProductServiceImpl) handle the business logic and interaction with the underlying data storage.

View:

- JSP files, such as login.jsp, register.jsp, userHome.jsp, adminViewProduct.jsp, orderDetails.jsp, etc., form the presentation layer of the application.
- These files contain HTML markup combined with JSP tags to dynamically generate content based on the data provided by the controller.
- The view is responsible for presenting data to users in a human-readable format and capturing user interactions, such as form submissions and clicks.

Controller:

- Servlets, including LoginSrv, RegisterSrv, LogoutSrv, AddProductSrv, RemoveProductSrv, UpdateProductSrv, OrderServlet, ShipmentServlet, etc., serve as the intermediary between the model and the view.
- These servlets receive requests from clients, interpret user input, and invoke appropriate actions on the model to perform business operations.
- The controller determines the appropriate view to render based on the outcome of these operations, managing the flow of information between the model and the view.

- It handles user requests, updates the model accordingly, and selects the view to display, thus facilitating the interaction between users and the application's underlying logic.

Design Principles:

Single Responsibility Principle (SRP):

- Each class embodies a single responsibility, fostering clarity and maintainability.
- The AddtoCart servlet, for instance, exclusively manages adding products to the cart and related operations.
- Service classes (CartServiceImpl, ProductServiceImpl, DemandServiceImpl) encapsulate precise business logic, ensuring responsibility allocation.
- The ProductBean class solely represents product data, while servlets handle HTTP requests related to product management.
- The UserBean class effectively focuses on representing user data, encapsulating user-related properties and manipulation methods.

Open/Closed Principle (OCP):

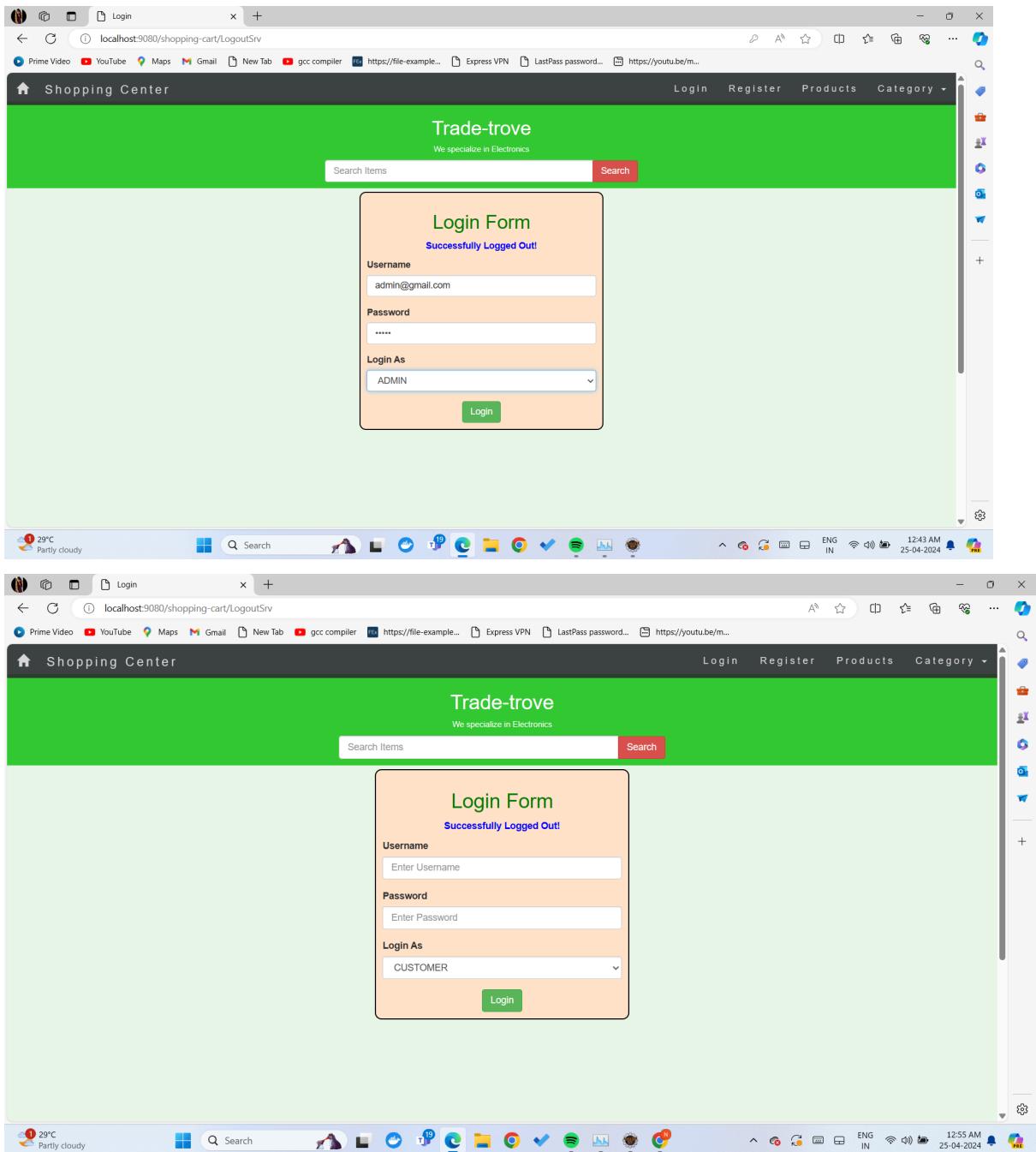
- The codebase facilitates extension without modification.
- Interfaces (CartService) define contracts for various operations, enabling seamless addition of new functionalities without altering existing code.
- New functionalities can be seamlessly integrated by implementing these interfaces.
- Registration of users and credential validation can be added without modifying the existing codebase.

