**EASTERN INTERNATIONAL UNIVERSITY**

**SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY**

**DEPARTMENT OF SOFTWARE ENGINEERING**



**PROJECT 1 REPORT**

**ANNA COSMETIC WEBSITE**

**Students**

Phan Ngoc Hanh Nhi– 2131209002

Tran Mai Anh – 2231200062

**Supervisor**

Ung Van Giau

**Binh Duong, December 2024**

ABSTRACT

ANNA Cosmetic is an innovative web application designed to transform the online beauty shopping experience by providing a seamless and personalized platform for users. By centralizing product information and customer preferences, the platform simplifies the discovery and purchase of cosmetics while ensuring a user-friendly and enjoyable experience.

For customers, ANNA Cosmetic offers intuitive navigation, detailed product descriptions, personalized recommendations, and secure transactions, making it easy to find products that meet their unique beauty needs. The platform also includes educational resources, empowering users with knowledge about skincare and makeup trends to make informed purchasing decisions.

From a business perspective, ANNA Cosmetic streamlines inventory management, optimizes customer interactions, and provides valuable insights into consumer behavior. With its commitment to innovation and excellence, the platform enhances customer satisfaction while fostering long-term brand loyalty.

With a sleek design and robust functionality, ANNA Cosmetic represents a significant step forward in e-commerce, redefining the way beauty enthusiasts connect with their favorite products and brands. It serves as a comprehensive solution for modern beauty needs, blending technology with the art of self-expression.

ACKNOWLEDGEMENT

We extend our deepest gratitude to our college faculty and project guide, Mr. Ung Van Giau, for his invaluable support and guidance throughout the development of the ANNA Cosmetic website. His profound expertise, insightful feedback, and consistent encouragement have been instrumental in steering our project towards its successful completion.

Additionally, we would like to acknowledge the numerous professionals and developers who have shared their knowledge through online tutorials, forums, and articles. Their contributions have significantly enriched our understanding of web technologies and e-commerce development, empowering us to create a robust and user-friendly platform. This collective wisdom has played a pivotal role in shaping ANNA Cosmetic into a dynamic solution for the modern beauty market.

TABLE OF CONTENTS

[ABSTRACT i](#_Toc186714535)

[ACKNOWLEDGEMENT ii](#_Toc186714536)

[LIST OF FIGURES v](#_Toc186714537)

[LIST OF TABLES v](#_Toc186714538)

[LIST OF ABBREVIATIONS vi](#_Toc186714539)

[CHAPTER 1. OVERVIEW 1](#_Toc186714540)

[1.1. Introduction 1](#_Toc186714541)

[1.2. Project objectives 1](#_Toc186714542)

[1.3. Challenges 2](#_Toc186714543)

[1.4. Report structure 3](#_Toc186714544)

[CHAPTER 2. INTRODUCTION TO TECHNOLOGIES 4](#_Toc186714545)

[2.1. HTML 4](#_Toc186714546)

[2.2. CSS 5](#_Toc186714547)

[2.3. JavaScript 6](#_Toc186714548)

[2.4. ASP.NET Core 7](#_Toc186714549)

[2.5. MySQL 8](#_Toc186714550)

[2.6. GitHub 9](#_Toc186714551)

[CHAPTER 3. APPLICATION ANALYSIS, DESIGN AND IMPLEMENTATION 11](#_Toc186714552)

[3.1. Requirements 11](#_Toc186714553)

[3.1.1. User Groups 11](#_Toc186714554)

[3.1.2. Core Features 11](#_Toc186714555)

[3.1.3. Use Case Diagram 11](#_Toc186714556)

[3.1.4. Scope 12](#_Toc186714557)

[3.2. Use case diagram 12](#_Toc186714558)

[3.2.1. Admin use case diagram 12](#_Toc186714559)

[3.2.2. Customer use case diagram 14](#_Toc186714560)

[3.3. Database 15](#_Toc186714561)

[3.3.1. ERM diagram 15](#_Toc186714562)

[3.3.2. Table in database 15](#_Toc186714563)

[3.4. Introduction to ASP.NET Core 17](#_Toc186714564)

[3.4.1. Installation 17](#_Toc186714565)

[3.4.2. ASP.NET Core Project Structure 17](#_Toc186714566)

[3.4.2.1. Root Directory 17](#_Toc186714567)

[3.4.2.2. Folders 17](#_Toc186714568)

[3.4.2.3. Key Files 18](#_Toc186714569)

[3.4.2.4. Creating and Initializing the Project 18](#_Toc186714570)

[3.4.2.5. Configuring ASP.NET Core 18](#_Toc186714571)

