A project report on

ECOMMERCE APPLICATION FOR SHOPPING

Submitted in partial fulfillment for the award of the degree of

Bachelor of Technology

In

Computer Science and Engineering with Specialization in Internet of Things

by

ANANYA ADITI (20BCT0170)

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

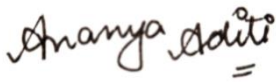
SCOPE

April, 2024

DECLARATION

I here by declare that the thesis entitled “ECOMMERCE APPLICATION FOR SHOPPING” submitted by me, for the award of the degree of Specify the name of the degree VIT is a record of bonafide work carried out by me under the supervision of Guide Name

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.



Place: Vellore

Date: 25/04/2024 **Signature of the Candidate**

Internship completion certificate

Chubb Business Services India Private Limited
(formerly known as Chubb Business Services India LLP)
Fifteenth floor, Unit No. 3, Parcel – 4, Octave Block,
Knowledge City, Plot No 2, Phase 1, Survey No.83/1,
Raidurg Village, Serilingampally Mandal,
Hyderabad – 500081

CHUBB

Date : Apr-23-2024

TO WHOMEVER IT MAY CONCERN

This is to certify that Ananya Aditi is currently undergoing her internship at Chubb India, Hyderabad, from December 26, 2023, and tentatively until June 30, 2024.

During the internship, he has demonstrated diligence and enthusiasm in learning and delivering projects on time. Ananya has successfully completed and delivered a capstone project as part of his internship.

We wish Ananya the very best for his career and future endeavors.

For Chubb Business Services India Private Limited

Lucky Misra
Lead Campus Program

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Lucky Misra, Program Manager, Chubb Business Services India LLP, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of IT.

I would like to express my gratitude to Dr. G. Viswanathan, Honourable Chancellor, Mr. Sankar Viswanathan, VP, Dr. Sekar Viswanathan, VP, Dr. G.V. Selvam, VP, Dr. V.S. Kanchana Bhaaskaran, Vice Chancellor, Dr. Partha Sharathi Mallick, Pro-Vice Chancellor, and Dr Ramesh Babu K, Dean, SCOPE for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to PAT - Placement and Training, Dr. V. Samuel Rajkumar, Dr. Sharmila Banu K, Head of the Department (HoD), IOT and all teaching staff and members working as limbs of our university for their not-self-centred enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would also like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last, but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: Vellore

Date: 25/04/2024 **Name of the student: Ananya Aditi**

ABSTRACT

The college campus eating experience is often limited to a small physical space, which can lead to overcrowding and long waiting times. As the demand for canteen services increases, it becomes crucial for the hospitality industry to adapt and provide convenient options for ordering and delivering food. This project aims to address these challenges by proposing the design and implementation of a website, "Underbelly Express," for the college canteen. The website will offer students the opportunity to order their favorite food and satisfy their cravings from anywhere on campus. This report analyzes the current shortcomings of the canteen system, identifies user needs and expectations, and explores the technology and infrastructure available for implementation. The proposed website design includes features that improve operational efficiency and enhance the overall customer experience. Through an implementation plan, the report outlines the steps necessary for successful development. Additionally, evaluation methods and feedback collection strategies are discussed to ensure continuous improvement and customer satisfaction. The website's potential benefits and recommendations for future expansion are also highlighted. Ultimately, the proposed design and implementation of the Underbelly Express canteen website aim to provide students with a seamless and convenient ordering platform that enhances their eating experience on campus.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1	Abstract	1
2	**CHAPTER-1:** **PROJECT DESCRIPTION AND OUTLINE** 1.1 Introduction 1.2 Motivation for the work 1.3 Problem Statement 1.4 Objective of the work 1.5 Summary	5 5 5 5 6 7
3	**CHAPTER-2: RELATED WORK INVESTIGATION** 2.1 Existing Approaches/Methods 2.2 Pros and cons of the stated Approaches/Methods	8 9

4	**CHAPTER-3: REQUIREMENT ARTIFACTS**	
	3.1 Introduction	10
	3.2 Hardware and Software requirements	10-11
	3.3 Specific Project requirements	11
	3.3.1 Data requirement	12
	3.3.2 Functions requirement	12
	3.3.3 Performance requirement	13
	3.3.4 Security Requirements	13
	3.3.5 Look and Feel Requirements	13
	3.4 Summary	13
5	**CHAPTER-4: DESIGN METHODOLOGY AND ITS NOVELTY**	
	4.1 Methodology and goal	14
	4.2 Functional modules design and analysis	15-16
	4.3 Software Architectural designs	18-20
	4.4 User Interface designs	20-27
	4.5 Summary	27
6	**CHAPTER-5: TECHNICAL IMPLEMENTATION & ANALYSIS**	
	5.1 Outline	28-30
	5.2 Technical coding and code solutions	30-31
	5.3 Prototype submission	31-33
	5.4 Summary	33-35

6	**CHAPTER-6:** **PROJECT OUTCOME AND APPLICABILITY** 6.1 key implementations outline of the System 6.2 Significant project outcomes 6.3 Project applicability on Real-world applications 6.4 Inference	 36-37 38-39 40-41 41
7	**CHAPTER-7:** **CONCLUSIONS AND RECOMMENDATION** 7.1 Outline 7.2 Limitation/Constraints of the System 7.3 Future Enhancements 7.4 Inference	 42 43 44 45
	APPENDIX A – Screen Shots	46-50
	APPENDIX B – Coding	51-67
	REFERENCES	68

CHAPTER 1
PROJECT DESCRIPTION AND OUTLINE

1.1 INTRODUCTION

The introduction to the e-commerce application project underscores the profound transformation e-commerce has brought to consumer shopping habits, making it more accessible and convenient than ever before. The project is driven by the growing demand for e-commerce applications, which has been fueled by the digital revolution and the increasing reliance on online shopping. This project aims to address the evolving needs of modern consumers by designing a comprehensive e-commerce solution that not only meets the current demands of online shopping but also sets a benchmark for future e-commerce solutions. The focus is on creating a platform that is user-friendly, secure, and efficient, leveraging technology to enhance the shopping experience for both consumers and businesses.

1.2 MOTIVATION FOR WORK

The motivation behind this project is deeply rooted in the increasing reliance on e-commerce platforms for shopping, a trend driven by a landscape characterized by a wide range of products, competitive pricing, and the need for efficient order management. The proposed e-commerce application seeks to address these challenges by offering a robust platform that facilitates easy navigation, secure transactions, and efficient order processing. This project is motivated by the desire to enhance customer satisfaction, reduce shopping time, and provide a convenient platform for both consumers and businesses, thereby streamlining the shopping experience. The project aims to leverage technology to create a seamless shopping experience, making it easier for consumers to find, purchase, and receive products, while also providing businesses with efficient tools to manage their online presence and customer relationships.

1.3 PROBLEM STATEMENT

The current college canteen system poses several challenges that hinder the overall eating experience for students. The limited physical space and increased demand often lead to overcrowding, long waiting times, and dissatisfaction among canteen-goers. Additionally, the lack of a convenient and efficient ordering mechanism restricts students' ability to place food orders and have them delivered to their preferred locations on campus. The absence of a dedicated website for the canteen, such as Underbelly Express, further exacerbates these issues, as students are compelled to visit the canteen in person, resulting in inconvenience and wasted time. To address these challenges, there is a need to design and implement a website that offers a seamless online ordering system, streamlining the canteen operations, reducing waiting times, and providing a more satisfying and convenient eating experience for college students. By tackling these issues, the proposed website aims to enhance customer satisfaction, improve operational efficiency, and meet the growing demand for an accessible and efficient canteen

service on campus.

1.4 OBJECTIVE OF THE WORK

The primary objective of this project is to design and build an e-commerce application that offers a comprehensive solution for online shopping. The specific objectives of the project include:

1. Admin Level Module:
    - Develop an account management system for different user levels, including parent accounts for organizations and sub-accounts for employees.
    - Implement an inventory management system that includes product categorization, customization options, and approval workflows.
    - Design a pricing system that supports various pricing strategies, including discounts, vouchers, and tax calculations.
    - Create a complete order management system that handles order statuses, error cases, and return/cancel scenarios.
    - Integrate various payment methods to cater to different customer preferences.
    - Manage shipping addresses and order flow with shipping status updates.

2. End User Level:
    - Develop a product listing system that supports quick search, filtering, and sorting based on various attributes.
    - Provide detailed product information through an API, including specifications and other attributes.
    - Implement a shopping cart system that allows users to manage their selections, add or remove products, and view order summaries.
    - Design a checkout process that includes user authentication, shipping information input, payment details, and order confirmation.

By achieving these objectives, the proposed e-commerce application will enhance the shopping experience for consumers, streamline operations for businesses, and contribute to the growth of the e-commerce sector.

1.5 **SUMMARY**

The proposed e-commerce application project is designed to address the evolving needs of consumers and businesses in the digital age by leveraging modern web development technologies. The project aims to create a comprehensive e-commerce solution that not only meets the current demands but also sets a benchmark for future e-commerce platforms.

The application will be developed using Angular v16 for the frontend and .NET for the backend, providing a robust and scalable solution. Angular v16 offers a powerful framework for building dynamic and responsive web applications, while .NET provides a robust backend framework that supports the development of secure, scalable, and efficient web services.

The project will utilize SQL Server as the database, offering a reliable and efficient data storage solution. SQL Server's robust features and performance make it an ideal choice for managing the application's data, including user information, product catalogs, and transaction records.

To facilitate communication between the frontend and backend, the project will implement REST API standards. This approach allows for a clear separation of concerns, with the frontend focusing on user interface and experience, and the backend handling data processing and business logic. The REST API will be developed using .NET, ensuring seamless integration with the SQL Server database and providing a scalable and secure way to access the application's data.

The project will also address the challenges of maintaining a heavy product catalogue, ensuring that the catalogue is searchable and that individual product pages contain rich information. The user experience will be enhanced through responsive web applications, providing a seamless shopping experience across multiple devices.

In summary, the proposed e-commerce application project is poised to revolutionize the online shopping experience by addressing the critical challenges faced by existing e-commerce platforms. Through the use of Angular v16 and .NET for frontend and backend development, SQL Server as the database, and REST API standards for communication, the project aims to create a robust e-commerce solution that meets the current needs of consumers and businesses and sets a new benchmark for future e-commerce solutions.

CHAPTER 2
RELATED WORK INVESTIGATION

2.1 EXISTING APPROACHES/METHODS

The existing approaches and methods related to our e-commerce application project may include:

- E-commerce Platforms: Platforms like BigCommerce, Shopify, and WooCommerce offer comprehensive solutions for building and managing online stores. They provide features such as inventory management, payment gateway integration, SEO tools, and customer relationship management (CRM) systems. These platforms are designed to cater to a wide range of businesses, from small startups to large enterprises [1].
- Software Integrations: The ability to integrate various software into an e-commerce site is crucial for scalability and functionality. Integrations can include payment gateways, shipping providers, marketing automation tools, and more. This flexibility allows businesses to tailor their e-commerce experience to their specific needs and to leverage the best tools in the market [1].
- SEO and User Experience: E-commerce platforms often prioritize SEO and user experience, offering features like CDN (Content Delivery Network) for fast loading times, mobile-optimized designs, and WYSIWYG (What You See Is What You Get) editors for easy content creation. These features are essential for attracting and retaining customers in a competitive online marketplace [1].
- Open-Source Solutions: Open-source e-commerce platforms, such as Magento and PrestaShop, offer a high degree of customization and control over the e-commerce site. They allow businesses to modify the source code to fit their specific requirements. However, they also come with the responsibility of managing security, compliance, and maintenance [1].
- CRM Integration: Integrating a CRM tool within an e-commerce site allows businesses to collect data on customer interactions, send personalized offers, and track each customer's journey through the sales funnel. This integration enhances customer engagement and can lead to increased sales and customer loyalty.

2.2 PROS AND CONS OF THE STATED APPROACHES/ METHODS

When it comes to the existing approaches/methods for an ecommerce website, there are several pros and cons to consider. Here are some of them:

Pros:

- Convenience and Accessibility: E-commerce platforms offer the convenience of shopping from anywhere, anytime, and on any device. This accessibility can significantly increase sales and customer satisfaction.
- Scalability: E-commerce platforms are designed to scale with the business, allowing for the addition of products and expansion into new markets without the limitations of physical space.
- Cost Efficiency: Operating an e-commerce store can be more cost-effective than running a physical store, as it eliminates overhead costs associated with physical locations.
- Data Analytics and Insights: E-commerce platforms provide tools for tracking sales, customer behavior, and market trends, which can inform business decisions and strategies.

Cons:

- Technical Challenges: Implementing and managing an e-commerce platform can be complex, requiring technical expertise and ongoing maintenance.
- Customer Expectations: Customers expect fast shipping and free delivery, which can increase operational costs and affect profitability.
- Security Risks: E-commerce sites are vulnerable to cyberattacks, which can lead to data breaches and financial losses. Ensuring robust security measures is crucial.
- Competition: The e-commerce market is highly competitive, with businesses constantly seeking ways to differentiate themselves and attract customers.

In conclusion, the existing approaches and methods for e-commerce applications offer a range of benefits and challenges. By carefully considering these factors, businesses can select the most suitable platform and strategies for their e-commerce needs, ensuring a successful and sustainable online presence.

CHAPTER-3
REQUIREMENT ARTIFACTS

3.1 INTRODUCTION

The requirements for the e-commerce website are pivotal in ensuring that the platform is functional, secure, and user-friendly. These requirements are crucial for providing customers with a convenient and reliable way to browse products, place orders, and pay online. The requirements for the e-commerce website can be categorized into three main areas: functional, non-functional, and usability requirements. Functional requirements outline the features and capabilities of the website, such as product listing, shopping cart management, and secure payment processing. Non-functional requirements define the performance, security, and scalability of the website, ensuring it can handle high traffic volumes and protect user data. Usability requirements focus on the ease of use and user experience, ensuring the website is intuitive and accessible to all users.

Meeting these requirements is essential for creating a successful e-commerce platform. By adhering to these standards, the e-commerce website can enhance the customer experience, increase operational efficiency, and expand its reach to a broader audience.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements

The hardware requirements for the e-commerce website include:

- Server: A robust server is necessary to host the website and manage incoming requests efficiently. This ensures the website can handle the expected traffic volume and provide a responsive user experience.
- Storage: A database system is required to store user information, order history, product details, and other relevant data. This ensures the website can function smoothly and securely.
- Network: A stable and reliable network connection is crucial for quick response times and to prevent downtime. This ensures users can access the website and complete transactions without interruption.

These hardware requirements are vital for the website's performance, stability, and security. By meeting these requirements, the e-commerce platform can offer a seamless and secure online shopping experience for its customers. Additionally, scalability is a key consideration for hardware requirements. As the website grows and experiences increased traffic, additional servers and hardware may be necessary to maintain performance and prevent downtime.

Software Requirements

The software requirements for the e-commerce website, developed using Angular v16 and .NET v6, include:

- Angular v16: A popular framework for building web applications, Angular v16 provides a robust foundation for developing the frontend of the e-commerce website. It supports the development of dynamic, single-page applications and offers a wide range of tools and libraries for building user interfaces.
- .NET v6: A high-performance, cross-platform framework for building modern, cloud-based, and internet-connected applications, .NET v6 is used for developing the backend of the e-commerce website. It supports the development of web APIs, microservices, and serverless applications.
- SQL Server: A relational database management system used for storing and managing the website's data. SQL Server provides a secure and scalable solution for handling large volumes of data and complex queries.
- Visual Studio Code: A lightweight but powerful source code editor that runs on your desktop, Visual Studio Code is used for writing and debugging code. It supports a wide range of programming languages and offers features like IntelliSense, debugging, and Git integration.
- Figma: A collaborative interface design tool used for creating user interfaces and user experiences. Figma allows for real-time collaboration, making it easier to design and iterate on the website's design.
- Programming Languages: The website is developed using programming languages such as TypeScript (for Angular), C# (for .NET), and SQL (for database operations). These languages are essential for implementing the website's functionalities and ensuring its performance and security.

These software requirements are essential for building a functional, secure, and scalable e-commerce platform. By adhering to these standards, the website can provide a seamless and secure shopping experience for its customers, while also allowing for easy management and maintenance of the website’s data and functionality.

3.3 SPECIFIC PROJECT REQUIREMENTS

3.3.1 Data Requirements:

- Customer Profiles: The application must store customer profiles, including name, email, and delivery address. This data is essential for personalized user experiences, order tracking, and communication.
- Product Catalog: The application must store a list of products, along with their descriptions, prices, and images. This catalog is the foundation of the e-commerce platform, allowing customers to browse and select products.
- Order History: The application must store order history for each customer, including order details, payment information, and delivery status. This history is crucial for customer svice, order tracking, and customer loyalty programs.

3.3.2 Functionality Requirements:

- User Accounts: The application must allow customers to create accounts and log in to access their order history, payment information, and personalized settings.
- Product Browsing and Cart Management: Customers should be able to browse the product catalog, add items to their cart, and view their cart contents.
- Checkout and Payment: The application must provide a secure checkout process, including payment options, order review, and confirmation.
- Order Confirmation and Receipts: Upon completing an order, customers should receive an order confirmation page and an email receipt detailing their purchase.

3.3.3 Performance Requirements:

- Responsiveness: The application must be responsive and load quickly on both desktop and mobile devices, ensuring a seamless user experience across all platforms.
- Scalability: The application must be able to handle multiple concurrent users without performance degradation, ensuring smooth operation even during peak shopping times.
- High Volume Handling: The application must be capable of processing a high volume of orders during peak hours, maintaining performance and availability.

3.3.4 Security Requirements:

- Authentication and Authorization: The application must implement secure authentication protocols to protect user accounts and personal information. It should also use authorization mechanisms to control access to sensitive information and operations.
- Data Protection: The application must employ measures to protect against common security threats, including SQL injection attacks, cross-site scripting (XSS), and cross-site request forgery (CSRF).

3.3.5 Looks and Feel Requirements:

- Design and Aesthetics: The application must have a clean, modern design that is visually appealing and reflects the brand identity. It should be easy to navigate, with clear labels and instructions to enhance the user experience.
- Customization: The application should allow for customization to match the brand's visual identity, including color schemes, logos, and themes.

3.4 SUMMARY

By adhering to these specific project requirements, our e-commerce application is designed to meet the needs of both business and users. It ensures high performance, security, and usability, providing a positive experience for customers. This approach not only enhances the functionality and user experience of the application but also supports the scalability and growth of the business.

CHAPTER-4
DESIGN METHODOLOGY AND ITS NOVELTY

4.1 METHODOLOGY AND GOAL

The design methodology employed for our e-commerce application project is User-Centered Design (UCD). UCD is an iterative design process that places the needs and preferences of users at the core of the design process, aiming to create products and services that are intuitive, efficient, and enjoyable to use.

The objective of utilizing UCD for this project is to develop an application that meets the needs of its target audience, which includes online shoppers seeking a convenient and user-friendly platform for purchasing products. By adhering to the UCD process, we were able to gather user feedback and insights throughout the design process and incorporate them into the final product.

The UCD process encompasses several key steps, including user research, persona development, prototyping, and usability testing. Through user research, we collected information about the target audience's needs, preferences, and pain points related to online shopping. The overall goal of using UCD is to create an application that meets the needs of its users and provides a positive user experience, which will help to ensure the success of the project.

The technical methodology used in our e-commerce application project is an agile software development approach. This methodology emphasizes iterative development and continuous improvement, involving frequent feedback and collaboration between the development team and target audience.

This methodology was chosen for our e-commerce application project because it allows for a flexible and adaptive approach to software development, which is particularly important for a project of this scope and complexity. It also promotes frequent communication and collaboration between the development team and audience, which helps to ensure that the final product meets the needs and requirements of all parties involved.

4.2 FUNCTIONAL MODULES DESIGN AND ANALYSIS

Functional Modules for this project are:

- Registration Module: This module allows new users to register with the application by providing their basic information like email address and password. The module includes a form for users to fill in their information, and a verification process to ensure that the user's information is accurate.

- Cart & About Us Module: The Cart module allows users to add items to their cart and place orders. It includes a shopping cart interface, checkout process, and delivery information. The About Us module provides insights about the e-commerce venture and its developers.

- Login Page Module: This module allows registered users to log in to the application and access their account information, order history, and other features. The login page includes a form for users to enter their email and password, and authentication processes to ensure that only registered users can access their account.

- Home Page Module: This module serves as the main interface for the application, providing users with a view of all available products, promotional offers, and other relevant information. It includes a search bar, featured items section, and filter options.

- Navbar & Side Panel Module: This module provides easy navigation for users to move between different pages and sections of the application. It includes a navigation bar and side panel with links to different pages and features.

- Contact Page & My Orders Module: The Contact page module allows users to contact customer service with any queries or complaints. The My Orders module provides users to preview their cart items and make payment through razorpay payment gateway..

- Admin Module: This module provides administrators with access to the backend of the application, where they can manage orders, update inventory, and monitor application activity. It includes a dashboard with statistics, order management tools, and user management functionality.

- Search Module: This feature allows users to search for products by name, category, or keywords. It can be implemented using Angular's form controls and HTTP requests to filter the product list based on the user's input. This functionality improves the user experience by making it easier to find specific products quickly.

- Order Return and Cancellation: This module enables users to request a return or cancellation of their orders. It includes a form for users to submit their return or cancellation request, specifying the order ID, reason for the request, and any additional information required by the seller. The module also provides a status update for the user, informing them of the outcome of their request.

- Precisely Order Items: This feature allows users to specify the exact quantity of each item they wish to purchase. It can be implemented using Angular's reactive forms, which allow for dynamic form fields based on the product selected. This ensures that users can order exactly what they need, reducing the risk of over-ordering or under-ordering

Design:

The design for our e-commerce application is modern and user-friendly, with a clean and simple layout. The color scheme is primarily black and white, with highlights of white and grey for emphasis. The application is fully responsive, meaning it is optimized for use on both desktop and mobile devices.

Product Listings

- Purpose: This component is responsible for displaying a list of products available for purchase. It acts as the entry point for users to explore the product catalog.
- Functionality: It should include features like filtering and sorting options to help users find products based on their preferences. It may also support pagination to efficiently manage large product catalogs.
- User Interaction: Users can click on individual products to view more details, add them to the cart, or apply filters and sorting options to refine their search.

Product Details

- Purpose: To provide detailed information about a specific product, including descriptions, images, specifications, and pricing.
- Functionality: This component should display comprehensive information about the product, including customer reviews and ratings. It may also include options for users to add the product to their cart or wishlist.
- User Interaction: Users can read detailed descriptions, view images, and make purchasing decisions based on the information provided.

Shopping Cart

- Purpose: To manage the products that users have added to their cart, allowing them to review their selections before proceeding to checkout.
- Functionality: It should display a summary of the products in the cart, including quantities, prices, and a total cost. Users can adjust quantities, remove items, or clear the cart.
- User Interaction: Users can modify their cart, apply discounts or coupons, and proceed to checkout when ready.

Checkout

- Purpose: To facilitate the final steps of the purchasing process, including shipping information, payment details, and order confirmation.
- Functionality: This component should collect necessary information from the user, such as shipping address, payment method, and any additional notes or instructions. It should also calculate the total cost, including taxes and shipping fees.
- User Interaction: Users enter their shipping and payment information, review their order, and confirm the purchase.

Confirmation Page

- Purpose: To provide users with a confirmation of their order after it has been successfully placed.
- Functionality: This component should display a summary of the order, including the products purchased, total cost, and any additional information such as order number or expected delivery date.
- User Interaction: Users can review their order details, print a receipt, or return to the homepage to continue shopping.

Payment

- Purpose: To handle the payment process securely, ensuring that users can safely complete their purchases.
- Functionality: This component should integrate with a trusted payment processor to handle transactions. It should support multiple payment methods, such as credit cards, PayPal, or other digital wallets.
- User Interaction: Users enter their payment details, confirm the payment, and receive a confirmation once the transaction is complete.

Home

- Purpose: To serve as the landing page of the e-commerce application, showcasing featured products, promotions, or other enticing content to attract and engage users.
- Functionality: This component may include a carousel of featured products, special offers, or user-generated content. It should be designed to quickly engage users and encourage them to explore the product catalog.
- User Interaction: Users can browse featured products, click on promotions to view more details, or navigate to other sections of the site.

Each of these components of design plays a crucial role in the e-commerce application, working together to provide a seamless and engaging shopping experience for users.

4.2 SOFTWARE ARCHITECTURAL DESIGNS

The software architecture encompasses the technologies and components utilized both on the client and server sides, as well as the development tools employed throughout the project.

4.2.1 Client-Side Architecture

The client-side architecture is responsible for presenting the user interface and managing interactions with the user.

