Independent University **Bangladesh (IUB)**

IUB Academic Repository

Internship Reports Summer 2023

2023-09-21

ShopHive E-Commerce Web Application

Bhuiyan, Faisal

Independent University, Bangladesh

https://ar.iub.edu.bd/handle/11348/689

Downloaded from IUB Academic Repository



ShopHive E-Commerce Web Application

By

Faisal Bhuiyan

Student ID: **1921298**

Summer, 2023

Supervisor:
Md. Mahmudul Kabir Peyal

Research & Development Officer

Department of Computer Science & Engineering

Independent University, Bangladesh

September 21, 2023

Dissertation submitted in partial fulfillment for the degree of
Bachelor of Science in Computer Science

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

I, Faisal Bhuiyan, a trainee.NET Developer at Nerd Castle Ltd, hereby certify that the material contained in this internship report accurately represents my work and experiences during my term.

I contributed actively to the development of the ShopHive ecommerce project, utilizing.NET Web API for backend development and Angular for frontend development. Team cooperation, API implementation, user interface design, and overall software quality assurance were within my responsibilities.

I am grateful to my mentors and colleagues at Nerd Castle Ltd for their advice and assistance in helping me progress professionally.



18-Oct-23

Signature Date

Faisal Bhuiyan

Name

Acknowledgement

I would like to express my heartfelt gratitude to everyone who assisted with making my internship and the ShopHive ecommerce project a success.

First and foremost, I am eternally thankful to Zohirul Alam Tiemoon, the CEO of Nerd Castle Ltd and my devoted mentor, whose guidance, insights, and constant encouragement were crucial in developing my understanding of software development and assisting me in navigating the project's complexities. His patience and desire to share his knowledge have been invaluable to me.

I want to thank the entire development team at Nerd Castle Ltd for welcoming me with open arms and creating a welcoming and cooperative atmosphere in which I could learn and grow. My internship experience was not only productive but also enjoyable due to the team's unity and desire to assist at every step.

I am also grateful to Zinia Hoque, Md Faisal, and the rest of the team for their ongoing encouragement, constructive criticism, and the opportunity to contribute meaningfully to the ShopHive project.

In addition, I'd like to thank my fellow interns and staff members who shared their experiences and perspectives, creating an environment of mutual growth and development.

I also like to thank faculty members of Independent University Bangladesh (IUB), for providing me with the knowledge and skills I needed to complete this internship and submit this report.

Finally, I want to express my heartfelt gratitude to my family and friends for their unfailing support, encouragement, and understanding throughout this journey.

I am truly honored to have been a part of the ShopHive project team, and I am confident that the lessons and experiences gained during this internship will continue to shape my future path.

Letter of Transmittal

Md. Mahmudul Kabir Peyal

Research & Development Officer

Department of Computer Science and Engineering

Independent University, Bangladesh

Subject: Internship Report

Dear Sir,

I hope this letter finds you in good health and high spirits. I am writing to transmit my internship report, which was completed as part of my undergraduate internship program at Nerd Castle Ltd.

As a.NET Developer Trainee, I got the pleasure to work with the great team at Nerd Castle Ltd for the past four months. I was able to obtain excellent hands-on experience in software development and improve my skills in numerous elements of web development throughout my time there. My internship was mostly focused on the development of ShopHive, an e-commerce platform built with.NET as the backend and Angular as the frontend.

In this report, I've given a thorough description of the work I did during my internship, including project objectives, methods, technical specifics, and obstacles faced. I've also emphasized the significance of my contributions to the overall project, demonstrating how I applied the theoretical information learned in my academic courses to practical, real-world settings.

In addition, the report contains a full analysis of the technologies, tools, and frameworks utilized in the development of ShopHive, as well as a discussion of best practices applied throughout the project. In addition, I have included my own comments on the learning experiences and professional growth that I gained throughout this internship.

I hope I was able to meet all your expectations in this report. Finally, I would like to thank you for assisting me and addressing all my queries with kindness.

Sincerely,
Faisal Bhuiyan
1921298
+8801790750458

Evaluation Committee

Supervision Panel

…… Academic Supervisor	…….. Industry Supervisor

Panel Members

…… Panel Member-1	…….. Panel Member-2

Panel Members

…… Panel Member-3	…….. Panel Member-4

Office Use

…… Program Coordinator Name: Subrata Kumar Dey	…….. Head of Department Computer Science & Engineering Name: Mahady Hasan

Abstract

In this three-month EDGE Project Internship, I will undertake a significant programming project concentrated on developing an advanced E-commerce web application. The primary objective is to cultivate my professional programming skills essential for real-world applications. The project will follow a systematic approach, encompassing stages such as gathering and analyzing functional requirements, designing a scalable database schema, implementing a secure .Net backend, crafting an intuitive Angular frontend, ensuring seamless integration and testing, and deploying the application to a reliable web server. Potential future enhancements include integrating multiple payment gateways, instituting intelligent product recommendations, creating native mobile applications, supporting multiple languages, and incorporating analytics tools for data-driven decisions. Overall, this internship offers an opportunity to cultivate practical programming skills and sets the foundation for a successful career in the programming domain.

Contents

Attestation 2

Acknowledgement 3

Letter of Transmittal 4

Evaluation Committee 5

Abstract 7

List of Figures 10

List Of Tables 11

Chapter 1 12

Introduction 12

1.1 Overview/Background of the Work 12

1.2 Objectives 12

1.3 Scopes 13

Chapter 2 14

Literature Review 14

2.1 Relationship with Undergraduate Studies 14

2.2 Related works 14

Chapter 3 16

Project Management & Financing 16

3.1 Work Breakdown Structure 16

3.2 Process/Activity wise Time Distribution 17

3.3 Gantt Chart 18

3.4 Estimated Costing 18

Chapter 4 19

Methodology 19

Chapter 5 20

Body of the Project 20

5.1 Work Description 20

5.2 Requirement Analysis 21

5.3 System Analysis 22

5.3.2 Feasibility Analysis 23

5.3.3 Problem Solution Analysis 25

5.3.4 Effect and Constraints Analysis 26

5.4 System Design 27

5.4.1 UML Diagrams 27

5.4.2 Architecture 30

5.5 Implementation 31

5.6 Testing 36
5.6.1 Input 36
5.6.2 Output 37
5.6.3 Designing Test Cases 37
5.6.4 Test Results 38
Chapter 6 39
Results & Analysis 39
Chapter 7 43
Project as Engineering Problem Analysis 43
7.1 Sustainability of the Project/Work 43
7.2 Social and Environmental Effects and Analysis 44
7.3 Addressing Ethics and Ethical Issues 45
Chapter 8 46
Lesson Learned 46
8.1 Problems Faced During this Period 46
8.2 Solution of those Problems 46
Chapter 9 47
Future Work & Conclusion 47
9.1 Future Works 47
9.2 Conclusion 47
Bibliography 48
References 48

List of Figures

Figure 1: Time Distribution 17
Figure 2: Gantt Chart 18
Figure 3: Rich Picture 21
Figure 4: Use Case Diagram 27
Figure 5: Activity Diagram for Customer checking out a cart 28
Figure 6: Activity Diagram for Admin Activities 29
Figure 7: System Architecture 30
Figure 8: Generating a JWT Token 31
Figure 9: User, CustomerBasket and Product Models 31
Figure 10: Basket Repository Code 32
Figure 11: Product Repository Code 32
Figure 12: Basket API Controller 33
Figure 13: Register User API Controller 33
Figure 14: Loading Categories, CurrentUser and Basket initially 34
Figure 15: Configuring Application Routes 34
Figure 16: ShopService methods 35
Figure 17: LoginComponent code 35
Figure 18: Home Page 40
Figure 19: Shop Page 40
Figure 20: Product Detail Page 41
Figure 21: Cart Page 41
Figure 22: Login Page 42
Figure 23: Sign Up Page 42

List Of Tables

Table 1: Six Element Analysis..... 23
Table 2: Test Input Cases..... 36
Table 3: Test Output Cases..... 37
Table 4: Validation Testing 37
Table 5: Authorization Testing..... 38
Table 6: Error Testing 38
Table 7: Connection Testing 38

Chapter 1

Introduction

1.1 Overview/Background of the Work

In this era of digital revolution, e-commerce continues to grow as a critical component of modern retail. The ShopHive e-commerce project, conceptualized and developed during my internship at Nerd Castle Ltd, demonstrates the significant benefits of an online shopping platform that caters to a wide range of products, including electronics, accessories, and food.

ShopHive leverages digital technology to provide consumers with a unique shopping experience while also providing businesses with an effective way to extend their market reach. As the need for online shopping grows, this project represents the exciting future of e-commerce, which bridges the gap between ease, accessibility, and a worldwide marketplace.

1.2 Objectives

The key goals of the ShopHive e-commerce initiative are as follows:

- ✓ **Platform Creation:** Build a strong and user-friendly e-commerce platform that supports a diverse range of product categories while assuring seamless navigation and a responsive user interface.
- ✓ **Enhanced Shopping Experience:** Incorporate features that boost the overall shopping experience, such as customized product recommendations, user reviews, and a secure checkout process.
- ✓ **Business Integration:** Develop a platform that enables businesses to efficiently list and manage their items by leveraging sophisticated inventory management and order processing capabilities.
- ✓ **Scalability:** As the business grows, ensure that the platform can scale to support increased user traffic and product listings.
- ✓ **Data Analytics:** Use data analytics tools to gain insights into customer behavior, which may then be used for targeted marketing and better inventory management.

1.3 Scopes

The ShopHive e-commerce project's scope includes the following:

Frontend Development: Using Angular, design and create a dynamic and user-friendly online interface to provide customers with an appealing and intuitive buying experience.

Backend Development: Using.NET, implement backend logic for things like product administration, user authentication, and secure payment processing.

Database Management: This involves setting up a strong database system to securely store product information, user profiles, and transaction records.

Safety Measures: Advanced safeguards are used to protect user data, financial transactions, and sensitive information.

User Authentication: Protecting user accounts by implementing secure user authentication processes such as password hashing and multi-factor authentication.

Testing and quality assurance: This comprises extensive evaluation to detect and resolve defects, as well as verifying the platform's dependability and performance.

Documentation: For future reference, thoroughly document the project's development process, technical specifications, and best practices.

Chapter 2

Literature Review

2.1 Relationship with Undergraduate Studies

I worked with two major languages/framework while building this project, which is Angular and .NET. While these two frameworks were not explicitly taught to me in any of my undergraduate courses, I was able to quickly grasp their concepts because of them.

Angular is a JavaScript framework that was created by Google while .NET was introduced my Microsoft. The expertise I gathered from CSE309 (Web Application and Internet) assisted me greatly while learning Angular. On the other hand, the CSE213 (Object Oriented Programming in Java) was helpful for .NET since .NET uses object oriented principles and it's very similar to Java.

I also had to work with a database to store products, categories, cart etc. where I had to work with Advanced SQL and I believe CSE303 (Database Management) was useful here. Lastly, CSE203 (Data Structures) and CSE211 (Algorithms) played an important role in optimizing my application for the best output and performance.

2.2 Related works

As numerous businesses require E-Commerce platforms for their day-to-day operations, there are lots of projects that are similar in scope.

Here are some of the highlights:

- **Shopizer**
    - Language/Framework: Java Spring Boot
    - GitHub Repository: https://github.com/shopizer-ecommerce/shopizer
- **Shuup**
    - Language/Framework: Python/Django
    - GitHub Repository: https://github.com/shuup/shuup

- **Angularspree**
    - Language/Framework: Angular
    - GitHub Repository: https://github.com/aviabird/angularspree
- **OpenCart**
    - Language/Framework: PHP
    - GitHub Repository: https://github.com/opencart/opencart
- **Shopping Cart**
    - Language/Framework: Python/Flask
    - GitHub Repository: https://github.com/HarshShah1997/Shopping-Cart

Chapter 3

Project Management & Financing

3.1 Work Breakdown Structure

Project Description:

An e-commerce web application which will have two major users:

1) Customers – Customers will have the ability to create accounts and log in securely. They will be able to browse a catalog of products with images, descriptions and prices. They will be able to filter the products by category and search for specific products. Additionally, they will be able to add Products to cart and manage orders.

2) Administrators – Administrators will be able to delete any customer account and add products to Inventory. They will also be able to update product

Front End Developer: Faisal Bhuiyan

UI/UX designer: Faisal Bhuiyan

Backend/Server side developers: Faisal Bhuiyan

Timeline of the Project:

● Starting date: 1st December 2021

● Ending date: 31st March 2022

Phases:

1. Requirement gathering: Backend

2. Implementation: Backend

3. Beta Testing: Backend

4. Requirement gathering and UI Design: Front End

5. Implementation: Front End

6. Beta Testing: Front End

7. Deployment

3.2 Process/Activity wise Time Distribution

The expected duration for delivering this project was a little less than two months. In order to make the best use of the time available, I divided the time based on the impact of each phase/activity on the entire project.