Dependency Inversion Principle (DIP):

- The codebase favors interfaces over concrete implementations, enhancing flexibility.
- Interfaces (e.g., ProductService) facilitate substitution of implementations, promoting adaptability.
- By depending on abstractions, the codebase adheres to DIP principles, enhancing maintainability and extensibility.

SNAPSHOTS

1. Login and logout as Admin:



2. Add products as Admin:

The screenshot shows a web browser window with the URL `localhost:9080/shopping-cart/addProduct.jsp`. The page title is "Add Product". The main content area displays the "Product Addition Form" for the "Trade-trove" website. The form fields include:

- Product Name:** samsung tv
- Product Type:** TV (selected from a dropdown)
- Product Description:** 52 inches smart tv
- Unit Price:** 42000
- Stock Quantity:** 2
- Product Image:** Choose File (tv.jpeg)

At the bottom of the form are two buttons: "Reset" and "Add Product".

The screenshot shows the same web browser window after the product has been added. The message "Product Added Successfully with Product Id: P20240425124942" is displayed above the form fields. The other fields remain the same as in the first screenshot.

3. Updating products as Admin:

The screenshot shows a web browser window with two tabs open. The active tab is titled "Update Product" and displays a "Product Update Form" for an iPhone 13 Pro. The form includes fields for Product Name (APPLE iPhone 13 Pro (Graphite, 512)), Product Type (MOBILE), Product Description (iPhone 13. boasts an advanced dual-camera system that allows you to click mesmerising pictures with immaculate clarity.), Unit Price (125999.0), and Stock Quantity (1000). There are "Cancel" and "Update Product" buttons at the bottom. The background shows a navigation bar with links like "Products", "Category", "Stock", "Shipped", "Orders", "Update Items", and "Logout". The header of the page reads "Trade-trove" and "We specialize in Electronics".

Product Name: APPLE iPhone 13 Pro (Graphite, 512)

Product Type: MOBILE

Product Description: iPhone 13. boasts an advanced dual-camera system that allows you to click mesmerising pictures with immaculate clarity.