1. **AngularJS Components**:
    - **Description:** AngularJS framework is utilized to create reusable components, each representing a specific part of the user interface.
    - **Implementation:** AngularJS components encapsulate HTML templates, JavaScript controllers, and CSS styles, promoting modularity and reusability throughout the frontend codebase.
2. **Material Design**:
    - **Description:** Material Design principles are employed for the user interface design, ensuring a cohesive and visually appealing experience for users.
    - **Implementation:** AngularJS Material library is leveraged to implement pre-designed UI components such as buttons, cards, and input fields, maintaining consistency and enhancing usability.
3. **Client-Side Routing**:
    - **Description:** Client-side routing is implemented to enable navigation between different views/pages within the application without full page reloads.
    - **Implementation:** AngularJS's built-in routing module is utilized to define routes and corresponding controllers, allowing users to seamlessly navigate through the e-commerce website.
4. **State Management**:
    - **Description:** State management is crucial for maintaining application state across various components and views.
    - **Implementation:** AngularJS's built-in services like **$rootScope** and custom services are employed to manage application-wide state, ensuring data consistency and synchronization between components.

4.2.2 Server-Side Architecture

The server-side architecture handles business logic, data processing, and communication with databases.

1. **.NET 6 and ASP.NET Core**:
    - **Description:** The server-side component is developed using the .NET 6 framework and ASP.NET Core for building web applications and APIs.
    - **Implementation:** ASP.NET Core MVC framework is utilized to structure the backend codebase into controllers, models, and services, facilitating the implementation of RESTful APIs for client-server communication.
2. **API Endpoints**:
    - **Description:** RESTful API endpoints are exposed by the server-side component to enable client-side interaction with backend functionalities.
    - **Implementation:** ASP.NET Core's routing mechanism is employed to define API routes, with each route corresponding to a specific action or resource, such as retrieving product listings or processing orders.
3. **Database Integration with Entity Framework Core**:
    - **Description:** Data persistence is achieved through integration with a relational database using Entity Framework Core.
    - **Implementation:** Entity Framework Core serves as the ORM (Object-Relational Mapper) for interacting with the underlying database, abstracting away the database-specific details and simplifying data access operations.
4. **Authentication and Authorization**:
    - **Description:** Authentication and authorization mechanisms are implemented to ensure secure access to protected resources and functionalities.
    - **Implementation:** ASP.NET Core Identity framework is utilized for user authentication and authorization, enabling features such as user registration, login, role-based access control, and token-based authentication for API endpoints.

4.2.3 Development Tools

Development tools facilitate the coding, design, and collaboration aspects of the project.

1. **Visual Studio Code (VS Code)**:
    - **Description:** VS Code serves as the primary code editor for writing and editing frontend and backend code.
    - **Implementation:** VS Code's rich ecosystem of extensions and features enhances productivity

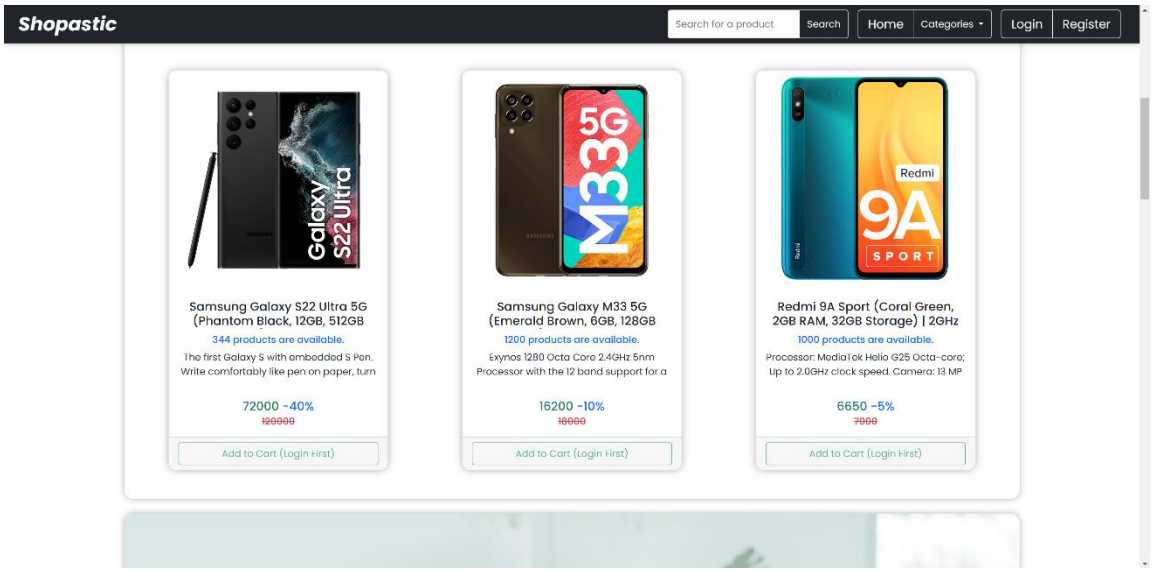
and supports various programming languages and frameworks used in the project.

2. **Figma**:
    - **Description:** Figma is employed for designing and prototyping the user interface of the e-commerce application.
    - **Implementation:** Figma's collaborative design tools enable team members to create, share, and iterate on UI mockups and wireframes, fostering effective communication and design validation.
3. **Git**:
    - **Description:** Git is utilized as the version control system for tracking changes to the codebase and facilitating collaborative development.
    - **Implementation:** Git repositories hosted on platforms like GitHub or GitLab provide centralized storage for project code, enabling version history management, branching, merging, and collaboration among team members.

4.3 USER-INTERFACE DESIGNS

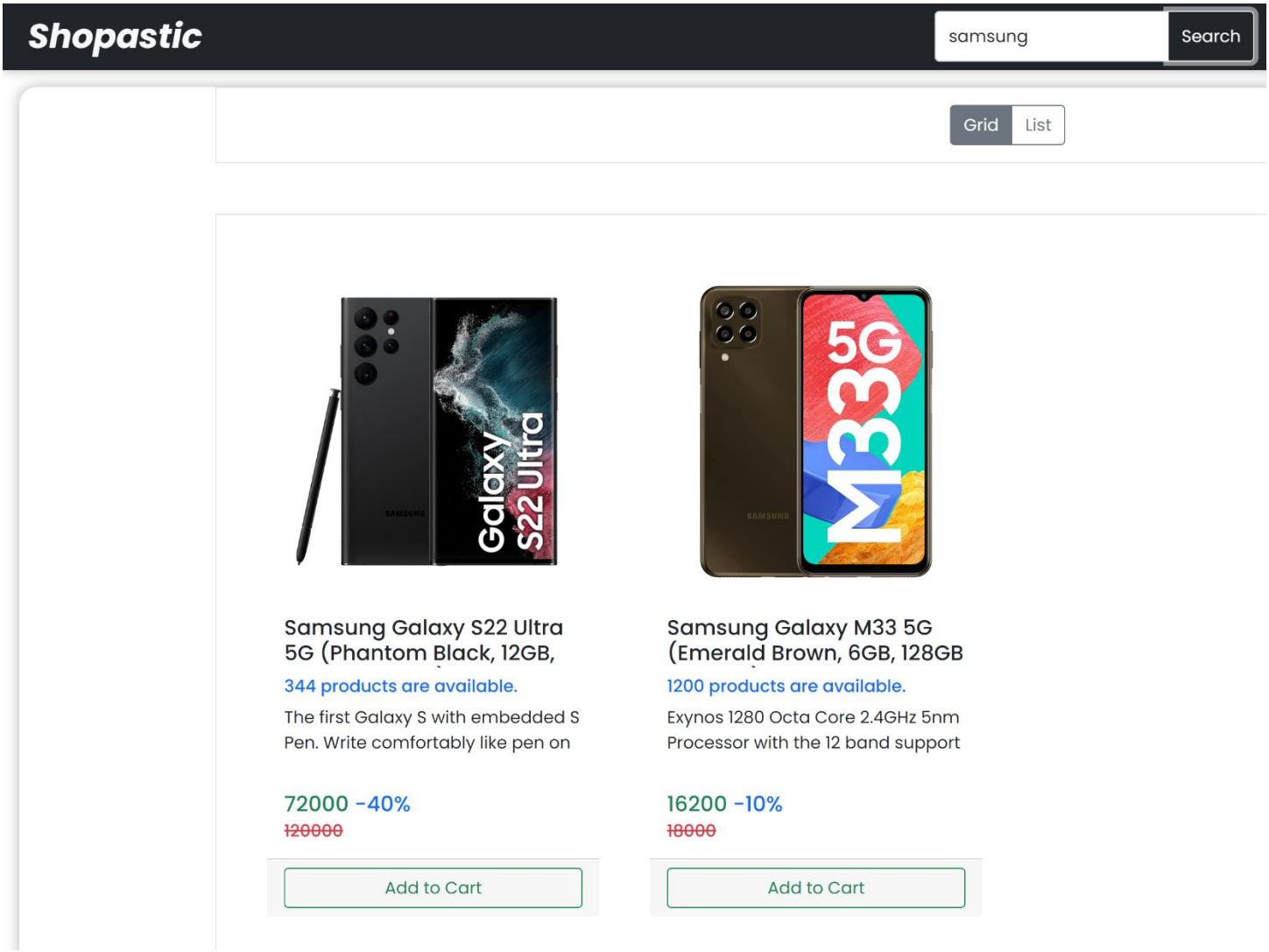
The user interface design for the e-commerce application aims to provide a seamless and intuitive experience for users. Incorporating modern design principles and leveraging UI with Angular, the interface emphasizes responsiveness, clarity, and ease of navigation.

4.3.1 Homepage



The homepage serves as the entry point to the e-commerce platform, offering users a glimpse of the available products and facilitating easy navigation to various sections of the website.

- **Search Bar**:



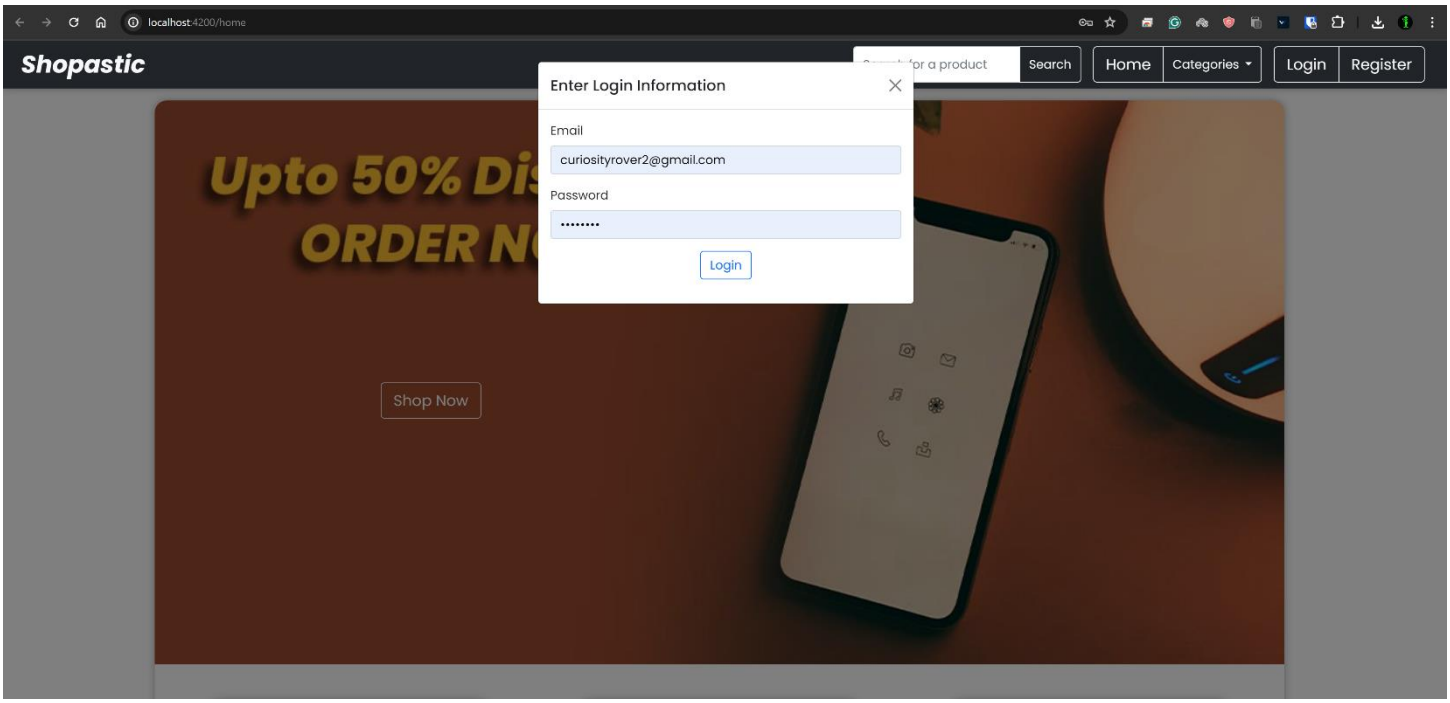
  - **Description:** A prominently displayed search bar allows users to quickly find specific items by entering keywords or phrases.
  - **Implementation:** The search functionality is seamlessly integrated with the backend to retrieve relevant product listings based on user input.

- **Category Banners**:

- **Description:** Visually appealing banners showcasing different food categories provide users with quick access to browse items based on their preferences.
- **Implementation:** Dynamic banners are implemented using React components, ensuring smooth transitions and responsive behavior across different devices.

- **Navigation Menu**:
  - **Description:** A concise navigation menu at the top of the page provides links to essential sections such as the menu, cart, about us, and contact pages.
  - **Implementation:** The navigation menu is implemented using Material UI components, offering consistent styling and easy access to key features.

4.3.2 Login Page

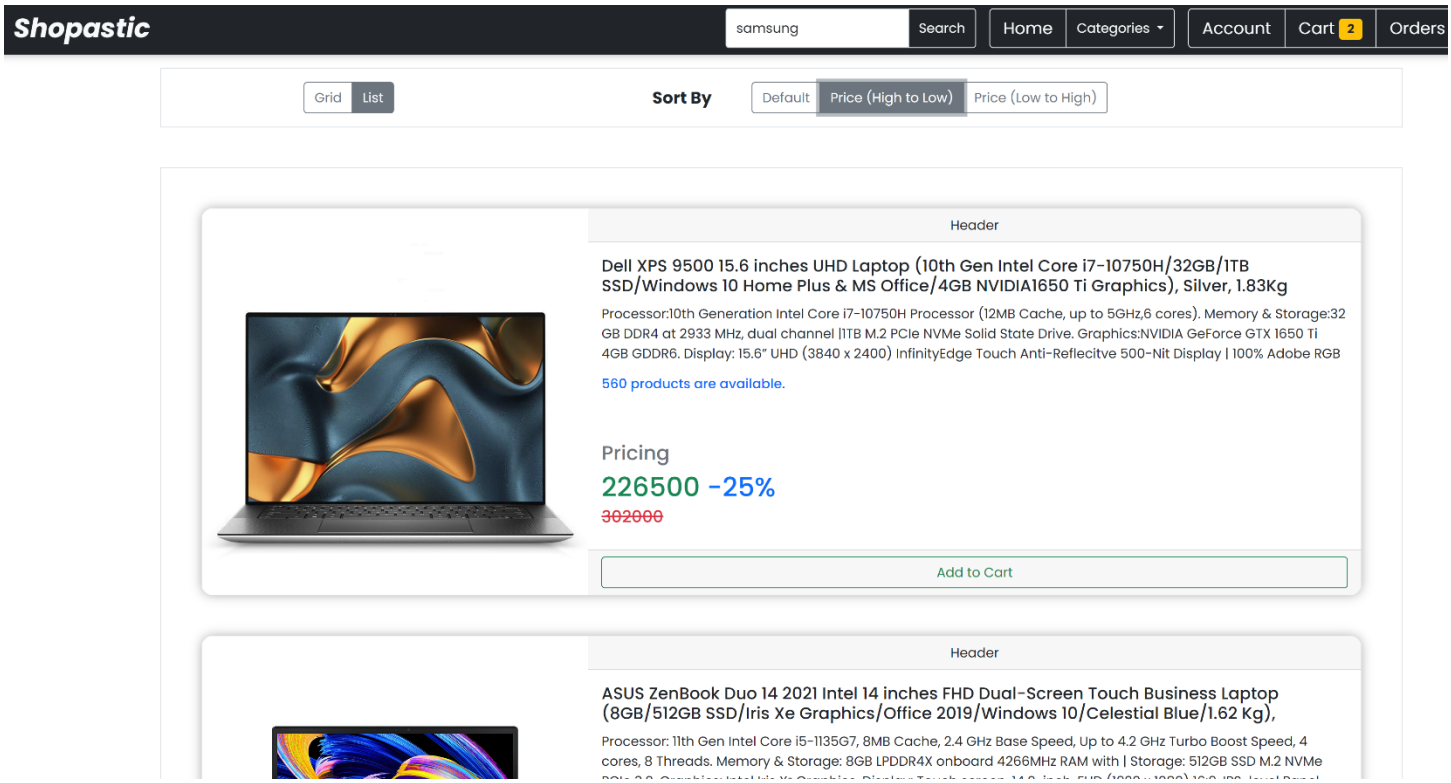


The menu page displays a comprehensive list of items and categories available for purchase, offering users the flexibility to explore and filter products based on their preferences.

- **Product Listings**:
  - **Description:** Clear and organized product listings present users with detailed information about each item, including description, image, and price.

- **Implementation:** React components render product cards dynamically, fetching data from the backend API to populate the menu page with up-to-date information.

- **Filtering and Sorting**:



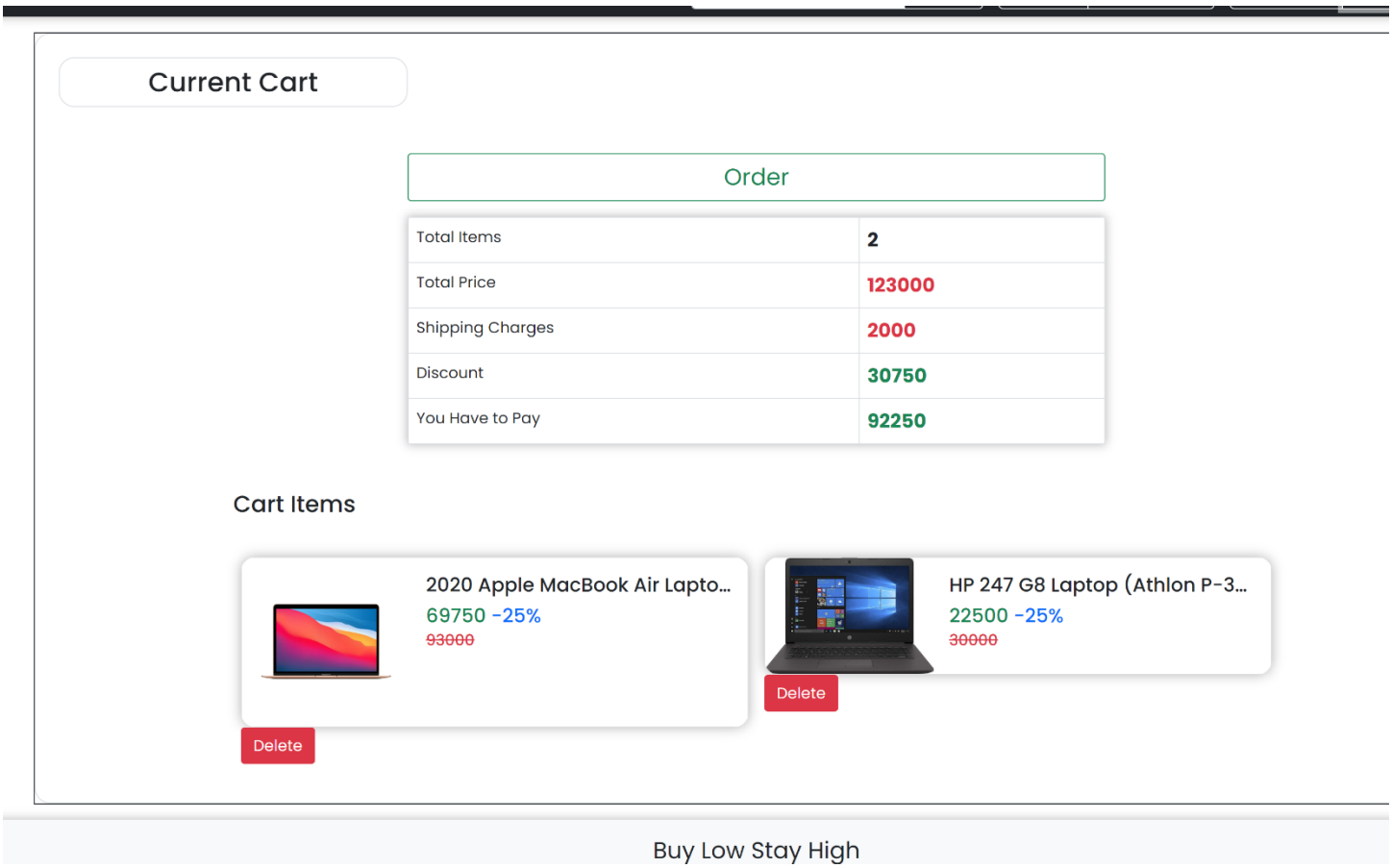
- **Description:** Interactive filtering and sorting options enable users to refine their search results based on criteria such as price, rating, or dietary requirements.

- **Implementation:** AngularJS directives are utilized to implement filter and sorting functionality, enhancing user experience by providing tailored product suggestions.

4.3.3 Cart and Checkout

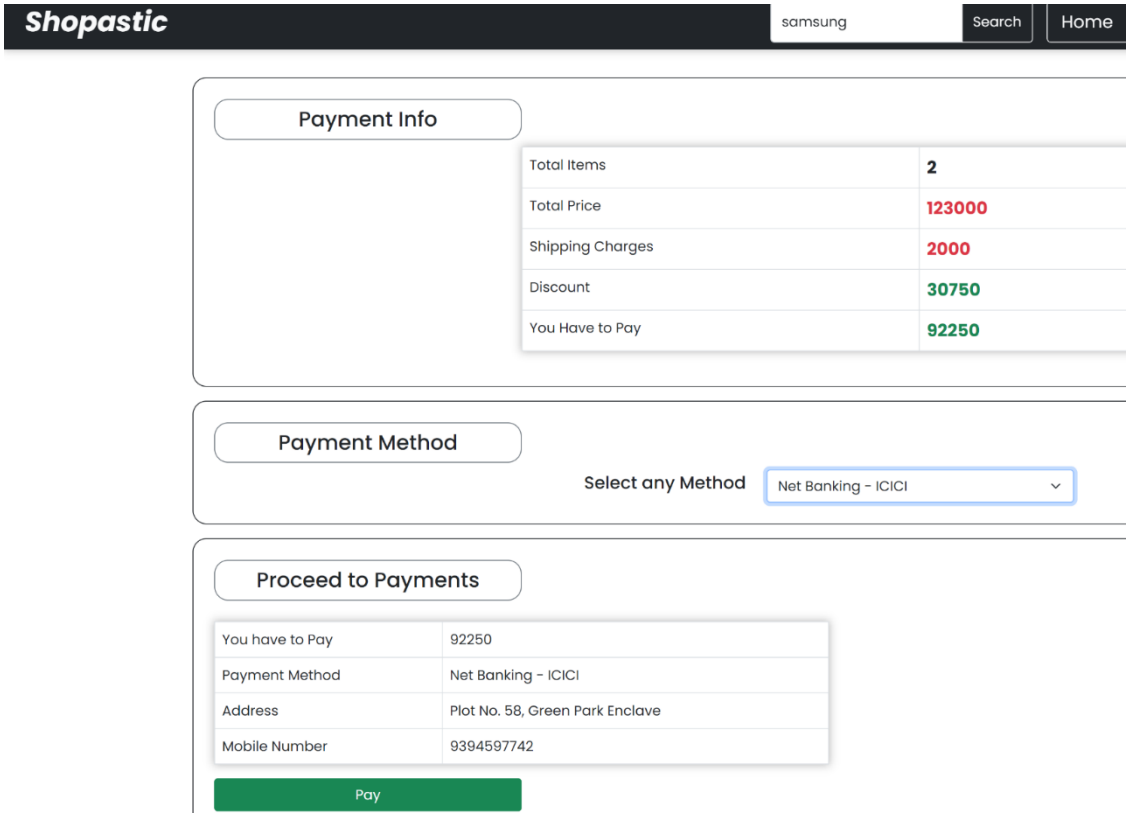
The cart and checkout process streamline the user's purchasing journey, offering convenient options for reviewing and finalizing their order.

- **Cart Page**:



- **Description:** The cart page displays a summary of items added by the user, allowing them to adjust quantities or remove items as needed.
- **Implementation:** AngularJS controllers manage the cart state, updating in real-time as users interact with the interface to modify their selections.

