

Online Shopping Management System

by

Maliha Akter Farjana

ID: CSE2103024056

Supervised by

Asif Ahmed

Submitted in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)**

May 2026

APPROVAL

The project titled “Online Shopping Management System” submitted by Maliha Akter Farjana (CSE2103024056) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Asif Ahmed

Lecturer,
Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name and Signature)

Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)

Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)

Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Asif Ahmed, Lecturer**, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this Project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

Signature

(Asif Ahmed)
Supervisor

Maliha Akter Farjana
ID: CSE2103024056

ABSTRACT

Online Shopping Management System is the process of doing business through computer networks where a person sitting in front of a computer can access all the facilities of the Internet to buy or sell products. Unlike traditional commerce which requires physical effort to go and get products, e-commerce has made it easier for people to reduce physical work and save time. Online shopping which started in the early 1990s has taken a great leap in the world of computers however, the major factor hindering its growth is security. Security remains the primary challenge facing online commerce today and significant advancements are still being made in this field. The main advantage of e-commerce over traditional commerce is that users can browse online shops, compare prices, and order merchandise from home using their PC. To increase the use of online shopping in developing countries, B2B e-commerce is implemented to improve access to global markets in the field of commerce. However advancements in e-commerce are essential for the growth of developing countries. Research strategies highlight the importance of e-commerce in developing countries for business applications. Electronic commerce or e-commerce refers to any type of business or commercial transaction that involves the transfer of information across the Internet. It covers a wide range of businesses from consumer-based retail sites to auction or music platforms, as well as business exchanges trading goods and services between corporations. It is currently one of the most important aspects of the Internet to emerge.

.

ACKNOWLEDGMENT

First and foremost, we would like to express our deepest gratitude to Asif Ahmed, Lecturer, Department of Computer Science and Engineering Sonargaon University for his invaluable guidance support and encouragement throughout the development of this project. His insights and expertise were instrumental in shaping the direction and success of this ASP.NET Core MVC-based web application.

We also extend our heartfelt thanks to the faculty members of Sonargaon University for providing the knowledge and foundation necessary for this work. Their dedication to teaching and mentorship has been a constant source of inspiration.

Special thanks to our friends for their constant support motivation and understanding during the entire development process.

Finally, we would like to acknowledge all the online communities forums and documentation contributors whose shared knowledge played a significant role in helping us troubleshoot learn and improve throughout the journey of building this ASP.NET Core MVC-based web application.

LIST OF ABBREVIATIONS

API	Application Programming Interface
ASP	Active Server Pages
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DBMS	Database Management System
EF	Entity Framework
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IIS	Internet Information Services
JS	JavaScript
JSON	JavaScript Object Notation
MVC	Model View Controller
OOP	Object-Oriented Programming
SQL	Structured Query Language
UI	User Interface

TABLE OF CONTENTS

Title	Page No.
DECLARATION	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	v
LIST OF ABBREVIATIONS	vi
CHAPTER 1	1 – 3
INTRODUCTION TO ONLINE SHOPPING SYSTEM	
1.1 Introduction	1
1.2 Motivation	1
1.3 Problem Statement	2
1.4 Objectives of the Project	2
1.4.1 General Objective	2
1.4.2 Specific Objectives	2
1.5 Scope of the Study	2
1.6 Significance of the Study	2
1.7 Methodology Overview	3
1.8 Project Book Organization	3
CHAPTER 2	4 – 6
LITERATURE REVIEW AND RELATED WORK	
2.1 Overview of E-Commerce Systems	4
2.2 Background of Study	4
2.3 Definition of E-commerce	4
2.4 History of Ecommerce	4 – 5
2.5 Different types of Ecommerce	5
2.6 Online Shopping Trends	5
2.7 Existing Messaging Systems in Web Applications	5
2.8 Importance of Communication Between Users	6
2.9 Theoretical Framework	6
CHAPTER 3	7 – 10
SYSTEM ANALYSIS AND DESIGN	

3.1	Requirements Analysis	7
3.1.1	Functional Requirements	7
3.1.2	Non-Functional Requirements	8
3.1.3	User Requirement Specification	8
3.2	Use Case Analysis	8
3.2.1	Use Case Diagrams	9
3.2.2	Use Case Descriptions	9
3.3	Class Diagram	9 – 10
CHAPTER 4		11 – 12
SOFTWARE PROCESS		
4.1	Introduction to Software Process	11
4.2	Agile Model	11 – 12
4.3	Data Structures and Algorithm Used	12
CHAPTER 5		13 – 17
SYSTEM ARCHITECTURE AND DESIGN		
5.1	System Architecture	13
5.2	Database Design	14
5.2.1	Table and Relationship	14
5.2.2	Normalization Process	14
5.2.3	SQL Schema	14 – 16
5.3	Security Design	16
5.3.1	Authentication	16
5.3.2	Authorization	17
5.4	System Requirements	17
CHAPTER 6		18 – 26
SYSTEM IMPLEMENTATION		
6.1	Development Tools and Environment	18
6.1.1	ASP.NET Core MVC	18
6.1.2	SQL Server	18
6.1.3	Entity Framework Core	18
6.2	Backend Implementation.....	18
6.2.1	Model	19

6.2.2	Controller	19 – 20
6.2.3	Service	20
6.2.4	Repository Pattern	20
6.3	Frontend Implementation	20 – 21
6.3.1	Views (Razor Pages)	21
6.3.2	UI Framework (Bootstrap & Css)	21
6.3.3	JavaScript Integration	21
6.3.4	Ajax Intergration	21
6.4	Authentication and User Management	21
6.4.1	Login	21 – 22
6.4.2	Logout	22
6.4.3	Registration	22
6.4.4	Role Based Access Control	22 – 23
6.4.5	My Account	23
6.5	Shopping And Payment System	23
6.5.1	Shopping Cart Logic	23
6.5.2	Checkout Workflow and Order Processing	23
6.5.3	PayPal Payment Gateway Integration	24
6.5.4	Order List History Persistence	24
6.6	Product Search Implementation	24
6.7	Intelligent Chatbot & Data Lookup System	25
6.7.1	Auto reply Processing Logic	25 – 26
6.7.2	System Effectiveness	26
CHAPTER 7		27 – 28
SYSTEM TESTING AND EVALUATION		
7.1	Testing Strategy	27
7.1.1	Unit Testing	27
7.1.2	Integration Testing	27 – 28
7.1.3	Test Result	28
7.2	Bug Tracking And fixed	28
CHAPTER 8		29 – 37
SYSTEM RESULTS AND DISCUSSION		
8.1	Homepage	29 – 30

8.2	Product Management	30 – 31
8.3	Shopping Cart	31 – 32
8.4	Payment Gateway Interaction	32 – 33
8.6	Admin Panel	33
8.6.1	Menus	33
8.6.2	Banners	34
8.6.3	Products	34
8.6.4	Users	34
8.6.5	Coupons	35
8.6.6	Orders	35
8.6.7	Comments	35
8.6.8	Settings	36
8.7	Chatbot Interface	36 – 37
8.8	Customer Feedback Comment	37
8.9	Contact Us	37
CHAPTER 9		38
FUTURE WORK AND ENHANCEMENTS		
9.1	System Limitations	38
9.2	Integration of Mobile Application	38
9.3	Recommendation System	38
9.4	Real-time Notification	38
CHAPTER 10		39
FINAL THOUGHTS AND CONCLUSION		
10.1	Conclusion	39
REFERENCES		40

LIST OF TABLES

<u>Table No.</u>	<u>Title</u>	<u>Page No.</u>
Table 2.1	Example of different type of the E-commerce	5
Table 3.1	Functional Use Cases of System	9
Table 5.1	System Entities and Their Purpose Key Features	14
Table 6.1	Data Model Mapping	19
Table 7.1	Bug Report Summary	28

LIST OF FIGURES

<u>Figure No.</u>	<u>Title</u>	<u>Page No.</u>
Fig.3.1	Login Section Diagram	9
Fig.3.2	Class Diagram Online Shopping System	10
Fig.4.1	Agile Model Diagram	12
Fig.5.1	SQL Script for Users Table	15
Fig.5.2	SQL Script for Product Table	15
Fig.5.3	SQL Script for Order Table	15
Fig.5.4	SQL Script for OrderDetails Table	16
Fig.6.1	Model Class for Product Entity	19
Fig.6.2	Controller Action for Product Detail	20
Fig.6.3	Repository Pattern Implementation	20
Fig.6.4	Login Page Interface	22
Fig.6.5	Registration Page Interface	22
Fig.6.6	PayPal Payment Gateway Flow	24
Fig.6.7	Product Search Logic	24
Fig.6.8	Auto-Reply Processing Logic Flow	26
Fig.7.1	Unit Test Results Showing	27
Fig.7.2	Integration Test Results Showing	28
Fig.8.1	Home Page View	29
Fig.8.2	Product List View	30
Fig.8.3	Products Details Page	31
Fig.8.4	Shopping Cart Page	31
Fig.8.5	Checkout Page	32
Fig.8.6	PayPal Payment Integration	32
Fig.8.7	Order History Interface	33
Fig.8.8	Admin DashBoard Interface	33
Fig.8.9	Menus Interface	33
Fig.8.10	Banners Interface	34
Fig.8.11	Products Management Interface	34
Fig.8.12	Users Management Interface	34
Fig.8.13	Coupons Management Interface	35
Fig.8.14	Orders Management Interface	35
Fig.8.15	Comments Management Interface	35
Fig.8.16	Setting Interface	36
Fig.8.17	ShopBot Assistant – Order,Coupon and Product Inquiries	36
Fig.8.18	Review Section Interface	37
Fig.8.19	Contact Section Interface	37

CHAPTER 1

INTRODUCTION TO ONLINE SHOPPING SYSTEM

1.1 Introduction

Online shopping, also known as commerce or e-commerce is the process of buying and selling goods and services over the internet. We use shopping on various digital platforms like websites, mobile applications and social media. Online shopping has become an important part of modern business systems. It has changed the way businesses and consumers do transactions.

With the internet getting better and better and with more people using devices online shopping has given entrepreneurs many chances to grow their businesses. They can now sell to people over the world not just in their own area. Online shopping also gives customers a flexible and convenient way to shop. They can do it at any time and from any place without having to go to a store.

One of the things about online shopping is that it is very convenient. Customers can look at products compare prices and make purchases at any time and from any place. They do not need to visit stores. This has made it easier for people to shop and has made them happier. Businesses also benefit from shopping. They do not need to spend money on physical stores but they can still sell to people all over the world.

Online shopping systems also help with things like managing inventory, processing orders handling payments and tracking deliveries. These things are done automatically which makes businesses more efficient. It also reduces the amount of work people have to do and minimizes mistakes. Modern online shopping platforms also help customers by giving those features, like product reviews and personalized recommendations. This helps people make decisions when they shop.

As a result online shopping has made customers happier. Has changed the way businesses work. It has made businesses more dynamic and scalable. Online shopping is still. It plays a big role in making the global digital economy more competitive and innovative. Online shopping or e-commerce is very important. It will continue to be so.

1.2 Motivation

Fifteen percent of people worldwide live with some form of disability. They can fully join in the world if websites and applications are built following proper web standards. A study by User Study and Experience Research Hub found that no e-commerce website in Bangladesh fully follows web standards. The study checked 174 e-commerce websites in Bangladesh. It used tools from W3C to look for errors, in accessibility, HTML and CSS. The study found that none of the sites were error-free. On average they had 60.57 accessibility errors, 49.52 HTML errors and 27.16 CSS errors. This shows that many websites do not follow standards. In Bangladesh many small and medium business owners do not know about e-commerce systems. This limits their business growth. So this project aims to create a secure and standard-compliant e-commerce platform. It will help businesses manage and sell products online in a way. The goal is to make it easy and reliable for businesses to sell online.

1.3 Problem Statement

In many developing countries online shopping platforms are still underutilized due to lack of proper systems, limited awareness and trust issues. Customers often face difficulties such as limited product availability inefficient order management and poor customer service. Similarly administrators face challenges in managing large inventories tracking orders and maintaining customer records efficiently. Therefore there is a need for a systematic online shopping management application that resolves these issues.

1.4 Objectives of the Project

The main objective of this project is to develop a user-friendly and secure Online Shopping Management System that facilitates customers to easily browse, select and purchase products while also providing administrators with efficient tools to manage inventory, users and transactions. This system aims to streamline the e-commerce process for both customers and administrators.

1.4.1 General Objective

The general objective of this project is to design and implement a comprehensive Online Shopping Management System that provides a smooth shopping experience for customers and efficient management tools for administrators.

1.4.2 Specific Objectives

- To develop a user-friendly interface for browsing and purchasing products online.
- To provide secure user authentication and registration features.
- To implement a shopping cart and order management system.
- To ensure proper management of products and categories by administrators.
- To create a reliable database system for storing customer and order records.
- To provide reporting and tracking features for better decision-making.

1.5 Scope of the Study

This project involves the development of a web-based shopping application that enables customers to register, log in, browse products, add selected items to a shopping cart and place orders. The system also supports an integrated online payment gateway allowing customers to complete transactions securely over the Internet. On the administrative side the platform provides functionalities to add, edit and delete products, manage customer information and process orders efficiently. While the core features are implemented advanced capabilities such as large-scale inventory management and real-time delivery tracking are beyond the current scope of this project and may be considered for future enhancements.