Unit Price: 125999.0

Stock Quantity: 1000

Cancel Update Product

The screenshot shows a second tab titled "Update Product" displaying a simplified "Product Update Form" with a single input field for "Product Id" and "Enter Product Id". Below the input field are "Cancel" and "Update Product" buttons. The rest of the interface is identical to the first screenshot, featuring the "Trade-trove" header and navigation bar.

Product Id: Enter Product Id

Cancel Update Product

The screenshot shows a "Contact" section with a note "Fan? Drop a note." and "Bitter Code". It includes input fields for "Name" and "Email". Below the input fields, it says "We love our fans!"

Fan? Drop a note.

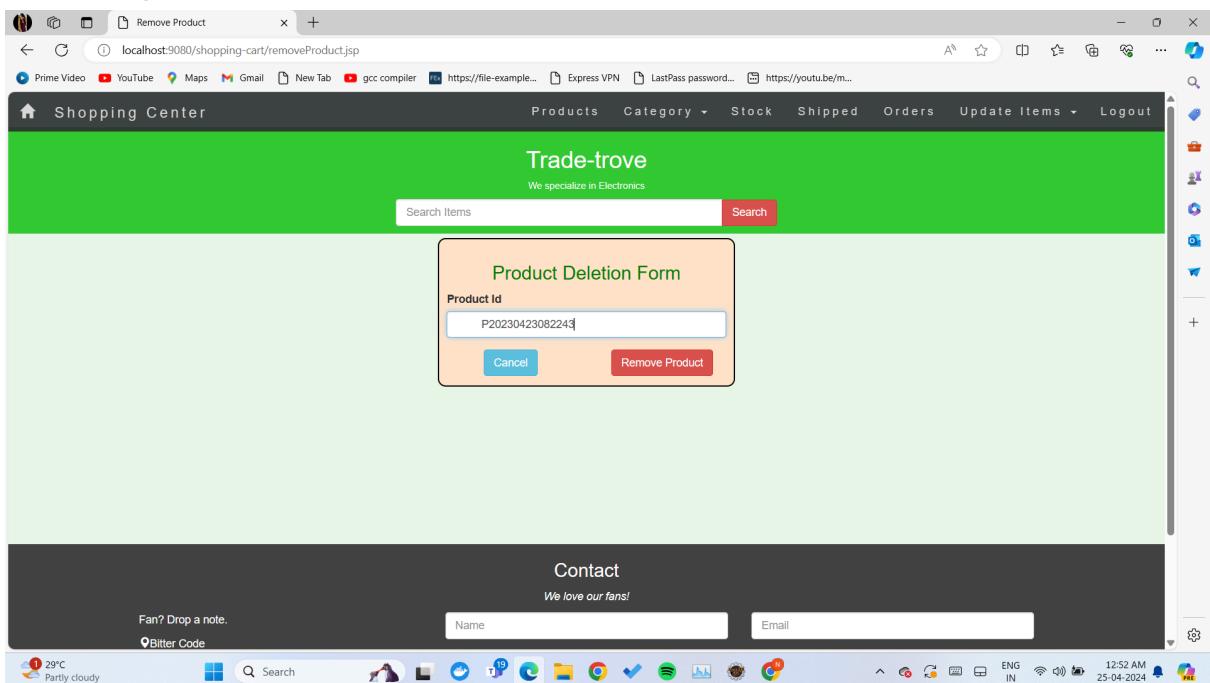
Bitter Code

Name

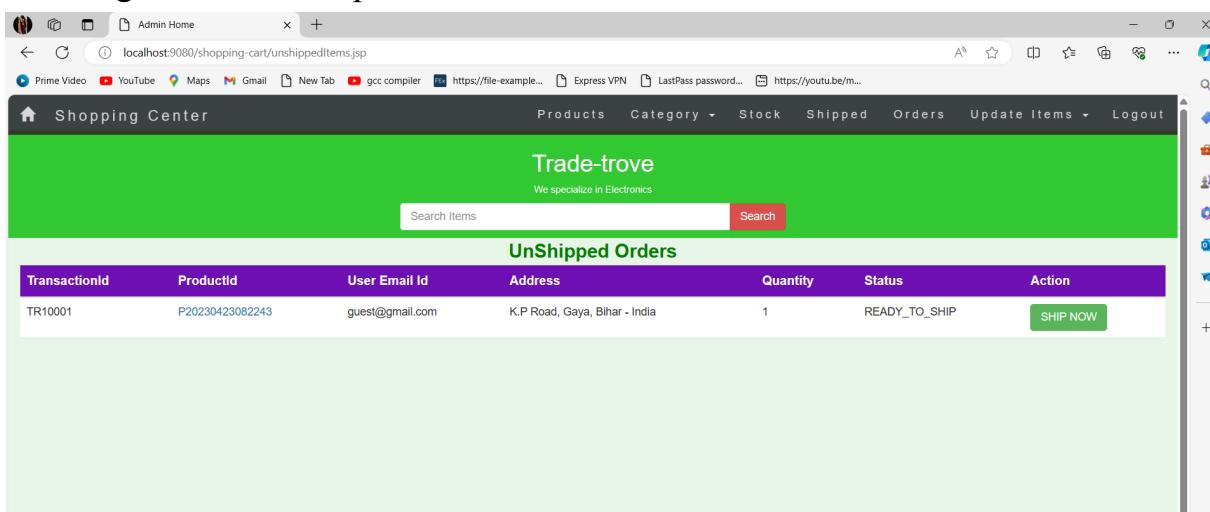
Email

We love our fans!

4. Removing Products as Admin:



5. Tracking orders and shipments as Admin:



UnShipped Orders

TransactionId	ProductId	User Email Id	Address	Quantity	Status	Action
TR10001	P20230423082243	guest@gmail.com	K.P Road, Gaya, Bihar - India	1	READY_TO_SHIP	<button>SHIP NOW</button>

6. Updating shipment status and product stock as Admin:

UnShipped Orders