- **Checkout Process**:

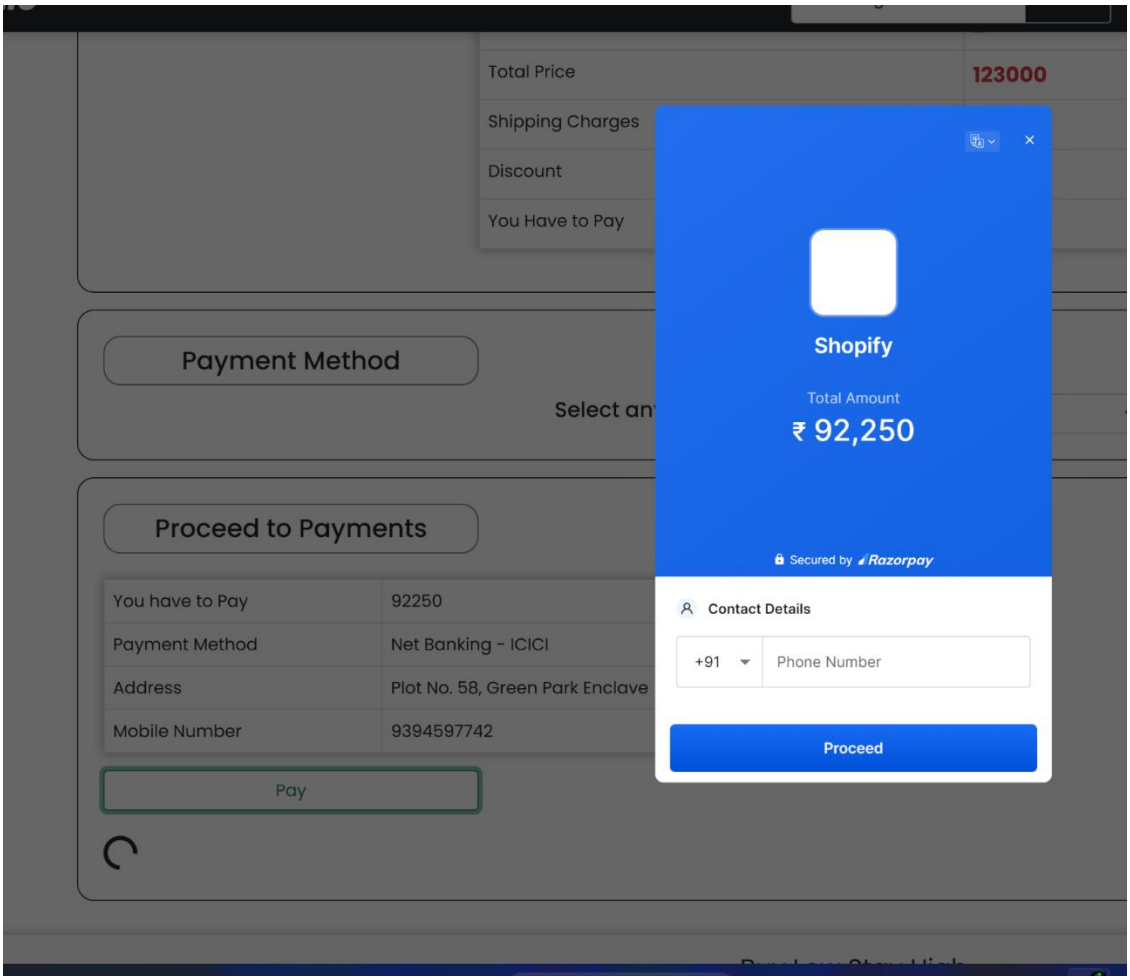

Shopastic
samsung
Search
Home
Payment Info
Total Items
2
Total Price
123000
Shipping Charges
2000
Discount
30750
You Have to Pay
92250
Payment Method
Select any Method
Net Banking - ICICI
Proceed to Payments
You have to Pay
92250
Payment Method
Net Banking - ICICI
Address
Plot No. 58, Green Park Enclave
Mobile Number
9394597742
Pay

- **Description:** A straightforward checkout process presents users with options for delivery or pickup, accompanied by a transparent breakdown of the total cost.

- **Implementation:** .NET 6 backend services handle the checkout logic, validating user inputs and processing payment transactions securely.

4.3.4 Payment Gateway Page

Razorpay payment gateway is integrated into the e-commerce shopping application to process lightning-fast payments from a range of accepted payment methods



- **About Us Page**:

  - **Description:** The About Us page offers insights into the food venture, highlighting its mission, values, and commitment to quality.

  - **Implementation:** Static content is served using AngularJS templates, supplemented with

dynamic data fetched from the backend to showcase team members and project milestones.

- **Contact Page**:
    - **Description:** The Contact page features a user-friendly form for submitting queries or feedback, along with contact information for reaching out to the restaurant and website developers.
    - **Implementation:** AngularJS directives manage form validation and submission, with .NET 6 backend APIs handling incoming requests and storing user messages in a database.

4.4 SUMMARY

The development of the e-commerce website follows a structured waterfall methodology, meticulously addressing requirements artifacts and software architectural designs. The proposed e-commerce platform represents a significant innovation in the realm of online shopping, tailored to meet the needs of our target audience.

With a focus on providing a seamless and intuitive shopping experience, the website offers a comprehensive range of features designed to enhance usability and convenience for users. By leveraging modern technologies such as AngularJS for the frontend and .NET 6 for the backend, the platform ensures optimal performance and scalability.

The project's novelty lies in its commitment to delivering a user-centric approach to online shopping, with a particular emphasis on usability, functionality, and visual appeal. Through meticulous attention to detail in both software architecture and user interface design, the e-commerce website sets a new standard for online retail experiences.

CHAPTER-5
TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 OUTLINE

I. **Front-End Implementation**

A. UI Design

- **Utilize Figma for UI Design**: Figma provides a collaborative interface design tool that allows the creation of wireframes, prototypes, and user interface designs. It enables the design team to iterate quickly, gather feedback, and create a visually appealing layout for the e-commerce website.

- **Create High-Fidelity Wireframes**: High-fidelity wireframes serve as detailed blueprints for the website's user interface, depicting the layout, navigation flow, and visual elements. These wireframes are developed in Figma, incorporating design elements such as typography, color schemes, and branding guidelines.

- **Implement Material Design**: Material Design principles are followed for the UI design to ensure consistency, usability, and aesthetic appeal across the website. Angular Material is utilized to implement Material Design components and design patterns.

B. Angular Implementation

- **Set Up Angular Project**: The Angular project is initialized using Angular CLI (Command Line Interface), which provides a convenient way to create, scaffold, and manage Angular applications. Angular CLI generates the necessary boilerplate code and configuration files to kickstart development.

- **Implement Angular Services for Business Logic**: Angular services are used to encapsulate reusable business logic and data manipulation tasks. Services facilitate code organization, separation of concerns, and dependency injection.

- **Utilize Angular Routing for Navigation**: Angular's built-in router module is used to define routes and navigation paths within the application. Routing allows users to navigate between different views and components without reloading the entire page.

- **Create Components for UI Elements**: The user interface of the e-commerce website is divided into Angular components, each responsible for rendering a specific UI element or feature. Components encapsulate HTML templates, CSS styles, and TypeScript logic.

- **Implement Reactive Forms for User Input**: Angular's reactive forms module is used to create forms with complex validation and data handling requirements. Reactive forms enable the creation of dynamic, data-driven forms that respond to user input in real-time.

II. **Back-End Implementation**

A. .NET Implementation

- **Set Up .NET Project**: The .NET project is initialized using .NET CLI or Visual Studio, providing a foundation for backend development with .NET technologies. Project structure and configuration files are generated to support the development of RESTful APIs and business logic.

- **Implement .NET Web API with ASP.NET Core**: ASP.NET Core is used to develop RESTful APIs that expose endpoints for client-server communication. Controllers, routes, and action methods are defined to handle incoming HTTP requests and generate appropriate responses.

- **Integrate Entity Framework Core for Data Access**: Entity Framework Core is a lightweight and extensible ORM framework for .NET, used to interact with databases in an object-oriented manner. EF Core simplifies data access operations such as querying, inserting, updating, and deleting records.

- **Implement Middleware for Request Processing**: Middleware components are used in the ASP.NET Core pipeline to intercept, handle, and modify HTTP requests and responses. Middleware can perform tasks such as logging, error handling, authentication, and authorization.

- **Secure API Endpoints with Authentication and Authorization**: Authentication and authorization mechanisms are implemented to secure access to API endpoints and protect sensitive data. ASP.NET Core Identity or JWT (JSON Web Tokens) authentication may be used to authenticate users and enforce access control policies.

III. **Testing and Validation**

A. Unit Testing

- **Write Unit Tests for Angular Components and Services**: Unit tests are written to verify the behavior and functionality of Angular components, services, and other application modules. Test cases are defined to cover various scenarios and edge cases, ensuring code correctness and reliability.

IV. **Deployment**

A. Set Up Deployment Environment

- **Choose Hosting Platform**: Select a suitable hosting platform such as Azure, AWS, or Heroku for deploying the e-commerce website. Consider factors such as scalability, reliability, and cost when choosing a hosting provider.

- **Configure Deployment Pipeline**: Set up a deployment pipeline using CI/CD (Continuous Integration/Continuous Deployment) tools such as Azure DevOps, GitHub Actions, or Jenkins. Automate the deployment process to streamline the release of new updates and features.

- **Optimize Performance and Scalability**: Implement performance optimizations such as

caching, minification, and compression to improve the website's responsiveness and load times. Configure scaling options to handle increasing traffic and demand effectively.

5.2 TECHNICAL CODING AND CODE SOLUTIONS

1. **Setting up the Development Environment:**

    - **Install Node.js**: Node.js was installed to provide the runtime environment for executing JavaScript code on the server-side, facilitating backend development.

    - **Install .NET**: .NET version 6 was installed to provide the backend APIs for receiving requests from the Angular Frontend, as well as communicating with the MySQL DB.

    - **Install Visual Studio Code (VSCode)**: VSCode was selected as the code editor for writing the project code, providing a lightweight and extensible development environment.

    - **Set up Angular**: Angular CLI was used to create a new Angular project, providing a foundational structure and tooling for frontend development.

2. **Front-end Development:**

    - **Create Angular Components**: Reusable and modular Angular components were developed for different sections of the application's user interface, promoting code reusability and maintainability.

    - **Utilize Angular Material**: Angular Material component library was leveraged to design and style the application's user interface elements, ensuring consistency and adherence to Material Design principles.

    - **Implement Routing**: Angular's built-in router module was used to implement client-side routing, enabling navigation between different pages of the application without full page reloads.

    - **Fetch Data from APIs**: Angular's HttpClient module was utilized to make HTTP requests to the server-side APIs, fetching data and updating the application's state dynamically.

3. **Back-end Development:**

    - **Create ASP.NET Core Web API**: An ASP.NET Core Web API project was created to handle incoming requests from the client-side and provide appropriate responses, serving as the backend infrastructure for the application.

    - **Define API Endpoints**: RESTful API endpoints were designed and implemented in ASP.NET Core to support various operations such as user authentication, product listing, order processing, and data manipulation.

    - **Connect to Microsoft SQL Server**: Entity Framework Core was utilized to establish a

connection between the ASP.NET Core Web API and the Microsoft SQL Server database, facilitating seamless data interaction and management.

- **Implement Data Models**: Data models were defined using Entity Framework Core to represent the structure of the data stored in the Microsoft SQL Server database, ensuring consistency and integrity.

4. **Database Management:**

- **Create Microsoft SQL Server Database**: A Microsoft SQL Server database was created to store data related to different entities in the application, such as users, products, orders, and transactions.

- **Implement CRUD Operations**: Entity Framework Core was used to implement Create, Read, Update, and Delete operations, allowing for efficient data manipulation and management within the Microsoft SQL Server database.

5.3 PROTOTYPE SUBMISSION

The prototype for your project is a web-based application built using Angular.js, .NET, and SQL Server. The application allows users to create an account and log in to access a variety of features.

Once logged in, users can create and edit their profiles, view a dashboard displaying important information related to their account, and manage their orders. They can also browse through a catalog of products, add items to their cart, and checkout with a secure payment system.

The application includes a search function and filters to help users find the products they are looking for quickly and easily. It also includes an order history page that allows users to view their past orders and track their current ones. The prototype has been designed with a clean, modern interface using Material UI components and responsive design to ensure it works seamlessly across different devices. It also includes form validation and error handling to ensure a smooth user experience.

Overall, the prototype provides a user-friendly and intuitive interface that allows users to easily navigate the application and complete tasks efficiently.

Technology Stack:

- **Frontend**: AngularJS is employed to create a dynamic and responsive user interface, offering an immersive browsing experience.
- **Backend**: Powered by .NET 6 and ASP.NET Core, providing robust server-side logic and RESTful API endpoints for seamless client-server communication.
- **Database**: Microsoft SQL Server serves as the database management system, ensuring efficient data storage and retrieval.

Key Features:

1. **User Authentication and Account Management**:
    - Users can register accounts and securely log in to unlock access to various application features.
    - Account management functionalities enable users to customize profiles, manage addresses, and review order history.
2. **Dashboard and Profile Editing**:
    - Upon login, users are greeted with a personalized dashboard displaying relevant account information and order statuses.
    - Profile editing capabilities allow users to update personal details, preferences, and saved information for seamless checkout experiences.
3. **Product Catalog Browsing and Shopping**:
    - A diverse catalog of products is available for users to explore, with intuitive search and filtering options to streamline product discovery.
    - Users can conveniently add items to their shopping carts, adjust quantities, and proceed to secure checkout for seamless transactions.
4. **Order Management and History**:
    - Users have access to an order management interface, enabling real-time tracking of current orders and review of past purchases.
    - Detailed order history provides insights into previous transactions, facilitating reorder and reference capabilities.

5. **Search Functionality and Filters**:
    - Robust search functionality and advanced filters empower users to quickly locate desired products based on specific criteria, enhancing overall browsing efficiency.
6. **Responsive Design and User Experience**:
    - The application boasts a sleek and modern interface, featuring Material UI components and responsive design principles to ensure optimal performance across devices.
    - Form validation and error handling mechanisms are implemented to enhance user experience and minimize potential data entry errors during interactions.

Overall Experience:

- The e-commerce application prototype delivers a seamless and intuitive user experience, empowering users to navigate the platform effortlessly and complete tasks efficiently.
- With its comprehensive feature set and robust technology stack, the prototype lays a solid foundation for the development of a scalable and feature-rich e-commerce platform tailored to meet the demands of modern online shoppers.

5.4 SUMMARY

The technical implementations and code solutions for Shopastic can be summarised as below:

I. **Frontend Development with AngularJS:**

- **AngularJS Framework**: Leveraged AngularJS for frontend development, offering a powerful framework for building dynamic and responsive user interfaces.
- **Component-based Architecture**: Implemented a component-based architecture to modularize and organize frontend code, enhancing maintainability and scalability.
- **Angular Material Design**: Utilized Angular Material Design components to create a sleek and modern interface, ensuring consistency and adherence to design principles.
- **Routing and Navigation**: Implemented Angular's built-in routing and navigation features to enable seamless navigation between different views and components.
- **Form Validation**: Integrated form validation techniques to enhance user experience and ensure data integrity, minimizing errors during user interactions.

II. **Backend Development with .NET 6 and ASP.NET Core:**

- **.NET 6 Framework**: Utilized .NET 6 for backend development, providing a robust and versatile framework for building scalable and high-performance applications.
- **ASP.NET Core Web API**: Developed RESTful API endpoints using ASP.NET Core Web API, facilitating communication between the frontend and backend components.
- **Entity Framework Core**: Leveraged Entity Framework Core for database access and management, simplifying data operations and ensuring efficient data handling.
- **Authentication and Authorization**: Implemented authentication and authorization mechanisms to secure API endpoints and protect sensitive data, ensuring user privacy and security.
- **Middleware and Error Handling**: Utilized middleware components and error handling techniques to intercept and handle HTTP requests and responses, ensuring application stability and reliability.

III. **Database Management with Microsoft SQL Server:**

- **Microsoft SQL Server**: Employed Microsoft SQL Server as the backend database management system, providing robust data storage and retrieval capabilities.
- **Database Design and Modeling**: Designed and modeled the database schema using SQL Server Management Studio, ensuring efficient organization and structuring of data.
- **CRUD Operations Implementation**: Implemented CRUD (Create, Read, Update, Delete) operations using Entity Framework Core, enabling seamless interaction with the database and manipulation of data.

IV. **Deployment and Environment Setup:**

- **Hosting Environment Setup**: Configured hosting environment for the application deployment, ensuring compatibility and scalability for future growth.
- **Deployment Pipeline**: Implemented deployment pipeline using continuous integration/continuous deployment (CI/CD) tools, automating the deployment process for efficiency and reliability.

- **Environment Configuration**: Configured environment variables for sensitive information such as database connection strings and API keys, ensuring security and confidentiality in production environments.

The technical implementation of this project involved utilizing these technologies and tools to develop the backend functionality, frontend user interface, and the database integration. The use of these technologies enabled efficient development, code reusability, and the creation of a responsive and interactive web application.

CHAPTER-6
PROJECT OUTCOME AND APPLICABILITY

6.1 KEY IMPLEMENTATION OUTLINES OF THE SYSTEM

User Management

- Implementation: Utilize Node.js, Express.js, and MongoDB to implement user registration and authentication. This ensures secure access to user accounts and personalized experiences.
- User Profiles: Develop user profiles to manage user data, including personal information, purchase history, and preferences. This personalization enhances user engagement and satisfaction.
- Security: Enable password hashing and encryption to securely store user credentials, protecting against unauthorized access.

Product Catalog

- Product Listing: Develop a dynamic product catalog to showcase available items, enhancing the shopping experience with detailed product information.
- Search and Filtering: Implement advanced search and filtering functionalities, allowing users to easily find products based on various criteria such as category, price, and brand.
- Product Details: Include detailed product information, including images, pricing, and reviews, providing users with all the necessary information to make informed purchasing decisions.

Shopping Cart

- Design and Implementation: Design and implement a user-friendly shopping cart functionality, enabling users to easily add products, update quantities, and remove items.
- Total Cost Calculation: Calculate and display the total cost of items in the cart, including any applicable discounts or taxes, ensuring transparency and accuracy in pricing.

Order Processing

- Order Placement: Enable users to place orders for selected items, streamlining the checkout process and ensuring a smooth transition from shopping to payment.

- Order Management: Implement an order processing system that handles order creation, payment processing, and order fulfillment, ensuring timely and efficient order handling.

Admin Dashboard

- Dashboard Creation: Create an admin dashboard for managing products, orders, and users, providing administrators with a centralized platform to manage the e-commerce platform.
- Role-Based Access Control: Implement role-based access control to restrict admin features to authorized users, ensuring security and preventing unauthorized access to sensitive information.

Reviews and Ratings

- Review Submission: Allow users to submit reviews and ratings for products, fostering a community of informed shoppers and enhancing the credibility of products.
- Display Ratings: Display average ratings and user reviews on product pages, providing transparency and helping other shoppers make informed decisions.

Performance Optimization

- Optimization: Optimize the system for performance and scalability, ensuring the platform can handle high traffic volumes and provide a responsive user experience.
- Caching: Implement caching mechanisms to improve response times, enhancing the platform's efficiency and user satisfaction.
- Database Optimization: Optimize database queries and indexes for efficient data retrieval, ensuring fast and reliable data access.

Security Measures

- Security Implementation: Implement security measures such as input validation, protection against cross-site scripting (XSS) attacks, and cross-site request forgery (CSRF) protection, safeguarding the platform against common web vulnerabilities.
- Secure Coding Practices: Follow best practices for secure coding and handling of sensitive data, ensuring the platform's security and the protection of user information.

By incorporating these key functionalities, the e-commerce platform will offer a robust, secure, and efficient ordering and delivery system, providing a seamless and enjoyable shopping experience for users.

6.2 SIGNIFICANT PROJECT OUTCOMES

Significant project outcomes are designed to enhance the shopping experience, streamline operations, and drive growth.:

Increased Efficiency

- Streamlined Process: The e-commerce platform can streamline the shopping process, reducing the time and effort required to browse, select, and purchase products. This efficiency benefits both customers and the business by speeding up transactions and freeing up resources.

Improved Accuracy

- Reduced Human Error: With a digital system, there's less chance of human error in the shopping process, such as misreading product descriptions or making incorrect selections. This accuracy ensures that customers receive exactly what they ordered, enhancing satisfaction and trust in the platform.

Enhanced Convenience

- Anytime, Anywhere Shopping: Customers can shop from the e-commerce website at any time and from any location, without the need to physically visit a store or wait in line. This convenience is a significant advantage, especially for busy individuals or those who prefer shopping from the comfort of their homes.

Improved Customer Experience

- Personalized Shopping: The e-commerce platform can offer personalized product recommendations, special promotions, and a seamless shopping experience, enhancing the overall satisfaction of customers. Features like wish lists, personalized product suggestions, and easy returns can significantly improve the shopping experience.

Increased Revenue

- Increased Sales Volume: With an easier and more convenient shopping process, more customers may choose to shop from the e-commerce platform, potentially leading to increased sales volume and revenue for the business. This growth can be sustained through effective marketing and customer retention strategies.

Better Data Tracking

- Valuable Insights: The e-commerce website can track customer behavior, product preferences, and shopping patterns, providing valuable insights that can inform future product offerings, marketing strategies, and promotions. This data-driven approach allows the business to stay ahead of trends and meet customer needs more effectively.

Improved Communication

- Efficient Order Fulfillment: The e-commerce platform can facilitate better communication between the business and customers, allowing for more efficient and accurate order fulfillment. Features like order tracking, real-time updates, and customer support can enhance the shopping experience and build trust.

Reduced Waste

- Accurate Inventory Management: With more accurate tracking of orders and inventory, the e-commerce business can reduce waste and save money on unnecessary stock. This efficiency not only benefits the environment but also the bottom line by optimizing resource allocation.

By focusing on these outcomes, an e-commerce website can significantly enhance its value proposition, driving growth, customer satisfaction, and operational efficiency.

6.3 PROJECT APPLICABILITY ON REAL-WORLD APPLICATIONS

The e-commerce application for shopping is a comprehensive solution that addresses various aspects of online retail, from account management to order processing. This application's applicability extends beyond the initial concept, offering a robust framework that can be adapted for a wide range of real-world applications. Here's how the project's applicability can be expanded:

Retail Stores

- Digital Transformation: Retail stores can adopt this e-commerce application to create an online presence, allowing customers to browse products, place orders, and make payments online. This can significantly enhance the shopping experience and increase sales.

Online Marketplaces

- Centralized Platform: The application can serve as a centralized platform for multiple vendors, facilitating the listing, selling, and shipping of products. This model can support a wide range of products and vendors, making it a versatile solution for online marketplaces.

Specialty Stores

- Niche Focus: Specialty stores, such as those selling art, antiques, or unique items, can use this application to reach a wider audience. The ability to list products with detailed descriptions and images can attract customers looking for specific items.

Subscription Services

- Recurring Orders: For businesses offering subscription services, the application can manage recurring orders, automating the process of shipping products to customers on a regular basis. This can streamline operations and improve customer satisfaction.

Event Tickets and Merchandise

- Event Management: Event organizers can use the application to sell tickets and merchandise online, simplifying the process for attendees and providing a convenient way to purchase items related to the event.

Educational Platforms

- Course Materials and Resources: Educational platforms can use the e-commerce application to sell course materials, textbooks, and other educational resources, making them easily accessible to students worldwide.

Health and Wellness Products

- Online Health Store: Health and wellness stores can leverage this application to sell products online, offering customers the convenience of shopping from home and receiving products directly at their doorstep.

Custom Applications for Specific Industries

- Industry-Specific Solutions: The application can be customized for specific industries, such as fashion, electronics, or home goods, offering tailored features and functionalities to meet the unique needs of each sector.

6.4 INFERENCE

The e-commerce application is not only a powerful tool for online retail but also a versatile platform that can be adapted for various real-world applications. Its modular design, which includes account management, inventory management, pricing, order processing, payment handling, and product listing functionalities, makes it suitable for a wide range of industries and business models. By focusing on user experience, scalability, and security, e-commerce application can provide a seamless and efficient shopping experience, driving growth and customer satisfaction across different sectors.

CHAPTER-7
CONCLUSIONS AND RECOMMENDATION

7.1 OUTLINE

The e-commerce website project has successfully met its objectives by providing a convenient and efficient platform for online shopping. The key outcomes and achievements of this project are as follows:

- Development of a Fully Functional Web Application: The project has resulted in the creation of a comprehensive e-commerce platform that allows users to browse products, add items to their cart, and complete purchases online. This web application is designed to be user-friendly and accessible, catering to a wide range of customers.

- User-Friendly Features: The e-commerce website incorporates several user-friendly features, including product categories for easy navigation, a search functionality that enables users to find products quickly, and a cart management system that streamlines the checkout process. These features are crucial for enhancing the shopping experience and encouraging users to complete their purchases.

- Integration with Online Payment Gateways: The platform is integrated with secure online payment gateways, ensuring that transactions are processed safely and efficiently. This integration not only provides convenience to users but also enhances the security of the e-commerce platform.