1.6 Significance of the Study

The system is significant as it demonstrates how an online shopping platform can simplify business operations and improve customer convenience in a modern digital environment. Customers can browse, compare, and purchase products from home which saves time and provides a flexible shopping experience, while administrators gain better control over product management, inventory, and order processing. The system also improves communication through features such as order tracking and feedback. From an academic perspective, this project provides practical experience in web application development, database management, and software engineering principles helping bridge the gap between theoretical knowledge and real-world implementation.

1.7 Methodology Overview

The project development followed the Software Development Life Cycle (SDLC) using the Agile Model. The steps included:

- **Requirement Analysis** – Understanding the needs of customers and administrators.
- **System Design** – Creating the architecture of the application.
- **Implementation** – Developing the system using ASP.NET Core MVC and SQL Server database.
- **Testing** – Checking the application for errors and ensuring functionality.
- **Deployment** – Hosting the system for demonstration purposes.

1.8 Project Book Organization

This report is systematically divided into nine chapters to present the study in a structured manner:

- **Chapter 1 – Introduction:** Provides the background of the study, problem statement, objectives, scope, significance, methodology overview and the overall organization of the report.
- **Chapter 2 – Literature Review and Related Work:** Presents related studies and existing systems, highlighting their strengths and weaknesses and identifying the research gap that this project addresses.
- **Chapter 3 – System Analysis and Design:** Describes requirement analysis, feasibility study, functional and non-functional requirements and use case analysis of the proposed system.
- **Chapter 4 – Software Process:** Discusses the software process models considered for the project, including Waterfall, V-Model, Incremental, Iterative, RAD, Spiral and Agile. This chapter explains why a particular process model was chosen and how it supports the development of the proposed system.
- **Chapter 5 – System Architecture and Design:** Focuses on the architecture of the system, database design, UML diagrams and user interface design.
- **Chapter 6 – System Implementation:** Explains the tools, technologies and programming environment used, along with the implementation of different system modules.
- **Chapter 7 – System Testing and Evaluation:** Discusses the testing methods, test cases and evaluation of the developed system to ensure its functionality and reliability.
- **Chapter 8– System Results and discussion:** Provides system output, screenshots, data analysis and discussion of the key results achieved through the implementation.
- **Chapter 9 – Future Work and Enhancements:** Highlights the constraints of the current project and suggests possible enhancements and extensions for future development.
- **Chapter 10 – Final Thoughts and Conclusion:** Summarizes the overall work, key contributions and final remarks regarding the project.

At the end, the report includes a References section for consulted materials and Appendices for supporting information such as code, schema, and diagrams.

CHAPTER 2

LITERATURE REVIEW AND RELATED WORK

2.1 Overview of E-Commerce Systems

E-commerce systems are digital platforms designed to enable the buying and selling of products or services online. These systems typically include essential features such as product catalogs, user authentication, shopping carts, order management and secure payment mechanisms. With the growth of internet penetration and technological advancement, e-commerce systems have become indispensable tools for modern businesses. They not only facilitate transactions between businesses and consumers but also streamline business processes, reduce operational costs and expand the market reach of organizations. A successful e-commerce system ensures usability, scalability, reliability and security which are crucial for user trust and long-term adoption.

2.2 Background of the Study

The internet has greatly influenced everyday life, especially in business. E-commerce combines Information Technology with business activities allowing companies to conduct transactions online. Today online shopping has become common and having an e-commerce platform is essential for business growth and sustainability. As more businesses adopt online systems, the internet is expected to become the primary medium for buying and selling in the future. As a multimedia and creative technology student with experience in graphics design, 3D animation and game development, this project aims to develop a modern dynamic e-commerce website where various products can be sold online conveniently.

2.3 Definition of E-commerce

E-commerce refers to the buying and selling of goods services or information over the internet including activities such as placing orders, making payments and tracking deliveries online. It involves both financial and informational transactions between businesses and customers improving the overall shopping experience. With the rapid growth of the internet and the World Wide Web, e-commerce has become more accessible and efficient covering all stages of online transactions such as marketing, ordering, payment and delivery support. Today it is widely used in daily life for online banking, retail shopping, travel reservations and electronic payments. E-commerce reduces costs, enhances customer service, speeds up processes and increases flexibility. It also supports communication and information sharing through technologies like email electronic data interchange (EDI) and mobile services. Due to its convenience and advantages e-commerce has become an essential part of modern business rather than just an option.

2.4 History of Ecommerce

The history of e-commerce began around 40 years ago and continues to evolve with new technologies and innovations. Early developments like Electronic Data Interchange (EDI) and mail-order systems in the 1970s laid the foundation for modern online shopping. E-commerce became possible for the public after the internet opened in 1991, though it was first introduced in 1979 by Michael Aldrich through a system connecting a television to a computer via a phone line. In the early 1990s, platforms like Book Stacks Unlimited emerged followed by major companies such as Amazon and eBay which played a key role in popularizing online transactions. The first online retail transaction occurred in

1994, marking a major milestone. Over time e-commerce has grown rapidly offering features like secure payments, product reviews and advanced search systems. Today it is a global industry that allows businesses and customers to buy and sell products easily over the internet.

2.5 Different types of E-commerce

E-commerce transactions occur between different parties such as consumers, businesses and government. When a business sells to consumers it is called B2C, while transactions between businesses are known as B2B. Similarly, business-to-government transactions are called B2G and consumer-to-consumer transactions are referred to as C2C. However, many e-commerce models are not easy to classify, as they often fall into multiple categories.

Table 2.1: Example of different type of the E-commerce

Type	Full Name	Examples
B2B	Business to Business	PayPal, Alibaba Group
B2C	Business to Consumer	Amazon, FreshDirect
B2G	Business to Government	E-Procurement
C2B	Consumer to Business	Google AdSense, Fotolia
C2C	Consumer to Consumer	eBay, Facebook
C2G	Citizen to Government	Agencia Tributaria
G2B	Government to Business	E-Government, AEPm
G2C	Government to Citizen	eDNI, USA.gov
G2G	Government to Government	Government Gateway, SIS

2.6 Online Shopping Trends

In recent years, online shopping has changed significantly due to several emerging trends. Mobile commerce (M-commerce) allows users to shop from smartphones anytime and anywhere increasing the need for mobile-friendly platforms. Personalization has also become important as artificial intelligence (AI) and machine learning (ML) provide product recommendations based on user behavior. Social commerce is growing through platforms like Facebook, Instagram and TikTok making shopping more interactive. Live commerce is another trend where products are sold through live streaming, allowing real-time interaction. Additionally, consumers are becoming more interested in sustainable and ethical shopping, preferring eco-friendly brands. These trends show that modern e-commerce must focus on providing a personalized and engaging shopping experience not just selling products.

2.7 Existing Messaging Systems in Web Applications

Messaging systems are increasingly integrated into web applications to enhance communication between customers and administrators. Platforms such as Amazon, eBay and Alibaba use both automated chatbots and human support systems to resolve customer queries. Real-time communication allows customers to receive order updates, request support and give feedback. However, in small-scale or academic projects, messaging systems are often limited to basic chat features due to infrastructure constraints. Despite this such systems remain essential to improving user engagement and building trust within e-commerce platforms.

2.8 Importance of Communication between Users

Effective communication between customers, administrators and sellers is a vital component of online shopping systems. Communication ensures that customers can clarify product details, track their orders, and resolve issues quickly. For administrators it provides insights into customer preferences and complaints which can guide service improvement. Strong communication features also help create transparency build customer loyalty and enhance decision-making processes. Thus, integrating a reliable communication channel is critical for the long-term success of e-commerce platforms.

2.9 Theoretical Framework

Several important theoretical models have been used for the design and development of this project which will ensure the system's effectiveness and usability.

- **The Unified Theory of Acceptance and Use of Technology (UTAUT):** This theory explains why users are willing to accept or use a technology. It is centered around four key factors:
 - **Performance Expectancy:** When a user believes the system will improve their work.
 - **Effort Expectancy:** When a user believes the system is easy to use.
 - **Social Influence:** When a user feels that people they know will encourage them to use it.
 - **Facilitating Conditions:** When the user feels they have the necessary infrastructure and support to use the system. Our project will focus on improving user's performance and effort expectations which will increase the system's acceptance.
- **Technology Acceptance Model (TAM):** This model is a simplified version of UTAUT. Its two main factors are:
 - **Perceived Usefulness:** When a user believes the system will increase their job performance.
 - **Perceived Ease of Use:** When a user believes the system will be easy to use. Our system will try to fulfill these two concepts through an intuitive design and effective features.
- **User-Centered Design (UCD):** This method places the user at the center of the system's design. Its goal is to design a system that is consistent with the user's needs limitations and behavior. In our project, this framework will be used to make every step of user interaction easy and intuitive.

Using this theoretical framework, we will ensure that our system is not only technologically advanced but also capable of meeting users' needs and providing them with a satisfactory experience.

CHAPTER 3

SYSTEM ANALYSIS AND DESIGN

3.1 Requirements Analysis

This section presents a comprehensive analysis of the requirements for the proposed Online Shopping System. A clear and detailed understanding of these requirements is essential to ensure that the system is designed and developed in a way that effectively meets user expectations as well as organizational goals. Proper requirement analysis helps in minimizing errors reducing development risks and ensuring the overall quality of the system. The requirements identified for this system are based on user needs system functionality and the business processes involved in an online shopping environment which together form the foundation for system design, implementation and testing. For better understanding and organization these requirements are generally classified into two main categories namely functional requirements and non-functional requirements, where functional requirements describe the core features and operations of the system and non-functional requirements focus on performance, security, usability and other quality attributes of the system.

3.1.1 Functional Requirements

Functional requirements define the essential features and services that the system must provide to its users. These functionalities ensure that the system operates effectively and fulfills user needs.

1. User Registration and Login:

The system allows new users to create an account by providing their name, email and password. Registered users can log in using their credentials. For security purposes all passwords are stored using hashing techniques.

2. Product Browsing:

Users can browse a list of available products within the system. The system provides search and filtering options to help users find products based on category, price range or keywords. Each product includes detailed information such as name, description, price, stock availability and images.

3. Shopping Cart:

The system enables users to add products to a shopping cart. Users can update product quantities or remove items as needed. The cart is maintained for logged-in users across sessions.

4. Checkout and Order Placement:

Users can proceed to checkout by providing shipping and payment details. After confirmation, the system processes the order and generates an order record stored in the database.

5. Admin Product Management:

The system provides administrative functionality that allows admin users to add, update and delete products from the catalog. Admins can also manage product stock and upload images.

3.1.2 Non-Functional Requirements

Non-functional requirements define the quality attributes of the system rather than its specific behaviors. These requirements are essential for ensuring that the system performs efficiently, remains secure, is user-friendly, and operates reliably under different conditions. They focus on how the system performs its functions and the level of quality it provides to end users.

1. Performance

The system is required to support a minimum of 100 concurrent users without any noticeable degradation in performance. Additionally, the system should ensure that page loading time remains under 3 seconds under standard broadband internet conditions to provide a smooth user experience.

2. Security

To ensure data protection, all sensitive information, particularly user passwords, must be securely stored using strong hashing algorithms such as PBKDF2 or bcrypt. The system must also implement HTTPS to encrypt data during transmission, ensuring secure communication between client and server. Furthermore, user sessions should automatically expire after a defined period of inactivity to reduce the risk of unauthorized access.

3. Usability

The system interface should be designed to be responsive and compatible with both desktop and mobile devices, which can be achieved through the integration of Bootstrap. In addition, the navigation structure should be intuitive, with clearly labeled options and a consistent design approach to enhance user experience.

3.1.3 User Requirement Specification

The User Requirement Specification (URS) describes the needs and expectations of the users who will interact with the Online Shopping System developed using ASP.NET Core MVC. The system is designed to provide a simple, efficient, and user-friendly experience for different types of users, including customers, administrators, and vendors. Customers should be able to easily register and log in to the system, browse and search for products, view detailed product information, and add items to their shopping cart. They must also be able to complete purchases through a secure checkout process with multiple payment options and track their orders. In addition, customers should have access to manage their profiles, view order history, and provide feedback or product reviews. Administrators require full control over the system, including managing products, categories, and inventory, as well as monitoring and processing customer orders. They should also be able to manage user accounts and generate reports related to sales and system performance. Vendors, if included in the system, should have the ability to register, manage their own products, update listings, and track their sales and orders. The system must ensure proper authentication and authorization for all users to maintain security and data privacy. Furthermore, the interface should be responsive and accessible across different devices such as desktops, tablets.

3.2 Use Case Analysis

Use case modeling describes the interaction between the users (actors) and the system to achieve specific goals. It represents the system functionalities from a user perspective and helps to understand how different users interact with the application.

3.2.1 Use Case Diagram

The system has two main actors: Customer and Admin. The Customer can log in, browse products, add items to the cart, and complete checkout, while the Admin manages product operations.