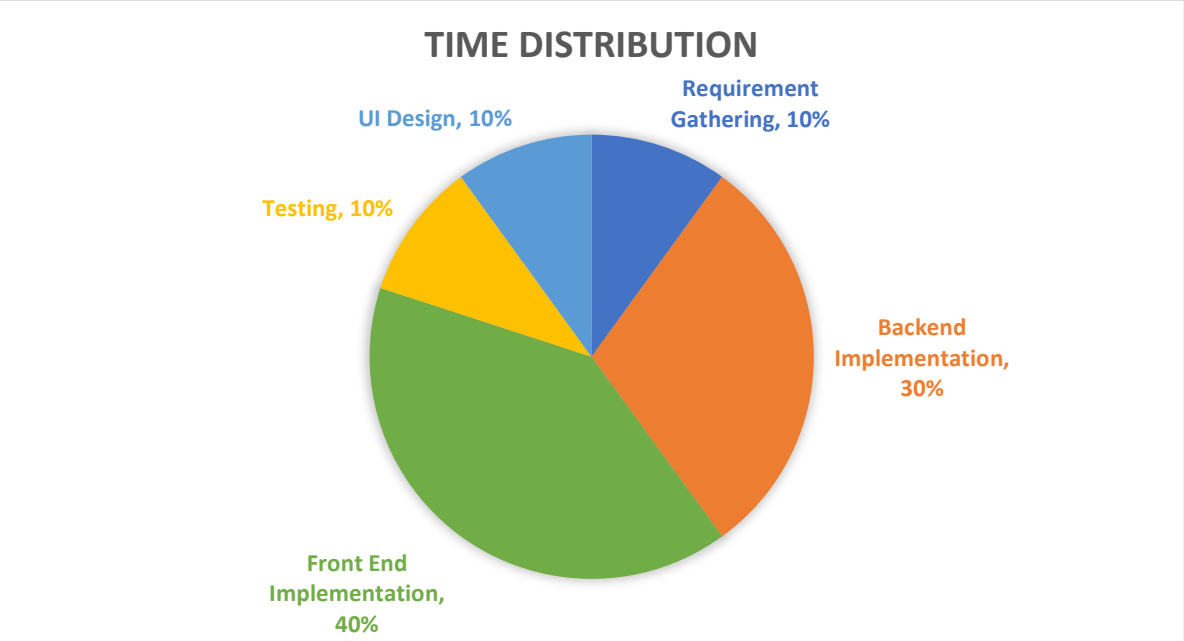


Figure 1: Time Distribution

3.3 Gantt Chart

A Gantt chart is a visual tool used for project planning. It shows tasks as bars on a timeline, making it easy to see when each task starts and finishes. It helps teams stay organized and on track with their projects.

- ✓ This Gantt chart is a high-level overview of the tasks involved in the e-commerce project. The actual tasks and timelines may vary depending on the specific project.
- ✓ The dependencies between tasks are shown here. For example, the "Front End UI design" task cannot start until the "Backend Testing" task is complete.

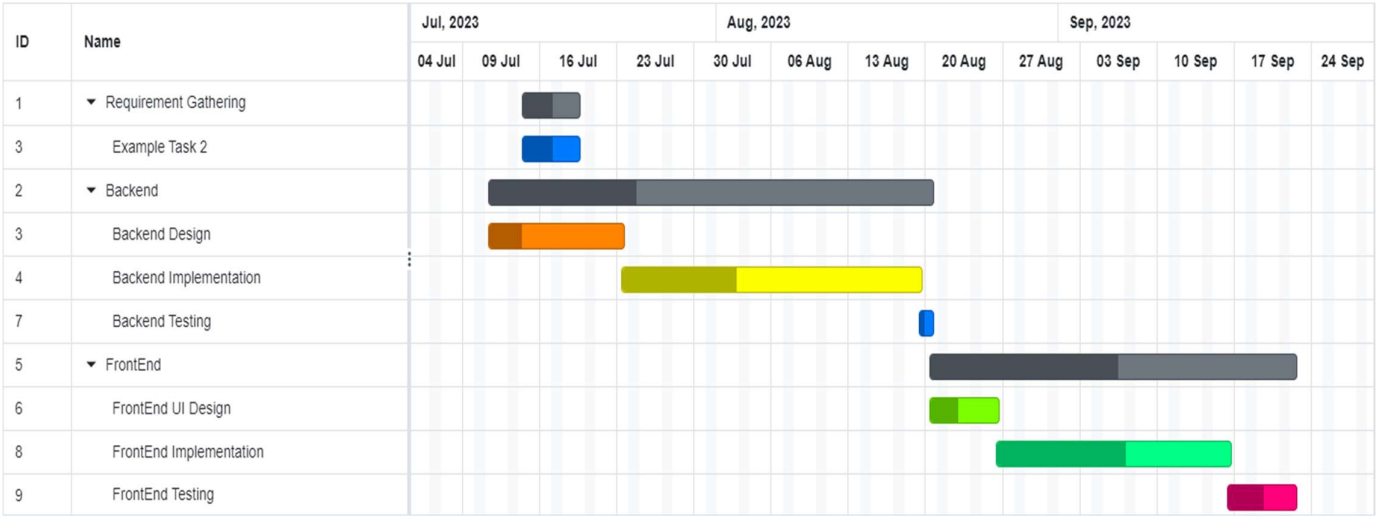


Figure 2: Gantt Chart

3.4 Estimated Costing

Since this is a testing project and no real user will use this project on a daily basis, the costing is negligible. However, in the case that this project is taken over and real users are involved, the costing will be as follows:

- ✓ Frontend development (Angular): 100,000 BDT
- ✓ Backend development (.NET Core): 200,000 BDT
- ✓ Database (Microsoft SQL Server): 50,000 BDT
- ✓ Payment service: 10,000 BDT
- ✓ Other costs (hosting, domain name, etc.): 50,000 BDT

Total: 360,000 BDT

Chapter 4

Methodology

Let me guide you through how I went about creating my e-commerce website. The procedure can be broken down into various stages:

1. *Planning the Project and Requirements Analysis:* I began by thoroughly researching the project's requirements. This phase involves determining the essential features and operations of the website and developing a plan for how I would proceed.

2. *Backend Development:* I laid the groundwork for the website here. I created the database structure to store critical data such as product information and user information. Then, using.NET Web API, I created the backend logic to handle things like user logins and order processing.

3. *Frontend Development:* Now that the backend was in place, I focused on the user interface. I used Angular to design and build the website's many components, such as product listings, shopping carts, and user profiles. Throughout the process, I tested and polished the user experience to ensure it was fluid and intuitive.

4. *Integration and Testing:* The goal of this phase was to ensure that everything worked together effortlessly. I linked the frontend and backend, thoroughly testing their interactions. I also ran tests to discover and correct any problems, ensuring that the website was as bug-free as possible.

5. *Optimization and Efficiency Tuning:* I concentrated on optimization to guarantee the website could manage an increasing number of users and products. This included duties such as enhancing database performance, developing caching methods, and improving code efficiency. I also ran load tests to check how the system handled high traffic.

6. *Project Evaluation and Conclusion:* After launching the website, I took a step back to assess the entire project. I learnt significant lessons, faced and overcame problems, and provided suggestions for future improvements.

Chapter 5

Body of the Project

5.1 Work Description

Assigned Role:
Trainee ASP.Net Developer

Joining Date:
June 17, 2023

Project Start Date:
July 10, 2023

Project Completion Date:
September 20, 2023

Day to Day Work:

As a trainee at Nerd Castle Ltd, my goal was to study ASP.Net development from the ground up, either through in-office sessions or through online courses.

I also had to learn, develop, and integrate the Angular front-end framework with ASP.Net projects. Before moving on to the main project, I began by creating tiny Windows Forms applications for learning purposes.

Everything was developed, implemented, coded, tested, and maintained from the ground up by me.

5.2 Requirement Analysis

There are two users of this application overall. 1 is an Admin who is able to add or remove products from the website. Second is the customers. Customers can register themselves through the front end. Then, the user information will be passed to the backend and the data will be stored in Microsoft sql server database. The Admin will be able to add new products to the database. The users can add products to their cart when needed only if they are logged in.

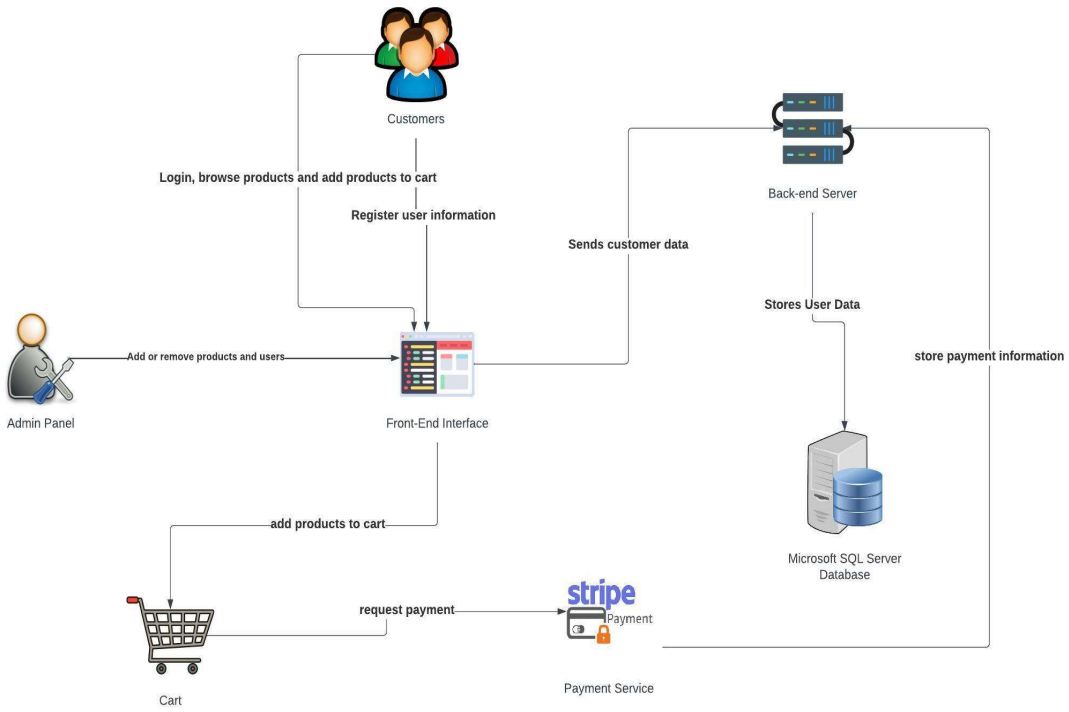


Figure 3: Rich Picture

Functional Requirements:

1. User Registration and Authentication: Enable users to create accounts and securely log in.
2. Product Catalog: Showcase a product catalog with features such as photos, descriptions, and prices.
3. Shopping Cart: Allow customers to add products to a virtual shopping cart, see and amend the cart's contents, and check out.
4. Product Search and Filtering: Make search capabilities available to assist consumers in finding products based on keywords, categories, brands, and other filters.

5. Payment Gateway Integration: Integrate with a secure payment gateway to enable online transactions and allow several payment methods (credit cards, PayPal, and so on).

6. Responsive Design: Make certain that the web application is mobile-friendly and optimized for various devices and screen sizes.

Non-Functional Requirements:

1. Performance: Quick page load speeds and responsiveness; concurrent user traffic handling at peak periods.
2. Scalability: Create horizontal scaling in order to support rising user bases.
3. Security: Strong data protection methods, encryption for sensitive data, and web vulnerability protection.
4. Reliability: Low downtime and errors, as well as regular data backups and disaster recovery strategies.
5. Load Handling: Capacity to handle expected traffic spikes; load testing.
6. Monitoring and logging: Detailed logging for troubleshooting and auditing; performance and security monitoring.

5.3 System Analysis

5.3.1 Six Element Analysis

Process	Human	Hardware (Computing)	Software	Database	Network
Login	All	Phone/Tablet/PC	ShopHive Web App	Retrieves user information	Internet
Update User	Admin	Phone/Tablet/PC	ShopHive Web App	Update user details	Internet
Delete User	Admin	Phone/Tablet/PC	ShopHive Web App	Delete User from database	Internet
Add items to Basket/Cart	Customer	Phone/Tablet/PC	ShopHive Web App	Add or remove products	Internet
Register User	Customer	Phone/Tablet/PC	ShopHive Web App	Add new user information	Internet

Add new Category	Admin	Phone/Tablet/PC	ShopHive Web App	Add new category information	Internet
Update Category	Admin	Phone/Tablet/PC	ShopHive Web App	Update category information	Internet
Add Product	Admin	Phone/Tablet/PC	ShopHive Web App	Add product information	Internet
Checkout	Customer	Phone/Tablet/PC	ShopHive Web App	Add order information	Internet

Table 1: Six Element Analysis

5.3.2 Feasibility Analysis

During my internship, I completed a feasibility analysis to assess the practicality and viability of constructing the ShopHive online application. This examination took into account a number of factors, including technological feasibility, market feasibility, and resource feasibility.

Technical Suitability

Backend Development: The ShopHive backend was built with.NET Core, Entity Framework, and Microsoft SQL Server. The use of known technology secured the project's technical sustainability. APIs for user registration, login, product administration, and cart functionality were successfully implemented.

Frontend construction: Angular was chosen as the frontend framework, allowing for the efficient development of user interfaces. User identification, product filtering, and cart management were all integrated flawlessly, confirming the technical capability of the selected technology stack.

Database Management: Using Microsoft SQL Server for the database and Redis for cart management confirmed the technical survival of the database technologies used. These options provided scalability as well as robust data management.

Market Suitability

ShopHive fills a rising need for online purchasing platforms. Given the growing desire for e-commerce solutions, the market feasibility analysis suggested a promising future. To develop a presence in the market, however, competitive research and marketing techniques would be required.

Feasibility of Resources

Development Resources: The availability of qualified developers trained in.NET Core, Angular, and database technologies secured the project's resource viability. Within the constraints of the available resources, the project timeframe and scope were efficiently handled.