- Attractive and Responsive User Interface: The design and development of the e-commerce website have focused on creating an attractive and responsive user interface. Utilizing Material UI and React, the website offers a visually appealing layout that is also optimized for various devices, ensuring a seamless shopping experience across all platforms.

- Admin Panel for Management: An admin panel has been implemented to manage orders, product listings, and user accounts. This panel provides administrators with the tools needed to efficiently manage the e-commerce platform, ensuring that customer orders are processed promptly and that the product catalog is up-to-date.

- Deployment on a Reliable Hosting Platform: The e-commerce website has been deployed on a reliable hosting platform, ensuring that users can access the platform at any time without interruptions. This deployment strategy is crucial for maintaining the availability and performance of the e-commerce platform.

Throughout the course of this project, several important lessons have been learned, including the importance of planning and organization, effective communication among team members, and the necessity of thorough testing to ensure the functionality and reliability of the web application.Areas for improvement for this project include Redis server for storage and optimizing the web application's performance to reduce page loading times and improve the user experience.Overall, this project has been a valuable learning experience, enabling the development of technical skills and gaining experience working on a real-world web application. The team is proud of the outcomes and achievements of this project and looks forward to implementing improvements in the future to enhance the e-commerce platform further.

7.2 LIMITATIONS/ CONSTRAINTS OF THE SYSTEM

- Limited Functionality: The e-commerce platform may initially offer basic functionalities such as product listing, shopping cart management, and checkout processes. As the platform evolves, it may need to expand its features to include more advanced functionalities like personalized recommendations, customer reviews, and loyalty programs.
- Limited Scope: Initially, the e-commerce website may focus on a specific product category or niche market. Expanding the product range to include a broader variety of items could be a challenge, requiring careful planning and execution.
- Limited Access: The platform may initially be accessible only within a specific geographical area or to a limited demographic. Expanding access to a wider audience, including international customers, would require addressing issues related to international shipping, currency conversion, and language support.
- Technical Constraints: The platform may face technical challenges such as scalability issues, security vulnerabilities, and performance bottlenecks. Continuous monitoring and optimization are necessary to ensure the platform can handle increased traffic and maintain high performance.
- Resource Constraints: The development and maintenance of the e-commerce platform may require significant resources, including skilled personnel, development tools, and hosting infrastructure. Managing these resources efficiently is crucial for the platform's success.
- Adoption Challenges: Encouraging users to adopt the e-commerce platform, especially in a competitive market, may be challenging. Strategies to increase user engagement, such as offering promotions, loyalty programs, and personalized content, may be necessary.
- Regulatory Constraints: The e-commerce platform must comply with various regulatory requirements, including data protection laws, e-commerce regulations, and accessibility standards. Ensuring compliance is essential for legal and ethical reasons.

7.3 FUTURE ENHANCEMENTS

There are several possible future enhancements for our ecommerce website, such as:

- Mobile Application: Developing a mobile app for the e-commerce platform would provide users with a more convenient and flexible shopping experience. It could also enhance the platform's functionality, such as offering push notifications for promotions or order updates.
- Loyalty Program: Implementing a loyalty program could encourage repeat purchases and increase customer loyalty. This could involve offering discounts, rewards, or exclusive access to new products for loyal customers.
- Online Payment Options: Expanding the range of online payment options, including digital wallets and cryptocurrencies, could make the checkout process more convenient and secure for users.
- Real-time Tracking: Providing real-time tracking of orders could improve customer satisfaction by allowing them to monitor the status of their orders more easily.
- Social Media Integration: Integrating the e-commerce platform with social media could increase its visibility and attract more users. This could involve sharing product promotions, user reviews, and engaging content on social media platforms.
- Customizable Menu: Allowing users to customize their shopping experience, such as by filtering products based on their preferences or creating personalized product lists, could enhance user satisfaction and encourage repeat visits.
- Integration with Other Services: Integrating the e-commerce platform with other services, such as delivery tracking or customer support platforms, could improve the overall user experience.
- Multi-language Support: Offering multi-language support could cater to a global audience, making the platform more accessible to international customers.
- Integration with the Student ID Card: For educational institutions, integrating the e-commerce platform with student ID cards could streamline the checkout process for students, making it more convenient for them to purchase educational materials or other products.

7.4 INFERENCE

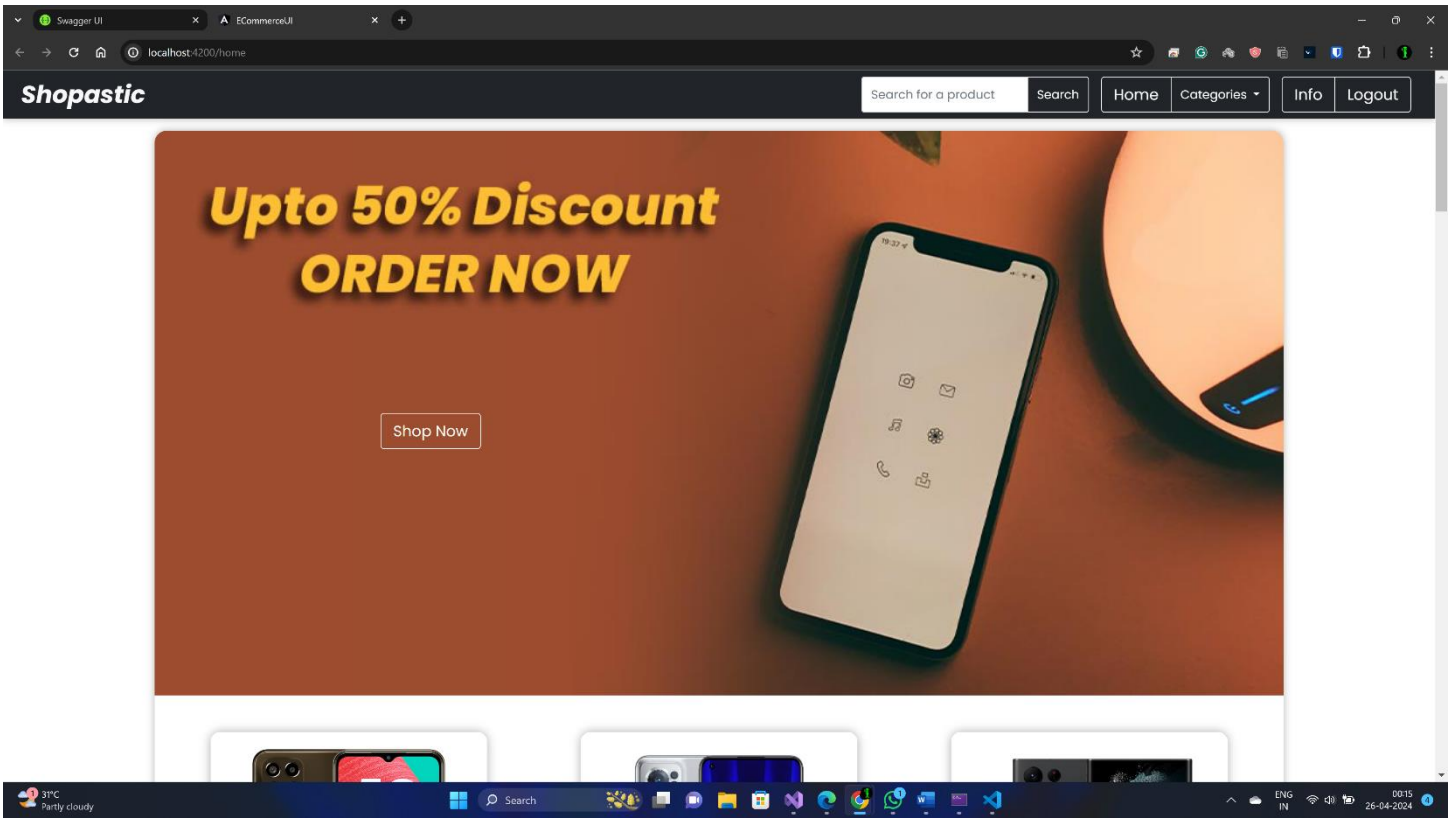
The e-commerce website project has demonstrated the potential of technology to provide a convenient and efficient shopping experience. The project's outcomes, including the development of a fully functional web application, user-friendly features, secure payment gateways, and an attractive user interface, have been significant. The platform has shown improved efficiency, convenience, and accessibility, as well as increased customer satisfaction and revenue.

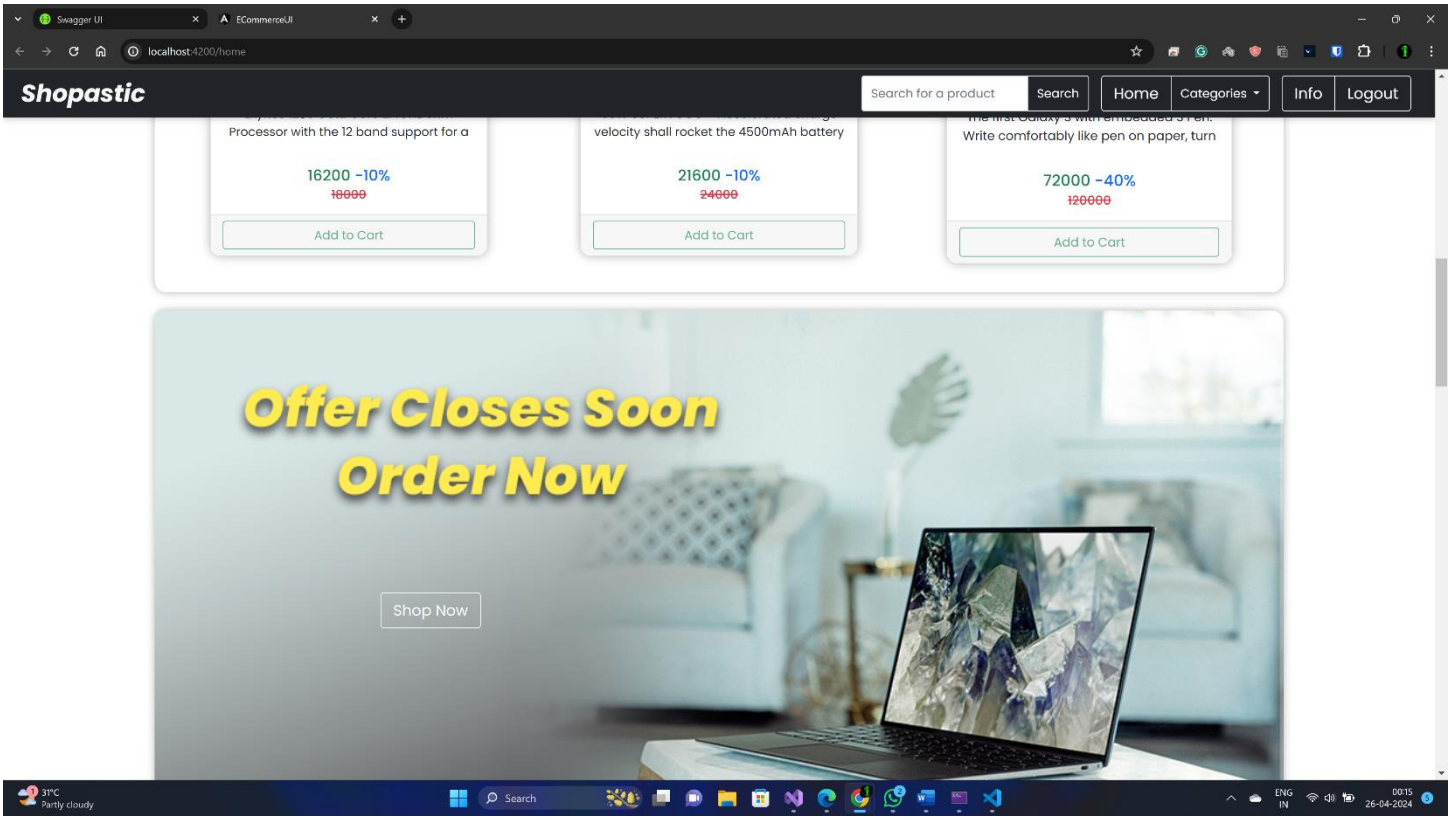
Future enhancements, such as implementing a mobile app, adding more payment options, integrating a loyalty program, and expanding the delivery range, can further improve the platform's functionality and user experience. These enhancements highlight the importance of continuous improvement and adaptation to meet the changing needs and expectations of users.

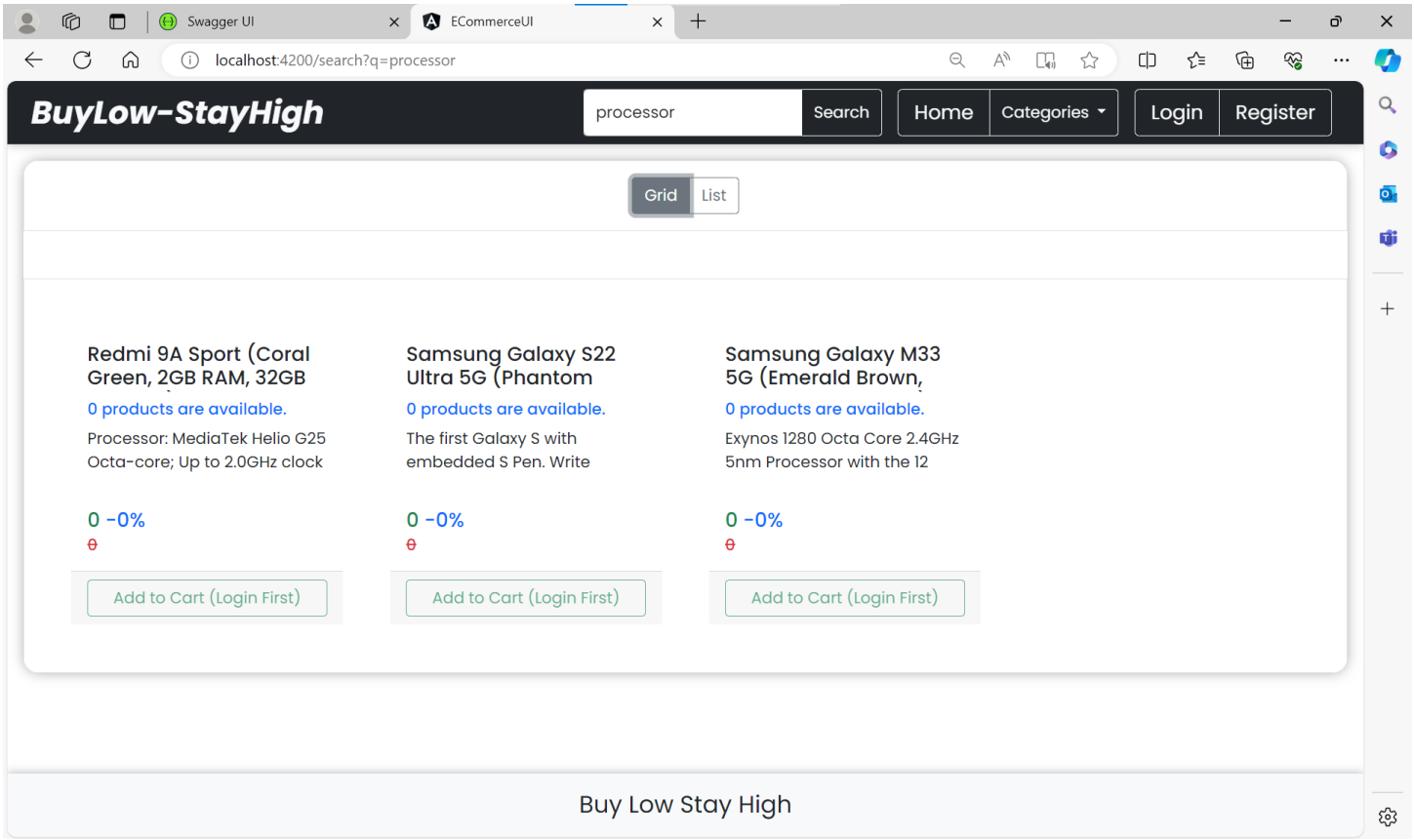
Overall, the project has demonstrated the potential of technology to solve real-world problems and improve the quality of life for people. It also underscores the importance of planning, organization, communication, and thorough testing in achieving project success.

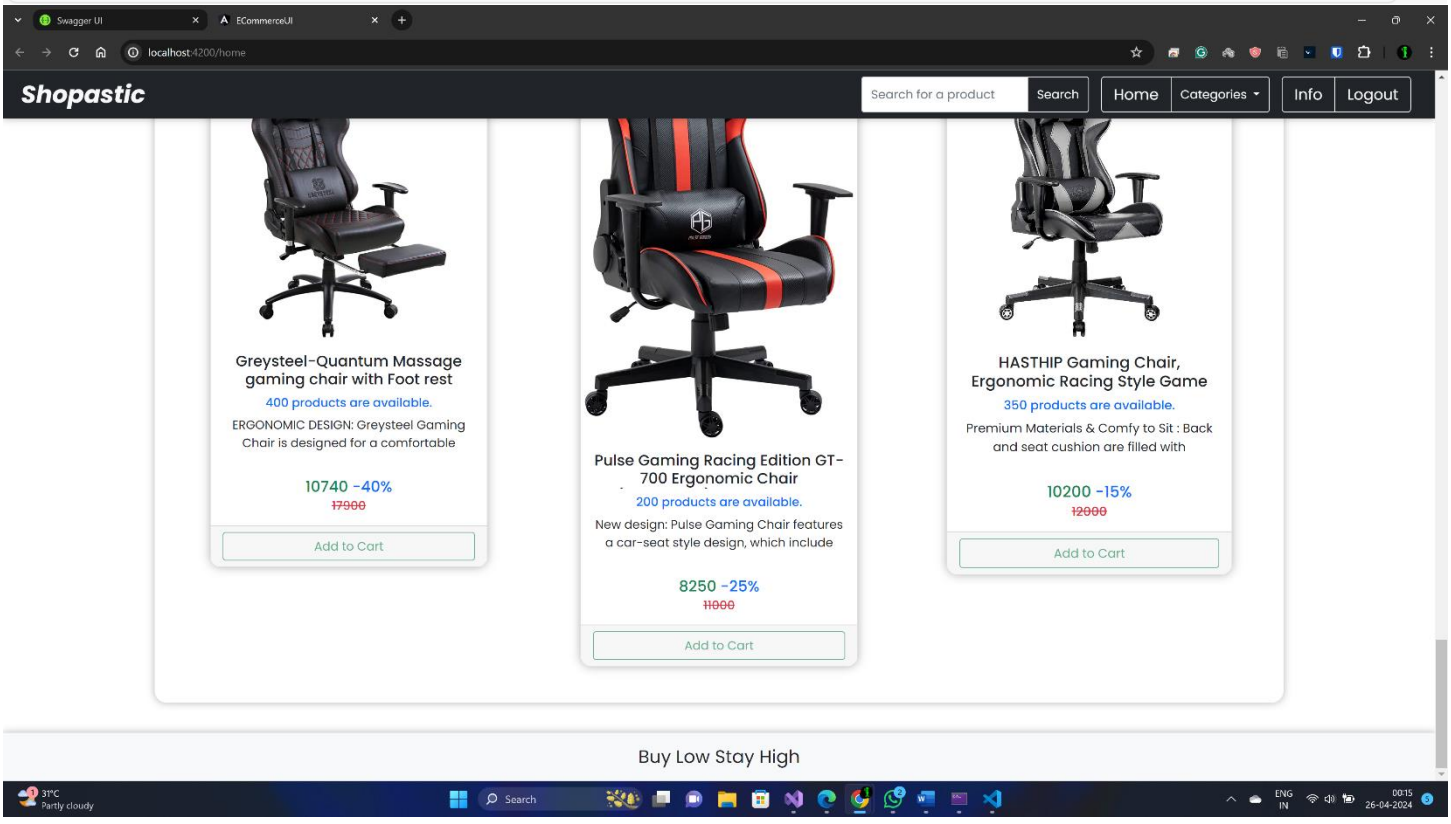
APPENDIX A(Screenshots)