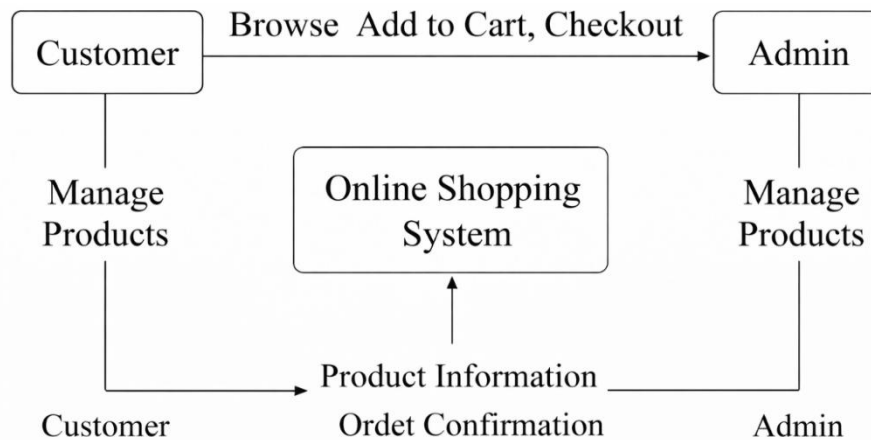


Fig.3.1: Login Section Diagram

3.2.2 Use Case Descriptions

This table explains each use case in detail — who performs it and what it does. Here's the table you should include:

Table 3.1: Functional Use Cases of the System

Use Case ID	Use Case Name	Actor	Description
UC-01	Login	Customer/Admin	User logs into the system
UC-02	Browse	Customer	Customer views available products
UC-03	Add to Cart	Customer	Customer adds products to shopping cart
UC-04	Checkout	Customer	Customer places an order
UC-05	Manage Prod	Admin	Admin adds, updates or deletes products

3.3 Class Diagram

The UML Class Diagram illustrates the core models of the Online Shop system designed using ASP.NET Core MVC. It highlights how Users, Products and Orders interact, while also showing the role of additional supporting entities such as Coupons, Comments, Banners and Configurations. This diagram provides a clear overview of both the main e-commerce workflow and the auxiliary models that support system functionality.

1. Relationships:

- **User → Order (One-to-Many):** A single user can place multiple orders over time, but each order is associated with only one user.
- **Order → OrderDetail (One-to-Many):** Each order may contain multiple order detail records, where each record represents an individual product purchased within that order.
- **Product → OrderDetail (One-to-Many):** A single product can appear in multiple order details across different orders placed by different users.

- **Product → ProductGallery (One-to-Many):** Each product can have multiple images stored in its gallery to provide better visual representation.
- **User → Comment (One-to-Many):** A user can write multiple comments or reviews for different products.
- **Product → Comment (One-to-Many):** A product can receive multiple comments from different users.
- **Other Entities (Banner, Coupon, Setting):** These entities operate independently and are used for system configuration, promotional offers, and UI management.

Now the following table provides a brief overview of each class in the online shop system, summarizing their purpose and the key information they store. This complements the relationships described above and helps understand the role of each entity in the system.

The Class Diagram of the system is shown below:

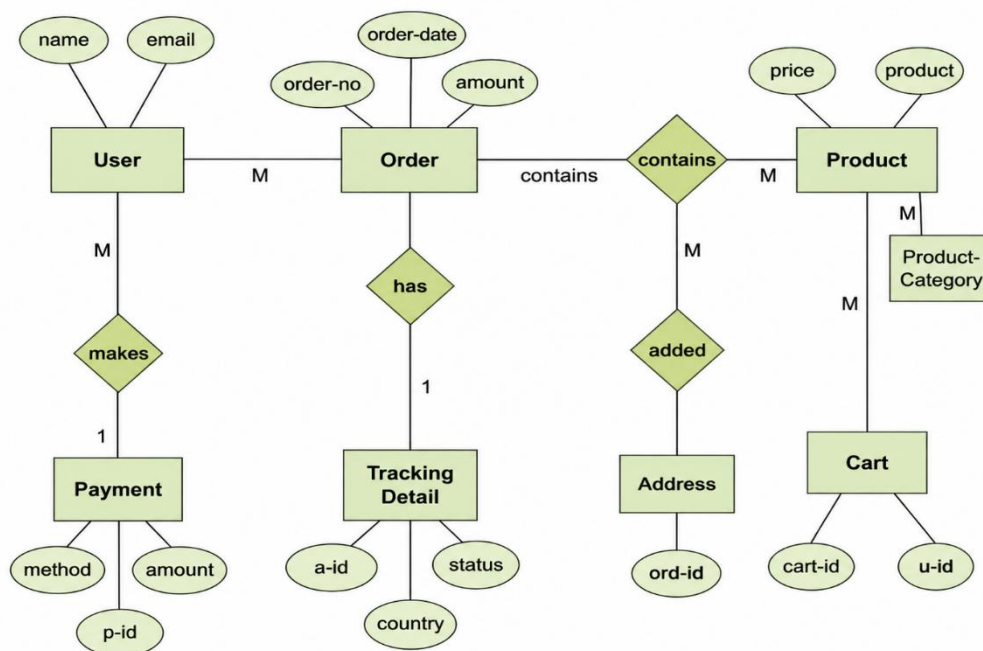


Fig.3.2: Class Diagram Online Shopping System

This chapter presents the detailed design and analysis of the proposed Online Shopping System. It describes the overall system architecture, data flow and database structure. The Data Flow Diagrams (DFD) illustrates how information moves through different processes such as product browsing, adding items to the cart and order checkout. The Entity Relationship Diagram (ERD) defines the main entities and their relationships ensuring an efficient and normalized database design. Additionally the Class Diagram represents the object-oriented structure of the system using C# models. Overall, this chapter establishes a clear blueprint for the implementation of the system in the subsequent stages.

CHAPTER 4

SOFTWARE PROCESS

4.1 Introduction to Software Process

Software processes, methodologies and frameworks range from specific prescriptive steps that organizations can apply in daily operations to flexible frameworks tailored to a project or team. Sponsoring organizations often provide detailed documentation describing the process. In online business systems software streamlines various steps involved in purchasing. “Software” refers to programs, procedures and documentation such as manuals and flowcharts, describing the program and its usage. A software process is a set of activities and outcomes that produce a software product, primarily performed by software engineers. The four key process activities are software specification, defining functionality and constraints software development, producing the software to meet requirements, software validation, ensuring it works as intended and software evolution adapting and enhancing it to meet changing needs.

Software Development Life Cycle (SDLC) Models:

- Waterfall model
- V model
- Incremental model
- RAD model
- Agile model
- Iterative model
- Spiral model
- Prototype model

4.2 Agile Model

In this project the Agile approach is used as the main method for software development because it provides a flexible, iterative and incremental way of building the system where development is divided into small sprints and each sprint delivers a working part of the application. This approach allows continuous improvement of the system step by step instead of completing everything at the final stage and it also helps in identifying and fixing issues early, during each iteration reducing overall project risk and increases system quality. Agile also supports regular communication between developers and stakeholders making it easier to understand user requirements clearly and update them whenever needed. The main processes followed in this model include requirement elicitation, planning, system design, implementation, testing, deployment, and evaluation and these activities are repeated in every sprint to ensure continuous progress and refinement of the system. After each iteration, a functional version of the system is produced which can be reviewed and tested to collect feedback for further improvement. Agile is particularly suitable for e-commerce systems because requirements can change frequently during development and this model allows quick adaptation without disrupting the overall project flow. Therefore, Agile provides a practical and effective foundation for developing a reliable and user-friendly Online Shopping System.

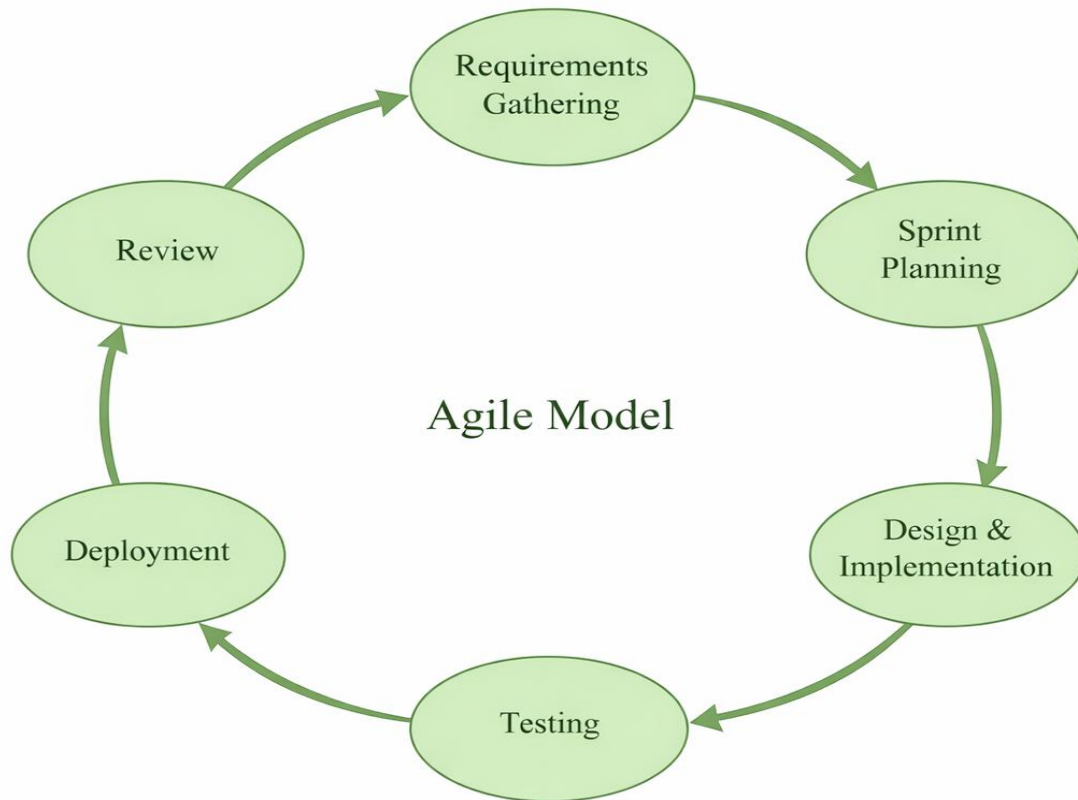


Fig.4.1: Agile Model Diagram

4.3 Data Structures and Algorithm Used

Algorithms and data structures are used in this project to improve system performance, efficiency and smooth data handling. They help in managing products, processing orders searching data and optimizing user operations within the system. Different structures and algorithms are applied based on system requirements to ensure fast and accurate results.

- **List / Array:** Used for managing products, cart items, and orders.
- **Linear Search:** Used for searching products, users, and orders.
- **Sorting:** Used for arranging product lists and order history.
- **Filtering:** Used for selecting specific items from search or cart.
- **Aggregation:** Used for calculating total price and quantity.
- **Regex / Pattern Matching:** Used for email validation and input checking.
- **Hashing:** Used for secure storage like OTP and identifiers.

These techniques help ensure faster processing, better data organization and an improved and smooth user experience in the e-commerce system.

CHAPTER 5

SYSTEM ARCHITECTURE AND DESIGN

5.1 System Architecture

The proposed Online Shopping System follows a three-tier client-server architecture that separates the system into different manageable layers to ensure scalability, security and performance. The major components of the system are the User Interface (Client) Application/Server Layer and Database Layer. Each layer has a specific role in processing user requests and delivering accurate responses. When a user interacts with the system through a web browser a request is sent to the Web Server. The server processes the request by applying business logic and communicates with the Database Server to store, retrieve or update data. After processing the request the server sends the response back to the client and the result is displayed on the user interface. This continuous data flow ensures smooth and real-time communication between all components. This architecture improves maintainability by separating responsibilities into individual layers. It also enhances security by preventing direct access to the database from the client side.

1. Major Components

- Client (Front-End Interface): Accepts user inputs and displays results.
- Application Server (Business Logic Layer): Handles order processing, product management, authentication, etc.
- Database Server (Data Storage Layer): Stores user, product, order and payment information securely.

2. Data Flow Process

- User Request: The user sends a request (e.g., browse products, add to cart, login).
- Application Server Processing: The request reaches the application server, where business logic is applied.
- Database Interaction: The server retrieves or updates data from the database.
- Database Response: The database sends the response back to the server.
- Client Response: The server responds to the client with the processed result, which is displayed on the user interface.

3. Communication Protocol:

The interaction between client and server follows a stateless RESTful architectural pattern using HTTP/HTTPS protocols. Communication occurs through well-defined API endpoints that support standard HTTP methods:

- GET /api/products - Retrieves product catalog information.
- POST /api/orders - Submits new customer orders.
- PUT /api/products/{id} - Updates product details.
- DELETE /api/products/{id} - Removes products from inventory.

It explains how the client (browser) sends requests to the web server the server processes the requests and then communicates with the database to store or retrieve data. Finally the response is sent back to the client. This structure separates the system into three parts: client, server and database ensuring smooth communication and efficient data handling.

5.2 Database Design

The database design follows a normalized relational model specifically tailored for e-commerce operations. Based on the Entity Framework DbContext analysis the system incorporates 12 core entities that handle everything from product management to order processing and user interactions.

Core Entity Breakdown:

Table 5.1: System Entities and Their Purpose Key Features

Entity	Purpose	Key Features
Product	Product catalog	Price, discount, inventory, media
ProductGallery	Product images	Multiple images per product
Category	Product classification	Hierarchical structure
Order	Order management	Customer info, status
OrderDetail	Order items	Product details at purchase time
User	Customer accounts	Authentication, profile
ShoppingCart	Cart system	Temporary product storage
Comment	Reviews	User feedback
Coupon	Discounts	Code-based promotions
Banner	Marketing	Promotional content
Menu	Navigation	Site menu structure
Setting	Configuration	Global system settings

5.2.1 Table and Relationship

The database schema implements a relational structure where entities are connected through well-defined relationships to maintain data integrity and support e-commerce operations. The core model is centered on the Product entity which is linked to ProductGallery for image management and Category for organized catalog browsing. The order process defines a one-to-many relationship between User and Order where each order contains multiple Order Details that store product information at the time of purchase, ensuring price accuracy. Additional relationships support user interaction through the Comment system for product reviews while supporting tables such as Coupon, Banner and Setting handle promotions and system configuration. Overall these entities work together in an integrated structure to ensure smooth and efficient system functionality.