TransactionId	ProductId	User Email Id	Address	Quantity	Status	Action
TR10001	P20230423082243	guest@gmail.com	K.P Road, Gaya, Bihar - India	1	READY_TO_SHIP	<button>SHIP NOW</button>

Stock Products

Image	ProductId	Name	Type	Price	Sold Qty	Stock Qty	Actions
	P20230423082243	APPLE iPhone 13 Pro (Grap..)	MOBILE	125999.0	1	1000	<button>Update</button> <button>Remove</button>
	P20230423083830	HP Intel Core i5 11th Gen..	LAPTOP	40990.0	0	1000	<button>Update</button> <button>Remove</button>
	P20230423084143	LED Smart Google TV ..	TV	41999.0	0	1000	<button>Update</button> <button>Remove</button>
	P20230423084144	MOTOROLA G32 Mobile..	MOBILE	11999.0	0	1	<button>Update</button> <button>Remove</button>
	P20230423084145	realme NEO 80 cm (32 inch..)	TV	11999.0	0	1000	<button>Update</button> <button>Remove</button>
	P20230423084146	REDMI Note 12 Pro 5G..	MOBILE	24999.0	0	1000	<button>Update</button> <button>Remove</button>
	P20230423084147	Google Pixel 6a (Charcoal..)	MOBILE	27999.0	0	1000	<button>Update</button> <button>Remove</button>

7. Login/ Register as User

The image contains two screenshots of a web browser window, likely Microsoft Edge, displaying user authentication pages for a website named "Trade-trove".