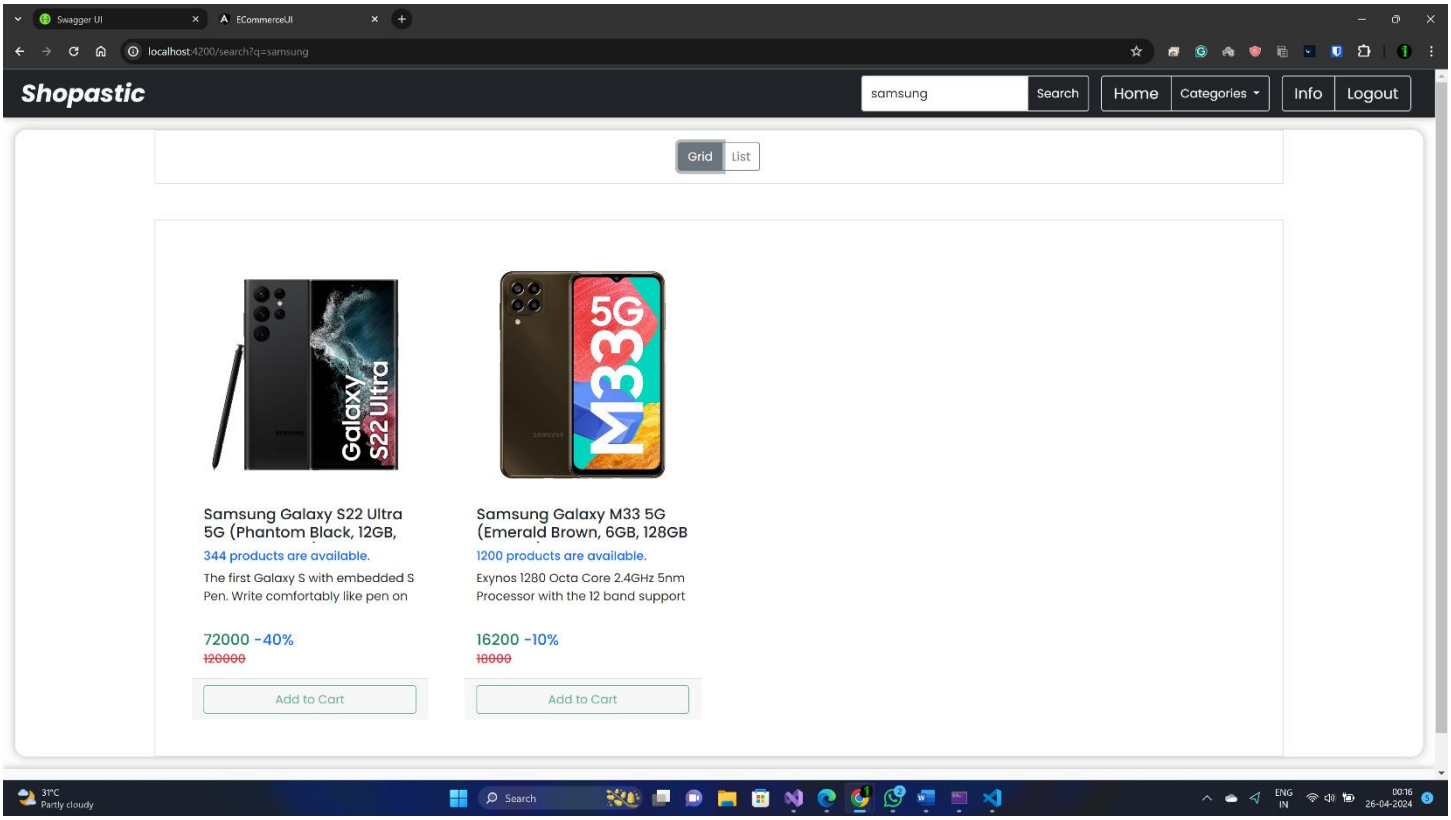
User Login

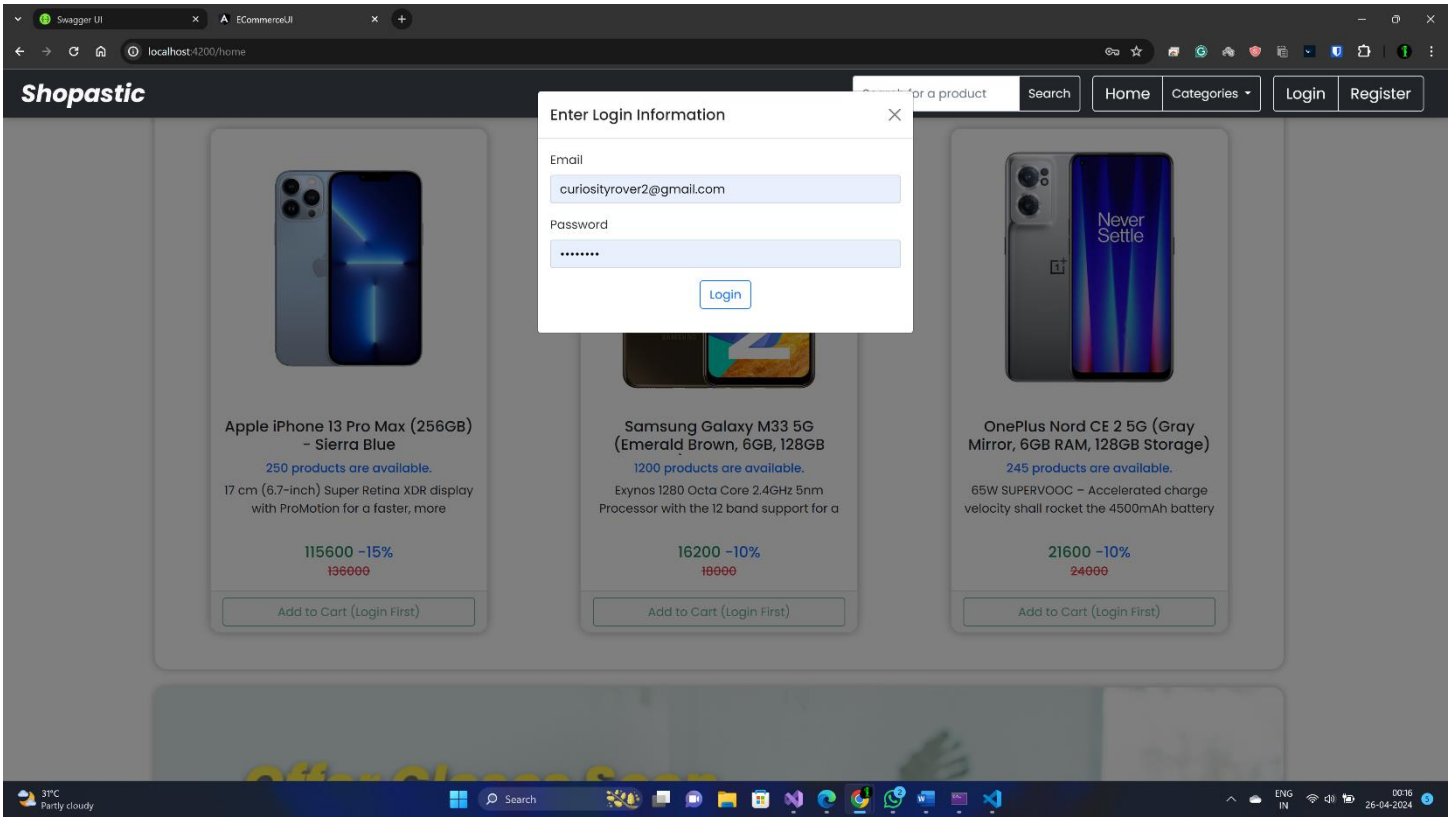


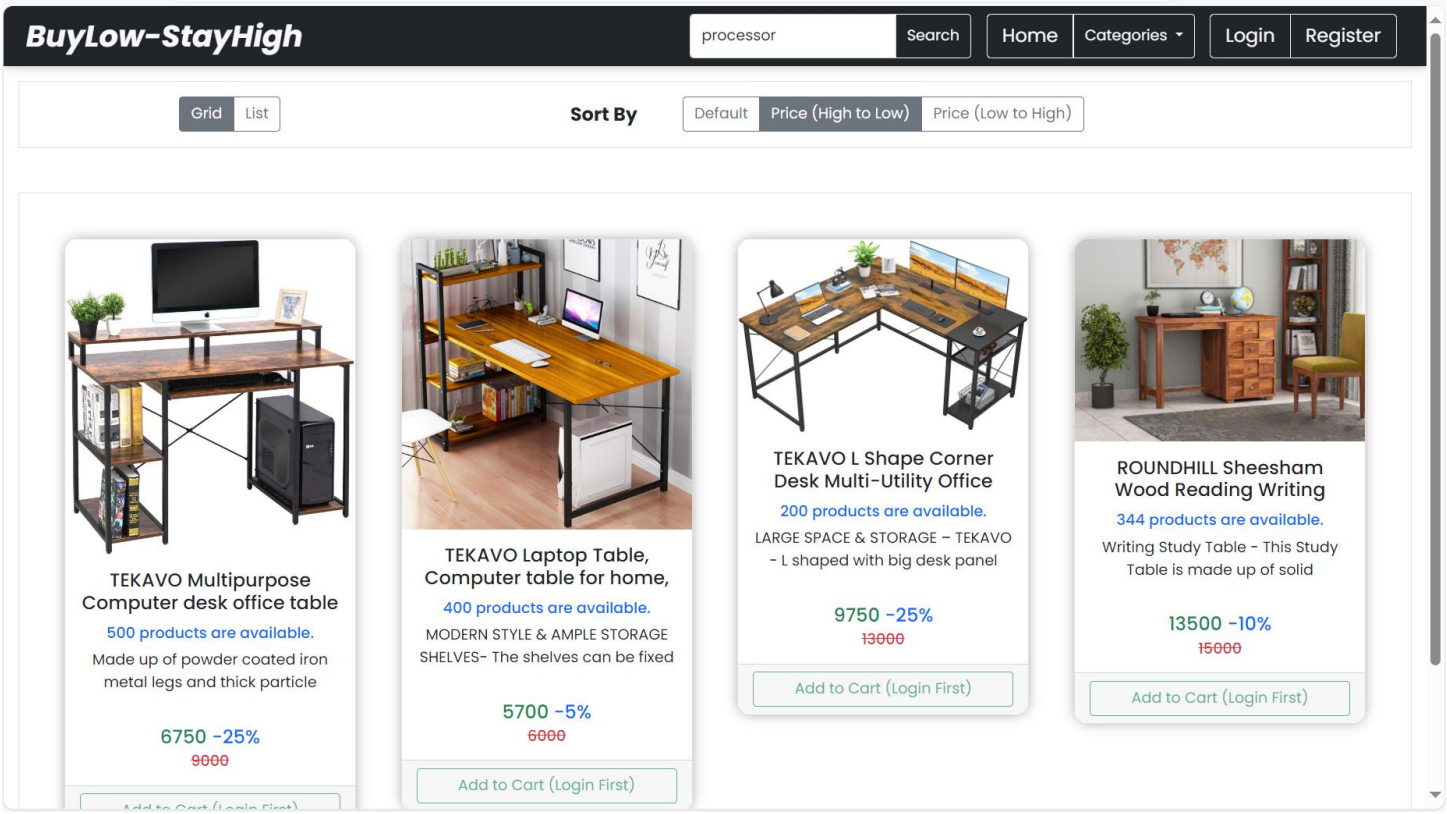

Swagger UI
ECommerceUI
localhost:4200/home
Shopastic
Search for a product
Search
Home
Categories
Info
Logout
Processor with the 12 band support for a
16200 -10%
18000
Add to Cart
velocity shall rocket the 4500mAh battery
21600 -10%
24000
Add to Cart
Write comfortably like pen on paper, turn
72000 -40%
120000
Add to Cart
Offer Closes Soon
Order Now
Shop Now
31°C
Partly cloudy
Search
ENG
IN
00:15
26-04-2024

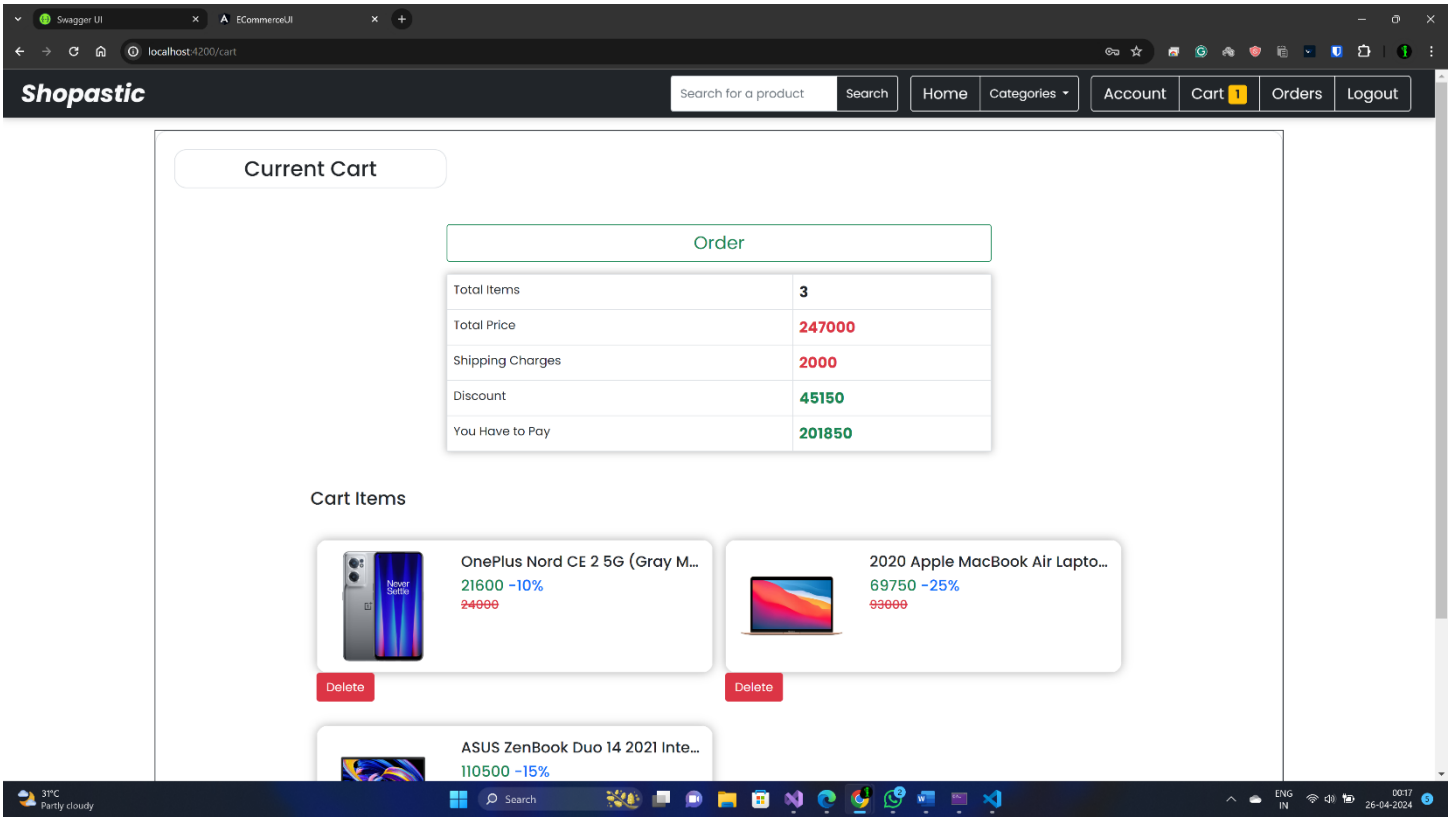

Swagger UI
ECommerceUI
localhost:4200/search?q=processor
BuyLow-StayHigh
processor
Search
Home
Categories
Login
Register
Grid
List
Redmi 9A Sport (Coral Green, 2GB RAM, 32GB
0 products are available.
Processor: MediaTek Helio G25 Octa-core; Up to 2.0GHz clock
0 -0%
0
Add to Cart (Login First)
Samsung Galaxy S22 Ultra 5G (Phantom
0 products are available.
The first Galaxy S with embedded S Pen. Write
0 -0%
0
Add to Cart (Login First)
Samsung Galaxy M33 5G (Emerald Brown,
0 products are available.
Exynos 1280 Octa Core 2.4GHz 5nm Processor with the 12
0 -0%
0
Add to Cart (Login First)
Buy Low Stay High

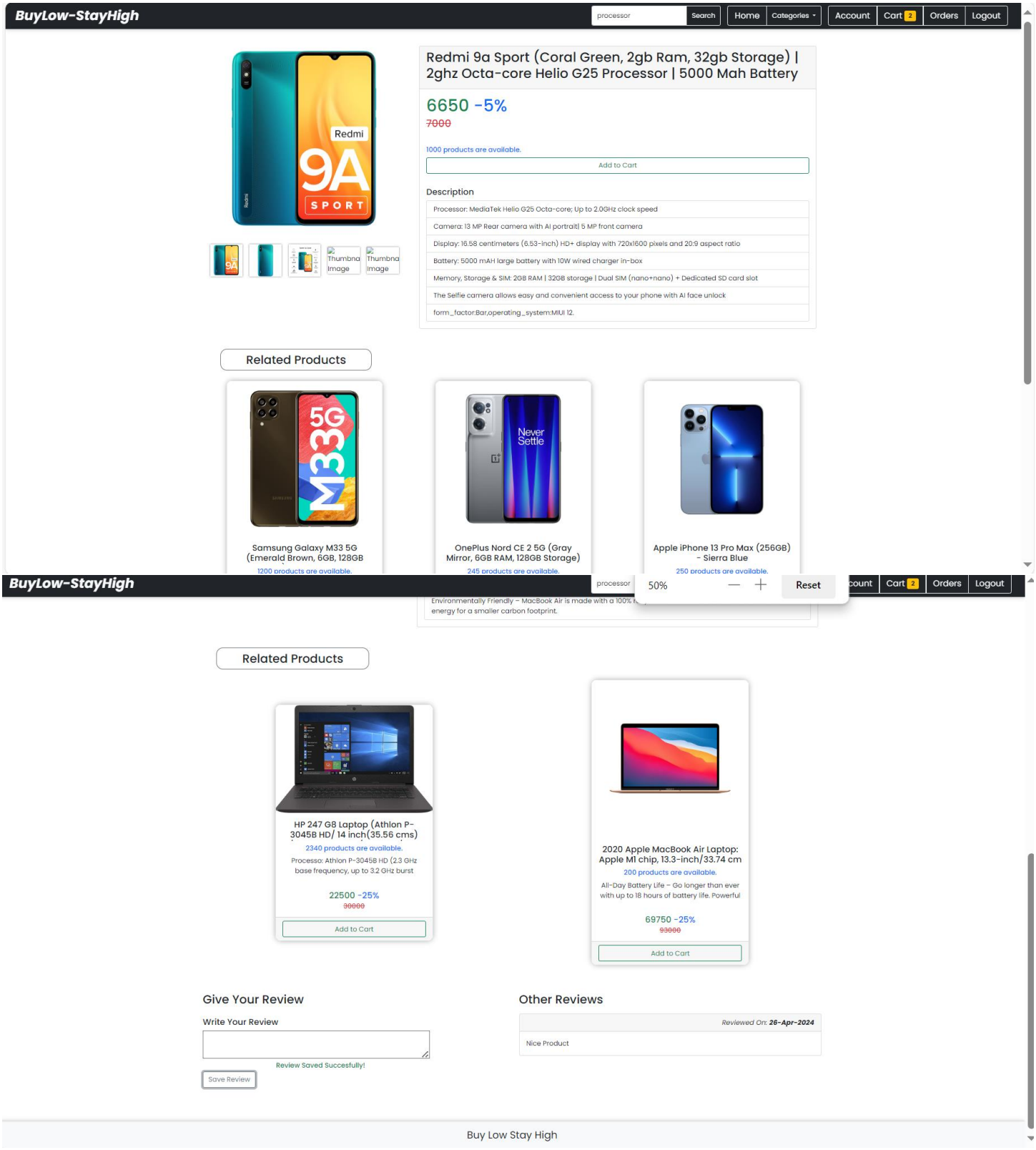

Swagger UI
ECommerceUI
localhost:4200/home
Shopastic
Search for a product
Search
Home
Categories
Info
Logout
Greysteel-Quantum Massage gaming chair with Foot rest
400 products are available.
ERGONOMIC DESIGN: Greysteel Gaming Chair is designed for a comfortable
10740 -40%
17900
Add to Cart
Pulse Gaming Racing Edition GT-700 Ergonomic Chair
200 products are available.
New design: Pulse Gaming Chair features a car-seat style design, which include
8250 -25%
11000
Add to Cart
HASTHIP Gaming Chair, Ergonomic Racing Style Game
350 products are available.
Premium Materials & Comfy to Sit : Back and seat cushion are filled with
10200 -15%
12000
Add to Cart
Buy Low Stay High

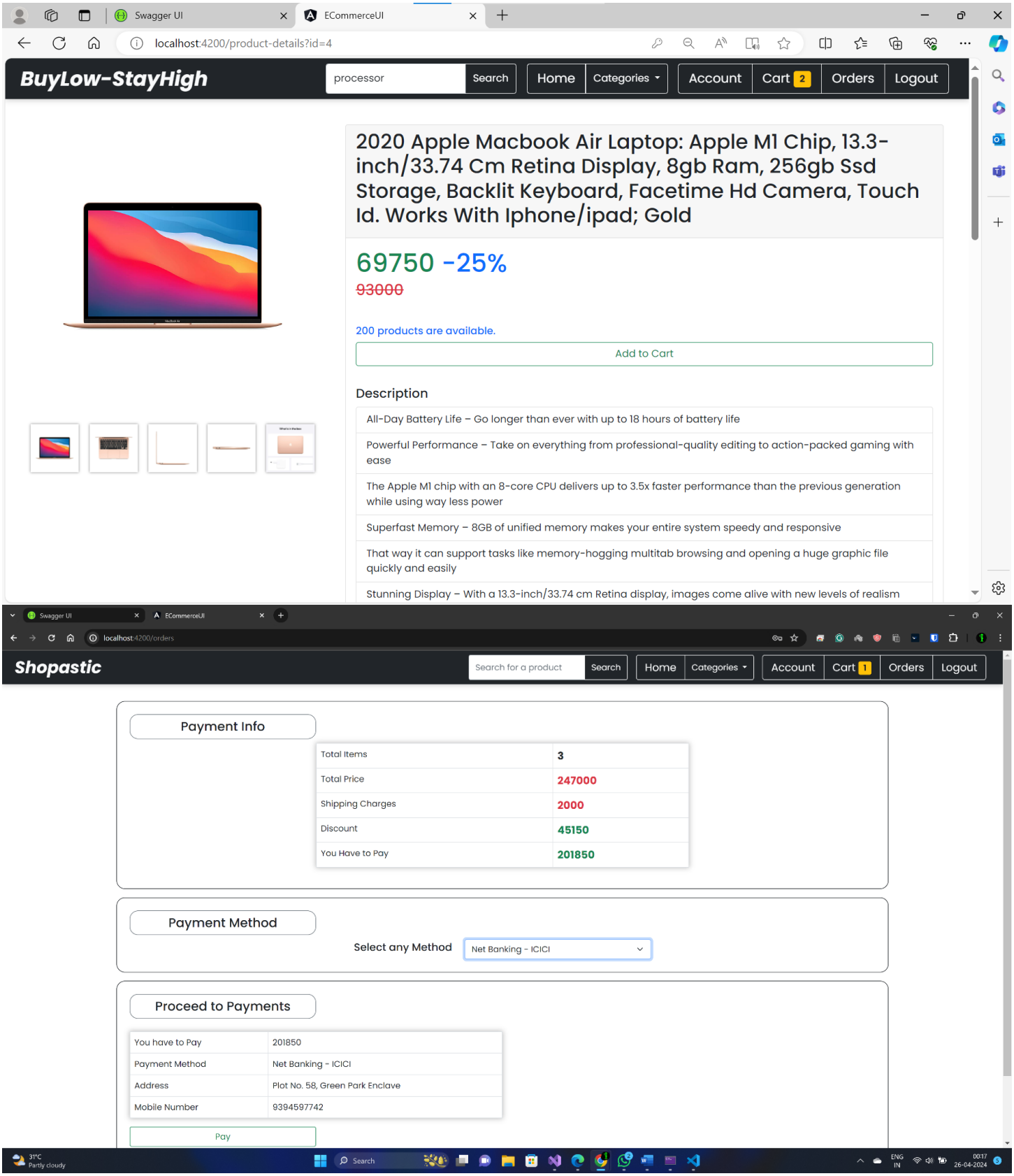

Swagger UI
ECommerceUI
localhost:4200/search?q=samsung
Shopastic
samsung
Search
Home
Categories
Info
Logout
Grid
List
Samsung Galaxy S22 Ultra 5G (Phantom Black, 12GB,
344 products are available.
The first Galaxy S with embedded S Pen. Write comfortably like pen on
72000 -40%
120000
Add to Cart
Samsung Galaxy M33 5G (Emerald Brown, 6GB, 128GB
1200 products are available.
Exynos 1280 Octa Core 2.4GHz 5nm Processor with the 12 band support
16200 -10%
18000
Add to Cart

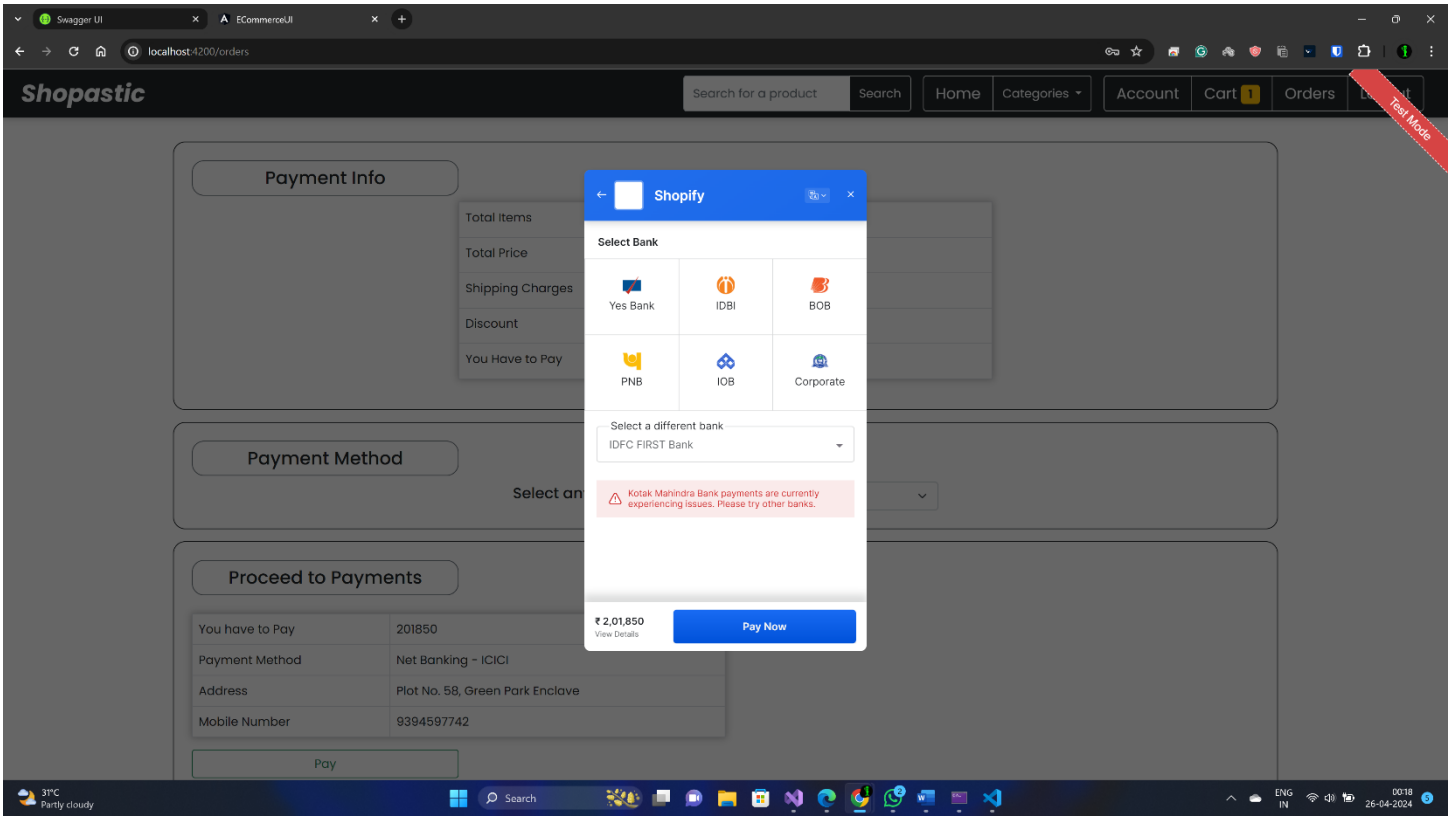

Swagger UI
ECommerceUI
localhost:4200/home
Shopastic
Search
Home
Categories
Login
Register
Enter Login Information
Email
curiosityrover2@gmail.com
Password
Login
Apple iPhone 13 Pro Max (256GB) - Sierra Blue
250 products are available.
17 cm (6.7-inch) Super Retina XDR display with ProMotion for a faster, more
115600 -15%
136000
Add to Cart (Login First)
Samsung Galaxy M33 5G (Emerald Brown, 6GB, 128GB
1200 products are available.
Exynos 1280 Octa Core 2.4GHz 5nm Processor with the 12 band support for a
16200 -10%
18000
Add to Cart (Login First)
OnePlus Nord CE 2 5G (Gray Mirror, 6GB RAM, 128GB Storage)
245 products are available.
65W SUPERVOOC – Accelerated charge velocity shall rocket the 4500mAh battery
21600 -10%
24000
Add to Cart (Login First)