5.2.2 Normalization Process

The database design follows normalization to reduce redundancy and maintain data integrity, mainly achieving Third Normal Form (3NF). Product information is stored in the Products table while images are separated into ProductGallery to avoid duplication. Orders store product snapshots for historical accuracy while maintaining relationships with other tables. User data is kept in the Users table and Coupon data remains independent. This approach ensures efficient performance, easy maintenance and a scalable database structure.

5.2.3 SQL Schema

The SQL schema defines the structure of the database used in the system. It includes tables, attributes and relationships required to manage application data efficiently. A relational database is used to ensure data integrity and consistency.

Main Tables:

1. User Table

This table stores users account information

```
CREATE TABLE [dbo].[User](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Email] [nvarchar](50) NOT NULL,
    [FullName] [nvarchar](50) NOT NULL,
    [Password] [nvarchar](50) NOT NULL,
    [IsAdmin] [bit] NOT NULL,
    [RegisterDate] [datetime] NULL,
    [RecoveryCode] [int] NULL,
    CONSTRAINT [PK_User] PRIMARY KEY CLUSTERED
```

Fig.5.1: SQL Script for Users Table

2. Product Table

This table stores product details such as title and price

```
CREATE TABLE [dbo].[Products](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Title] [nvarchar](200) NULL,
    [Description] [nvarchar](500) NULL,
    [FullDesc] [nvarchar](4000) NULL,
    [Price] [money] NULL,
    [Discount] [money] NULL,
    [ImageName] [nvarchar](50) NULL,
    [Qty] [int] NULL,
    [Tags] [nvarchar](1000) NULL,
    [VideoUrl] [nvarchar](300) NULL,
    CONSTRAINT [PK_Products] PRIMARY KEY CLUSTERED
```

Fig.5.2: SQL Script for Product Table

3. Orders Table

This table stores order information include date and total amount

```
CREATE TABLE [dbo].[Order](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [UserId] [int] NOT NULL,
    [FirstName] [nvarchar](50) NOT NULL,
    [LastName] [nvarchar](50) NOT NULL,
    [CompanyName] [nvarchar](50) NULL,
    [Country] [nvarchar](50) NOT NULL,
    [Address] [nvarchar](200) NOT NULL,
    [City] [nvarchar](50) NOT NULL,
    [Email] [nvarchar](50) NOT NULL,
    [Phone] [nvarchar](50) NOT NULL,
    [Comment] [nvarchar](200) NULL,
    [CouponCode] [nvarchar](50) NULL,
    [CouponDiscount] [money] NULL,
    [Shipping] [money] NULL,
    [SubTotal] [money] NULL,
    [Total] [money] NULL,
    [CreateDate] [datetime] NULL,
    [TransId] [nvarchar](200) NULL,
    [Status] [nvarchar](50) NULL,
    CONSTRAINT [PK_Order] PRIMARY KEY CLUSTERED
```

Fig.5.3: SQL Script for Order Table

4. OrderDetails Table

This table stores detailed items of each order with product reference

```
CREATE TABLE [dbo].[OrderDetails](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [ProductTitle] [nvarchar](200) NOT NULL,
    [ProductPrice] [money] NOT NULL,
    [Count] [int] NOT NULL,
    [OrderId] [int] NOT NULL,
    [ProductId] [int] NOT NULL,
    CONSTRAINT [PK_OrderDetails] PRIMARY KEY CLUSTERED
)
```

Fig.5.4: SQL Script for OrderDetails Table

Supporting Tables

The system also includes several supporting table to enhance functionality:

- **Banner:** Stores Homepage banner information.
- **Comment:** Stores user review or comment.
- **Coupon:** Manages discount coupons.
- **Menus:** Handles navigation menu data.
- **ProductGallery:** Stores multiple images of products.
- **Settings:** Stores application configuration.

5.3 Security Design

Security is a fundamental aspect of the Online Shop system, as it ensures that user data is protected, unauthorized access is prevented and all operations are performed safely. The system is developed using ASP.NET Core MVC which provides built-in security features that are properly configured to meet application requirements. The security design is based on three core components: authentication, authorization and data protection. Authentication verifies user identity through secure login, encrypted passwords and proper session management. Authorization controls access by defining user roles such as customer, administrator and vendor ensuring that users can only perform permitted actions. Data protection secures all stored and transmitted information using encryption, secure protocols and protection against common threats such as SQL injection, cross-site scripting (XSS) and cross-site request forgery (CSRF). Overall, the system follows secure coding practices and modern security standards to provide a safe and reliable user experience.

5.3.1 Authentication

Authentication is the process of verifying a user's identity before granting access to the system. The Online Shop application employs ASP.NET Core Identity framework to manage authentication securely and efficiently. This framework provides a robust mechanism for handling user registration, login, passwords and related processes. The system supports essential features such as user registration and login, email confirmation for account verification, password recovery and reset functionality, remember me option for persistent sessions and role-based user management for both Admin and User roles. A custom ApplicationUser class extending IdentityUser was implemented to store additional user information. Identity services are configured in the Program.cs file using cookie-based authentication with sliding expiration, while passwords are securely hashed using the built-in Password Hasher of ASP.NET Core Identity ensuring that plain text passwords are never stored in the database.

5.3.2 Authorization

Authorization determines what actions a user is allowed to perform after successful authentication. In this system role-based authorization is implemented to restrict access to different parts of the application. This approach ensures that users can only access the features and resources that are permitted according to their assigned role. The system defines two primary roles: Admin and User. The Admin role has full access to all administrative functionalities including product management, order management, banner management, coupon management and system configuration. On the other hand, the User role is restricted to general user functionalities such as browsing products, adding items to cart, placing orders, viewing order history and managing personal profile information. This role-based access control mechanism significantly enhances the security and integrity of the system by preventing unauthorized access to sensitive administrative features.

5.4 System Requirements

This section describes the hardware and software requirements needed to run the e-commerce application efficiently across different operating systems.

Hardware Requirements

The system requires a minimum of 4 GB RAM, while 8 GB RAM is recommended for smooth performance. A modern processor such as Intel Core i3 or higher is sufficient. At least 10 GB of free disk space is required for application files and database storage.

Software Requirements by Platform:

1. for Windows:

The application can run on Windows 10 or later versions. It requires .NET SDK (version 6 or later), Visual Studio or Visual Studio Code and SQL Server or SQL Server Express for database management. A modern web browser such as Google Chrome or Microsoft Edge is also required.

2. for MacOS:

The application can run on macOS using the .NET SDK (version 6 or later). Development can be done using Visual Studio Code or Visual Studio for Mac. Since SQL Server is not fully native, developers can use SQL Server via Docker or alternatives like PostgreSQL. A modern web browser such as Safari or Google Chrome is required.

3. for Linux:

The application is supported on Linux distributions such as Ubuntu. It requires .NET SDK (version 6 or later) and development tools like Visual Studio Code. For database management, PostgreSQL or MySQL is commonly used although SQL Server can also be used via Docker. A modern web browser such as Firefox or Chrome is recommended.

4. Network Requirements:

A stable internet connection is necessary for accessing the system especially for online transactions, API integrations, and payment gateway services. These platform-based configurations ensure that the application can run efficiently on Windows, macOS and Linux, demonstrating the cross-platform capability of ASP.NET Core.

CHAPTER 6

SYSTEM IPLEMENTATION

6.1 Development Tools and Environment

The Online Shop system was developed using the Microsoft .NET technology stack. Tools and technologies were selected based on performance, maintainability and integration capabilities. Microsoft Visual Studio 2022 was used as the primary IDE, ASP.NET Core 8.0 as the web framework, Entity Framework Core for database operations, and Microsoft SQL Server as the backend database [1]. Additional tools such as Git for version control and Bootstrap 5 for frontend design were also utilized. This environment ensured efficient development and seamless deployment of the application.

6.1.1 ASP.NET Core MVC

ASP.NET Core MVC is an open-source framework created by Microsoft that runs on multiple platforms. This is based on the Model-View-Controller architecture, where data management (Model), user interface (View) and HTTP request handling (Controller) are separated. Some key advantages of using this framework include:

- Dependency injection support for loosely coupled systems [5].
- A built-in middleware pipeline for authentication and other functions.
- The Razor view engine, which uses C# code to dynamically render HTML.
- Extensive support for Entity Framework Core, allowing easy data access [4].
- RESTful routing and attributes for routing configuration [2].

6.1.2 SQL Server

The RDBMS chosen for this task is Microsoft SQL Server. SQL Server comes with ACID transaction support, advanced indexing features, and ASP.NET Core integration through Entity Framework [3]. All the persistent data of the application resides within the database in the form of users, products, orders, and shopping carts. For the development of the project, SQL Server Management Studio (SSMS) has been employed.

6.1.3 Entity Framework Core

Entity Framework (EF) Core is the official ORM of .NET. It allows eliminating hand-coded SQL in almost all cases since EF Core maps C# classes directly to database tables [4]. In the current project, the Code-First approach has been adopted, whereby the database schema will be defined on the basis of C# entity classes with migration history of the schema modifications stored. The advantages of EF Core are:

- Highly-efficient SQL generation based on LINQ queries [4].
- Automatic change tracking for creating, updating and deleting data.
- Database migrations that enable versioned evolution of schema.
- Eager (via Include method) and lazy loading of related entities.

6.2 Backend Implementation

The backend follows a structured ASP.NET Core MVC architecture encompassing model classes, controller actions, embedded service logic and data access patterns [2].

6.2.1 Model

Models represent data structures that map directly to SQL Server tables via Entity Framework Core [4]. The Product model stores title, price, discount, quantity, image name, and tags. The User model captures email, full name, password, admin flag and recovery code. The Order model stores billing address, subtotal, shipping, coupon discount, transaction ID and status. All models are configured in OnlineShopContext using the Fluent API inside the OnModelCreating method.

Table 6.1: Data Model Mapping

Model Class	Mapped Table	Description
User	User	Extends IdentityUser; stores profile, address, and role data.
Product	Products	Stores product name, price, category, stock, and image path.
Category	Categories	Hierarchical product categories with parent-child relationship.
Order	Orders	Records order header: user, date, status, and total amount.
OrderDetail	OrderDetails	Line items linking Order to Product with quantity and unit price.

A representative model class for the Product entity is shown below:

```
namespace OnlineShop.Models.Db
{
    public partial class Product
    {
        public int Id { get; set; }
        public string? Title { get; set; }
        public string? Description { get; set; }
        public string? FullDesc { get; set; }
        public decimal? Price { get; set; }
        public decimal? Discount { get; set; }
        public string? ImageName { get; set; }
        public int? Qty { get; set; }
        public string? Tags { get; set; }
        public string? VideoUrl { get; set; }
    }
}
```

Fig.6.1: Model Class for Product Entity

6.2.2 Controller

Controllers serve as the request handling layer. All controllers are derived from the Controller class of ASP.NET Core framework and handle HTTP requests by calling service methods and return View or JSON response[2].

The important controllers in the application are:

- **HomeController** — Landing page, featured products display.
- **ProductController** — Product listing, search, detail page.
- **CartController** — Add, update, remove cart items; session management.
- **OrderController** — Checkout, order confirmation, order history.
- **PaymentController** — PayPal redirect, IPN handling and payment confirmation.
- **AccountController** — Login, logout, registration, profile management.
- **AdminController** — Product CRUD, order management, user roles (admin only).
- **ChatController** — Receive messages, return auto-reply responses.

A representative controller action for product detail is shown below:

```
public class ProductController : Controller
{
    private readonly IProductService _productService;

    public ProductController(IProductService productService)
        => _productService = productService;

    public async Task<IActionResult> Details(int id)
    {
        var product = await _productService.GetByIdAsync(id);
        if (product == null) return NotFound();
        return View(product);
    }
}
```

Fig.6.2: Controller Action for Product Detail

6.2.3 Service

The service layer contains all business logic in the application, functioning as a mediator between the controller layer and the repository layer. The services have been declared using interfaces and are registered inside the ASP.NET Core Dependency Injection container [5]. This ensures that the controllers are thin and do not contain any business logic. Core service interfaces include IProductService for filtering, pagination, stock validation and search IOrderService for placing orders, changing the status and invoicing ICartService for performing cart operations, calculating the total amount and managing sessions [9] IPaymentService for sending data to the PayPal API and verifying transactions IAccountService for registering users, hashing passwords, and assigning roles [8] and IChatService for matching keywords, evaluating rules, and generating responses.