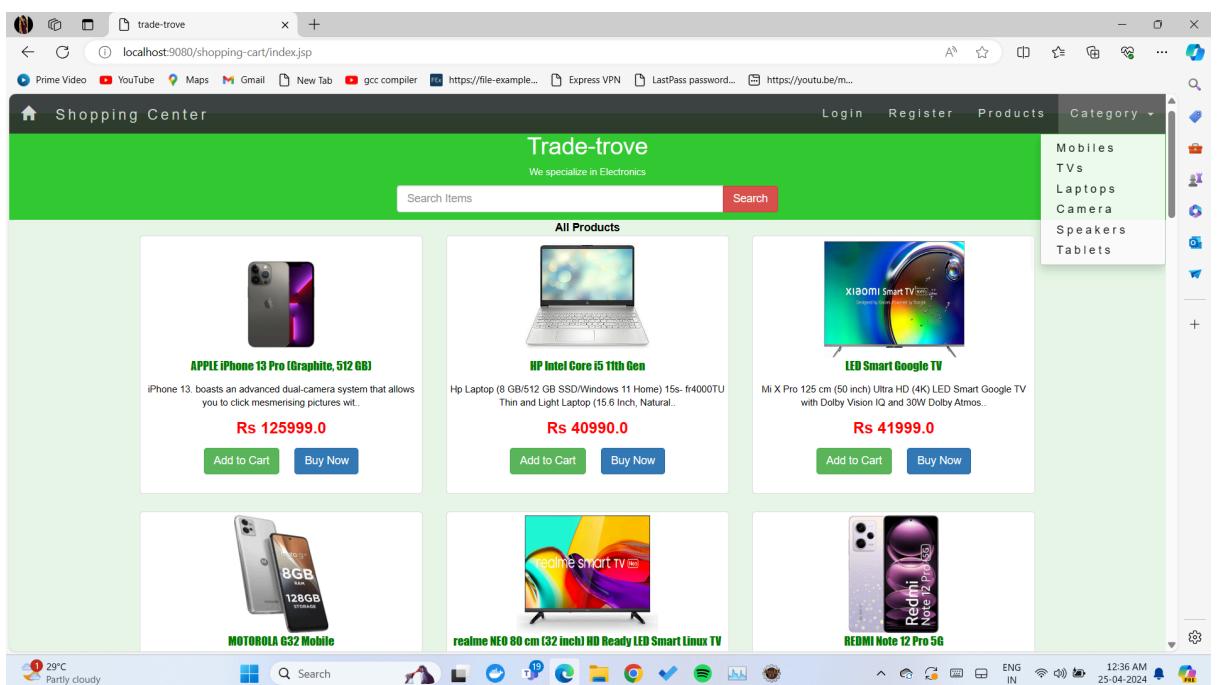
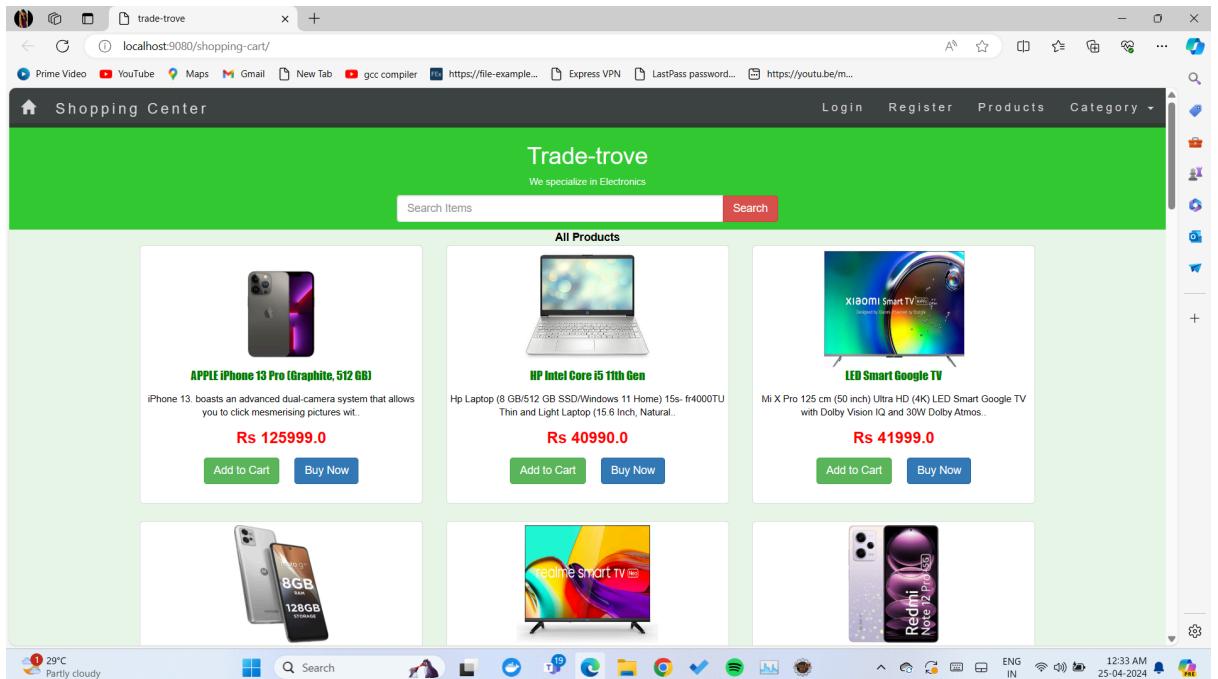
Screenshot 1: Login Form