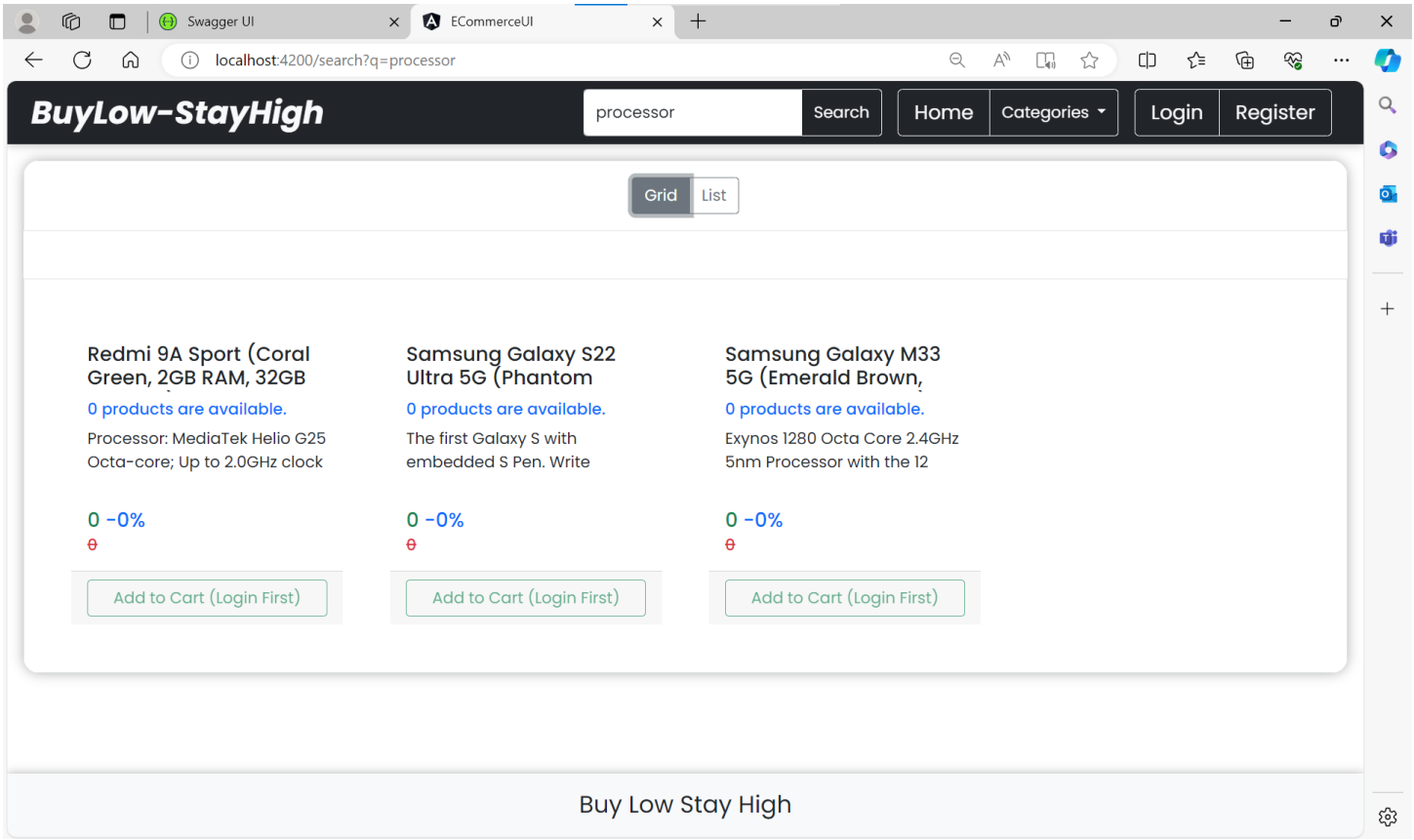

BuyLow-StayHigh
processor
Search
Home
Categories
Login
Register
Grid
List
Sort By
Default
Price (High to Low)
Price (Low to High)
TEKAVO Multipurpose Computer desk office table
500 products are available.
Made up of powder coated iron metal legs and thick particle
6750 -25%
9000
TEKAVO Laptop Table, Computer table for home,
400 products are available.
MODERN STYLE & AMPLE STORAGE SHELVES- The shelves can be fixed
5700 -5%
6000
Add to Cart (Login First)
TEKAVO L Shape Corner Desk Multi-Utility Office
200 products are available.
LARGE SPACE & STORAGE – TEKAVO - L shaped with big desk panel
9750 -25%
13000
Add to Cart (Login First)
ROUNDHILL Sheesham Wood Reading Writing
344 products are available.
Writing Study Table - This Study Table is made up of solid
13500 -10%
15000
Add to Cart (Login First)

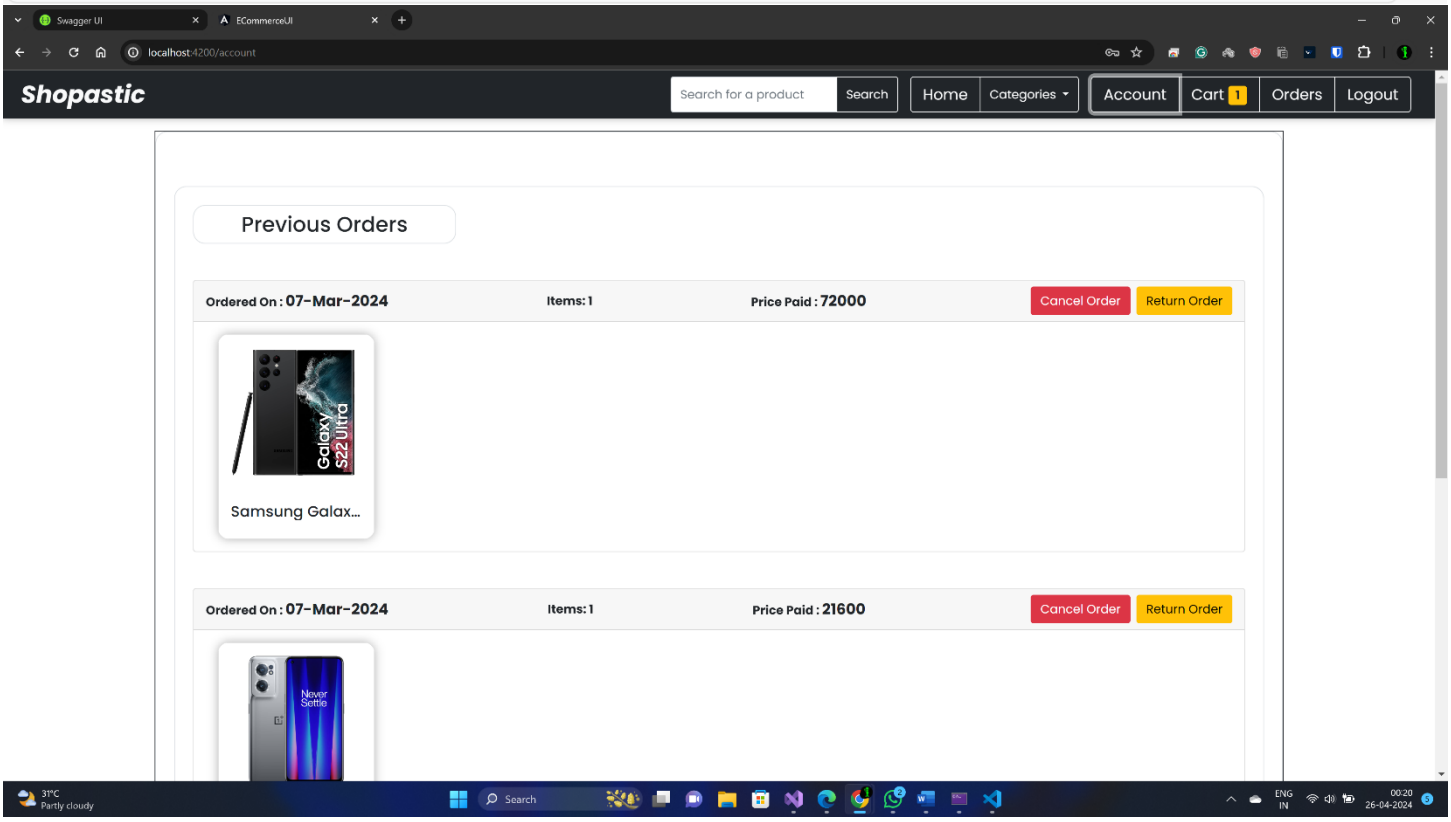

Swagger UI
ECommerceUI
localhost:4200/cart
Shopastic
Search for a product
Search
Home
Categories
Account
Cart 1
Orders
Logout
Current Cart
Order
Total Items 3
Total Price 247000
Shipping Charges 2000
Discount 45150
You Have to Pay 201850
Cart Items
OnePlus Nord CE 2 5G (Gray M...
21600 -10%
24000
Delete
2020 Apple MacBook Air Lapto...
69750 -25%
93000
Delete
ASUS ZenBook Duo 14 2021 Inte...
110500 -15%

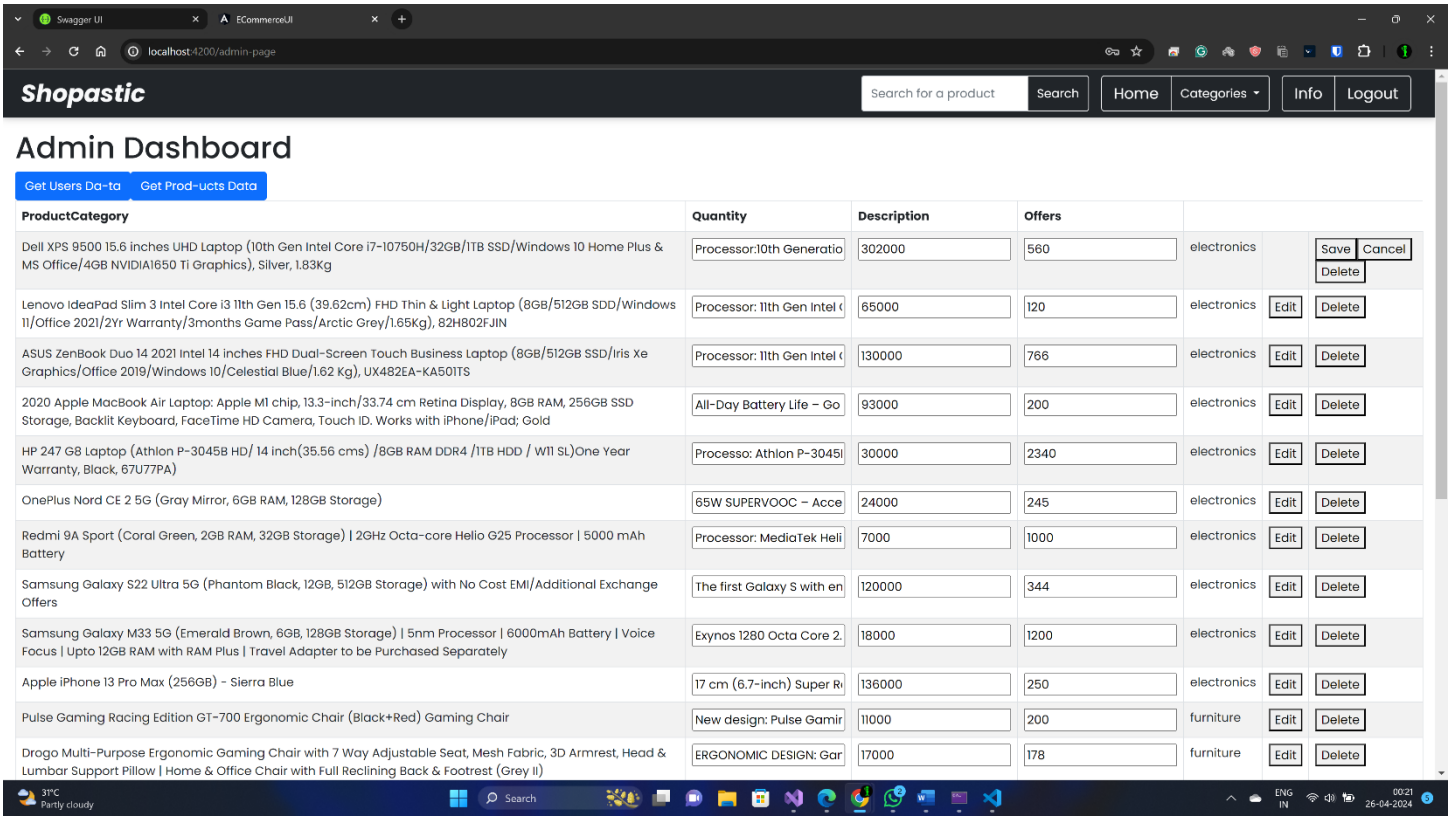

BuyLow-StayHigh
processor
Search
Home
Categories
Account
Cart 2
Orders
Logout
Redmi 9a Sport (Coral Green, 2gb Ram, 32gb Storage) | 2ghz Octa-core Helio G25 Processor | 5000 Mah Battery
6650 -5%
7000
1000 products are available.
Add to Cart
Description
Processor: MediaTek Helio G25 Octa-core; Up to 2.0GHz clock speed
Camera: 13 MP Rear camera with AI portrait| 5 MP front camera
Display: 16.58 centimeters (6.53-inch) HD+ display with 720x1600 pixels and 20:9 aspect ratio
Battery: 5000 mAH large battery with 10W wired charger in-box
Memory, Storage & SIM: 2GB RAM | 32GB storage | Dual SIM (nano+nano) + Dedicated SD card slot
The Selfie camera allows easy and convenient access to your phone with AI face unlock
form_factor:Bar,operating_system:MIUI 12.
Related Products
Samsung Galaxy M33 5G (Emerald Brown, 6GB, 128GB
1200 products are available.
OnePlus Nord CE 2 5G (Gray Mirror, 6GB RAM, 128GB Storage)
245 products are available.
Apple iPhone 13 Pro Max (256GB) - Sierra Blue
250 products are available.
50%
Reset
Related Products
HP 247 G8 Laptop (Athlon P-3045B HD/ 14 inch(35.56 cms)
2340 products are available.
Processo: Athlon P-3045B HD (2.3 GHz base frequency, up to 3.2 GHz burst
22500 -25%
30000
Add to Cart
2020 Apple MacBook Air Laptop: Apple M1 chip, 13.3-inch/33.74 cm
200 products are available.
All-Day Battery Life – Go longer than ever with up to 18 hours of battery life. Powerful
69750 -25%
93000
Add to Cart
Give Your Review
Write Your Review
Review Saved Succesfully!
Save Review
Other Reviews
Reviewed On: 26-Apr-2024
Nice Product
Buy Low Stay High


Swagger UI
ECommerceUI
localhost:4200/product-details?id=4
BuyLow-StayHigh
processor
Search
Home
Categories
Account
Cart 2
Orders
Logout
2020 Apple Macbook Air Laptop: Apple M1 Chip, 13.3-inch/33.74 Cm Retina Display, 8gb Ram, 256gb Ssd Storage, Backlit Keyboard, Facetime Hd Camera, Touch Id. Works With Iphone/ipad; Gold
69750 -25%
93000
200 products are available.
Add to Cart
Description
All-Day Battery Life – Go longer than ever with up to 18 hours of battery life
Powerful Performance – Take on everything from professional-quality editing to action-packed gaming with ease
The Apple M1 chip with an 8-core CPU delivers up to 3.5x faster performance than the previous generation while using way less power
Superfast Memory – 8GB of unified memory makes your entire system speedy and responsive
That way it can support tasks like memory-hogging multitab browsing and opening a huge graphic file quickly and easily
Stunning Display – With a 13.3-inch/33.74 cm Retina display, images come alive with new levels of realism
Shopastic
Search for a product
Search
Home
Categories
Account
Cart 1
Orders
Logout
Payment Info
Total Items 3
Total Price 247000
Shipping Charges 2000
Discount 45150
You Have to Pay 201850
Payment Method
Select any Method
Net Banking - ICICI
Proceed to Payments
You have to Pay 201850
Payment Method Net Banking - ICICI
Address Plot No. 58, Green Park Enclave
Mobile Number 9394597742
Pay


Swagger UI
ECommerceUI
localhost:4200/orders
Shopastic
Search for a product
Search
Home
Categories
Account
Cart 1
Orders
Test Mode
Payment Info
Total Items
Total Price
Shipping Charges
Discount
You Have to Pay
Payment Method
Select an
Proceed to Payments
You have to Pay 201850
Payment Method Net Banking - ICICI
Address Plot No. 58, Green Park Enclave
Mobile Number 9394597742
Pay
Shopify
Select Bank
Yes Bank
IDBI
BOB
PNB
IOB
Corporate
Select a different bank
IDFC FIRST Bank
Kotak Mahindra Bank payments are currently experiencing issues. Please try other banks.
₹ 2,01,850
View Details
Pay Now
31°C
Partly cloudy
Search
ENG
IN
00:18
26-04-2024


Swagger UI
ECommerceUI
localhost:4200/search?q=processor
BuyLow-StayHigh
processor
Search
Home
Categories
Login
Register
Grid
List
Redmi 9A Sport (Coral Green, 2GB RAM, 32GB
0 products are available.
Processor: MediaTek Helio G25 Octa-core; Up to 2.0GHz clock
0 -0%
0
Add to Cart (Login First)
Samsung Galaxy S22 Ultra 5G (Phantom
0 products are available.
The first Galaxy S with embedded S Pen. Write
0 -0%
0
Add to Cart (Login First)
Samsung Galaxy M33 5G (Emerald Brown,
0 products are available.
Exynos 1280 Octa Core 2.4GHz 5nm Processor with the 12
0 -0%
0
Add to Cart (Login First)
Buy Low Stay High


Swagger UI
ECommerceUI
localhost:4200/account
Shopastic
Search for a product
Search
Home
Categories
Account
Cart 1
Orders
Logout
Previous Orders
Ordered On : 07-Mar-2024
Items: 1
Price Paid : 72000
Cancel Order
Return Order
Samsung Galax...
Ordered On : 07-Mar-2024
Items: 1
Price Paid : 21600
Cancel Order
Return Order

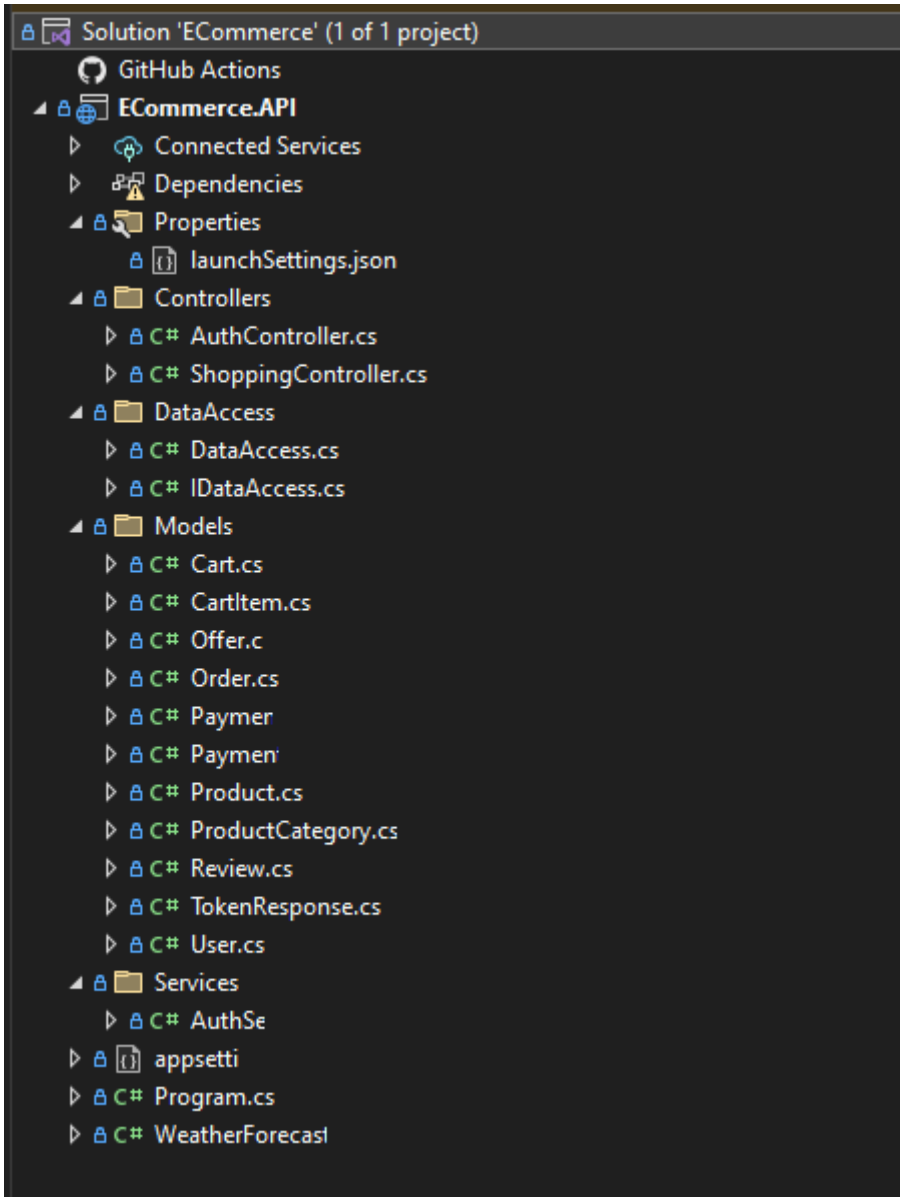

Swagger UI
ECommerceUI
localhost:4200/admin-page
Shopastic
Search for a product
Search
Home
Categories
Info
Logout
Admin Dashboard
Get Users Da-ta
Get Prod-ucts Data
ProductCategory
Quantity
Description
Offers
Dell XPS 9500 15.6 inches UHD Laptop (10th Gen Intel Core i7-10750H/32GB/1TB SSD/Windows 10 Home Plus & MS Office/4GB NVIDIA1650 Ti Graphics), Silver, 1.83Kg
Processor:10th Generatio
302000
560
electronics
Save
Cancel
Delete
Lenovo IdeaPad Slim 3 Intel Core i3 11th Gen 15.6 (39.62cm) FHD Thin & Light Laptop (8GB/512GB SDD/Windows 11/Office 2021/2Yr Warranty/3months Game Pass/Arctic Grey/1.65Kg), 82H802FJIN
Processor: 11th Gen Intel (
65000
120
electronics
Edit
Delete
ASUS ZenBook Duo 14 2021 Intel 14 inches FHD Dual-Screen Touch Business Laptop (8GB/512GB SSD/Iris Xe Graphics/Office 2019/Windows 10/Celestial Blue/1.62 Kg), UX482EA-KA501TS
Processor: 11th Gen Intel (
130000
766
electronics
Edit
Delete
2020 Apple MacBook Air Laptop: Apple M1 chip, 13.3-inch/33.74 cm Retina Display, 8GB RAM, 256GB SSD Storage, Backlit Keyboard, FaceTime HD Camera, Touch ID. Works with iPhone/iPad; Gold
All-Day Battery Life – Go
93000
200
electronics
Edit
Delete
HP 247 G8 Laptop (Athlon P-3045B HD/ 14 inch(35.56 cms) /8GB RAM DDR4 /1TB HDD / W11 SL)One Year Warranty, Black, 67U77PA)
Processo: Athlon P-3045I
30000
2340
electronics
Edit
Delete
OnePlus Nord CE 2 5G (Gray Mirror, 6GB RAM, 128GB Storage)
65W SUPERVOOC – Acce
24000
245
electronics
Edit
Delete
Redmi 9A Sport (Coral Green, 2GB RAM, 32GB Storage) | 2GHz Octa-core Helio G25 Processor | 5000 mAh Battery
Processor: MediaTek Heli
7000
1000
electronics
Edit
Delete
Samsung Galaxy S22 Ultra 5G (Phantom Black, 12GB, 512GB Storage) with No Cost EMI/Additional Exchange Offers
The first Galaxy S with en
120000
344
electronics
Edit
Delete
Samsung Galaxy M33 5G (Emerald Brown, 6GB, 128GB Storage) | 5nm Processor | 6000mAh Battery | Voice Focus | Upto 12GB RAM with RAM Plus | Travel Adapter to be Purchased Separately
Exynos 1280 Octa Core 2.
18000
1200
electronics
Edit
Delete
Apple iPhone 13 Pro Max (256GB) - Sierra Blue
17 cm (6.7-inch) Super R
136000
250
electronics
Edit
Delete
Pulse Gaming Racing Edition GT-700 Ergonomic Chair (Black+Red) Gaming Chair
New design: Pulse Gamir
11000
200
furniture
Edit
Delete
Drogo Multi-Purpose Ergonomic Gaming Chair with 7 Way Adjustable Seat, Mesh Fabric, 3D Armrest, Head & Lumbar Support Pillow | Home & Office Chair with Full Reclining Back & Footrest (Grey II)
ERGONOMIC DESIGN: Gar
17000
178
furniture
Edit
Delete
31°C
Partly cloudy
Search
ENG
IN
00:21
26-04-2024

Appendix-B(Coding)

GitHub link for the whole project is attached.