Budgetary Considerations: The budget for the project was carefully calculated to meet real life development expenditures, license fees, and infrastructural requirements. The funding was within the limits of the internship project's resources.

In conclusion, the feasibility analysis indicates that the ShopHive web application is technically viable, is comparable with market trends, and can be produced effectively within the resources available. However, continued market research and marketing efforts will be required to reach the company's full potential in a competitive environment.

5.3.3 Problem Solution Analysis

An in-depth problem analysis was carried out in the context of the ShopHive project in order to identify important challenges and demonstrate suitable fixes.

Issue 1: Limited User Participation in Initial Requirements Gathering

Problem: Gathering complete user requirements proved difficult during the project's first phase. Survey and questionnaire participation was low, resulting in insufficient data collection.

Solution: To overcome this challenge, the project used an all-inclusive strategy. The survey participant sample was extended to include a more diversified mix of potential consumers. This technique attempted to improve the correctness of the system's design by taking into account a greater range of user viewpoints.

Issue 2: Scalability and Performance Enhancement

Problem: As the project progressed, issues about scalability and performance optimization surfaced. The initial architecture was insufficient to accommodate the expected increase in user traffic and data volume.

Solution: To address these problems head on, the project underwent a thorough evaluation of its design and infrastructure. This resulted in the use of scalable design principles and performance optimization strategies. As a result, the project assured that ShopHive could handle increasing customer activity and data without sacrificing speed.

Issue 3: Vulnerabilities in Security

Problem: It is critical to ensure strong security in an e-commerce platform. The research revealed possible vulnerabilities that needed to be addressed in order to safeguard user data and transactions.

Solution: To strengthen the platform's security, the project enlisted the help of online security resources and learned from extensive vulnerability analyses and penetration testing. The findings influenced the development of strong security measures, such as encryption protocols, authentication methods, and frequent security audits.

Issue 4: Integration Difficulties

Problem: Integrating the system's many components, such as the frontend, backend, and third-party services, caused integration issues and possible bottlenecks.

Solution: To ensure modularity and flexibility, the project used microservices architecture to speed integration. This method allowed for smooth communication across system components and decreased integration complexity.

The ShopHive project aims to improve its development processes, optimize performance, reinforce security, and create a scalable and efficient e-commerce platform for its users by solving these different difficulties with effective solutions.

5.3.4 Effect and Constraints Analysis

A complete testing of several limitations and their related consequences has been done within the context of the ShopHive project.

Budgetary Impact: Financial restrictions have a significant impact on the project's resource allocation and length of time. Budgetary constraints affect the extent to which project staff may be hired and have a substantial impact on the development path. The project's scale, scope, and availability of critical resources are all determined by the amount of money.

Time constraints: The relationship between money and time restrictions is boundless. The project timeframe has a connection to the budget allocation. In the instance of ShopHive, a time schedule was established, along with a precise deadline for project completion. This time limitation instills a feeling of urgency in the development team, imposing strict planning and streamlined execution with the goal to accomplish project objectives within the timeframe specified.

Scope Definition: The scope constraint serves as a starting point for defining the project's boundaries and requirements. It serves as a checkpoint to ensure that established project criteria are met. Throughout the development process, careful examination of project requirements is required. As project deadlines approach, it may be necessary to rebalance the project scope, delay some components, or prioritize particular parts. This limitation highlights the relevance of continuous discussions, reviews, and meetings in order to maintain the project's quality standards.

User Adoption Rate: End-user adoption of the ShopHive platform is critical to its success. While the system's features and functionality have been carefully created, the rate at which consumers migrate from traditional purchasing methods to the digital platform has an impact on the project's impact. This limitation highlights the importance of appropriate user training and continuous user assistance to ensure a seamless transition.

Market Competition: The e-commerce domain's competitive environment imposes a limitation that influences the project's market placement. The project has to constantly evaluate and adapt to rivals' strategy and advancements. This constraint needs to be responsive to market changes, consequently driving marketing strategies and product developments.

Technological Evolution: Rapid technological improvements and changing client preferences represent an impact restriction for the ShopHive initiative. It is critical to remain technologically current and adaptive to evolving trends. This limitation

emphasizes the significance of flexibility in the project's design and continuing development activities.

Cybersecurity Risks: The growing incidence of cyber-attacks and data breaches is an enormous obstacle. It is critical to protect user data and ensure the security of online transactions. This restriction needs constant awareness, regular security assessments, and active risk management techniques.

5.4 System Design

5.4.1 UML Diagrams

Use Case Diagram:

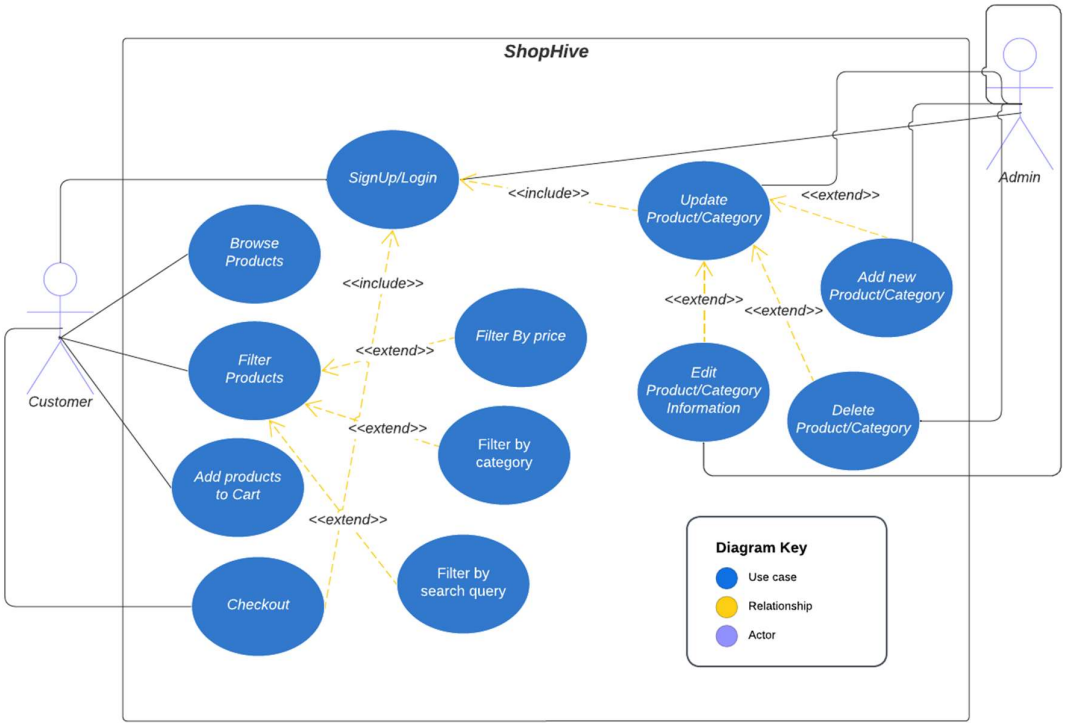


Figure 4: Use Case Diagram

Activity Diagrams:

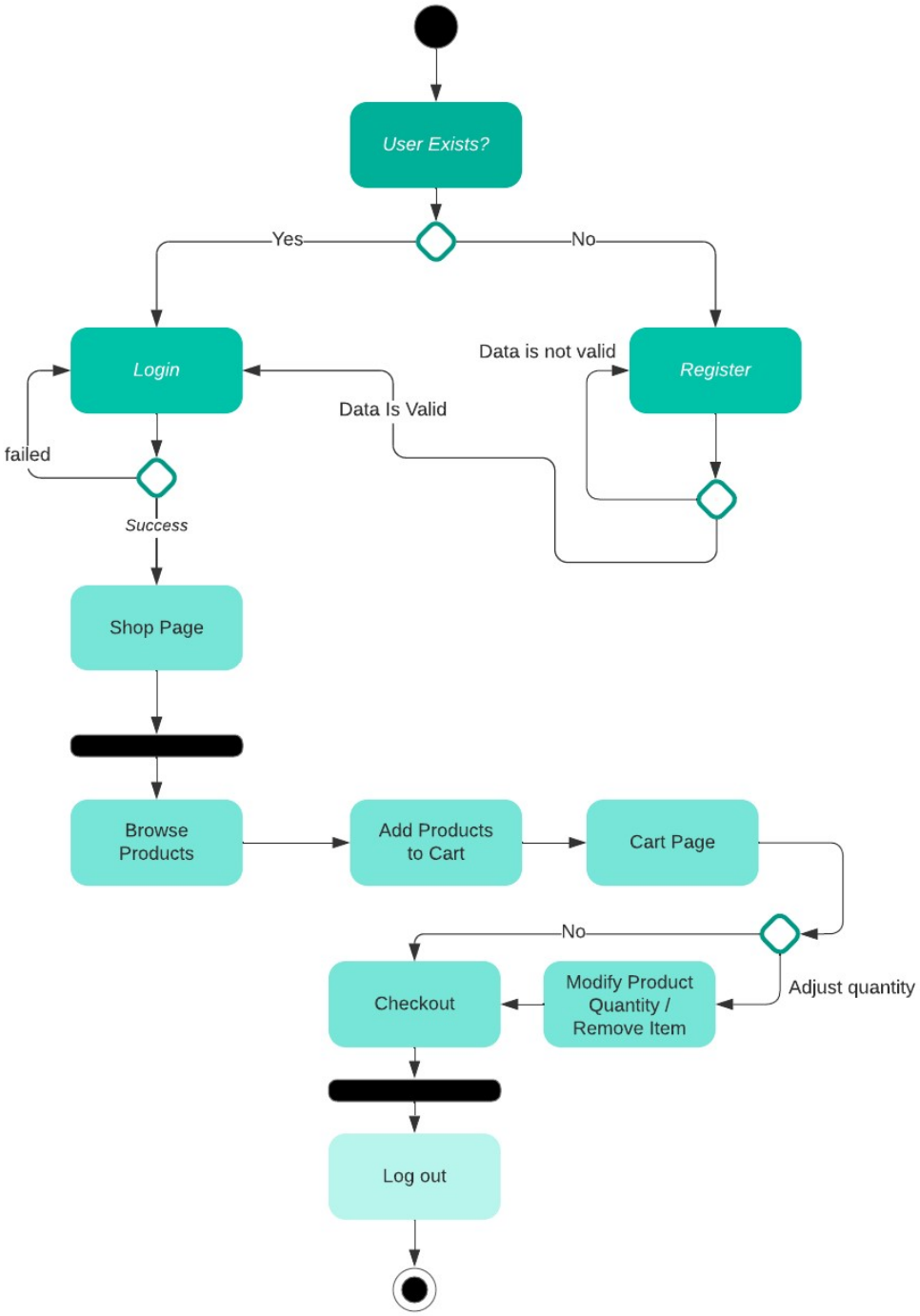


Figure 5: Activity Diagram for Customer checking out a cart

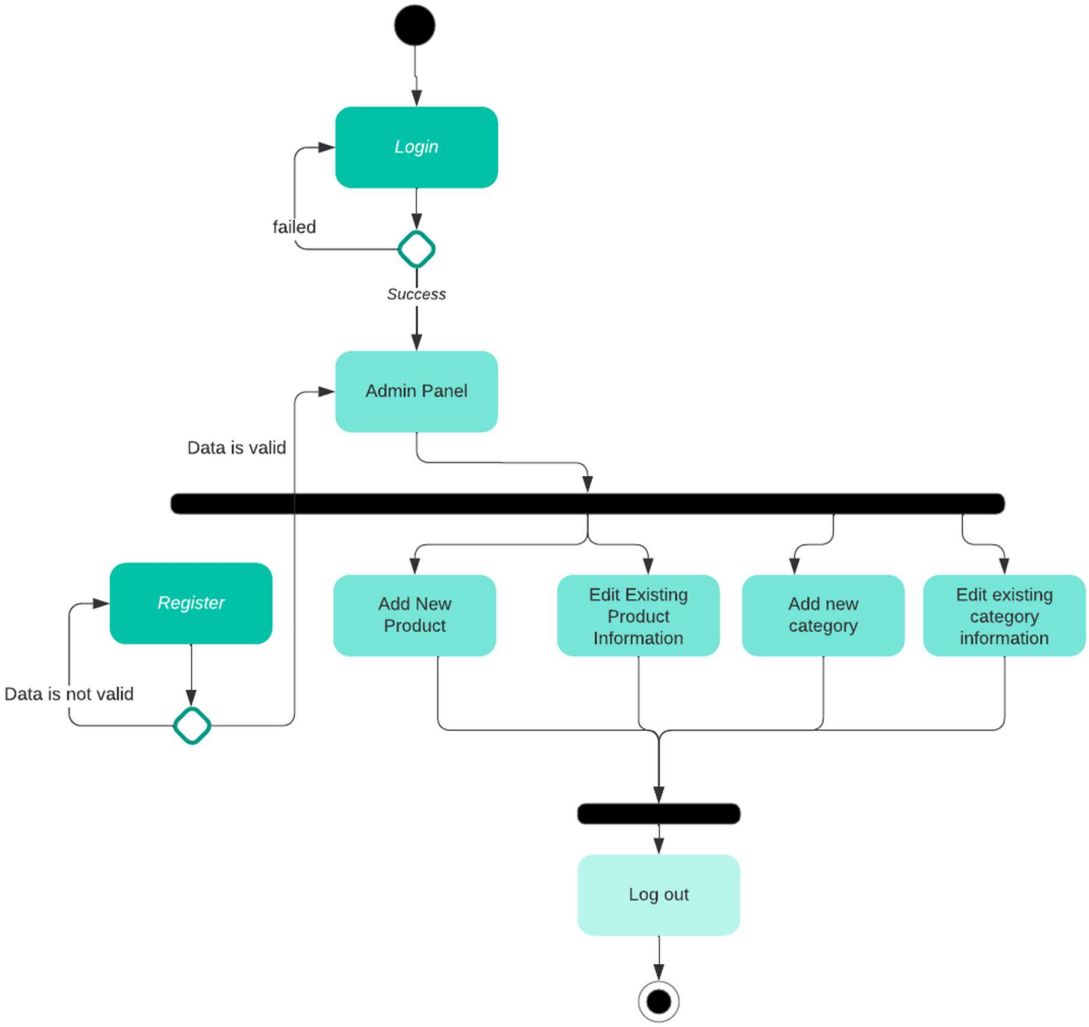


Figure 6: Activity Diagram for Admin Activities

5.4.2 Architecture

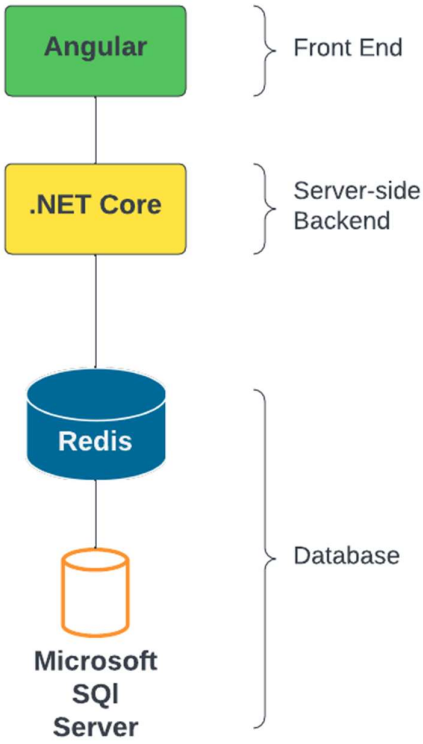


Figure 7: System Architecture

The architecture indicated above has been used in the development of ShopHive. The frontend was built in Angular, the server-side backend in.NET core, and Redis was used to store Cart information locally, with Microsoft SQL Server serving as the primary database.

This design is commonly referred to as a three-tier architecture since it leverages Angular for the GUI layer, .NET core for the application tier, and Redis and Microsoft SQL Server for the database tier.

5.5 Implementation

The backend for ShopHive was developed initially, before proceeding to the front-end framework. The screenshots below shows a glance at how the required features are developed.

Backend Implementation:

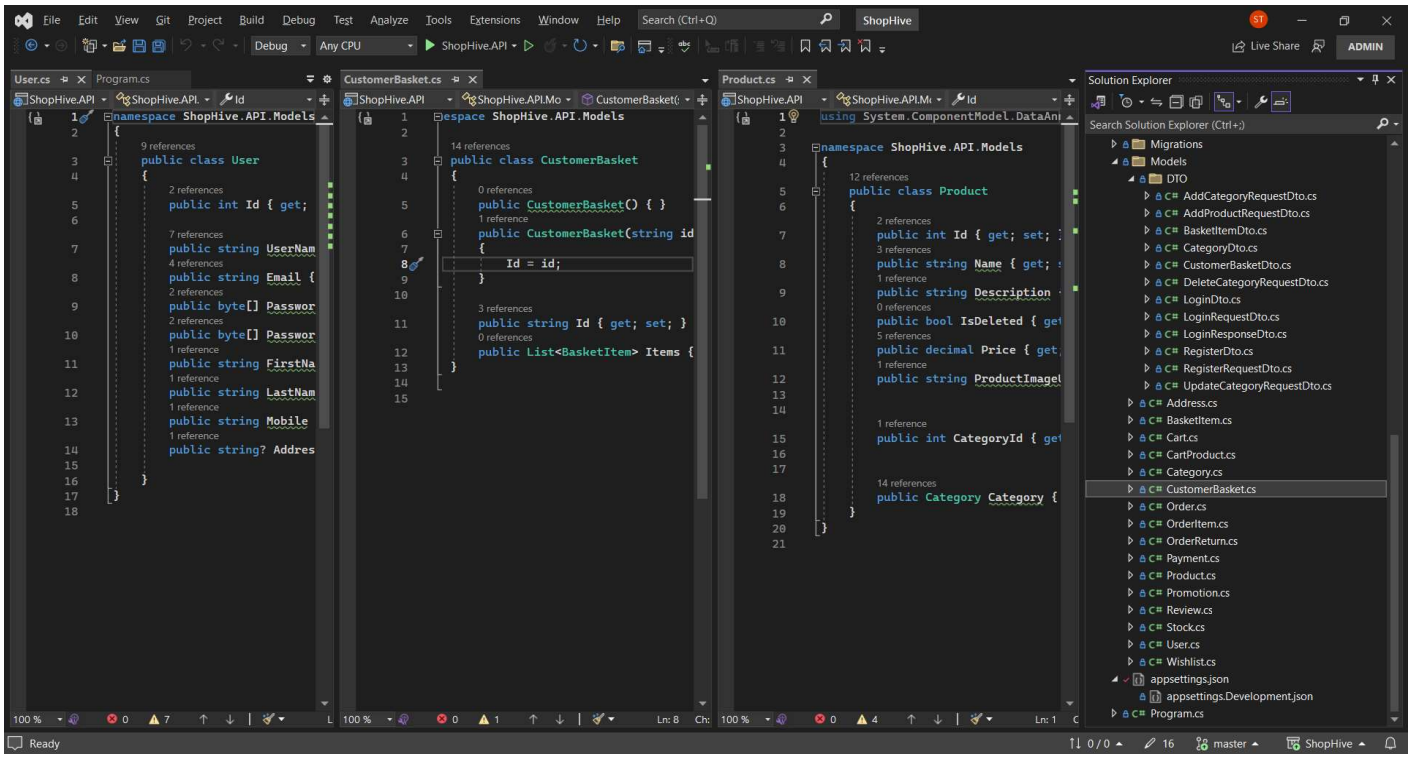
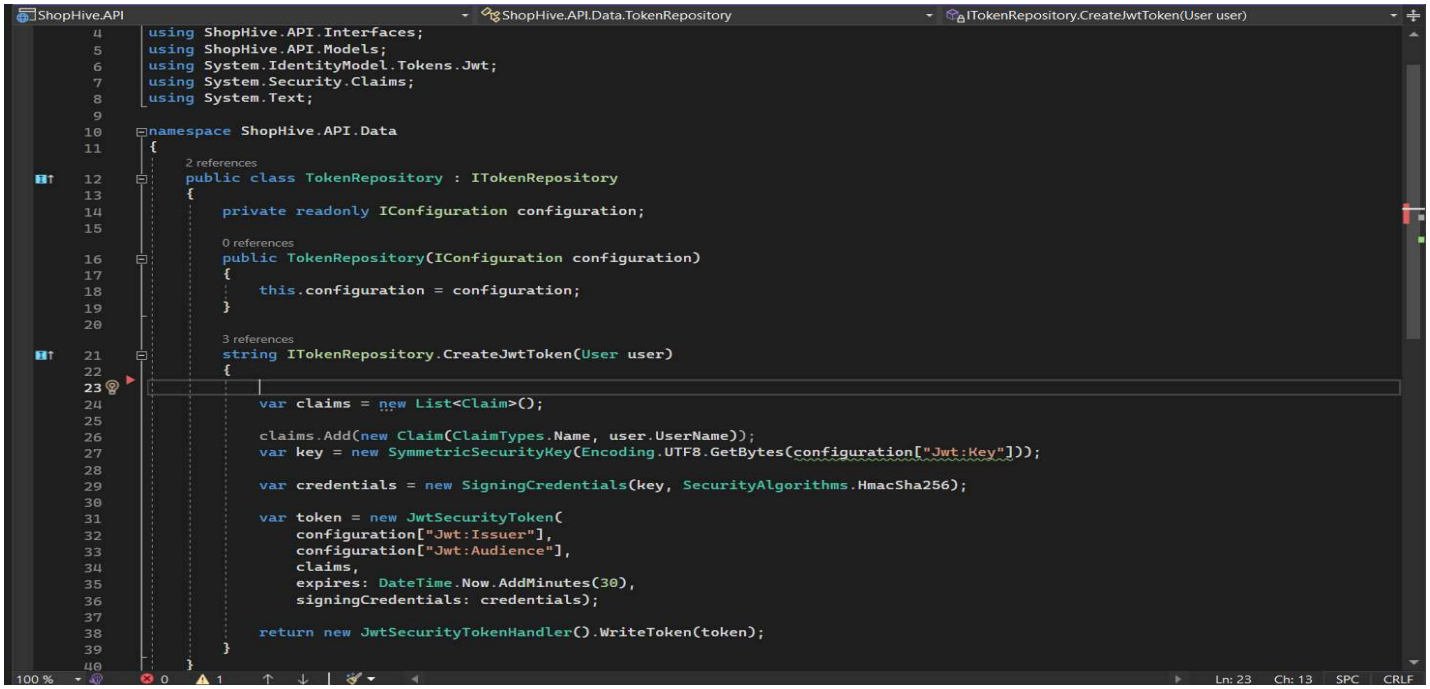


Figure 9: User, CustomerBasket and Product Models



```
using ShopHive.API.Interfaces;
using ShopHive.API.Models;
using System.IdentityModel.Tokens.Jwt;
using System.Security.Claims;
using System.Text;

namespace ShopHive.API.Data
{
    public class TokenRepository : ITokenRepository
    {
        private readonly IConfiguration configuration;

        public TokenRepository(IConfiguration configuration)
        {
            this.configuration = configuration;
        }

        string ITokenRepository.CreateJwtToken(User user)
        {
            var claims = new List<Claim>();

            claims.Add(new Claim(ClaimTypes.Name, user.UserName));
            var key = new SymmetricSecurityKey(Encoding.UTF8.GetBytes(configuration["Jwt:Key"]));

            var credentials = new SigningCredentials(key, SecurityAlgorithms.HmacSha256);

            var token = new JwtSecurityToken(
                configuration["Jwt:Issuer"],
                configuration["Jwt:Audience"],
                claims,
                expires: DateTime.Now.AddMinutes(30),
                signingCredentials: credentials);

            return new JwtSecurityTokenHandler().WriteToken(token);
        }
    }
```