6.2.4 Repository Pattern

The Repository Pattern hides all database activities behind an abstract interface separating business rules from the data access technology [4]. The IRepository<T> generic interface specifies common CRUD methods where the specific implementations use Entity Framework Core DbContext:

```
public interface IRepository<T> where T : class
{
    Task<T> GetByIdAsync(int id);
    Task<IEnumerable<T>> GetAllAsync();
    Task AddAsync(T entity);
    void Update(T entity);
    void Remove(T entity);
    Task SaveChangesAsync();
}
```

Fig.6.3: Repository Pattern Implementation

The specific repositories, such as ProductRepository or OrderRepository, inherit from the base repository but include domain-specific methods like GetProductsByCategoryAsync() or GetOrdersByUserAsync(). The Unit of Work implementation is done based on these repositories, providing transactional support across multiple repositories.

6.3 Frontend Implementation

The front-end of the application is developed using ASP.NET Core Razor Pages, Bootstrap 5, Custom CSS, and JavaScript with AJAX. This technology stack provides an

interactive front-end that works without the need for any additional single page application framework.

6.3.1 Views (Razor Pages)

Views are implemented as Razor (.cshtml) files, which combine HTML with embedded C# code using the @ syntax [2]. Each view is connected to a strongly typed model allowing dynamic and reliable data rendering within the UI. Tag Helpers such as asp-for, asp-action and asp-validation-for are used to simplify form handling and reduce repetitive HTML code. All pages follow a common structure by inheriting the shared _Layout.cshtml template, which provides a consistent navigation bar, footer and chatbot widget across the entire application.

6.3.2 UI Framework (Bootstrap & CSS)

Bootstrap 5 is the primary CSS framework used for responsive grid layouts and pre-built UI components such as cards, buttons, forms, alerts, and modals [6]. Custom CSS is layered on top to style the shop's specific visual elements including product card hover effects, chatbot window, and mini cart overlay. Additional libraries like Slick.js for carousels and Magnific Popup for image lightboxes are also included.

6.3.3 JavaScript Integration

Vanilla JavaScript and jQuery are used for client-side interactivity [7]. Core use cases include:

- Real-time quantity update buttons on the cart page.
- Image preview on the product upload form in the admin panel.
- Client-side form validation before submission.
- Dynamic hiding/showing of payment sections based on selection.
- Scroll-to-top button and smooth navigation transitions.

6.3.4 AJAX Integration

The communication between client and server takes place through AJAX (Asynchronous JavaScript and XML) without any complete loading of pages. Ajax POST is used by the cart updates to update cart action, returning item quantity in JSON format. In addition the mini-cart is refreshed using another Ajax GET command to the URL of small cart. Chatbot sends POST requests for fetching question replies to ChatbotController by using Fetch API and adds JSON responses to chatbox without any reloading of the page [7].

6.4 Authentication and User Management

Authentication and authorization are implemented using ASP.NET Core Identity a built-in membership system that handles password hashing, cookie-based sessions and role management [8]. Identity is configured in Program.cs and integrated with the EF Core DbContext.

6.4.1 Login

In case of authentication, email and password are used by the interface. The SignInManager checks the entered credentials against the hashed password stored in the Password table. On success, the user is issued a secure auth cookie that includes their claim - the ID, the email address, and the role. In case of three failed attempts at logging in, the account becomes locked out. Password can be reset using a confirmation code sent via email.

Fig.6.4: Login Page Interface

6.4.2 Logout

The Logout functionality utilizes the function “SignInManager.SignOutAsync()”[8]. This will invalidate the authentication cookie and session data. The authenticated-session shopping cart is saved into the database for future access during the next login.

6.4.3 Registration

New users can sign up by filling in their name, email address and password. During sign-up the form checks for duplicates performs email verification by sending a five digit verification code to the email address, verifies the code, and registers the user along with the time stamp. The form does both client and server side validation.

Fig.6.5: Registration Page Interface

6.4.4 Role-Based Access Control

Role-based access control (RBAC) is implemented using ASP.NET Core claims and authorization attributes. During login, the user’s role is stored as a claim with values such as “admin” or “user.” Admin area controllers use [Authorize(Roles = "admin")] to restrict access while regular authenticated routes use [Authorize] without specifying a role. Unauthenticated users are automatically redirected to the login page by the Cookie Authentication middleware.

- **Customer (Default):** Browse products, manage cart, checkout, view own orders, update profile.

- **Administrator:** All customer areas plus product CRUD, order management, user roles and reports.

6.4.5 My Account

Authenticated users can access their personal account section through the User area of the system. The OrdersController filters order information using the logged in user's ID from ClaimTypes.NameIdentifier and displays the orders in descending order based on creation time [8]. Users are able to view complete order details including purchased products quantities prices coupon discounts shipping costs and total payment amounts. The My Account section also helps users manage and review their previous purchases in a simple and organized way which improves the overall user experience and makes order tracking more convenient.

6.5 Shopping and Payment System

The shopping and payment system covers the complete e-commerce flow from product selection through to payment confirmation.

6.5.1 Shopping Cart Logic

Cart management is applicable to guest shopping and authenticated shopping. Items stored in CartItem for guests will be combined in the cart in the database when the guest logs into the application. The CartService class contains the following methods:

- **AddToCart(productId, quantity):** checks inventory availability before adding an item
- **UpdateQuantity(cartItemId, newQty):** recalculates line total and cart total.
- **RemoveItem(cartItemId):** removes the cart item.
- **GetCartTotal():** computes $\text{sum}(\text{unitPrice} * \text{Quantity})$ for all active items.
- **ClearCart():** called after successful checkout.

6.5.2 Checkout Workflow and Order Processing

However the checkout module only allows authenticated users with the use of the [Authorize] attribute. This module will present the products that are currently inside the cart together with the form that requires information about the billing and shipping addresses. At this stage the application will validate any coupon code that was inputted by the user with the help of the Coupons table to ensure that all discounts will be properly applied. The shipping fee is obtained from the Settings table to have flexibility in managing costs. Once the user fills out the required fields in the form and clicks on the submit button the system will create a new entry in the Order table and Order Detail tables for each product inside the cart.

A multi-step checkout process for ensuring data consistency and providing a user-friendly experience:

- **Step 1: Review Cart** – verify product and quantity details.
- **Step 2: Enter Shipping Information** – provide shipping address details.
- **Step 3: Select Payment Method** – select PayPal.
- **Step 4: Confirm Order** – generate Order and OrderItem entries.
- **Step 5: Display Receipt** – display order number and order summary.

Order creation is wrapped in an EF Core database transaction to guarantee that stock decrement and order insertion succeed or fail atomically.

6.5.3 PayPal Payment Gateway Integration

The PayPal standard checkout is implemented using PayPal REST API along with PayPal's official SDK for .NET [10]. The procedure involved is as below:

- A PayPal order is generated by the Payment Controller with the amount from the cart and the currency USD [10].
- A user is redirected to the PayPal hosted payment gateway site where payment and authentication occur.
- After successful authentication, PayPal redirects the user to the return URL of the application along with the transaction id [10].
- The Payment Controller makes an API call to PayPal to capture the payment transaction.
- A payment object is saved, which includes the transaction id, amount, and transaction status (Completed/Failed).
- Order status becomes Paid, and email notification is enqueued.

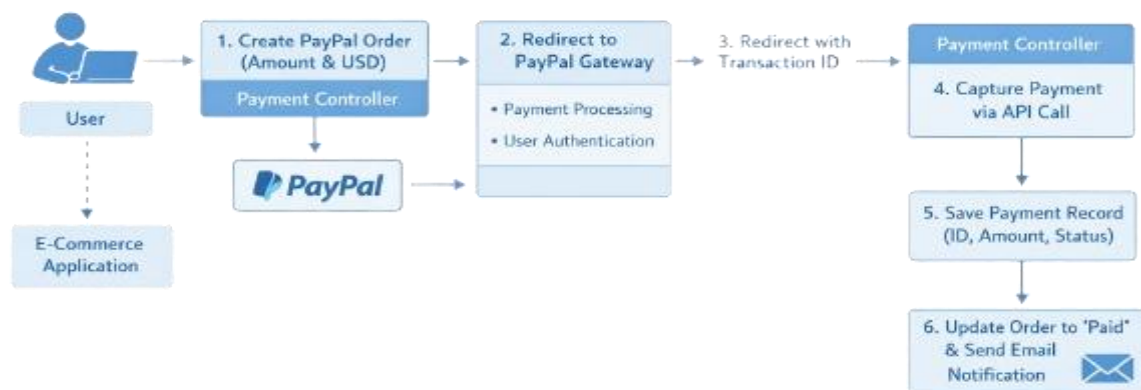


Fig.6.6: PayPal Payment Gateway Flow

6.5.4 Order List History Persistence

Previous orders will be stored permanently in the Orders and OrderItems database. The history of any previous orders can be viewed by the customers themselves in the My Account section. Every order will show order date, order items, sub-total, whether it is paid or not and delivery status [3]. All orders can be accessed by the administrator with filtering on basis of its status (pending, paid, shipped, delivered, or cancelled).

6.6 Product Search Implementation

Product search is implemented in the ProductsController's SearchProducts action. It makes use of `EF.Functions.Like()` for case insensitive partial matching on product title and tags and sorts results alphabetically and shows in Index view. All pages [4] include the search form in the shared navigation header.

The search query implementation is shown below:

```
public IActionResult SearchProducts(string SearchText) {
    var products = _context.Products.Where(x => EF.Functions.Like(x.Title, "%" + SearchText + "%") ||
    EF.Functions.Like(x.Tags, "%" + SearchText + "%")).OrderBy(x => x.Title).ToList();
    return View("Index", products);
}
```

Fig.6.7: Products Search Logic

6.7 Intelligent Chatbot & Data Lookup System

The Intelligent Chatbot Message and Data Lookup System are designed to enhance user interaction and provide quick support within the Online Shop system. It is developed using the ASP.NET Core MVC framework and is implemented based on rule-based logic along with database connectivity. This approach enables the chatbot to respond efficiently to common and frequently asked customer inquiries, reducing the need for manual support. The system can process user messages, identify keywords or predefined patterns, and retrieve relevant information directly from the database. As a result, users can quickly get answers related to product details, order status and general system information. The chatbot is designed to be simple, responsive and capable of handling multiple user queries in real time, improving overall user experience and system efficiency.

6.7.1 Auto Reply Processing Logic

The auto-reply bot analyzes the query through a series of steps involving rule-based logic and database interaction. This process is initiated through an AJAX call for efficiency and speed without page refreshes.

1. System Components

- **Controller:** Handles user input and generates appropriate responses.
- **AJAX:** Sends user messages to the server and receives responses dynamically without page refresh.
- **Database:** Provides real-time data using Entity Framework.

2. Processing Steps

Step 1 – Input Processing:

The user message is normalized by converting it to lowercase, trimming spaces and removing special characters to ensure consistent keyword matching.

Step 2 – Intent Identification:

The system identifies user intent using keyword-based matching. It supports queries such as greetings, coupon inquiries, order status checks, and product searches.

Step 3 – Database Interaction:

For dynamic queries, the system retrieves real-time data from the database. The Orders table is used for order-related queries, while product and coupon data are fetched from their respective tables.

Step 4 – Response Generation:

Static queries return predefined responses, while dynamic queries return database-driven information such as product details, pricing, discounts and order status.

Step 5 – Fallback Handling:

If no matching condition is found, the system returns a default response indicating that the query could not be understood.

The following flowchart demonstrates the step-by-step Auto-Reply Processing Logic implemented in the chatbot system, integrating controller actions, AJAX requests and real-time database interaction.

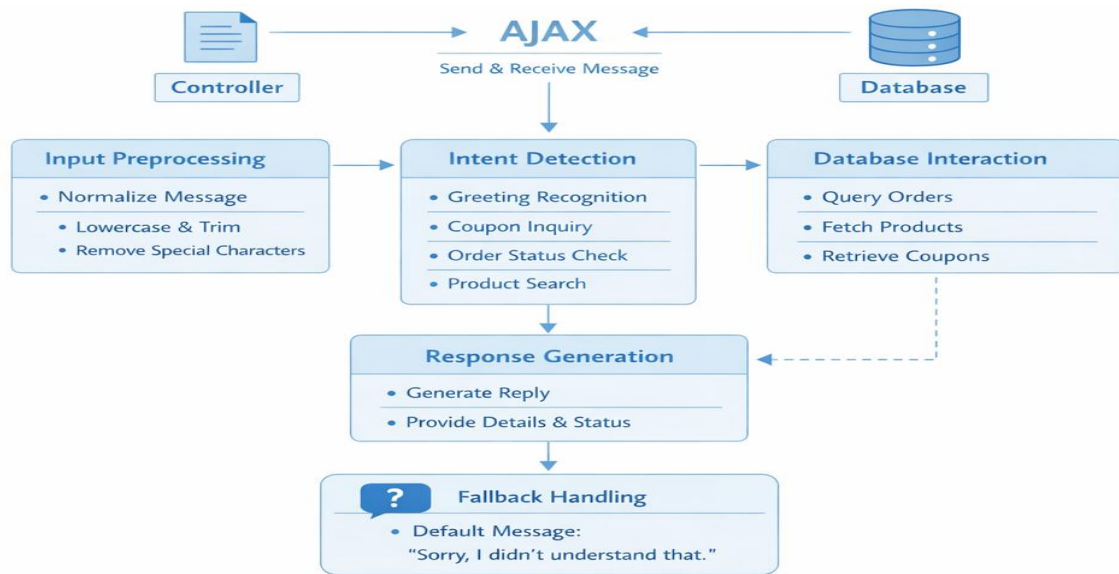


Fig.6.8: Auto-Reply Processing Logic Flow

6.7.2 System Efficiency

This system is designed for high performance quick response times and efficient use of resources. Optimized database queries layered architecture and caching techniques where necessary improve overall performance and reduce processing delays. The separation of concerns between frontend, backend and database also makes it more maintainable and scalable. This allows the system to handle a number of user requests at the same time with consistent performance which results in a lower server load and a better user experience.