Backend (.NET)

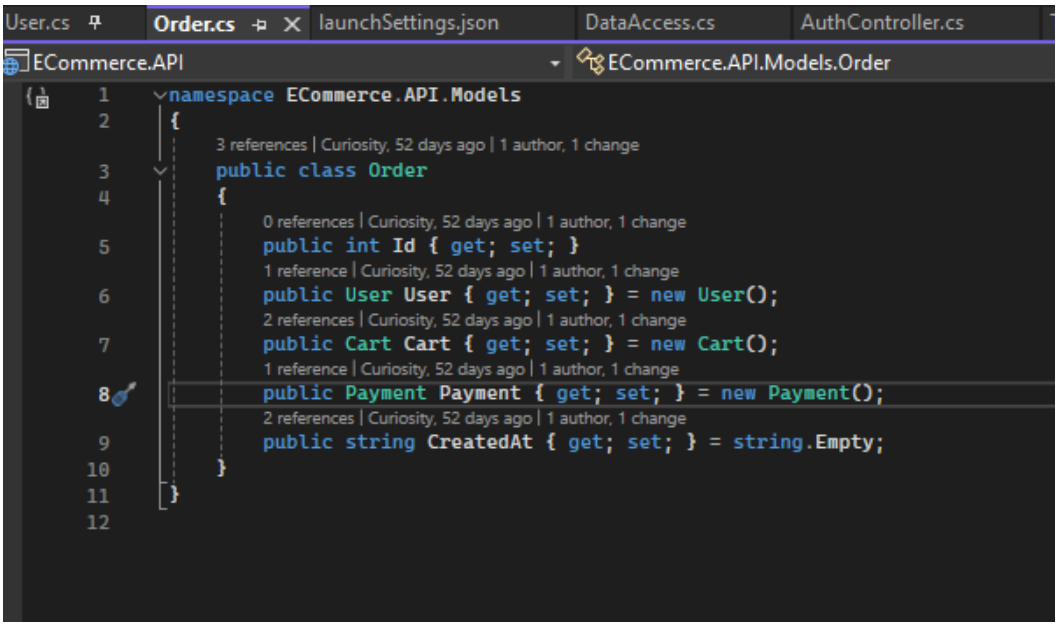
- Folder Structure :

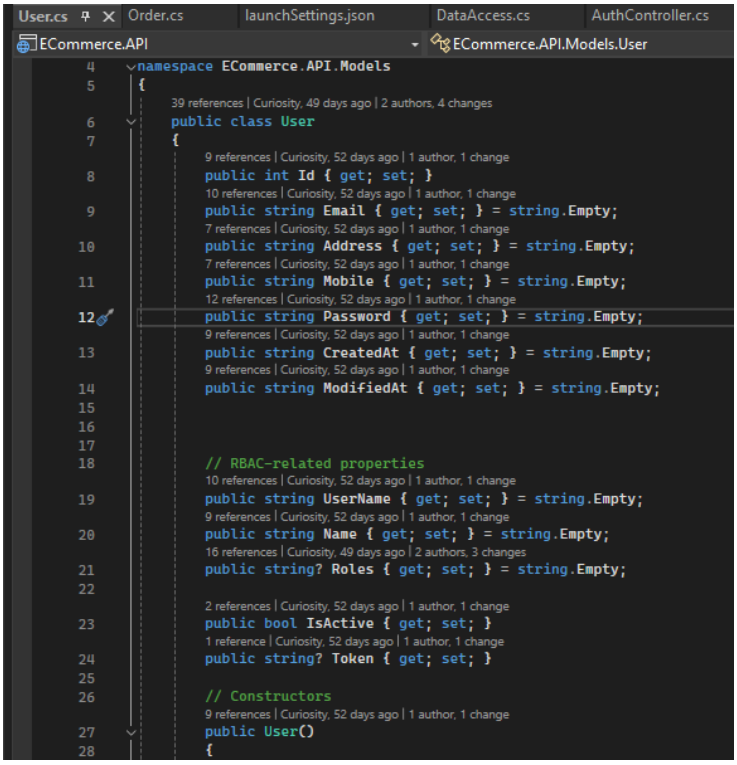


Let's break down the purpose and interaction of the key components:

Models

- Models: The Models folder contains classes that represent the data structures used in the application. These classes define the shape of the data, such as Product, User, Cart, etc. Each model class corresponds to a table in the database if you're using Entity Framework for data access. These models are used to transfer data between the database and the application's business logic.





Controllers

- Controllers: The Controllers folder contains classes that handle HTTP requests. Each controller corresponds to a specific resource or functionality in the application, such as AuthController for authentication-related actions and ShoppingController for shopping-related actions. Controllers are responsible for processing incoming HTTP requests, interacting with the models to perform business logic, and returning HTTP responses. They act as the intermediary between the client (e.g., a web browser or mobile app) and the server.

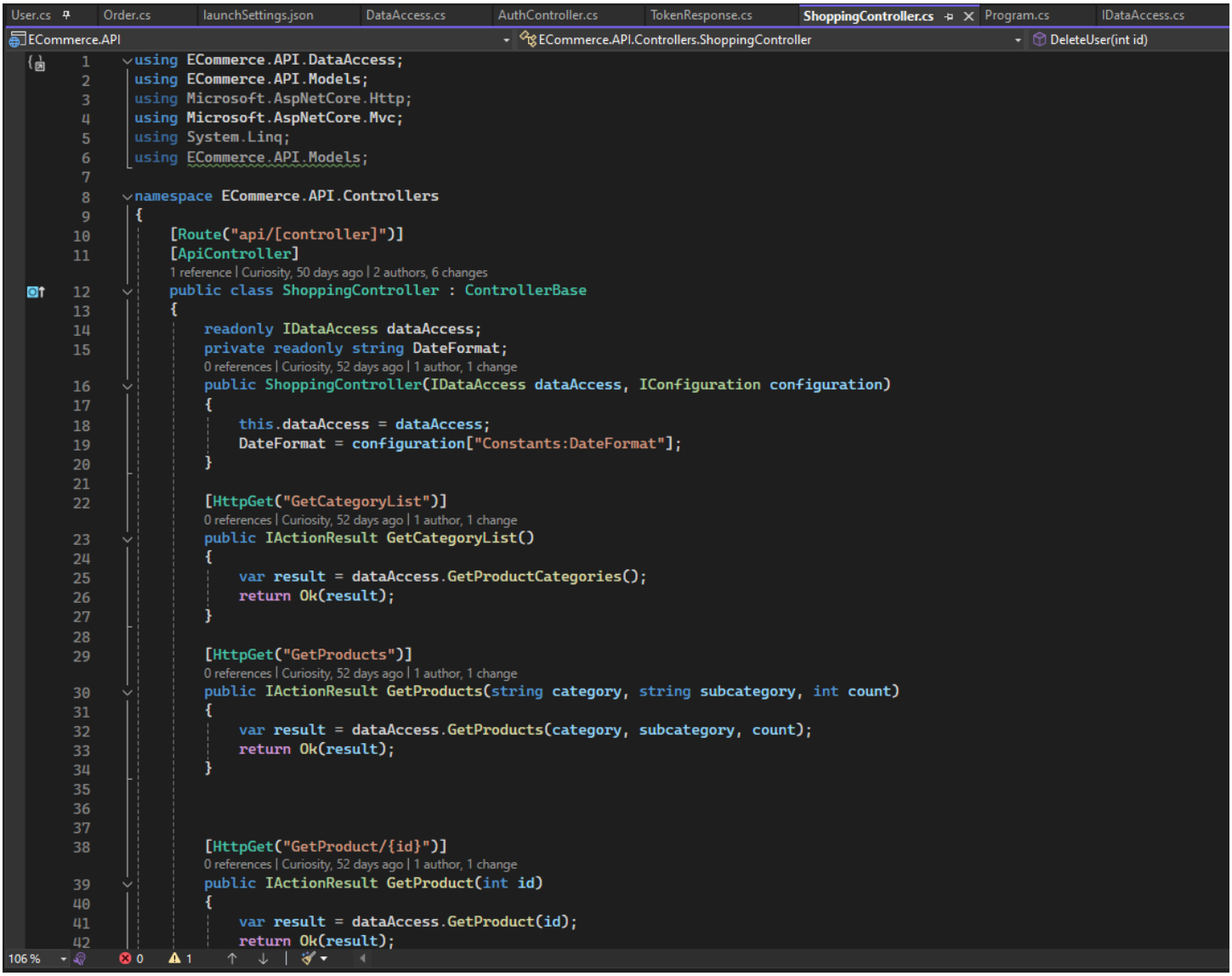


Figure 1: Shopping Controller

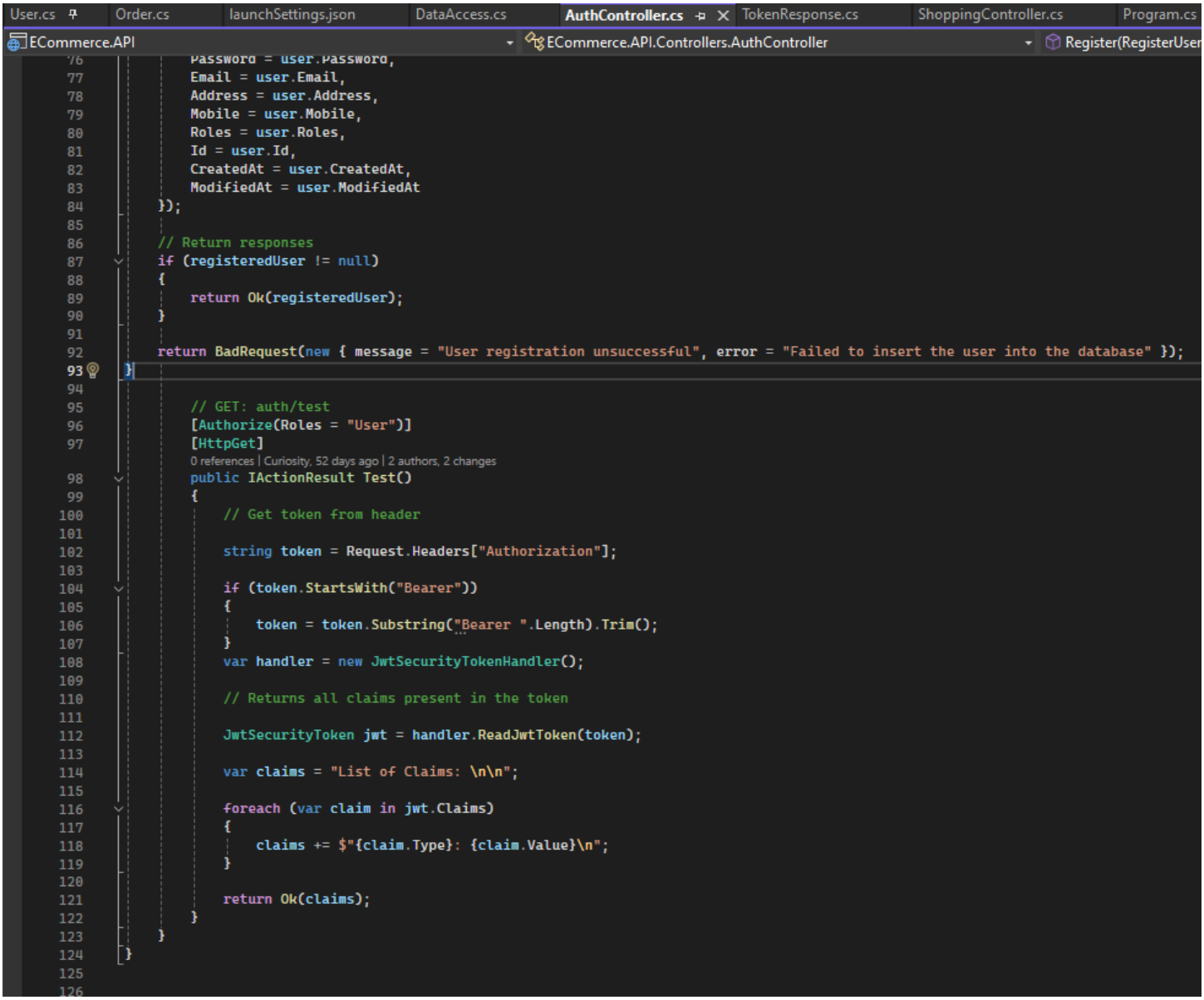
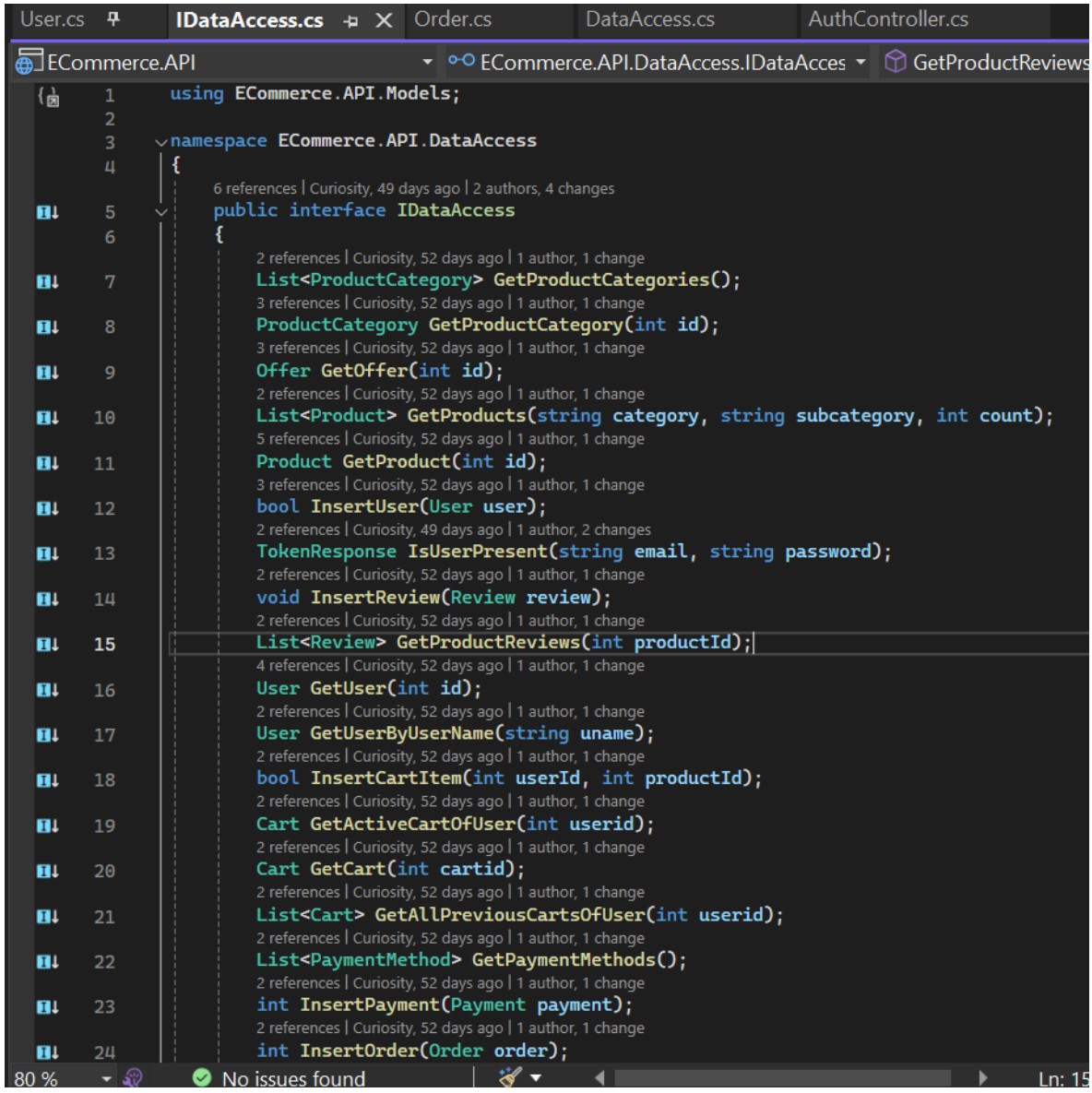


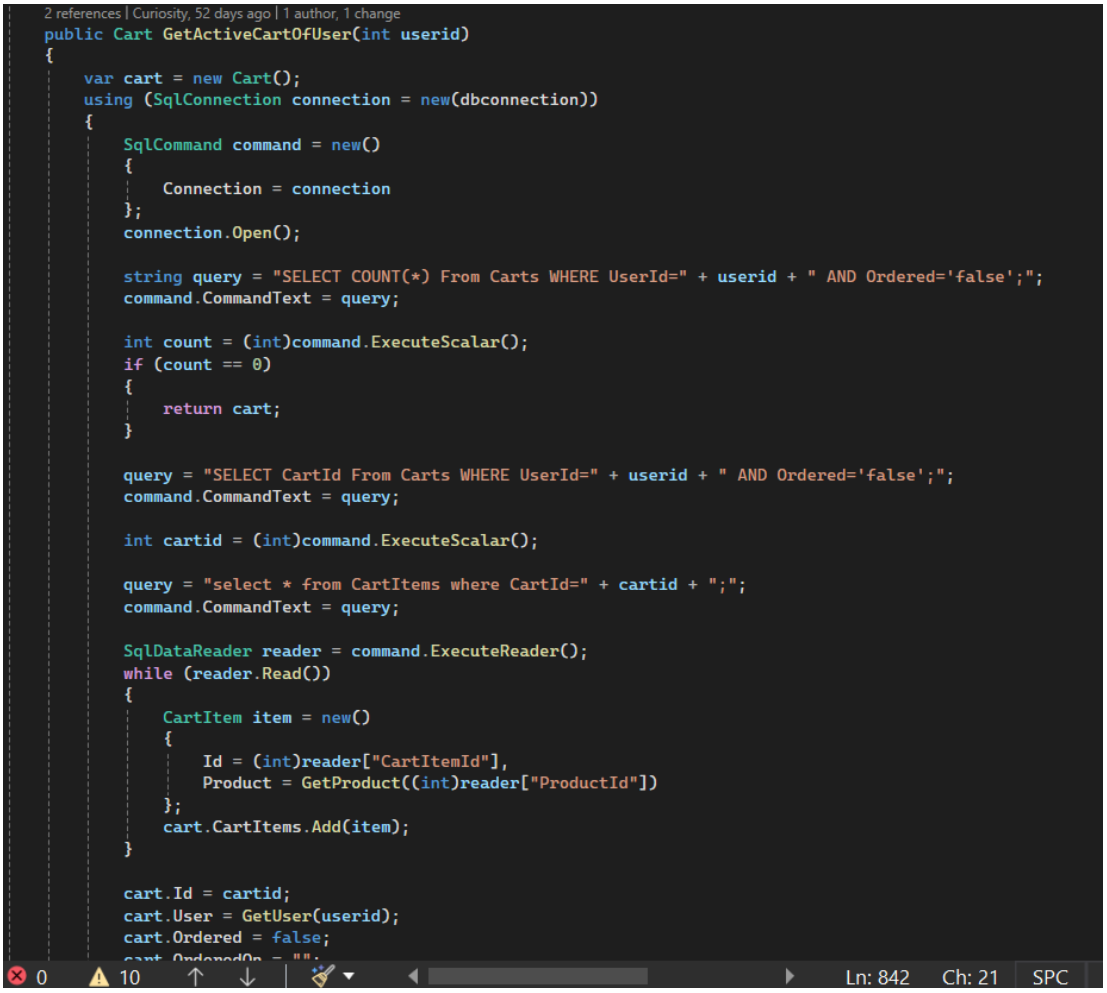
Figure 2:AuthController

DataAccess

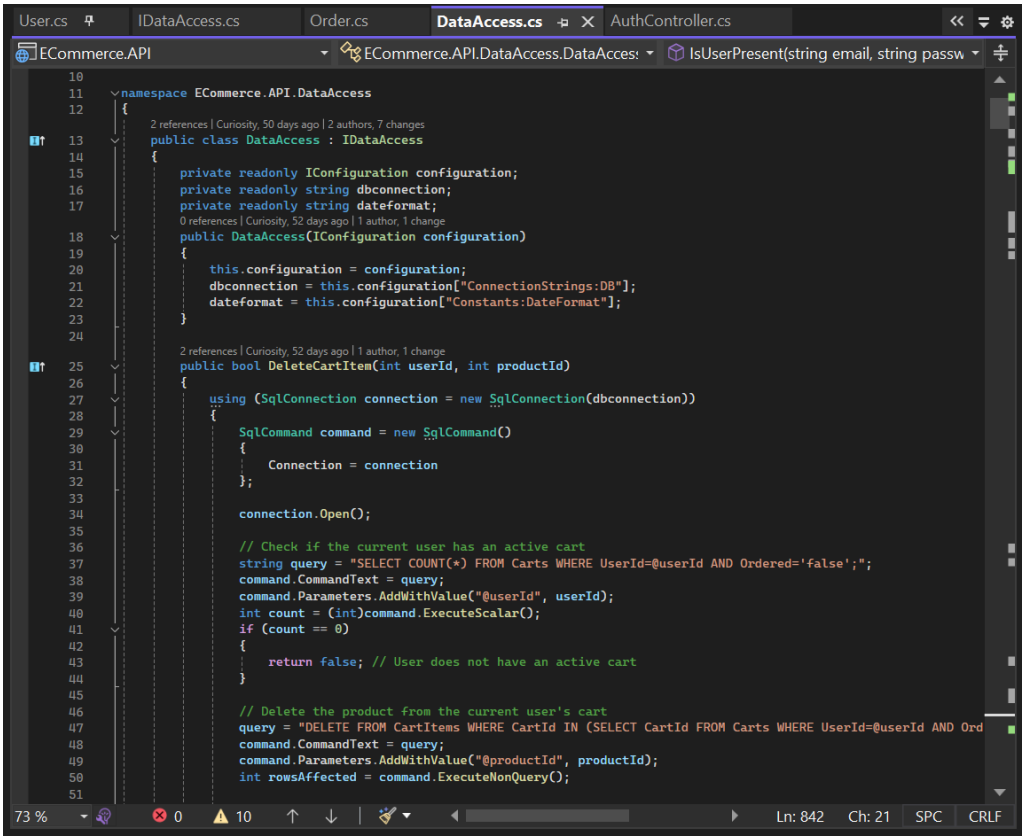
- DataAccess: This folder contains classes and interfaces related to data access. The DataAccess.cs file contains the implementation of data access logic, such as database operations for creating, reading, updating, and deleting records. The IDataAccess.cs file defines an interface that outlines the methods for data access, which can be implemented by the DataAccess class. This separation allows for easier testing and maintenance of the data access logic.