Figure 8: Generating a JWT Token

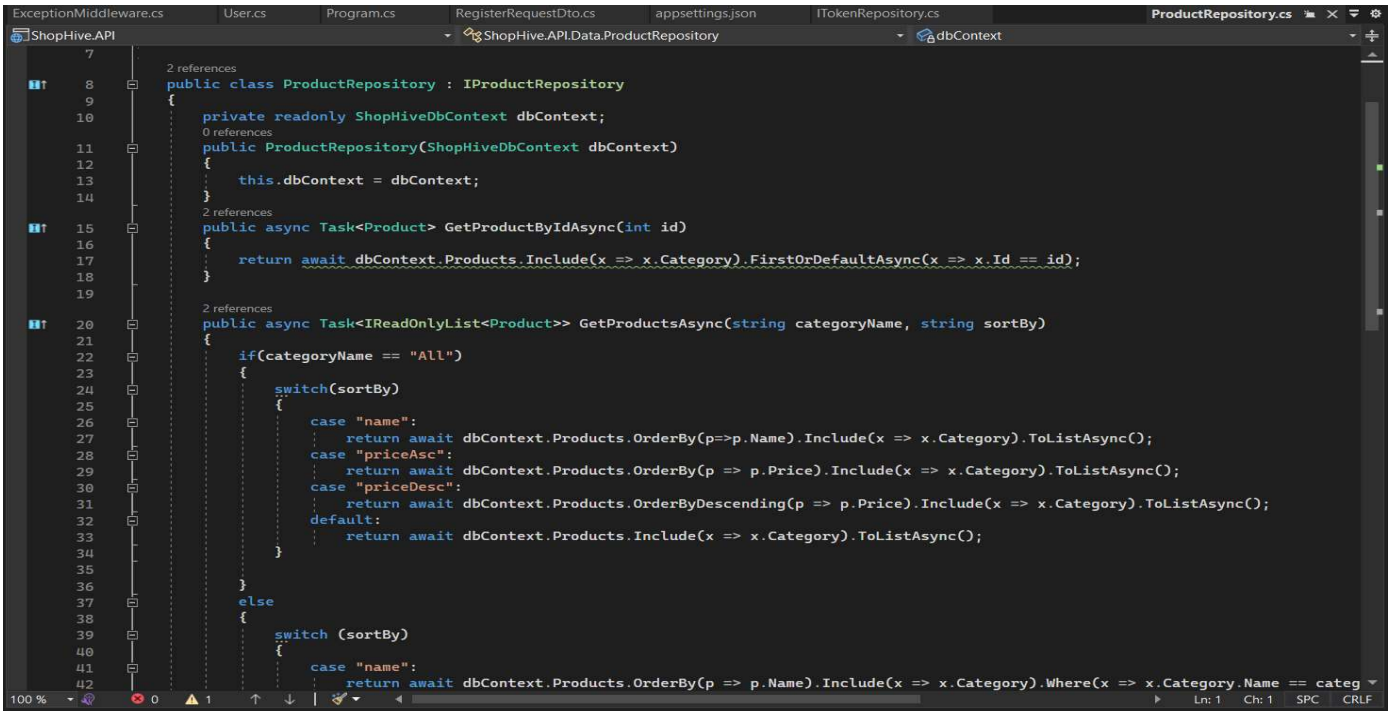


Figure 11: Product Repository Code

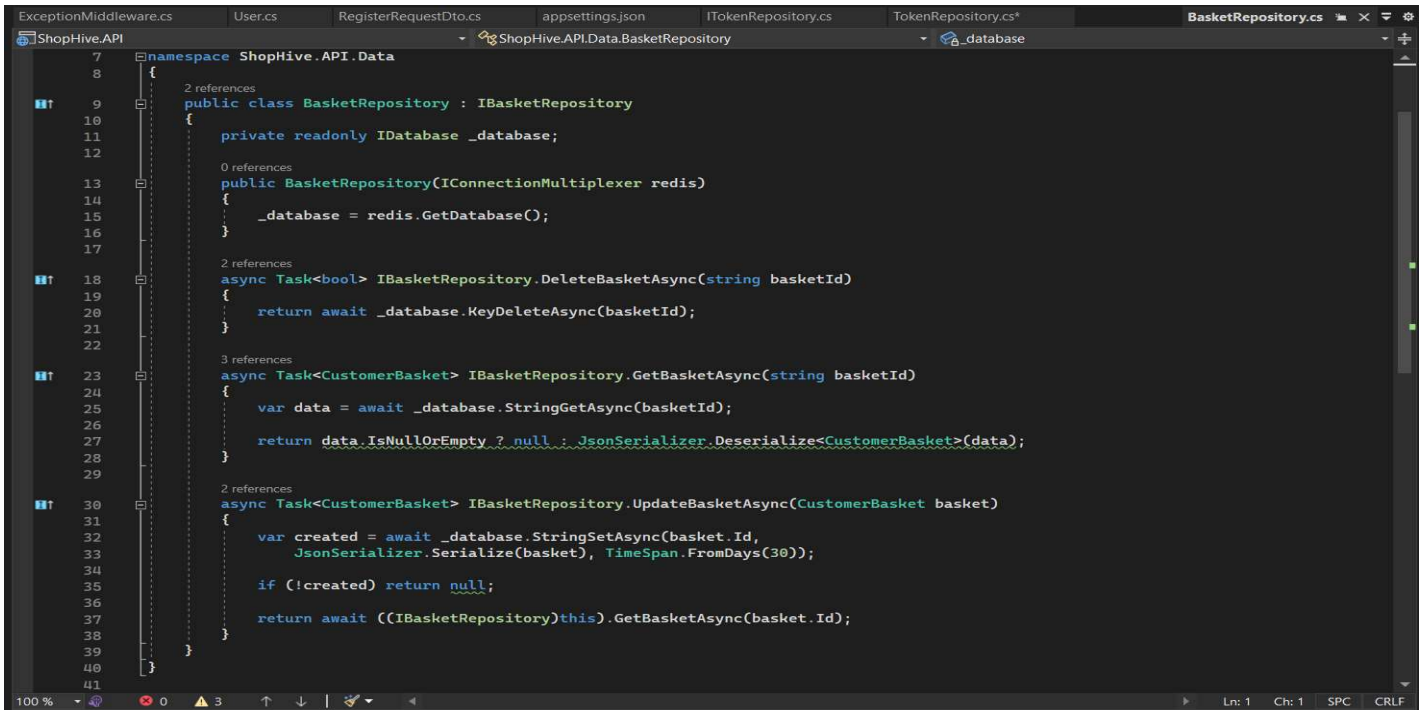


Figure 10: Basket Repository Code

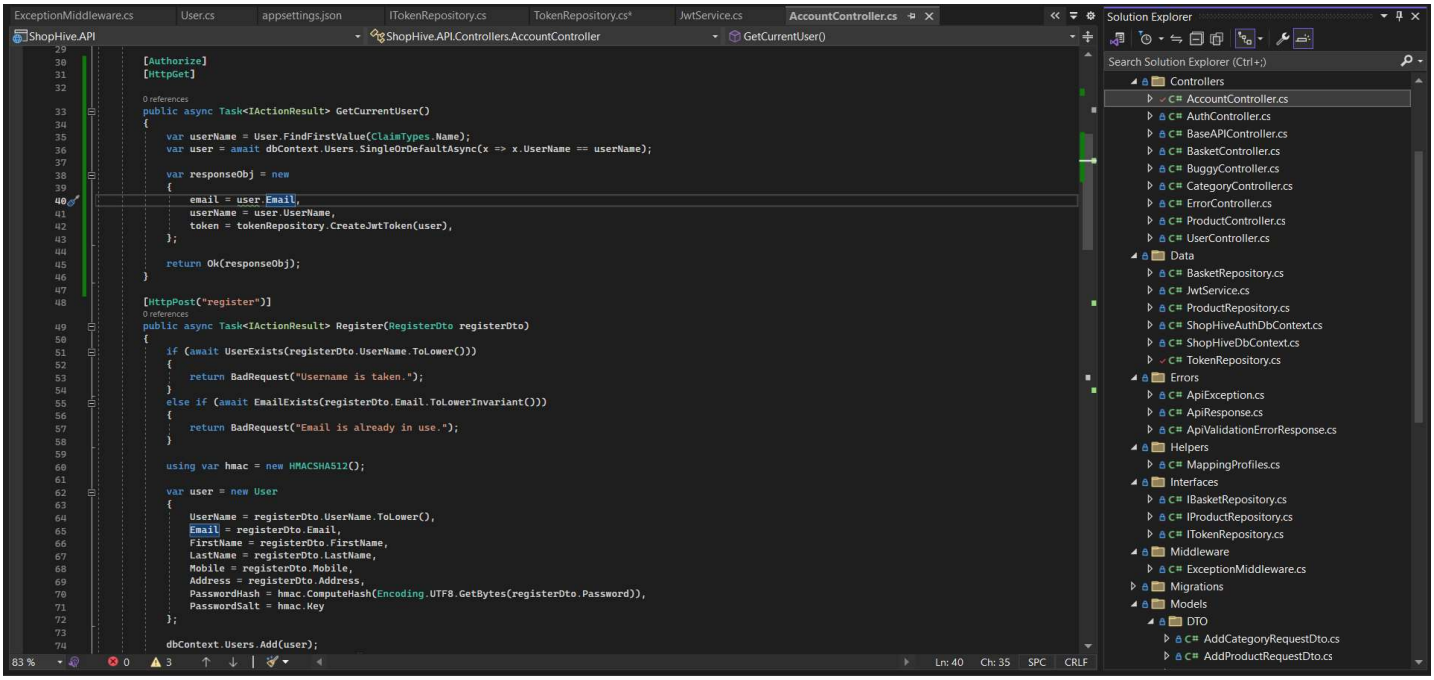


Figure 13: Register User API Controller

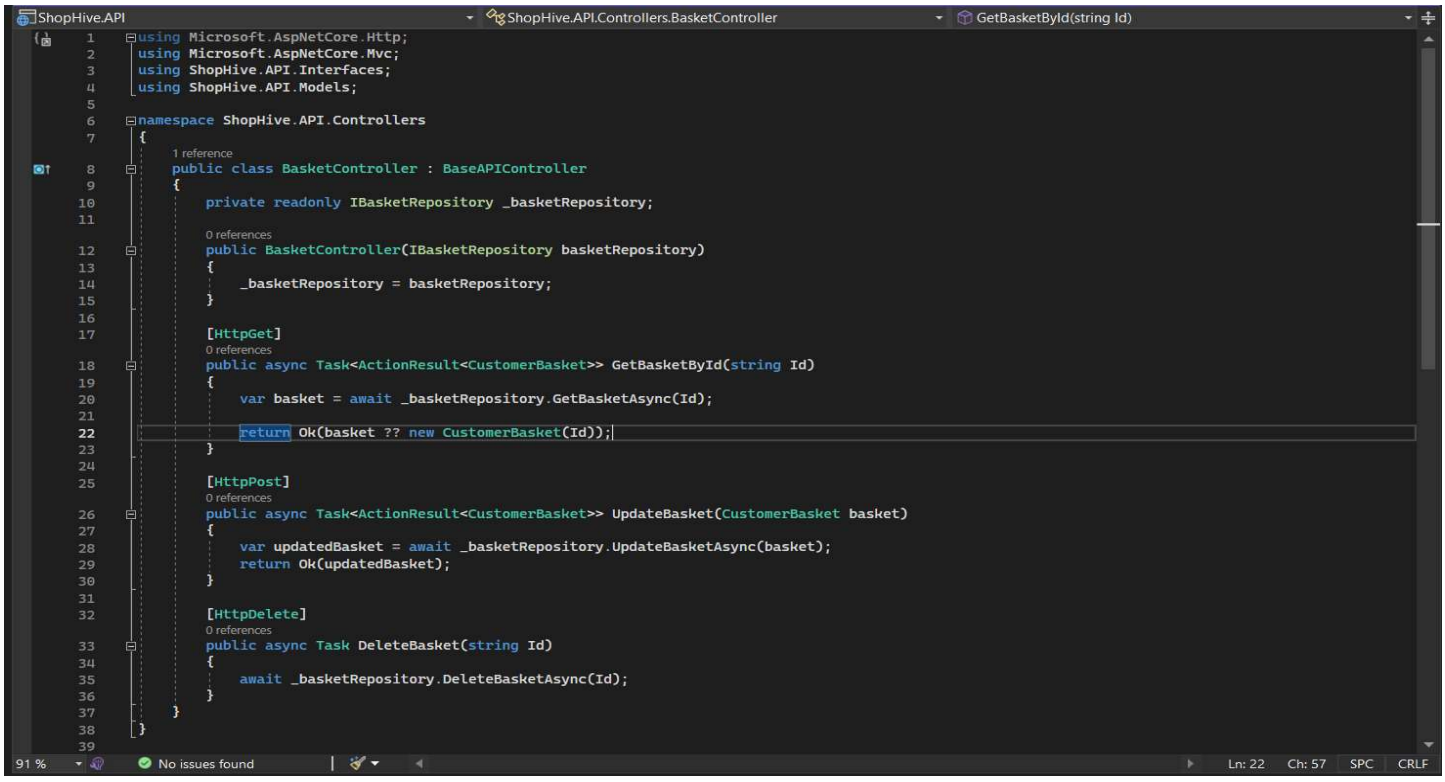


Figure 12: Basket API Controller

Frontend Implementation:

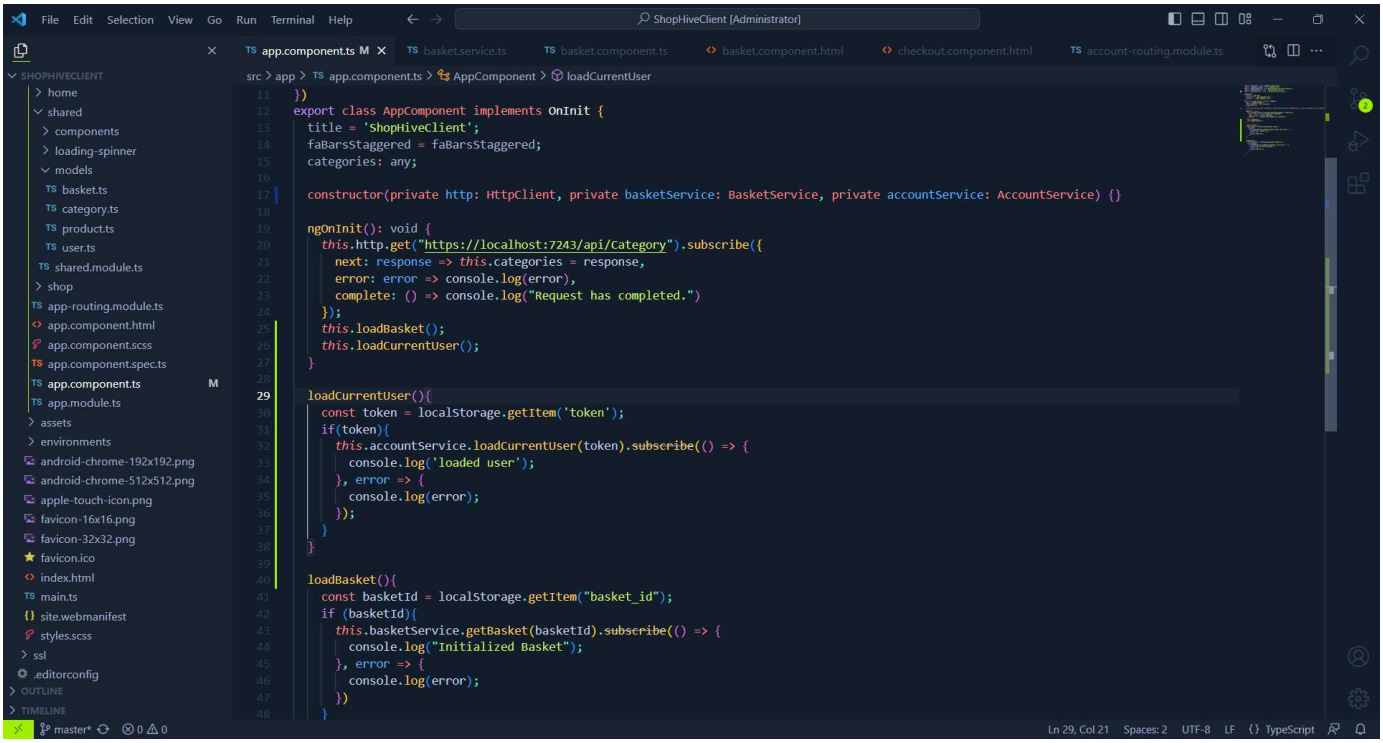


Figure 14: Loading Categories, CurrentUser and Basket initially

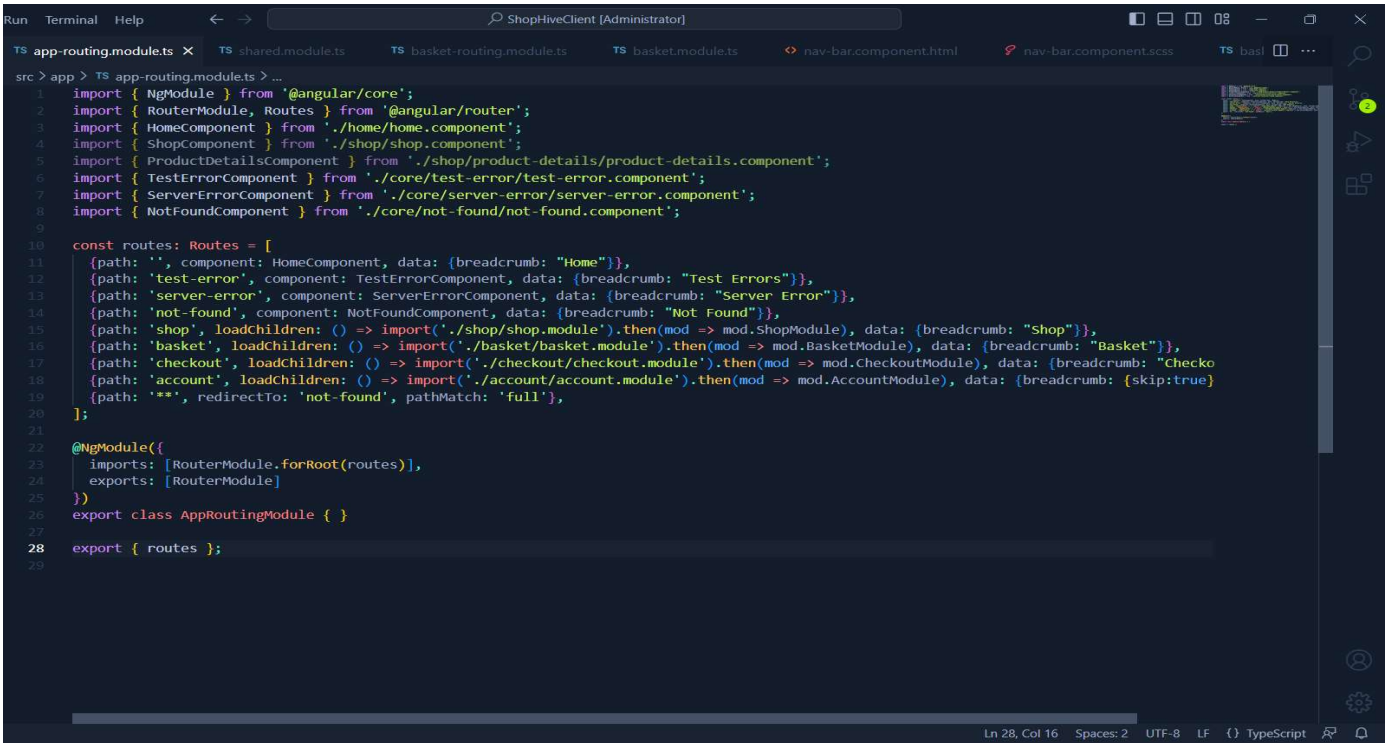


Figure 15: Configuring Application Routes

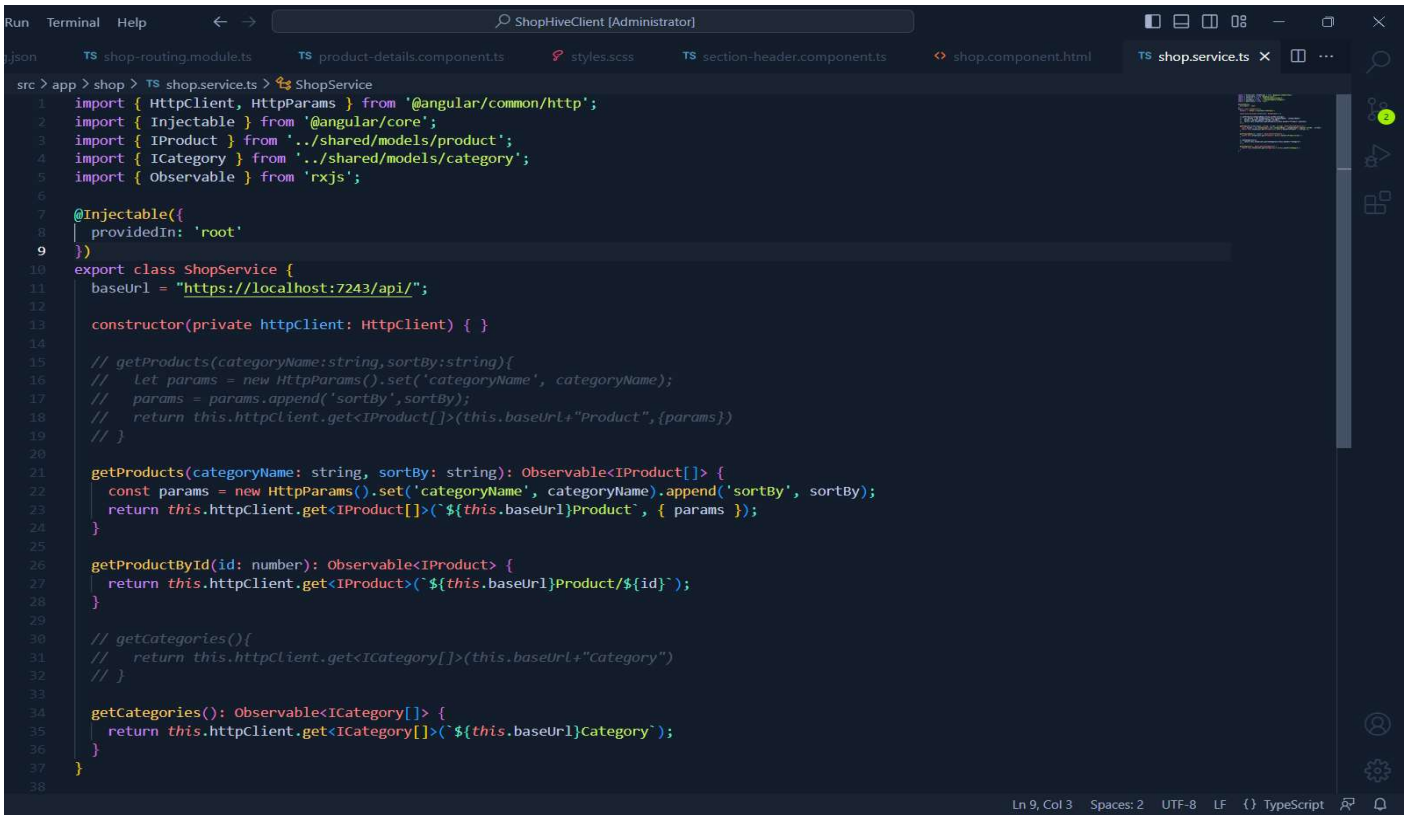


Figure 16: ShopService methods

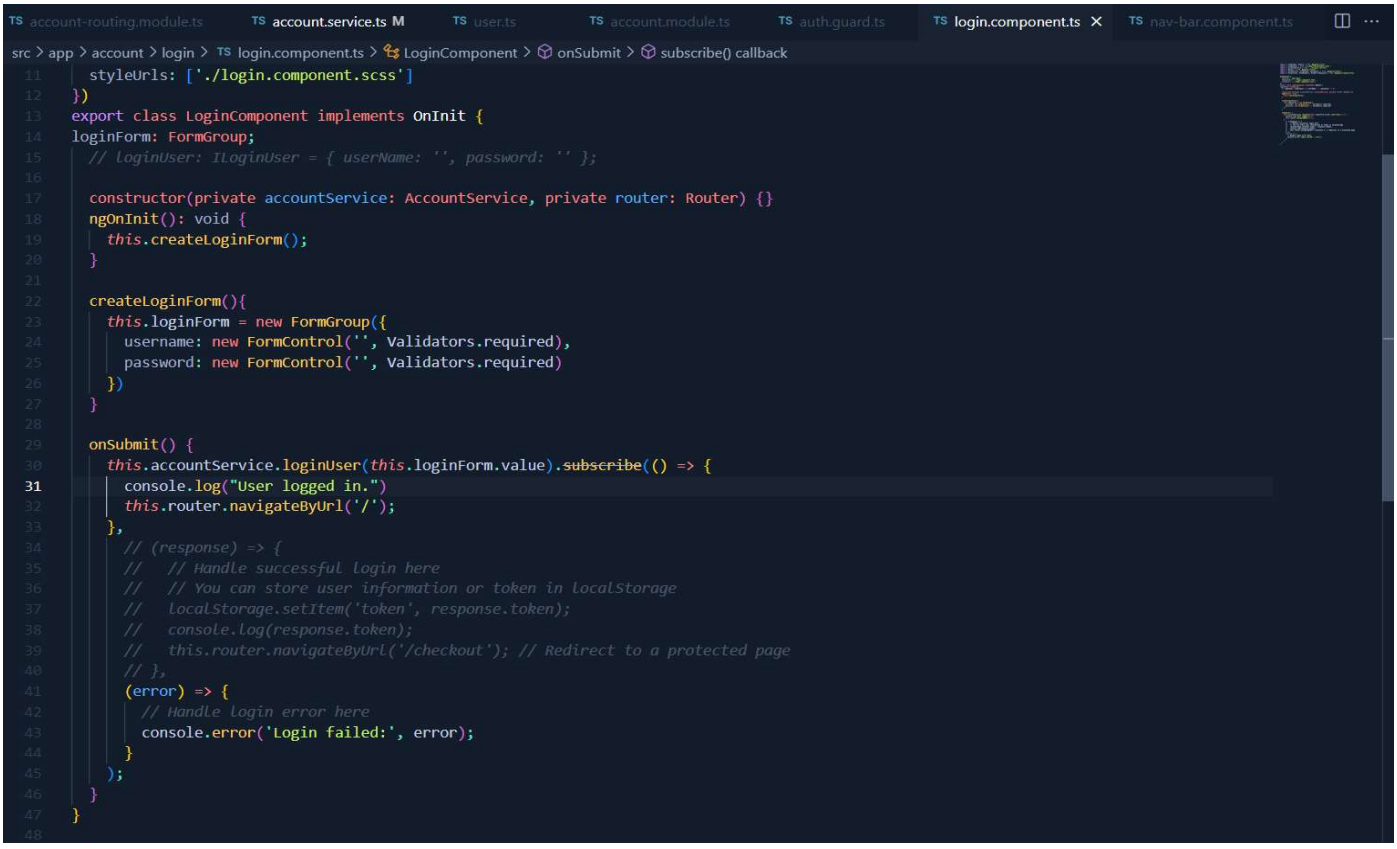


Figure 17: LoginComponent code

5.6 Testing

A crucial phase in development that often gets overlooked is the testing phase. Evaluating an application's limits through a series of inputs and outputs can help developer's solve bugs, and fix unexpected issues. It's also a necessary step to determine if the specified requirements have been met.

5.6.1 Input

Test Case	Description	Input Data
TC001	Register user	Email, password, name
TC002	Login user	Email, password
TC003	Create category	Category name
TC004	Update category	Category id, new name
TC005	Delete category	Category id
TC006	Get categories	None
TC007	Create product	Name, description, price, image, category
TC008	Get product	Product id
TC009	Get products	None
TC010	Add to cart	Product id, quantity
TC011	Remove from cart	Product id
TC012	Get cart items	None

Table 2: Test Input Cases

5.6.2 Output

Test Case	Description	Expected Output
TC001	Register user	Success message
TC002	Login user	JWT token, user info
TC003	Create category	Category id
TC004	Update category	Success message
TC005	Delete category	Success message
TC006	Get categories	List of categories
TC007	Create product	Product id
TC008	Get product	Product info
TC009	Get products	List of products
TC010	Add to cart	Updated cart info
TC011	Remove from cart	Updated cart info
TC012	Get cart items	List of items in cart

Table 3: Test Output Cases

5.6.3 Designing Test Cases

It's crucial to evaluate different aspects of ShopHive to ensure the quality and reliability of the application. With that in mind, the following test cases were designed.

1. Validation Testing

Input validation needs to be tested for boundary conditions and different types of valid and invalid data.

Test Case	Description	Input Data	Expected Output
TC001	Email format	Invalid email	Error message
TC002	Email format	Valid email	Success
TC003	Password length	4 chars	Error message
TC004	Password length	16 chars	Success
TC005	Product price	Negative	Error message
TC006	Product price	Valid positive	Success

Table 4: Validation Testing

2. Authorization Testing

Authentication and authorization mechanisms must be tested extensively.