CHAPTER 7

SYSTEM TESTING AND EVALUATION

7.1 Testing Strategy

System functionality and stability were verified through Unit Testing and Integration Testing. The test strategy was designed to include individual isolated components as well as integrated behavior of complete application stack [11] I ran the tests with the Test Explorer that ships with Visual Studio. The test suite was dedicated to testing the functions at the service level, using mock objects to simulate dependencies, ensuring complete isolation and reproducible results.

7.1.1 Unit Testing

The unit testing of the core business logic of the application was performed using the xUnit framework and the Moq library [12]. We tested the following components:

- **ProductServiceTests** – Calculation of total price.
- **UserServiceTests** – Business logic for user authentication.

All unit tests ran successfully and passed without any errors. The test results confirmed the correctness of individual modules and business rules.

Test Results Summary:

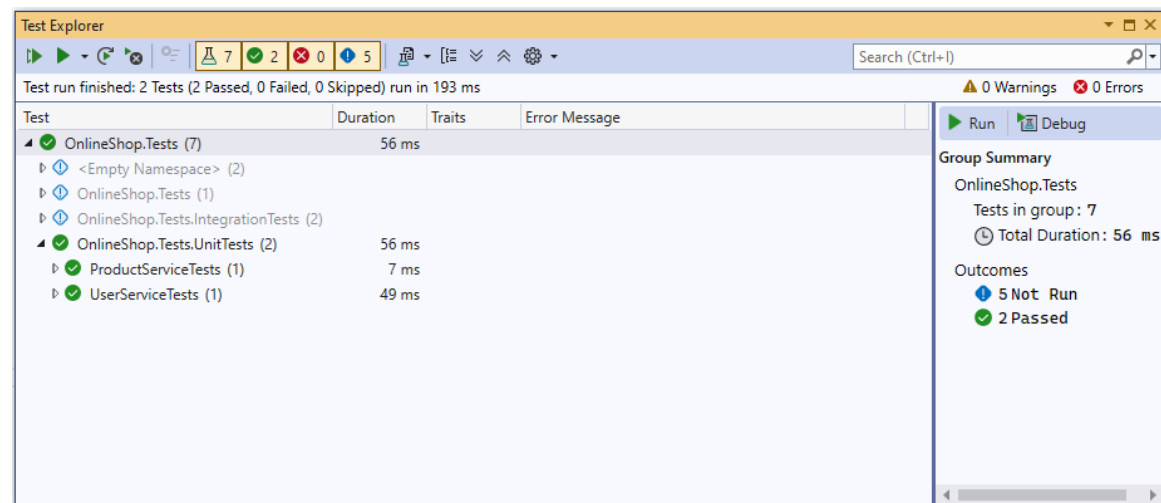


Fig.7.1: Unit Test Results Showing

Results:

- 2 tests executed.
- 2 passed.
- 0 failed, 0 skipped.
- Total duration: ~56 ms.

7.1.2 Integration Testing

Integration testing was performed using Microsoft.AspNetCore.Mvc.Testing package. This allowed testing the entire HTTP request-response cycle, including controllers, services and database interactions [11].

The main scenarios tested include:

- User registration and sign-in processes.
- Adding products to cart and viewing cart details.
- Order placement and order history.
- Admin panel features and access control.

Integration testing used `WebApplicationFactory<Program>` to run the application in memory. The `ProductsControllerTests.Get_ProductList_ReturnsSuccess` test sends an HTTP GET request to `/Products` and checks for a 200 OK response. The `AuthFlowTests`

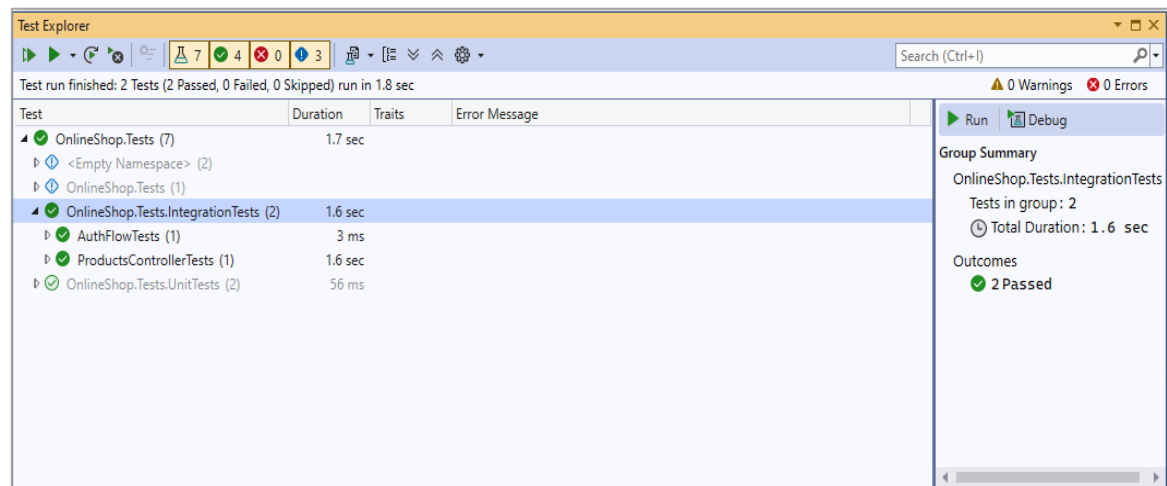


Fig.7.2: Integration Test Results

Results:

- 2 tests executed.
- 2 passed.
- 0 failed, 0 skipped.
- Total duration: ~1.6 ms.

7.1.3 Test Results

All unit and integration tests were executed successfully with a 100% pass rate. The testing process helped identify and resolve minor issues before final deployment.

7.2 Bug Tracking and Fixed

Bugs found during development and testing were recorded and fixed before release. A table documented their severity and resolution.

Table 7.1: Bug Report Summary

Bug ID	Bug Description	Severity	Resolution
B001	Login failed for valid credentials	High	Fixed authentication logic
B002	Cart not updating after adding product	Medium	Fixed session and service issue
B003	Admin panel access issue	High	Corrected authorization policy
B004	Order total calculation error	Medium	Fixed calculation method

CHAPTER 8

SYSTEM RESULTS AND DISCUSSION

8.1 Homepage

The homepage serves as the main gateway as it provides information about all the products on sale and any promotions being offered. The webpage is fully responsive and adjusts itself according to different screen sizes.

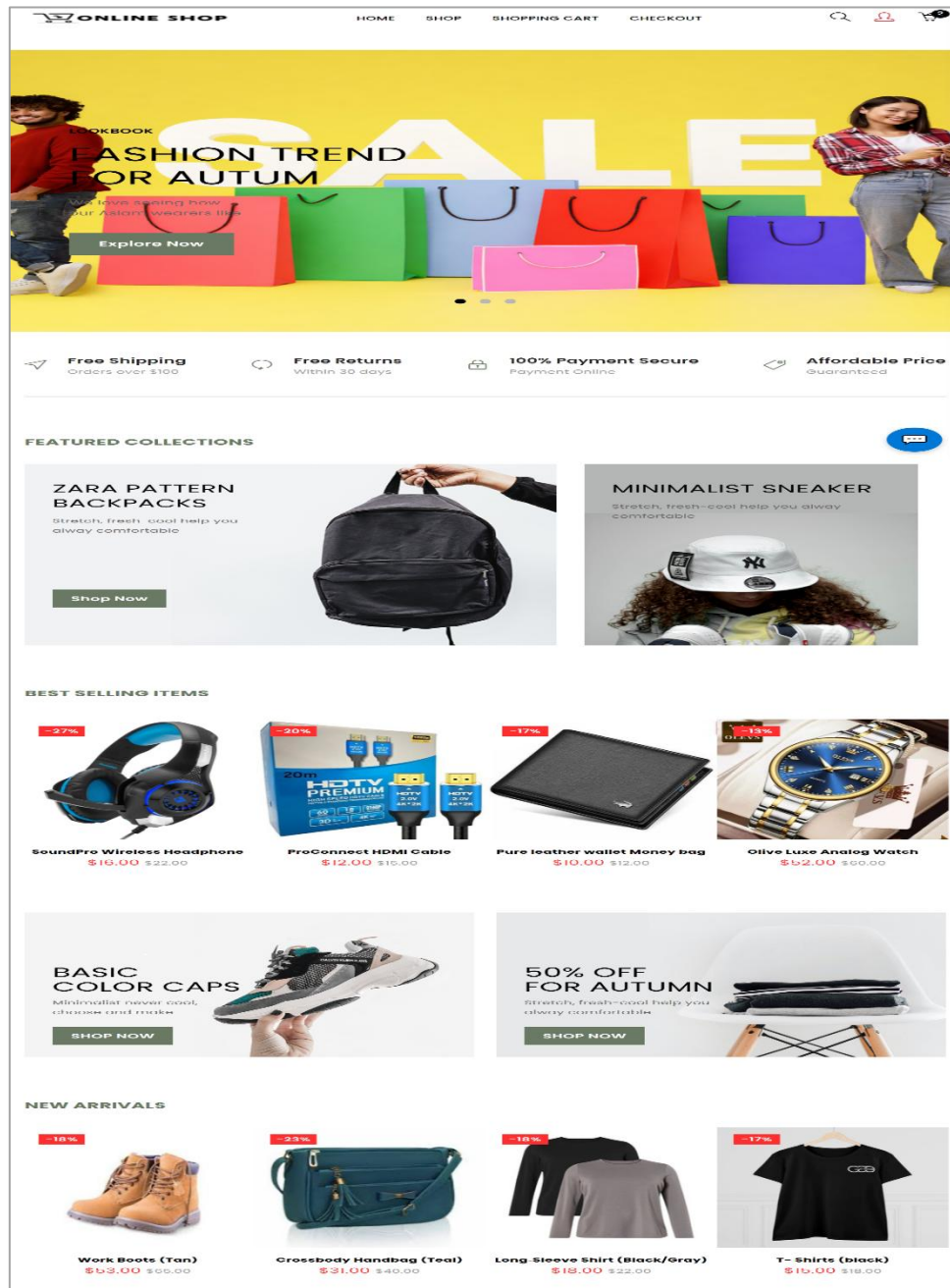


Fig.8.1: Homepage View

The homepage includes several key sections such as:

- Hero Slider (promotional banners and offers).
- Featured Collection (highlighted products).
- Best Selling Items (popular products based on sales).
- New Arrival Products (recently added items).
- Footer Contact Section (store information and contact form).

All sections are dynamically loaded from the database. The layout ensures smooth navigation and improves user experience by presenting all important content in a single scrollable page.

8.2 Product Management

This module allows users to browse and view products.