[3.4.2.6. Running the Application 19](#_Toc186714572)

[3.4.2.7. Advanced Configurations 19](#_Toc186714573)

[3.5. CRUD for Admin 20](#_Toc186714574)

[3.5.1. Create new category 20](#_Toc186714575)

[3.5.2. Read category 21](#_Toc186714576)

[3.5.3. Update category 22](#_Toc186714577)

[3.5.4. Delete category 24](#_Toc186714578)

[CHAPTER 4. EXPERIMENT AND DISCUSSION 26](#_Toc186714579)

[4.1. Installation Environment 26](#_Toc186714580)

[4.2. Results and Discussion 27](#_Toc186714581)

[4.2.1. Home page for customer 27](#_Toc186714582)

[4.2.1.1. Interface and Features Implemented 28](#_Toc186714583)

[4.2.1.2. Overall Evaluation 29](#_Toc186714584)

[4.2.2. CRUD for admin 29](#_Toc186714585)

[4.2.2.1. Create Category 30](#_Toc186714586)

[4.2.2.2. Read 30](#_Toc186714587)

[4.2.2.3. Update 31](#_Toc186714588)

[4.2.2.4. Delete 32](#_Toc186714589)

[4.2.2.5. Additional Features 33](#_Toc186714590)

[4.2.2.6. Overall Evaluation 33](#_Toc186714591)

[CHAPTER 5. CONCLUSION AND FUTURE WORKS 34](#_Toc186714592)

[5.1. Conclusion 34](#_Toc186714593)

[5.1.1. Knowledge Gained 34](#_Toc186714594)

[5.1.2. Product Outcomes 34](#_Toc186714595)

[5.1.3. Skills Developed 35](#_Toc186714596)

[5.1.4. Team Growth 35](#_Toc186714597)

[5.2. Future works 35](#_Toc186714598)

[REFERENCES 37](#_Toc186714599)

LIST OF FIGURES

[Figure 1. HTML Logo 4](#_Toc186714600)

[Figure 2. CSS Logo 5](#_Toc186714601)

[Figure 3. JavaScript Logo 6](#_Toc186714602)

[Figure 4. ASP.NET Core Logo 7](#_Toc186714603)

[Figure 5. MySQL Logo 8](#_Toc186714604)

[Figure 6. GitHub Logo 9](#_Toc186714605)

[Figure 7. Admin use case diagram 12](#_Toc186714606)

[Figure 8. Customer use case diagram 14](#_Toc186714607)

[Figure 9. ERM diagram 15](#_Toc186714608)

[Figure 10.Create activity diagram 20](#_Toc186714609)

[Figure 11.Read activity diagram 21](#_Toc186714610)

[Figure 12.Edit activity diagram 22](#_Toc186714611)

[Figure 13.Delete activity diagram 24](#_Toc186714612)

[Figure 14.Home page 27](#_Toc186714613)

[Figure 15.Create category interface 30](#_Toc186714614)

[Figure 16.Read category interface 30](#_Toc186714615)

[Figure 17.Update(Edit) category interface 31](#_Toc186714616)

[Figure 18.Delete category interface 32](#_Toc186714617)

[Figure 19.Delete category successfully interface 32](#_Toc186714618)

LIST OF TABLES

[Table 1. Table Customer 15](#_Toc186712207)

[Table 2. Table Product 16](#_Toc186712208)

[Table 3. Table Order 16](#_Toc186712209)

[Table 4. Table OrderDetail 16](#_Toc186712210)

[Table 5. Table Admin 17](#_Toc186712211)

LIST OF ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| **No.** | **Term** | **Meaning** |
| 1 | HTML | HyperText Markup Language |
| 2 | CSS | Cascading Style Sheets |
| 3 | SQL | Structured Query Language |
| 4 | API | Application Programming Interface |
| 5 | HTTPS | Hypertext Transfer Protocol Secure |
| 6 | ASP | Active Server Pages |
| 7 | UI | User Interface |
| 8 | CRUD | Create, Read, Update and Delete |

# OVERVIEW

This chapter provides an overview of the ANNA Cosmetic Website project, highlighting its purpose, objectives, challenges encountered during development, and the structure of this report.

## Introduction

Online shopping in the fast-growing digital world has become an integral feature of modern consumer behavior, therefore, they have changed the retail scenario for almost all industries. The cosmetic industry, which has shown strong growth and a wide consumer base, is also becoming part of this transition by implementing e-commerce solutions to reach consumers with their ever-changing preferences. Online channels are increasingly preferred over traditional retailing models due to the convenience and variety of products that customers can access from their comfort zone.