Test Case	Description	Input	Expected Output
TC007	Login without credentials	Empty	Error message
TC008	Login with invalid credentials	Invalid creds	Error message
TC009	Access API without token	Missing token	401 Unauthorized
TC010	Access API with invalid token	Invalid token	401 Unauthorized

Table 5: Authorization Testing

3. Error Condition Testing

Important error scenarios must be validated.

Test Case	Description	Input	Expected Output
TC011	Delete non-existing category	Invalid ID	404 Not Found
TC012	Get non-existing product	Invalid ID	404 Not Found

Table 6: Error Testing

4. Connection Testing

Test system behavior for connection issues.

Test Case	Description	Input	Expected Output
TC016	API server unavailable	Send request	Timeout error

Table 7: Connection Testing

5.6.4 Test Results

I ran the test scenarios to see if the ShopHive app was functioning properly, and here's what I discovered:

1. **Validation Testing:** The tests to see if the app handles different types of input were successful. When we provided it with inaccurate data, such as an invalid email format or values that were completely out of boundaries, the program did the right thing and rejected them.

2. **User login and Authorization Testing:** I also investigated how the app handles user login and authorization. I attempted to bypass the login and access portions of the app that I wasn't meant to, but the program refused to let me in. It appears to be quite powerful in this location.

3. **Testing for Error Conditions:** I also tried several error circumstances, such as attempting to remove data that did not exist. The software handled these circumstances admirably, returning error codes such as 404 NotFound, as it should.

4. **Connection Testing:** I even disabled the section of the program that communicates with the server, and the app didn't just sit there indefinitely. It ultimately timed out. This implies that it can detect when there is an issue with the connection.

Overall, the testing went about as well as we expected. During this round of testing, I discovered no major issues with the app. There were a few minor difficulties with how items appeared on the screen, but I've noted them to address in the future. I also intend to add more tests to ensure that the software continues to function properly.

Chapter 6

Results & Analysis

This internship's goal was to expand my knowledge and skills in designing a program utilizing ASP.NET and Angular. The objective was to develop a modern online application that prioritized security and performance.

I learned a lot while working on this project, and working on the front end and back end using Angular and ASP.NET both presented their own set of challenges. Google owns Angular, a well-known web application framework built on typescript. It works by generating single page applications (SPA), and as a result, the majority of my e-commerce website is constructed using components.

ASP.NET, on the other hand, was built by Microsoft and is another cross-platform web framework like Angular. I built the backend with ASP.NET, which handles API calls and works with the database, and the front end with Angular.

Because ASP.NET saw some substantial changes in recent upgrades, I encountered several challenges working with obsolete code, as well as troubleshooting and correcting them. The database queries were effortlessly handled, due to ASP.NET's built-in package Entity Framework Core, which builds tables and handles API requests without requiring developers to type a single SQL query.

Working with Angular presented me with substantially more challenges. It should be highlighted that Angular is updated every six months, which is excessive given the changes that typically follow releases. As a result, learning from earlier Angular projects was challenging because they frequently contained obsolete code. I eventually figured it out.

Because I was in charge of both the front end and back-end development, I was continuously switching back and forth between the two projects to ensure a smooth integration and collaboration between the two apps. Once the essential components had been evaluated and proved to perform successfully under various scenarios and test cases, the code was streamlined and refactored.

The list below displays what I have accomplished since the start of the project.