The top screenshot shows the "Login Form" page. The URL in the address bar is `localhost:9080/shopping-cart/login.jsp`. The page header includes "Shopping Center" and navigation links for "Login", "Register", "Products", and "Category". The main content area is titled "Login Form" and contains fields for "Username" (with value `nisha@gmail.com`) and "Password" (with value `.....!`). A dropdown menu labeled "Login As" is set to "CUSTOMER". A green "Login" button is at the bottom.

Screenshot 2: Registration Form

The bottom screenshot shows the "Registration Form" page. The URL in the address bar is `localhost:9080/shopping-cart/register.jsp`. The page header includes "Shopping Center" and navigation links for "Login", "Register", "Products", and "Category". The main content area is titled "Registration Form" and contains fields for "Name" and "Email" (both empty), "Address" (empty), "Mobile" and "Pin Code" (both empty), and "Password" and "Confirm Password" (both empty). A red "Reset" button and a green "Register" button are at the bottom.

8. Navigate the homepage using categories:



9. Search for products as User:

Screenshot of a web browser showing the Trade-trove shopping cart page. The search bar at the top shows 'mobiles'. Below it, a message says 'No items found for the search "mobiles"'. There are three product cards displayed:

- APPLE iPhone 13 Pro (Graphite, 512 GB)**: Price Rs 125999.0. Buttons: Add to Cart, Buy Now.
- HP Intel Core i5 11th Gen**: Price Rs 40990.0. Buttons: Add to Cart, Buy Now.
- XIAOMI Smart TV**: Price Rs 41999.0. Buttons: Add to Cart, Buy Now.