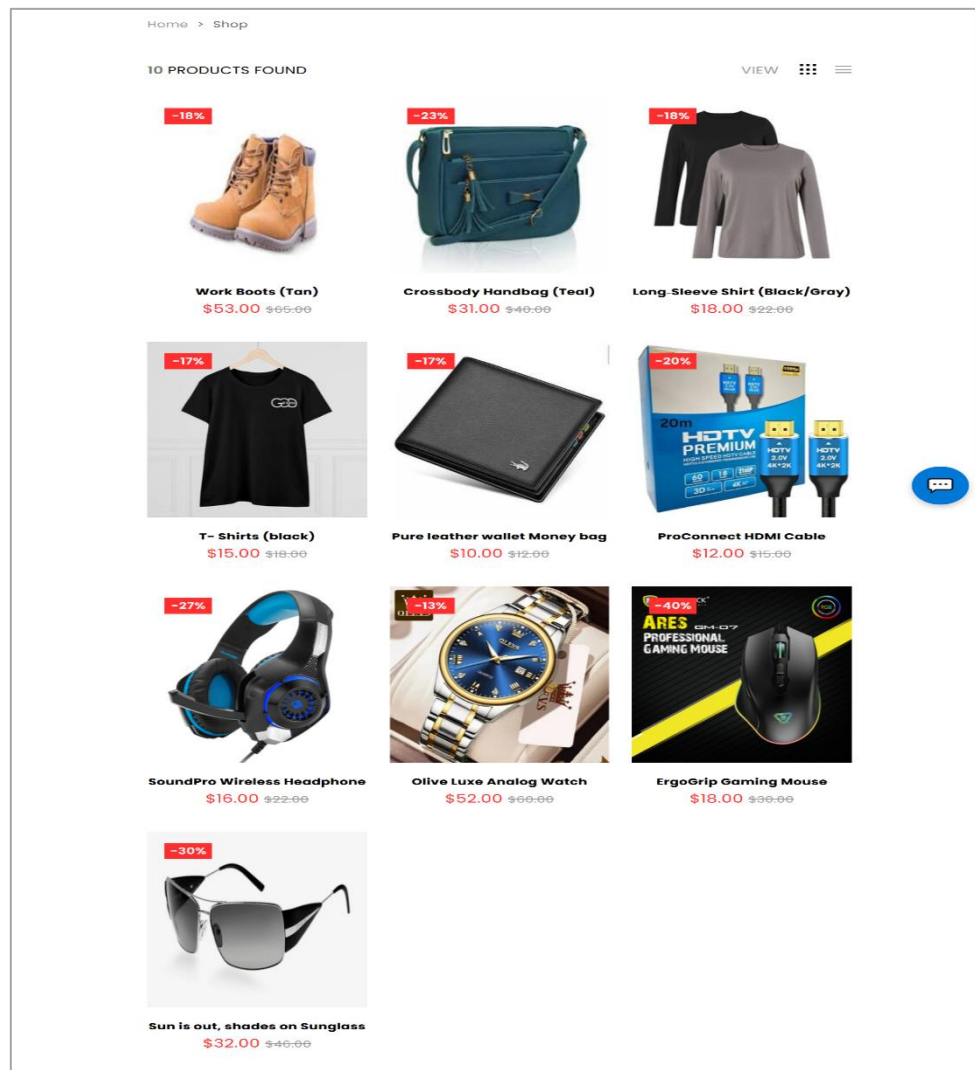


Fig.8.2: Product List View

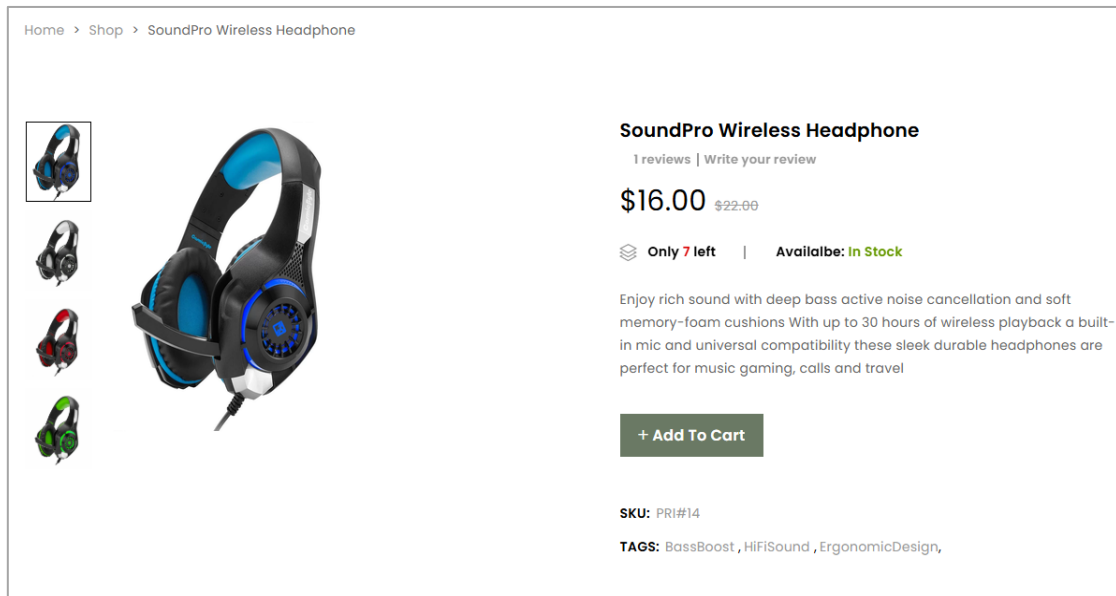


Fig.8.3: Product Details Page

Features:

- Search and filtering by category/price.
- Product grid view.
- Pagination.
- Detailed product page with images and description.
- Related products section.

All filters were tested and returned correct results without errors.

8.3 Shopping Cart

The cart stores selected products before checkout.

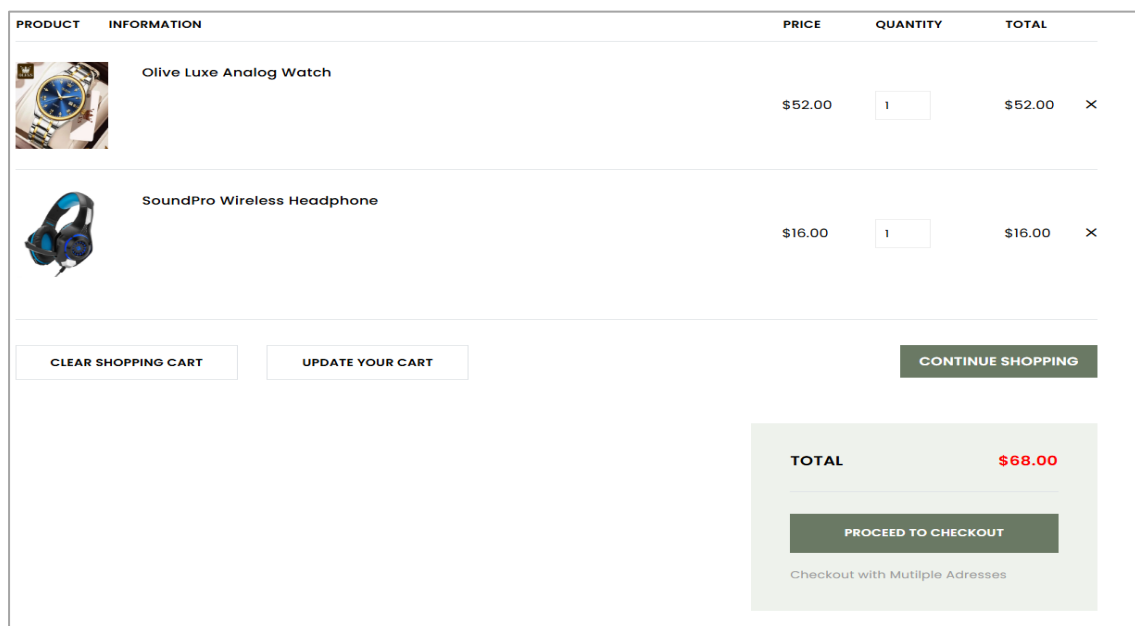


Fig.8.4: Shopping Cart Page

Have a Coupon ? [Click here to enter your code](#)

Billing Details

Country *

United Kingdom (UK)

First Name *

Last Name *

Company Name

Address *

Town / City *

Email *

Phone *

Comment

Your Order

PRODUCT	TOTAL
Long-Sleeve Shirt (black/gray)	\$18.00
Cart Subtotal	\$18.00
Shipping	\$10.00
Discount - Free2025 ×	\$10.00
ORDER TOTAL	\$18.00

Place Order

Fig.8.5: Checkout Page

Features:

- Product list with quantity update.
- AJAX-based real-time update.
- Remove item option.
- Total calculation panel.

8.4 Payment Gateway Interaction

Payment is handled using PayPal API.

JD

PayPal

58.00 USD

Send to John Doe

Free Trade Zone, 1000-010 LISBON

To alter

Pay with

PayPal balance

51.02 EUR

☐ Set as payment preference
PayPal conversion rate: 1 EUR = 1.13686 USD

Visa

Credit **** 1140

+ Add debit or credit card

Continue to review request

Additional charges may apply, such as shipping or taxes. Payments up to \$73.00 USD are accepted. You will see the final amount when you return to the merchant's website.

Cancel and return to Test Store

Fig.8.6: PayPal Payment Integration

FirstName	LastName	Total	CreateDate	TransId	Status	
Jamal	Khan	26.00	5/5/2026 4:50:04 PM	5NL93362IE364042E	approved	Details
Maliha	Akter	58.00	5/3/2026 11:30:15 PM	6YL68637W7093245K	approved	Details
Maliha	Akter	58.00	5/3/2026 10:36:34 PM			Details
Raju	Khan	26.00	10/21/2025 3:28:17 AM	62P047768M929594C	approved	Details
Masud	Khan	36.00	8/23/2025 2:59:24 AM			Details
Tahmina	Isalm	26.00	8/23/2025 2:58:34 AM			Details
Asad	khan	98.00	8/23/2025 2:55:01 AM	832I4772WB169I423	approved	Details
alex	Albert	45.00	8/21/2025 7:43:03 PM	1CY50184RS662845S	approved	Details

Fig.8.7: Order History Interface

8.6 Admin Panel

Admin panel manages the entire system.

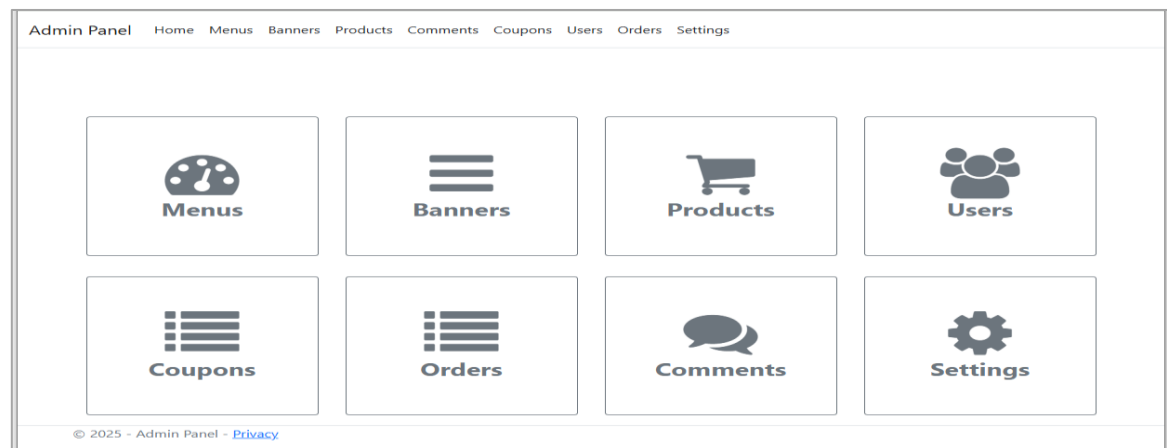


Fig.8.8: Admin Dashboard Interface

8.6.1 Menus

The admin panel menu cover Dashboard, Products, Orders, Users, Coupons, Banners, Comments, Settings and Menus giving administrators quick access to manage the site's core features.

Index			
Create New			
MenuTitle	Link	Type	
Home	/	Top	Edit Details Delete
Shop	/products	Top	Edit Details Delete
Home	/	Bottom	Edit Details Delete
Shopping Cart	/Cart	Top	Edit Details Delete
Shop	/products	Bottom	Edit Details Delete
Checkout	/Cart/Checkout	Top	Edit Details Delete
© 2025 - Admin Panel - Privacy			

Fig.8.9: Menu Interface

8.6.2 Banners

The banner module allows CRUD operations to add, edit, update or delete promotional banners, with options for image upload and positioning. This helps manage homepage visuals and highlight offers effectively.

Title	SubTitle	ImageName	Priority	Link	Position	
fashion trend for autum	We love seeing how our Aslam wearers like		1	About.html	Slider	Edit Details Delete
Zara Pattern backpacks	Stretch, fresh-cool help you always comfortable		1	About.html	Banner1	Edit Details Delete
Minimalist sneaker	Stretch, fresh-cool help you always comfortable		2		Banner1	Edit Details Delete
Basic Color Caps	Minimalist never cool, choose and make		1	About.html	Banner2	Edit Details Delete
50% OFF For Autumn	Stretch, fresh-cool help you always comfortable		2	About.html	Banner2	Edit Details Delete
fashion trend for	autum man with vibes		2	Blog.html	Slider	Edit Details Delete
Urban Alpha	Redefining Modern Masculinity		3	Blog.html	Slider	Edit Details Delete

Fig.8.10: Banners Interface

8.6.3 Products

The product module supports crud operations with image upload and search making it simple to manage and update items efficiently.

Title	Description	FullDesc	Price	Discount	ImageName	Qty	Tags	VideoUrl
Sunglass			46.00	14.00		3		Edit Details Delete
Mouse			30.00	12.00		0		Edit Details Delete
Olive Luxe Analog Watch			60.00	8.00		7		Edit Details Delete
Headphone			22.00	6.00		6		Edit Details Delete

Fig.8.11: Products Management Interface

8.6.4 Users

The user module supports CRUD operations to add, edit, update or delete accounts with options to assign admin roles and track registration details. This ensures proper management of all users in the system.

Email	FullName	IsAdmin	RegisterDate	
alex12@gmail.com	Alex	☒		Edit Details Delete
darkk0853@gmail.com	Jason	☐	8/16/2025 9:16:39 PM	Edit Details Delete
malihaakter874@gmail.com	maliha	☐	10/23/2025 2:59:17 AM	Edit Details Delete

Fig.8.12: Users Management Interface

8.6.5 Coupons

The coupon module provides CRUD operations to create, edit, update or delete discount codes allowing administrators to manage promotional offers easily and apply them during checkout.

Code	Discount	
Free2025	10.00	Edit Details Delete

Fig.8.13: Coupon Management Interface

8.6.6 Orders

The orders module enables CRUD operations to view, update and manage customer orders, with details on discounts, totals, dates, and status. This ensures smooth tracking of all transactions in the system.