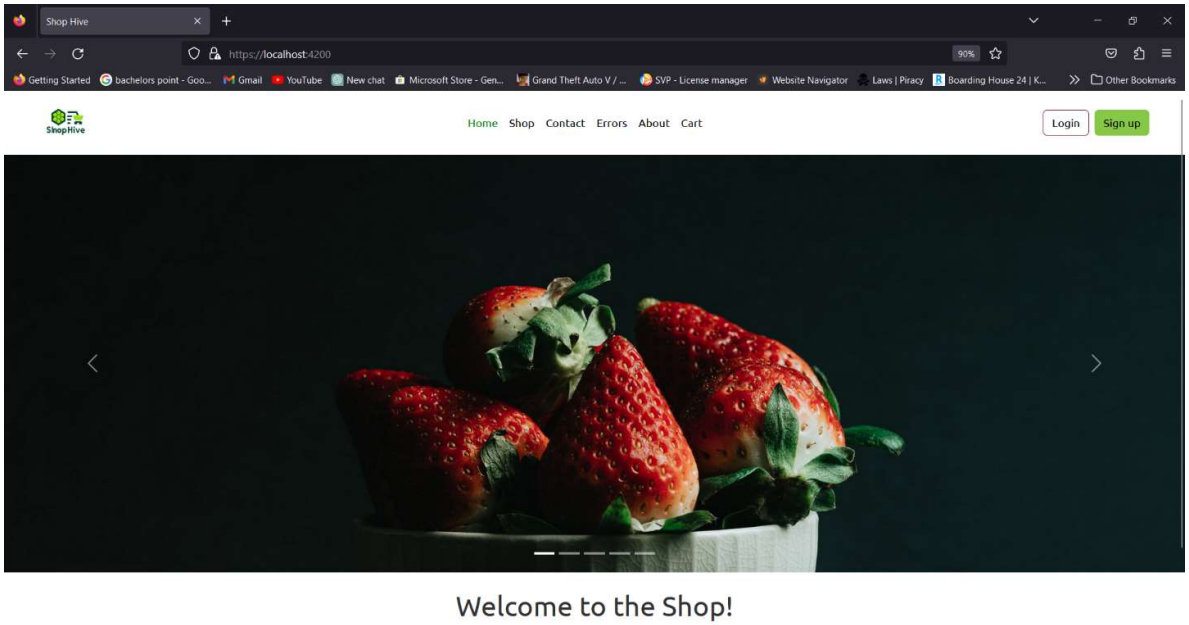


Figure 18: Home Page

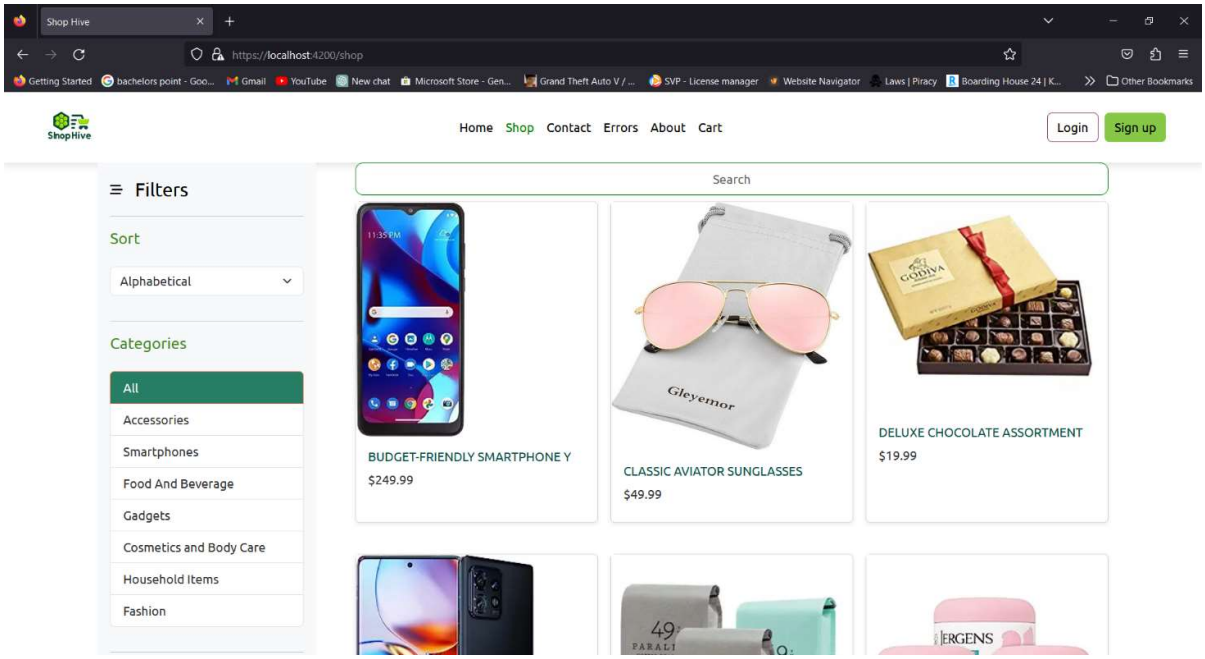


Figure 19: Shop Page

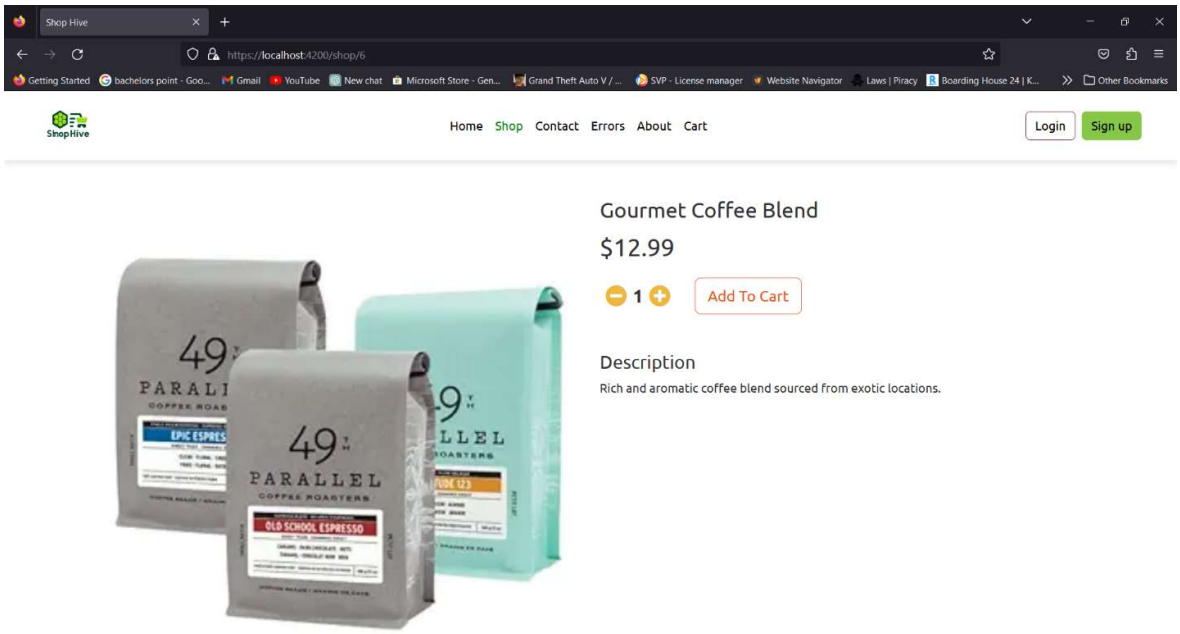


Figure 20: Product Detail Page

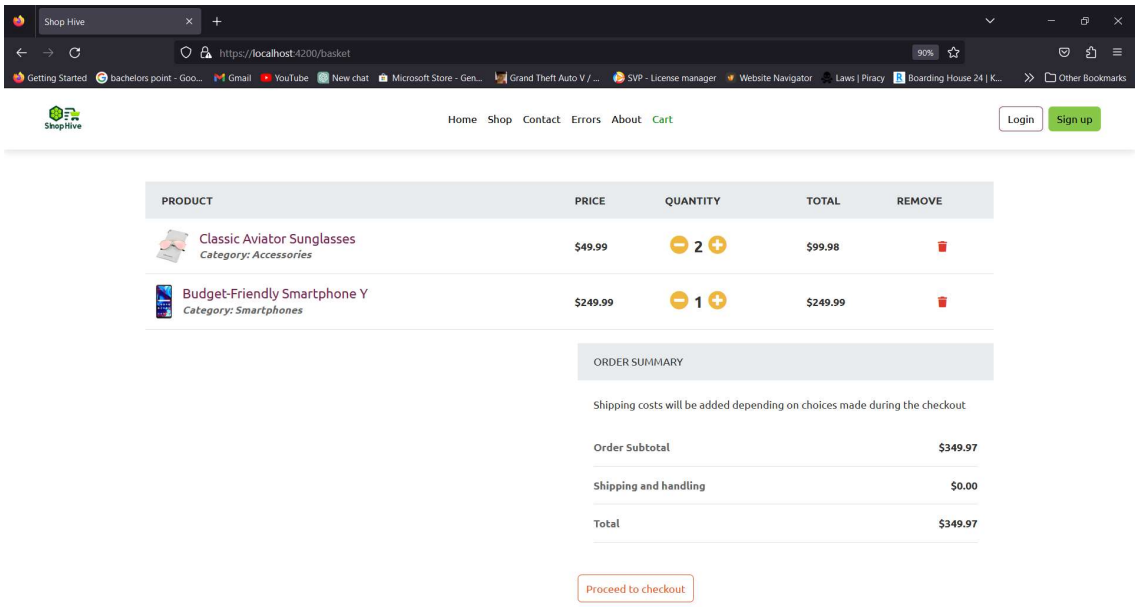


Figure 21: Cart Page

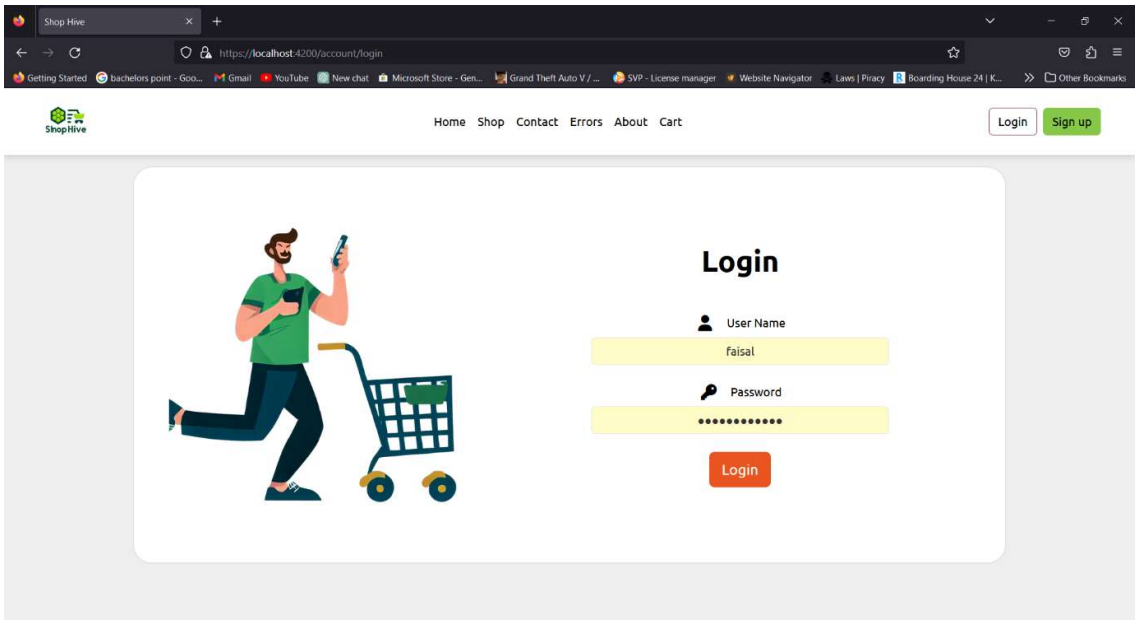


Figure 22: Login Page

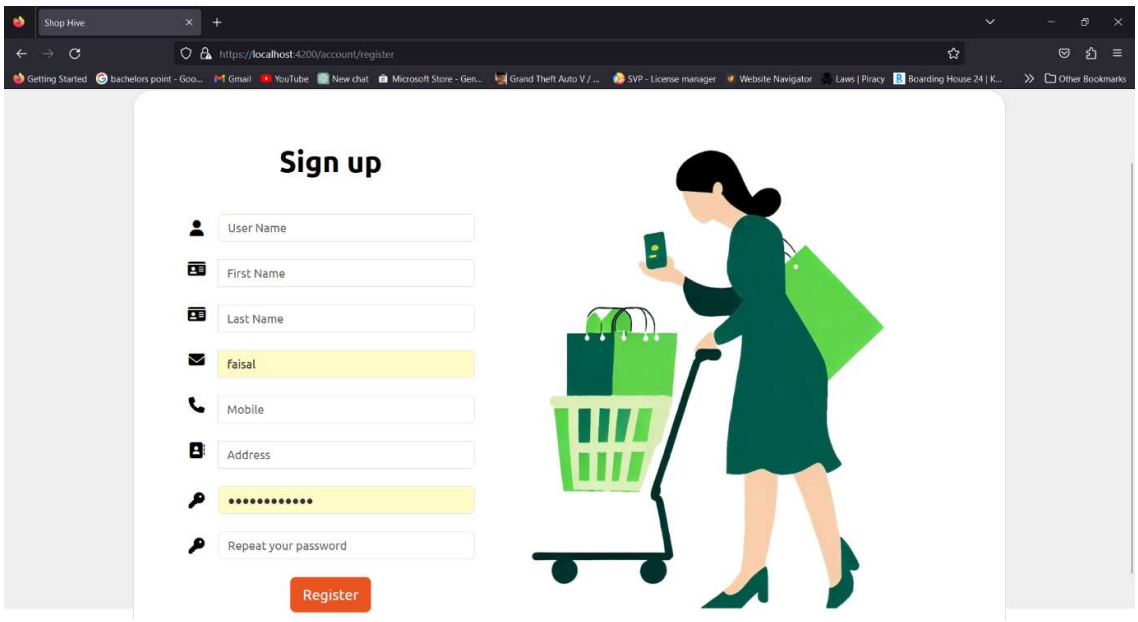


Figure 23: Sign Up Page

Chapter 7

Project as Engineering Problem Analysis

7.1 Sustainability of the Project/Work

During my internship at Nerd Castle Limited, I spent a large amount of time and effort constructing an e-commerce website. While the project is still in its early stages and is hosted locally on my laptop, it is critical to consider its long-term viability from a variety of angles:

Technical Durability:

Scalability: The ASP.NET backend of the e-commerce website uses Microsoft SQL Server for storage of data. This technological stack is strong and widely used throughout the industry, making it easy to scale as the project proceeds. If the necessity arises, the website can be simply deployed to production servers.

Frontend Framework: The frontend is developed with Angular and Node.js, both of which are well-known for their flexibility and long-term support. These frameworks have strong communities, ensuring that updates, security patches, and enhancements are always available.

Code Maintainability: Code maintainability and best practices were prioritized throughout the development process. Proper documentation, version control with Git, and coding standards were followed, enabling future developers' work on and extension of the project easy.

Data Administration and Security:

Data Backup and Recovery: To store critical data, a Microsoft SQL Server database is utilized. Regular backup and recovery methods are critical for data integrity and company continuity.

Security Measures: The project incorporates industry-standard security methods such as encryption, authentication, and authorization to ensure data security. To address evolving risks, regular security audits and updates will be required.

Continuity of Operations:

Collaboration and Support: Nerd Castle Limited may consider future collaboration with myself or other resources to continue the development and eventual implementation of the project. This might involve more internships or employing developers who are knowledgeable about the technology being used.

Roadmap and Future Enhancements for Long-Term Vision: To ensure the project's long-term viability, a roadmap outlining future additions and features is

recommended. This will help the project retain its relevance and competitiveness in the ever-changing e-commerce industry.

Considerations for Cost:

Costs of Infrastructure: Hosting and maintenance charges will be incurred if the website is deployed to production servers. Nerd Castle Limited should assess these expenses and plan accordingly.

To summarize, while the e-commerce website project is hosted locally, it has a high potential for sustainability and expansion. Nerd Castle Limited can continue to build on the foundation created during this internship by following to best practices, maintaining data security, documenting processes, and considering long-term vision and expenses. This will eventually lead to a profitable and sustainable e-commerce platform.

7.2 Social and Environmental Effects and Analysis

Despite the fact that the project was not deployed in a production environment during my internship, it is crucial to examine its possible social and environmental impacts.

Social Implications:

Usability and Diversity: It is essential to create websites that are accessible to people with disabilities. While this was not specifically taught during my internship, it is an important factor in web development. Nerd Castle Limited may look into accessibility features further while offering the website in order to make it more accessible to a wider audience.

User Privacy and Data Security: Because the website involves transactions and user data, maintaining the highest levels of user privacy and data security is vital. In today's digital world, ensuring that user data is protected and handled properly is a societal responsibility. To protect user information, comprehensive data security procedures should be established throughout the production deployment phase.

Environmental Implications:

Hosting and Energy Efficiency: While the project was developed locally on a laptop, the website will require energy for hosting and server maintenance when it is launched to production servers. Hosting choices that prioritize energy efficiency and environmental sustainability should be considered by Nerd Castle Limited. Green hosting companies and server optimization techniques might help to lower the carbon footprint of a website.

Sustainable Practices: Nerd Castle Limited should consider adopting sustainable practices in its IT operations in the long run. This could involve using renewable energy, reducing electronic waste, and following eco-friendly hardware procurement practices.

7.3 Addressing Ethics and Ethical Issues

Ethical difficulties in software development and e-commerce can affect both the organization and its users, therefore they must be addressed carefully.

User Privacy and Data Handling:

The handling of user data is one of the most important ethical considerations in web development. It was critical to ensure that user information, including personal and financial data, was managed with the highest care and integrity during the creation of the e-commerce website. The following efforts were taken to address this concern:

Data Encryption: To secure sensitive user information during transmission and storage, the website used strong data encryption algorithms. Nerd Castle Limited committed to developing and prominently displaying privacy rules outlining how user data would be collected, used, and protected. These policies were created to be clear and easy to understand. Ethical considerations also included creating data retention and deletion procedures to guarantee that user data was not maintained for longer than necessary.

Transparency and Honesty: It is an ethical imperative to maintain transparency and honesty with users. During my internship, we made certain that:

Clear Product Information: All product information, including pricing, specifications, and availability, was displayed on the website in a clear and accurate manner. Changes to the terms of service, privacy policies, or product offerings needs to be communicated to users in a clear and timely manner.

Fair Business procedures:

Ethical behavior includes fair business procedures, thus Nerd Castle should take steps to:

Avoid Deceptive Marketing and Sales tactics: Nerd Castle Limited has vowed to avoid deceptive marketing and sales tactics that may mislead users.

Customer service: It was deemed necessary to provide responsive and helpful customer service in order to answer user issues and concerns ethically.

Future Considerations:

As the e-commerce website nears completion, Nerd Castle Limited must maintain a proactive position on ethical issues. This includes conducting frequent ethical reviews, following ethical coding and business standards, and remaining up to date on new ethical considerations in e-commerce and online development.

Chapter 8

Lesson Learned

8.1 Problems Faced During this Period

My internship at Nerd Castle Ltd was amazing and gratifying. Despite the short time frame of three months, the experience was unique and one that I will never forget.

During my internship, the most difficult challenge was running many projects and IDEs in my laptop at the same time. I just have one laptop that is capable of performing most high-performance jobs pretty effectively. But, operating Visual Studio 2022 and Visual Studio Code at the same time, as well as maintaining two servers operating, proved difficult due to my slow processor. Although my office offered computers for work use, they were far inferior to mine and useless.

Aside from that, I had problems configuring the Redis database for storing Cart data with Docker. I had to learn these on my own because they were not clearly taught at my university or company.

There were also other projects at Nerd Castle Ltd that I had to oversee at the same time, which added to my mental stress. It became difficult for me to choose which project to work on, especially since the timeframes were not clearly stated.

8.2 Solution of those Problems

Fortunately, I discovered some effective solutions for the issues I encountered.

Because upgrading my laptop was not an option because of financial limitations, I researched techniques to optimize my laptop for increased speed. I opted to install an efficient and lightweight operating system. I also updated my RAM and installed an SSD. These enabled me to fully utilize the capabilities of my laptop, and I no longer encountered any issues at work.

After seeking assistance on online forums, I was able to resolve the Redis problem. The community was quite helpful, and any other little issues were readily resolved with the use of several chatbots such as Google Bard or ChatGPT.

Concerning the final issue, I discussed it with my supervisor at work, and we were able to set some deadlines and make an outline so that I could focus on one project at a time and present them all when the time came.

Chapter 9

Future Work & Conclusion

9.1 Future Works

The ShopHive project's future is entirely influenced on Nerd Castle Ltd's decisions. However, one activity that must be completed shortly is to deploy the project to the web server. This demands some expense planning, so it is entirely dependent on the company.

In addition, a UI change may be required. If users have any extra requirements, those will need to be added as well, albeit this will take some time.

Nerd Castle Ltd has opted not to hire any interns because all of the relevant roles are already filled, however they have offered to notify us if any vacancies become available. After my semester is through, I intend to apply to some companies via LinkedIn and BdJobs. I also just attended the BDJobs Job Fair 2023, where I handed out my resume to 10 different companies. I'm hoping to hear back from them.

9.2 Conclusion

To summarize, throughout my Internship at Nerd Castle Limited, I overcame some hurdles and got accustomed with a real-life work atmosphere. Speaking with active full stack developers was eye-opening, and it was fascinating to hear about their experiences and suggestions. In the future, I hope to advance as a developer by working on more important and diversified projects.

Bibliography

[1] System Development Life Cycle: PPT - System Development Life Cycle (SDLC) PowerPoint ... - Once upon a time, software development consisted of a programmer writing code to solve a problem or automate a procedure. - kaliacon

[2] What Is Agile Methodology? (A Beginner's Guide) [2023] • Asana

[3] Software Engineering | System Design Strategy - GeeksforGeeks

[4] Functional vs. nonfunctional requirements in software engineering | TechTarget

References

[1] https://www.projectmanager.com/work-breakdown-structure

[2] https://www.projectmanager.com/gantt-chart

[3] https://www.projectengineer.net/the-6-steps-of-resource-allocation/ [4] https://www.guru99.com/agile-scrum-extreme-testing.html

[5] https://www.w3computing.com/systemsanalysis/

[6] https://www.simplilearn.com/feasibility-study-article

[7] https://www.betterevaluation.org/en/evaluation-options/richpictures [8] https://www.smartdraw.com/uml-diagram/

[9] https://www.altexsoft.com/blog/non-functional-requirements



An Undergraduate Internship/Project on Topic

Shop Hive E-Commerce System

By

Faisal Bhuiyan

Student ID: **1921298**

Summer, 2023

Consent from Supervisor

The student modified the internship final report as per the recommendation made by his/her academic supervisor and/panel members during and/or before final viva, and the department can use this version for archiving as well as the OBE course material for CSE499.

This internship report is checked with Turnitin and/or Ithenticate plagiarism checker, and the score is:

Turnitin Score (%) : 5%

Ithenticate Score (%) :

(Signature of the Supervisor)

Md. Mahmudul Kabir Peyal
Research and Development Officer
Department of Computer Science & Engineering
Independent University, Bangladesh