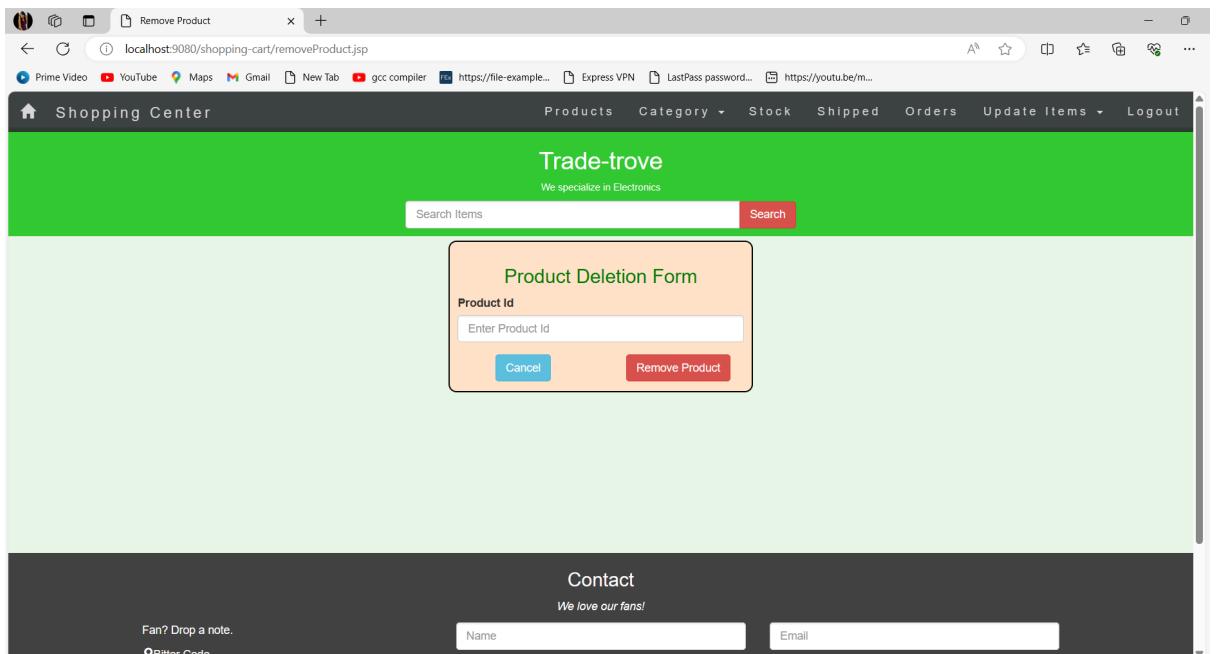
The browser's taskbar at the bottom shows various icons and the date/time: 25-04-2024, 12:38 AM.

10. Add items to Cart:

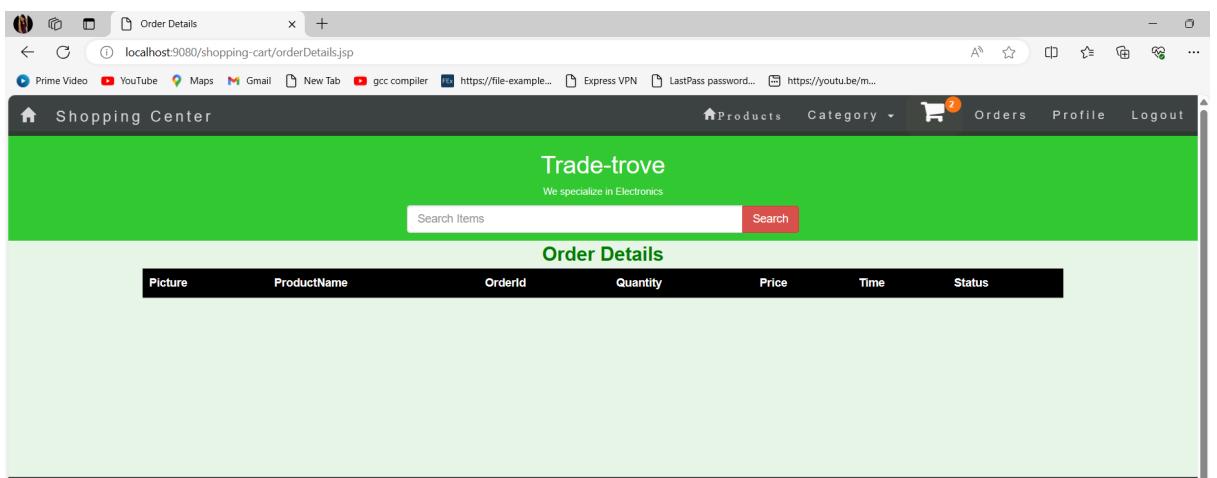
Screenshot of a web browser showing the 'Cart Details' page. The cart contains two items:

Picture	Products	Price	Quantity	Add	Remove	Amount	
	Realme Pad 3 GB	13999.0	<input type="text" value="1"/> Update			13999.0	
	Mivi Roam2 5 W Bluetooth Speaker	899.0	<input type="text" value="1"/> Update			899.0	
Total Amount to Pay (in Rupees)						14898.0	

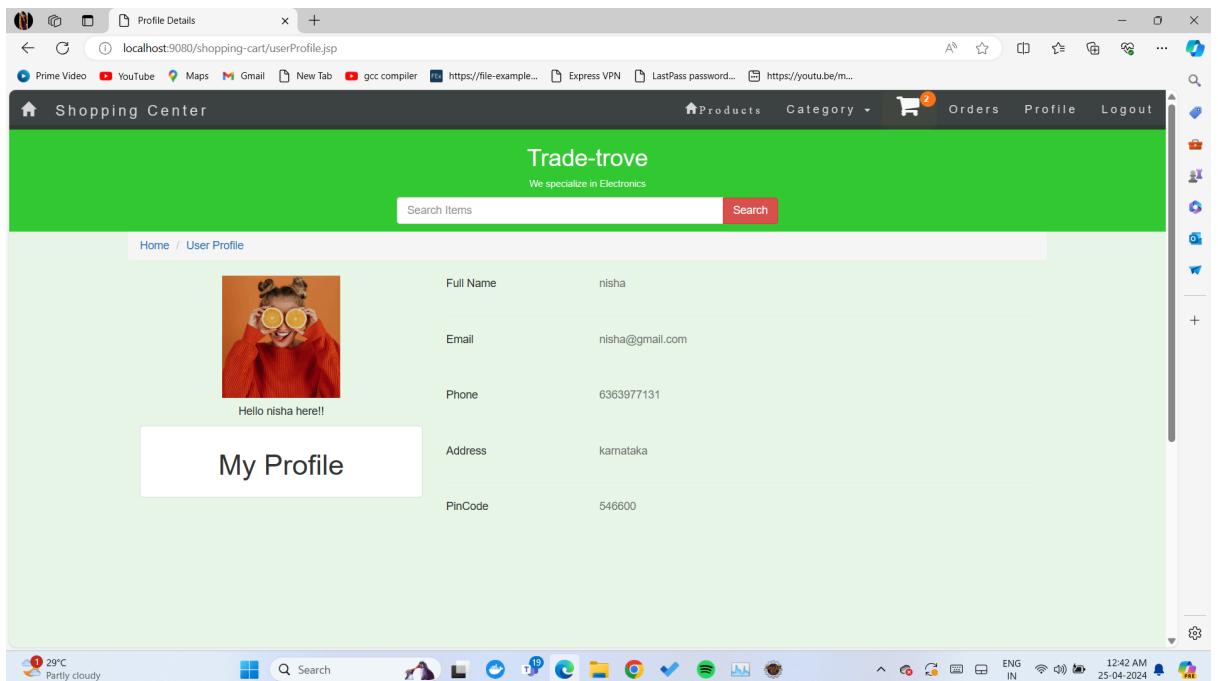
The browser's taskbar at the bottom shows various icons and the date/time: 25-04-2024, 12:40 AM.



11. Track order status:



12. View Profile as User:



REPO LINK: <https://github.com/NishaEBK/tradetrove>

REFERENCES

- Internet & World Wide Web: How to Program Deitel, PJ Deitel.
- Bitter Code YouTube Channel.
- Database System Concepts, by Silberschatz, Sudarshan, and Korth.
- Fundamentals of Database Systems, RamezElmasri and Shamkant B. Navathe, 7th Edition. 2017, Pearson.