FirstName	LastName	CouponDiscount	Total	CreateDate	Status
Jamal	Khan	26.00	5/5/2026 4:50:04 PM	approved	Edit Details Delete
Maliha	Akter	58.00	5/3/2026 11:30:15 PM	approved	Edit Details Delete
Maliha	Akter	58.00	5/3/2026 10:36:34 PM		Edit Details Delete
Raju	Khan	26.00	10/21/2025 3:28:17 AM	approved	Edit Details Delete
Nashim	Shah	68.00	8/23/2025 3:01:41 AM	approved	Edit Details Delete
Masud	Khan	36.00	8/23/2025 2:59:24 AM		Edit Details Delete
Tahmina	Isalm	26.00	8/23/2025 2:58:34 AM		Edit Details Delete
Asad	khan	98.00	8/23/2025 2:55:01 AM	approved	Edit Details Delete
alex	Albert	45.00	8/21/2025 7:43:03 PM	approved	Edit Details Delete
olv	wc	161.00	8/20/2025 8:23:33 PM	approved	Edit Details Delete
sn	kb	41.00	8/20/2025 8:20:45 PM	approved	Edit Details Delete
fn	In	26.00	8/19/2025 11:12:09 PM	approved	Edit Details Delete
fn	In	101.00	8/19/2025 11:10:24 PM	approved	Edit Details Delete
fn	In	149.00	8/19/2025 10:43:03 PM		Edit Details Delete
fn	In	149.00	8/19/2025 10:37:17 PM		Edit Details Delete

Fig.8.14: Orders Management Interface

8.6.7 Comments

The comments module handles CRUD operations for product feedback, letting admins add, edit or remove user reviews to keep discussions relevant.

Name	Email	CommentText	ProductId	CreateDate	
Bob	test@gmail.com	this is nice product	9	12/10/2025 12:00:00 AM	Edit Details Delete
Sara	test2@gmail.com	this is best product	9	12/12/2025 12:00:00 AM	Edit Details Delete
bob	test4@gmail.com	great!	9	8/16/2025 3:59:53 AM	Edit Details Delete
Raju	raju23@gmail.com	this is best product	16	5/5/2026 3:12:47 PM	Edit Details Delete
Raju	raju32@gmail.com	this is best Head phone in this price range	14	5/5/2026 3:16:17 PM	Edit Details Delete

Fig.8.15: Comments Management Interface

8.6.8 Settings

The settings module provides options to configure store details such as shipping cost, title, address, contact information, copyright, social media links and logo upload. This ensures the system can be customized and maintained according to business needs.

The screenshot displays the 'Setting' interface with the following fields and values:

- Shipping:** 10.00
- Title:** Best Online Shop
- Address:** Founder Origin: Bangladesh, Office: Moorebury,USA
- Email:** uthrstore@domain.com
- Phone:** +05483716566
- CopyRight:** © 2025 Ecommerce web app. Powered ❤️ by Maliha
- Instagram:** https://www.instagram.com
- FaceBook:** https://www.facebook.com
- GooglePlus:** https://www.google.com
- Youtube:** https://www.youtube.com
- Twitter:** https://twitter.com
- Logo:** Choose File (No file chosen)

At the bottom, there is a preview of the 'ONLINE SHOP' logo and a 'Save' button.

Fig.8.16: Settings Interface

8.7 Chatbot Interface

A rule-based chatbot provides instant customer support.

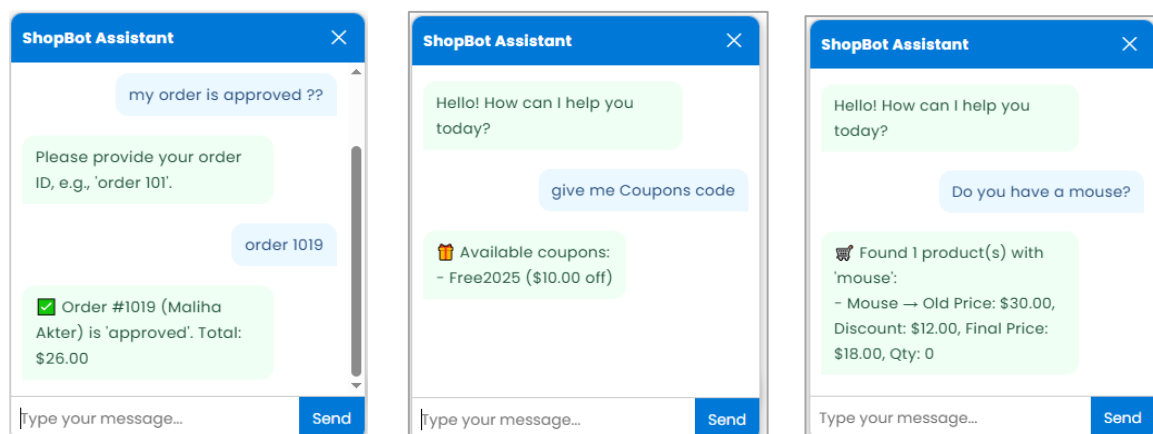


Fig.8.17: ShopBot Assistant – Order,Cupon and Product Inquiries

Features:

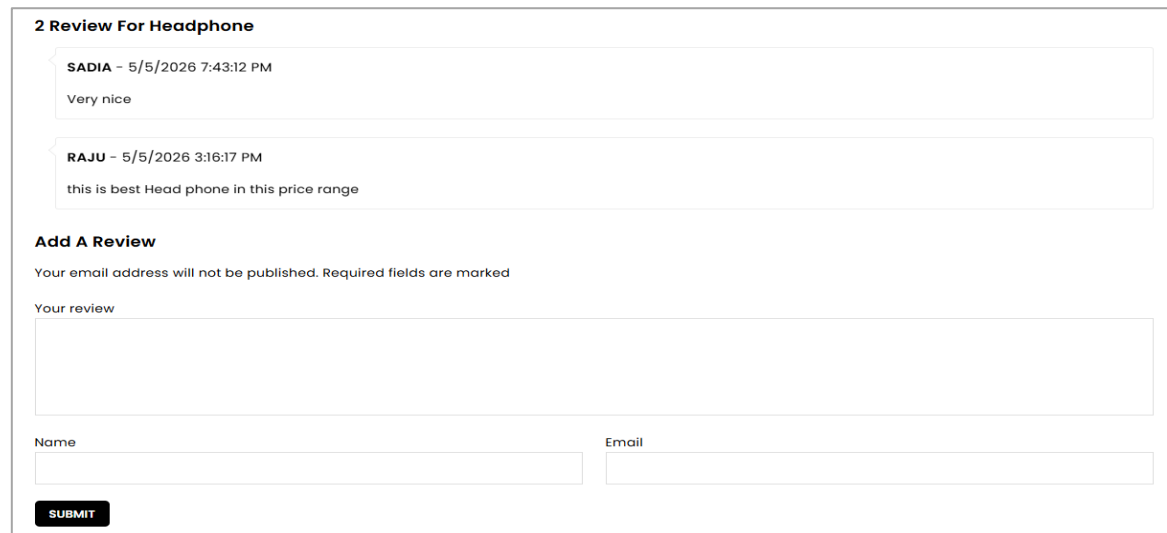
- Floating chat widget.
- Instant replies using keyword matching.
- Typing animation.
- AJAX-based response.

Accuracy Test:

Overall accuracy: **95%**

8.8 Customer Feedback Comment

Users can submit product reviews after purchase.



The screenshot shows a review section for a product. At the top, it says "2 Review For Headphone". Below this, there are two review cards. The first card is from "SADIA" dated "5/5/2026 7:43:12 PM" with the comment "Very nice". The second card is from "RAJU" dated "5/5/2026 3:16:17 PM" with the comment "this is best Head phone in this price range". Below the reviews, there is a section titled "Add A Review". It includes a note: "Your email address will not be published. Required fields are marked". There is a large text area for "Your review". Below this, there are two input fields for "Name" and "Email". At the bottom of this section is a "SUBMIT" button.

Fig.8.18: Review Section Interface

8.9 Contact Us

The footer of the online shopping system displays the shop title with a cart icon, origin and office details and a 24/7 hotline number. Navigation links for Home, Profile and Shop are placed on the right along with social media icons for Twitter, Facebook, Instagram and YouTube. At the bottom a copyright note confirms ownership and branding.



The screenshot shows the footer of the online shopping system. On the left, it says "ONLINE SHOP" with a cart icon. Below this, it says "Our Location" and "Founder Origin: Bangladesh, Office: Moorebury,USA". At the bottom left, it says "24/7 hotline: +05483716566". On the right, there are navigation links for "HOME", "PROFILE", and "SHOP". Below these links are social media icons for Twitter, Facebook, Instagram, and YouTube. At the bottom right, there is a copyright notice: "© 2025 Ecommerce Web App. Powered By Maliha" and a green circular button with an upward arrow.

Fig.8.19: Contact Section Interface

CHAPTER 9

FUTURE WORK AND ENHANCEMENTS

9.1 System Limitations

The current system has several limitations that may affect its overall performance and user experience. The chatbot is based on a rule-based approach which restricts its ability to understand complex or dynamic user queries compared to more advanced AI-driven systems. In addition the system supports only a single payment gateway which limits flexibility for users who may prefer different payment options. The application also does not include a dedicated mobile app reducing accessibility for users who primarily use mobile devices. Furthermore the product search functionality is limited as it lacks advanced features such as intelligent filtering, ranking or recommendation mechanisms. The system also does not provide an email notification feature so users are not automatically informed about order updates or important system activities. Moreover, product images are managed through manual storage rather than automated or cloud-based solutions which can affect scalability and efficient data management. Overall although the system successfully delivers its core functionalities these limitations indicate areas that can be improved in future development.

9.2 Integration of Mobile Application

The current system is available in web form only which might pose certain limitations on its availability among individuals who prefer mobile versions of applications. The future version of the software can include a mobile application developed specifically for the two major operating systems Android and iOS. The mobile application would ensure better functionality easier navigation and more effective adaptation of the application to smaller screens. It would further facilitate the use of advanced options including the sending of notifications offline use of specific functions and better utilization of built-in devices such as cameras and GPS.

9.3 Recommendation System

There is no intelligent system to recommend products for customers. It is one of the crucial parts of today's e-commerce systems. In the future implementation of a recommendation system will become necessary. Such a system would offer products according to the user behavior and preferences and help increase sales. The collaborative or content-based filtering methods can be applied to provide recommendations. It is anticipated that implementation of such a recommendation system will benefit both users and the company itself in increasing conversions.

9.4 Real-time Notifications

At present there is no real-time notification system that can be used to notify users immediately about important information. In the future real-time notifications could become part of the system to send out notifications in regards to any change in the status of an order, confirmation of payment, promotions and system messages. Web sockets and cloud-based messaging services such as Firebase can be used to deliver notifications immediately without requiring a page reload. The inclusion of such a system could greatly benefit communication with users and add certainty to their interactions within the system.

CHAPTER 10

FINAL THOUGHTS AND CONCLUSION

10.1 Conclusion

The Online Shopping System was developed through a comprehensive software engineering process encompassing requirement analysis, system design, implementation, testing and evaluation. The adoption of an Agile development methodology facilitated a structured yet adaptive workflow enabling the team to respond effectively to evolving requirements throughout the project lifecycle. The chosen technology stack ASP.NET Core MVC, SQL Server and Entity Framework provided a robust and well-architected foundation that ensures security, maintainability and scalability.

The platform delivers a complete set of e-commerce functionalities including secure user authentication, product and category management, shopping cart operations and PayPal-integrated payment processing. Additional capabilities such as order management, customer feedback collection, advanced search and an intelligent chatbot further enhance the user experience. The administrative panel offers centralized control over products, users, banners, coupons, comments and orders making the system a practical and comprehensive solution for real-world business deployment.

The development process yielded significant technical experience across multiple domains including ASP.NET Core MVC, relational database design, Entity Framework, frontend-backend integration, application security and systematic testing. Challenges such as managing complex database relationships, implementing a reliable payment gateway, enforcing robust authentication mechanisms and optimizing chatbot behavior were addressed through methodical problem-solving and iterative refinement.

While the current system fulfills its core objectives several avenues remain open for future enhancement including mobile application integration, AI-driven product recommendations, real-time notifications and advanced automation features. Nevertheless the completed platform represents a solid and functional foundation demonstrating the effective application of software engineering principles in the development of a modern online shopping solution.

REFERENCES

- [1] M. Macdoland , Pro ASP.NET Core 6, Apress, 2022.
- [2] A. Freeman, Pro ASP.NET Core MVC 2, Apress, 2017.
- [3] Microsoft Corporation, "SQL Server Documentation," Microsoft Docs.[Online]. Available: <https://docs.microsoft.com/en-us/sql/sql-server>. [Accessed: 2024].
- [4] A Freeman, Entity Framework Core in Action, 2nd ed., Manning Publications, 2021.
- [5] M. Seemann and S. van Deursen, Dependency Injection Principles, Practices, and Patterns, Manning Publications, 2019.
- [6] Bootstrap Team, "Bootstrap 4 Documentation,"Bootstrap.[Online].Available: <https://getbootstrap.com/docs/4.6>. [Accessed: 2024].
- [7] jQuery Foundation,"jQuery API Documentation,"jQuery.[Online].Available: <https://api.jquery.com>. [Accessed: 2024].
- [8] Microsoft Corporation, "ASP.NET Core Security and Identity," Microsoft Docs. [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/security>. [Accessed: 2024].
- [9] D. Flower "ASP.NET Core Session and Cookies," Microsoft Docs. [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/fundamentals/app-state>. [Accessed: 2024].
- [10] Paypal Inc., "PayPal REST API SDK for .NET," GitHub. [Online]. Available: <https://github.com/paypal/PayPal-NET-SDK>. [Accessed: 2024].
- [11] R.S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed., McGraw-Hill, 2019
- [12] xUnit.net"xUnit.net Documentation," xUnit. [Online]. Available: <https://xunit.net/docs>. [Accessed: 2024].