IDataAccess:

- 

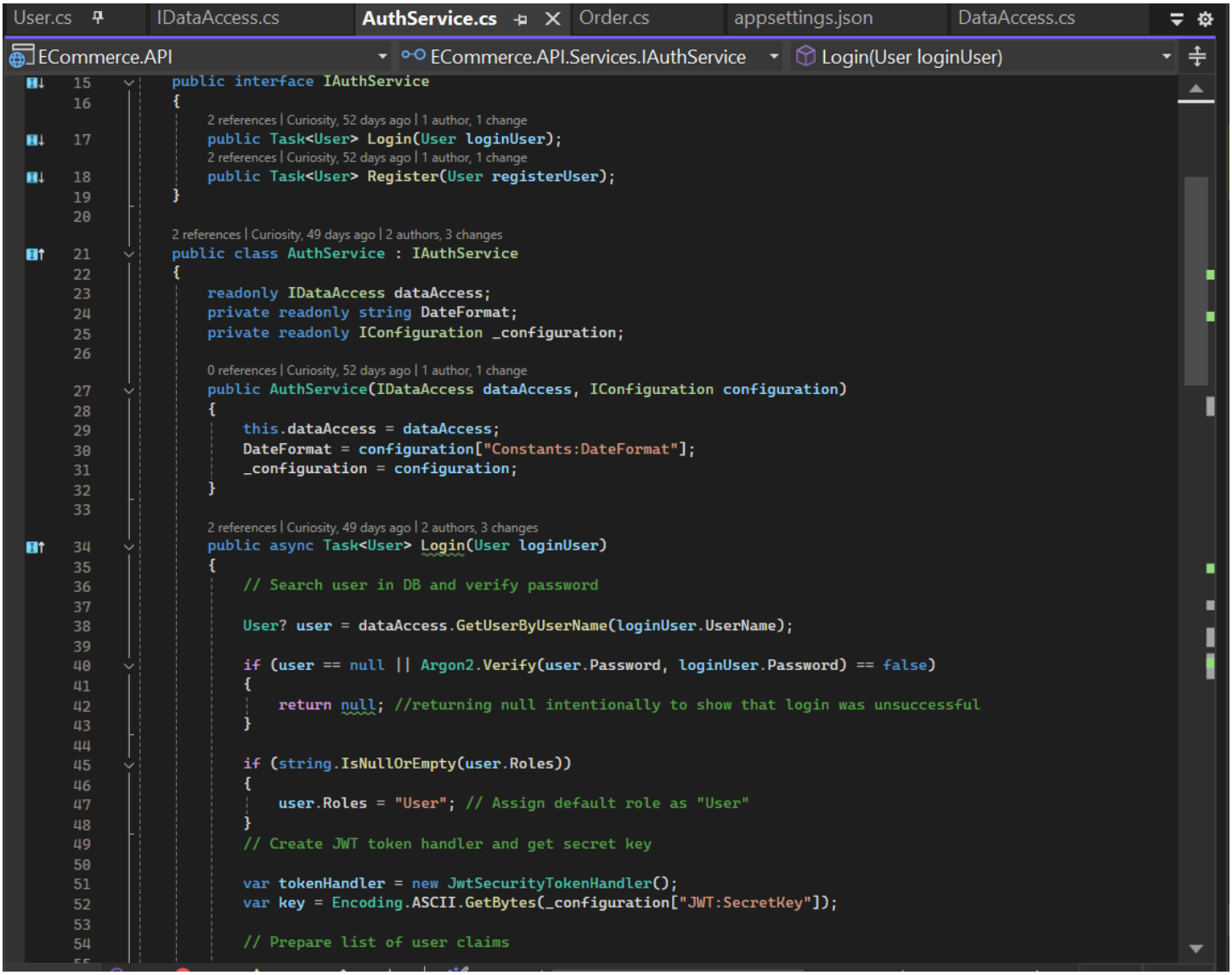


```
2 references | Curiosity, 52 days ago | 1 author, 1 change
public Cart GetActiveCartOfUser(int userid)
{
    var cart = new Cart();
    using (SqlConnection connection = new(dbconnection))
    {
        SqlCommand command = new()
        {
            Connection = connection
        };
        connection.Open();

        string query = "SELECT COUNT(*) From Carts WHERE UserId=" + userid + " AND Ordered='false';";
        command.CommandText = query;

        int count = (int)command.ExecuteScalar();
        if (count == 0)
        {
            return cart;
        }

        query = "SELECT CartId From Carts WHERE UserId=" + userid + " AND Ordered='false';";
        command.CommandText = query;

        int cartid = (int)command.ExecuteScalar();

        query = "select * from CartItems where CartId=" + cartid + ";";
        command.CommandText = query;

        SqlDataReader reader = command.ExecuteReader();
        while (reader.Read())
        {
            CartItem item = new()
            {
                Id = (int)reader["CartItemId"],
                Product = GetProduct((int)reader["ProductId"])
            };
            cart.CartItems.Add(item);
        }

        cart.Id = cartid;
        cart.User = GetUser(userid);
        cart.Ordered = false;
```

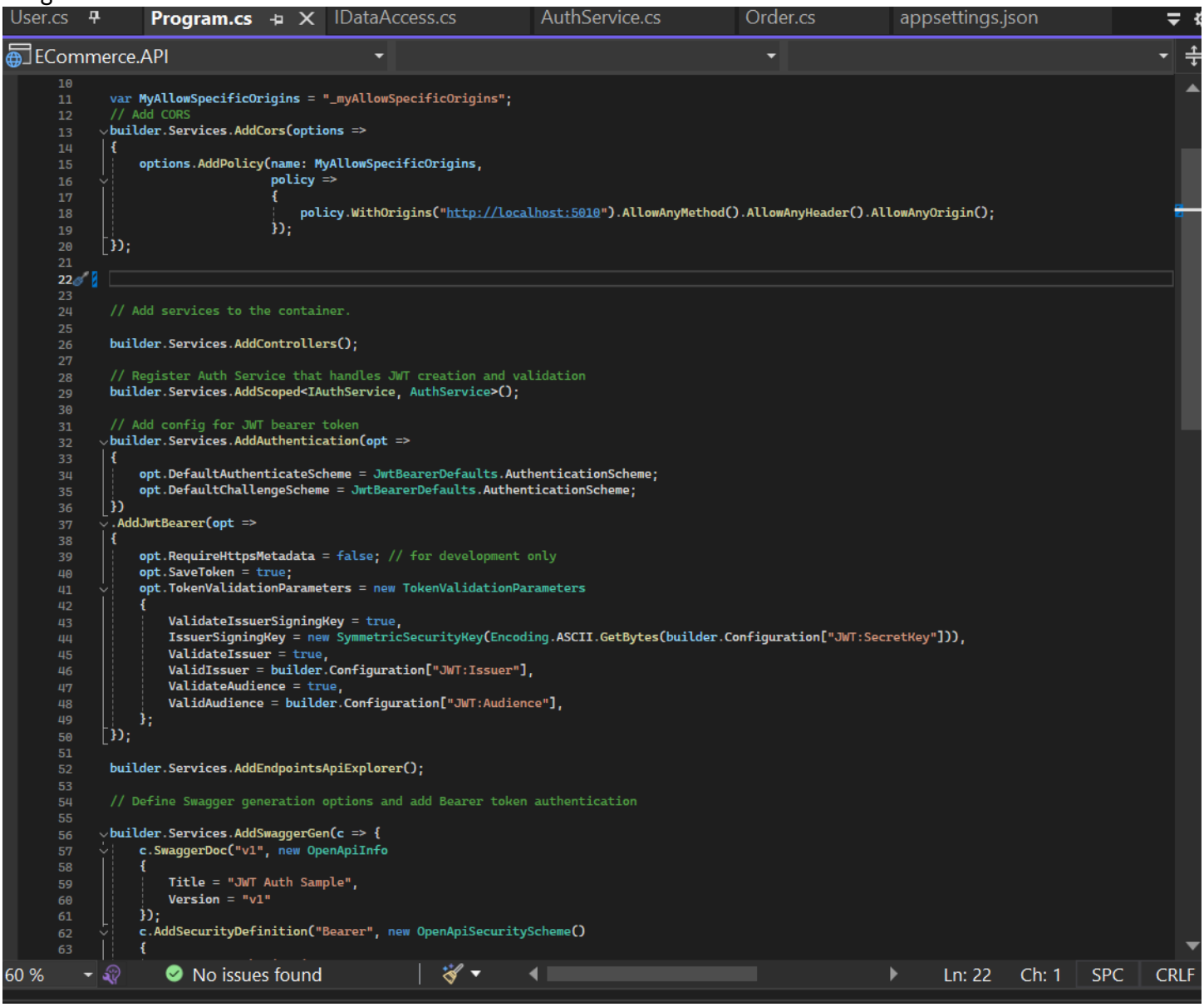

User.cs
IDataAccess.cs
Order.cs
DataAccess.cs
AuthController.cs
ECommerce.API
ECommerce.API.DataAccess.DataAcces
IsUserPresent(string email, string passw
namespace ECommerce.API.DataAccess
{
2 references | Curiosity, 50 days ago | 2 authors, 7 changes
public class DataAccess : IDataAccess
{
private readonly IConfiguration configuration;
private readonly string dbconnection;
private readonly string dateformat;
0 references | Curiosity, 52 days ago | 1 author, 1 change
public DataAccess(IConfiguration configuration)
{
this.configuration = configuration;
dbconnection = this.configuration["ConnectionStrings:DB"];
dateformat = this.configuration["Constants:DateFormat"];
}
2 references | Curiosity, 52 days ago | 1 author, 1 change
public bool DeleteCartItem(int userId, int productId)
{
using (SqlConnection connection = new SqlConnection(dbconnection))
{
SqlCommand command = new SqlCommand()
{
Connection = connection
};
connection.Open();
// Check if the current user has an active cart
string query = "SELECT COUNT(*) FROM Carts WHERE UserId=@userId AND Ordered='false';";
command.CommandText = query;
command.Parameters.AddWithValue("@userId", userId);
int count = (int)command.ExecuteScalar();
if (count == 0)
{
return false; // User does not have an active cart
}
// Delete the product from the current user's cart
query = "DELETE FROM CartItems WHERE CartId IN (SELECT CartId FROM Carts WHERE UserId=@userId AND Ord
command.CommandText = query;
command.Parameters.AddWithValue("@productId", productId);
int rowsAffected = command.ExecuteNonQuery();
73 %
0
10
Ln: 842
Ch: 21
SPC
CRLF

Services

- Services: The Services folder contains classes that encapsulate business logic and rules. For example, AuthService.cs contain methods for user authentication, password hashing, and token generation. Services are used by controllers to perform operations that involve business logic. This separation of concerns makes the code more modular and easier to maintain.



Program.cs File:



Here's a brief description of each part:

CORS Policy

- CORS Policy: The application defines a CORS policy named `_myAllowSpecificOrigins` that allows requests from `http://localhost:5010`. This policy is configured to allow any method, any header, and any origin, which is suitable for development purposes. In a production environment, you would typically restrict the allowed origins to specific, trusted domains to enhance security.

Services and Middleware

- Services: The application registers services with the dependency injection container. It adds controllers, an authentication service (`IAuthService`), and a data access service (`IDataAccess`). The authentication

service is scoped, meaning a new instance is created for each request, while the data access service is a singleton, meaning only one instance is created and shared across the application.

- Authentication: The application configures JWT bearer token authentication. It specifies that the default authentication and challenge schemes are JWT bearer tokens. The token validation parameters are configured to validate the issuer and audience, and the signing key is obtained from the application's configuration.

- Swagger: The application adds Swagger for API documentation. It configures Swagger to document the API and adds a security definition for JWT bearer tokens. This allows API consumers to easily understand and test the API endpoints.

- HTTP Request Pipeline: The application configures the HTTP request pipeline. It uses Swagger and Swagger UI in development mode, enforces HTTPS redirection, applies authentication and authorization middleware, applies the CORS policy, and maps controllers to routes.

Running the Application

- Running: Finally, the application is built and run. The `app.Run()` method starts the application and listens for incoming HTTP requests.

This setup provides a solid foundation for building a secure, well-documented, and cross-origin accessible web API

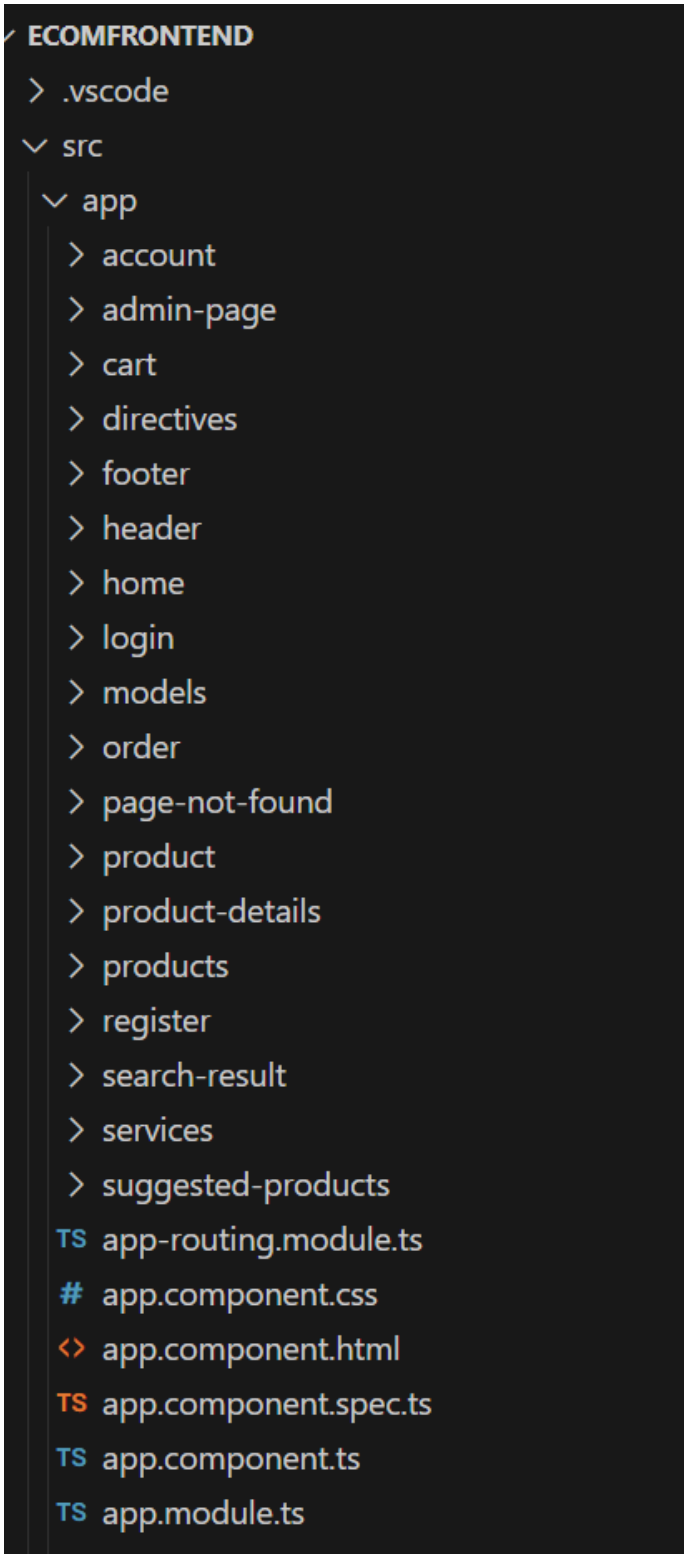
How the Flow Works

1. Client Request: A client (e.g., a web browser or mobile app) sends an HTTP request to the server.
2. Routing: The request is routed to the appropriate controller based on the URL and HTTP method. This is managed by the ASP.NET Core routing system, which can be configured using attributes in the controllers or convention-based routing in the `Startup.cs` file.
3. Controller Action: The controller's action method corresponding to the request is invoked. This method contains the business logic to handle the request.
4. Data Access: If the action requires data from the database, it calls methods in the `DataAccess` layer to retrieve or manipulate data.
5. Model Interaction: The controller may interact with model classes to represent the data being worked with. This could involve creating new instances of model classes, updating existing ones, or querying the database for data.
6. Response: The controller prepares an HTTP response, which could be a view (in MVC applications) or data (in API applications), and sends it back to the client.

This structure allows for a clean separation of concerns, making the application easier to develop, test, and maintain. Each component has a specific role, and changes in one area are less likely to affect others.

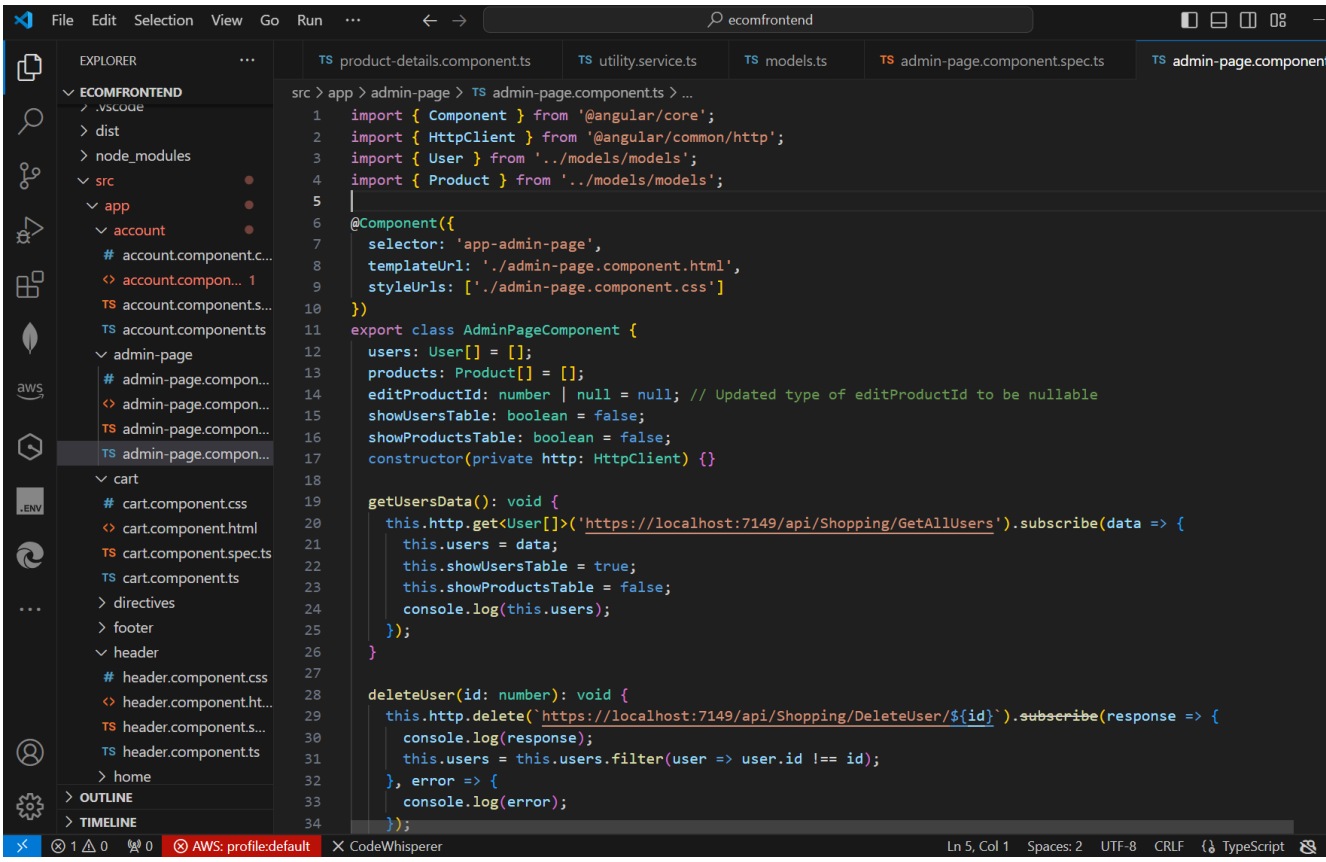
Frontend(Angular)

- Folder Structure :



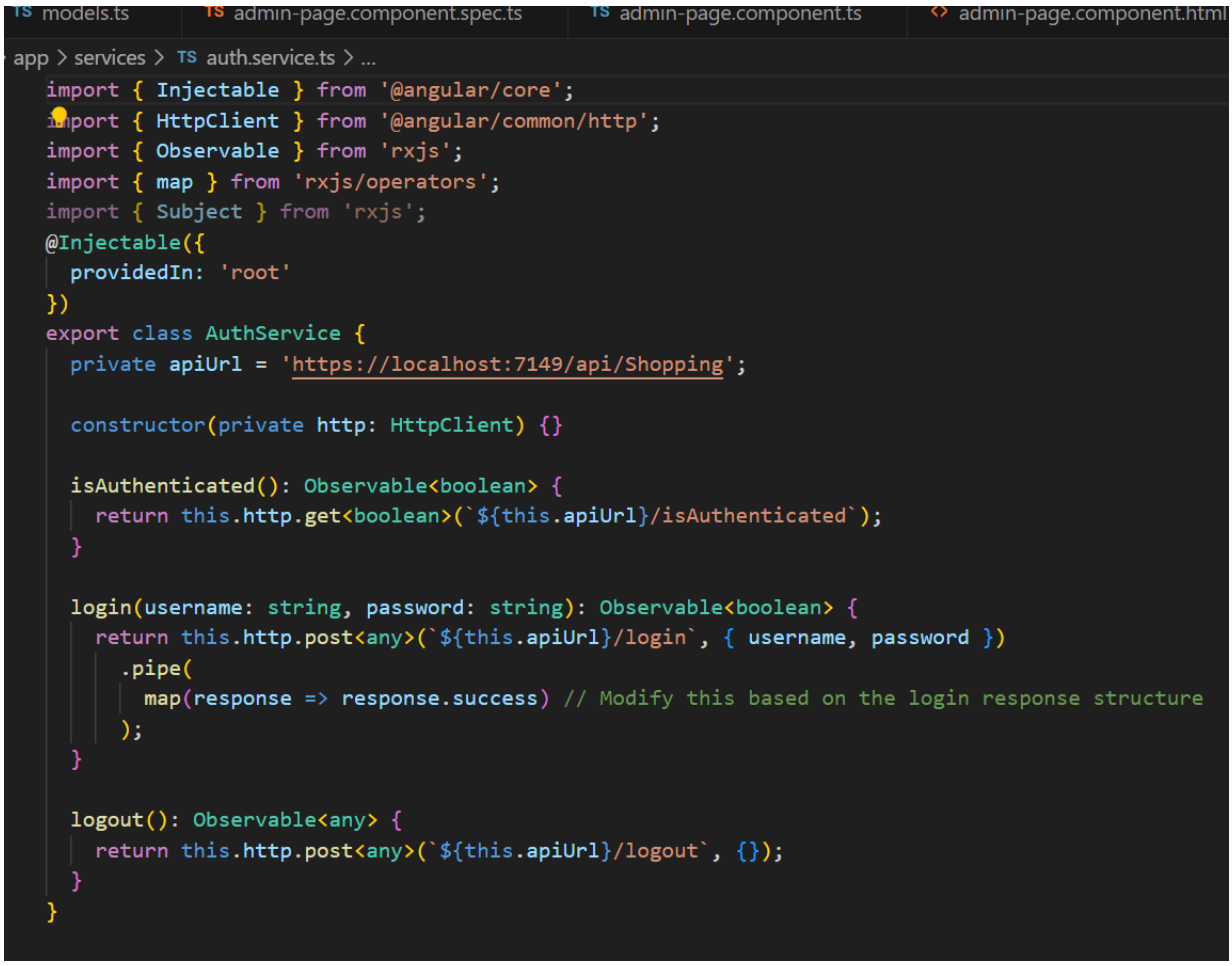
- src: The source directory where all the application's TypeScript, HTML, CSS, and other assets are stored. It's the heart of the Angular application.

- app: Contains the core components, services, and modules of the application. This is where the main logic and functionality of the application reside.
    - account: Components related to user account management, such as login, registration, and account settings.
    - admin-page: Components for administrative functionalities, likely used for managing products, orders, and user accounts.



    - cart: Components for the shopping cart functionality, allowing users to add items to their cart and proceed to checkout.
    - directives: Custom Angular directives that encapsulate specific behaviors or functionalities that can be reused across the application.
    - footer, header: Components for the application's footer and header, which are likely to be present on every page.
    - home: The homepage component, showcasing featured products or promotions.
    - login, register: Components for user authentication and registration.
    - models: TypeScript interfaces or classes that define the structure of data used in the application, such as product or user information.

- order, product, product-details, products, search-result: Components related to product listing, details, and search functionalities.
- services: Services for handling business logic, such as API calls to a backend server



- suggested-products: Components for displaying suggested products to users.
- app-routing.module.ts: The main routing module that defines the navigation paths within the application.
- app.component.css, app.component.html, app.component.spec.ts, app.component.ts: The root component of the application, which typically includes the main layout and navigation.
- app.module.ts: The root module that bootstraps the application and declares the components, services, and modules used in the application.
- auth.guard.ts: A guard that protects routes and ensures that only authenticated users can access certain parts of the application.

- assets: Contains static files like images, icons, and other assets used by the application.

- environments: Contains environment-specific configuration files, such as development and production settings.

REFERENCES

1. Angular & .NET Core Ecommerce Website: A fully functional e-commerce website developed using Angular 8 and .NET Core. It includes features like product listing, search, cart management, and payment integration. The source code and a detailed tutorial can be found on GitHub: https://github.com/ecommerce-website-dotnet/Angular-Azure-DevOps

2. MEAN Stack Ecommerce Application: A comprehensive e-commerce application built using the MEAN stack (MongoDB, Express.js, Angular, and Node.js). It showcases the power of Angular for the front-end and integrates with a robust .NET backend API for handling order processing and inventory management. Source code and documentation: https://github.com/onehungrymind/angular-meanstack-supertable

3. Build an e-commerce app with Angular 11 and .NET 5: Step-by-step tutorial on creating a scalable e-commerce application using the latest versions of Angular and .NET. This tutorial covers topics like authentication, product management, shopping cart functionality, and deployment. The tutorial along with the complete source code can be accessed here: https://www.positronx.io/build-angular-ecommerce-app-with-authentication-in-net-core/

4. Building an E-commerce Application with ASP.NET Core and Angular: A free eBook that guides you through the process of building an end-to-end e-commerce application using ASP.NET Core and Angular. It covers topics like database design, RESTful API development, client-side routing, and user authentication. The eBook can be downloaded here: https://dotnet.microsoft.com/download/e-book/aspnet/angular-apps