The “Cosmetics E-commerce Website” project aims to target the creation of a Web-based platform for browsing, selecting, and purchasing high-quality cosmetic products. Apart from being a regular e-commerce website, it will also engage customers in product reviews, identification of new products, and promotion campaigns, thereby improving their shopping experience. This translates to developing an intuitive and secure online shopping experience to meet the needs of today’s digitally skilled consumers. The main features to be included in the platform are an advanced shopping cart, online payment security, and product reviews. The design of the website will also ensure responsiveness for user experiences on both desktop and mobile devices.

The admin panel is a critical component of the system, serving as the central hub for managing the website’s operations. Therefore, this admin panel shall provide a means for administrators to inventory products, process orders, manage customer accounts, and view sales analytics with the intent of smooth system performance and efficiency in business. The technology stack used in this project is .NET, which is a very powerful, flexible framework that scales well and is secure and robust. The choice of use ensures the system will be able to handle increasing traffic and complex functionalities while ensuring data security at high levels. The website will also be integrated with modern web technologies for efficient data management and seamless integration with third-party services like payment gateways.

## Project objectives

The primary objective of this project is to design and develop a foundational web application for Anna Cosmetics, focusing on creating an intuitive user interface for customers and an administrative dashboard for backend management. This project serves as the first phase of a larger initiative and aims to establish a solid foundation for subsequent development.

**Specific Goals:**

* **User Interface (UI) Design for End Users**

The project prioritizes creating a visually appealing and user-friendly interface that enables customers to browse products effortlessly. The design will emphasize simplicity, usability, and responsiveness to ensure accessibility across various devices, enhancing the shopping experience.

* **Admin Panel Development**

For administrative operations, a separate dashboard will be developed to allow streamlined management of the website’s content. Admins will be able to view, edit, and manage key data efficiently, ensuring smooth backend functionality.

* **Implementation of Basic CRUD Features**

The project will focus on implementing Create, Read, Update, and Delete (CRUD) functionalities. These features will provide the essential framework for managing product data, categories, and user accounts. The scope includes building robust and secure operations that are easy to maintain.

* **Exploration of Web Development Techniques**

This phase will also involve researching and applying appropriate tools, frameworks, and technologies for web development. Emphasis will be placed on gaining practical knowledge and experience in structuring a project effectively for scalability and maintainability.

* **Limitations and Scope:**

This project is limited to the development of basic functionalities and design. Advanced features such as order management, payment gateways, and analytics will not be included at this stage. These will be addressed in a future phase (Project 2) to complete the website’s functionalities and enhance the overall user experience.

By focusing on these core objectives, this project aims to establish a foundation for a fully functional and scalable e-commerce platform, ensuring smooth progression into the subsequent development phases.

## Challenges

While developing the ANNA Cosmetic Website, my team faced a lot of challenges that required hefty, hard-won efforts to overcome them. One major challenge was getting used to ASP.NET Core, which was new to the entire team. It was a daunting task to get accustomed to the framework's syntax, tools, and development practices within such a short period. The steep learning curve necessitated an extra time investment for the team in studying the framework, playing around with its features, and resolving problems that cropped up during implementation.

Another challenge concerned the design and development of the admin interface, which was something entirely new for the team. This required us to learn how we could develop a functional interface along with a user-friendly interface, conforming to the overall objectives of the website. The admin interface needed to be an intuitive solution so that administrators could effectively manage inventory, orders, and customer information. Achieving simplicity while being functional demanded great attention as well as several iterations to reach the desired standards.

Such challenges ensured that the team members went out of their comfort zones, thereby enhancing technical skills and problem-solving abilities to ensure success in project completion.

## Report structure

Chapter 1: Introduce motivation, objectives and project overview.

Chapter 2: Introduce some libraries and techniques that play a vital role in our project.

Chapter 3: System analysis, design, and implementation.

Chapter 4: Result and discussion.

Chapter 5: Conclusion and future works.

# INTRODUCTION TO TECHNOLOGIES

This chapter introduces the core technologies used to develop the project. Each technology is described with its key advantages and contributions to the overall system, demonstrating how they work together to deliver a fully functional web application.

## HTML



Figure . HTML Logo

HTML (HyperText Markup Language) is the standard markup language used to create the structure and content of web pages. It consists of a series of elements, each enclosed in angle brackets, which define the different parts of a webpage's content. These elements can represent headings, paragraphs, images, links, forms, and more.[1]

**Key points**

* **Universal Compatibility:** HTML is supported by all major web browsers, ensuring that web pages built with HTML can be accessed by a wide range of users regardless of their browser or device.
* **Simple and Easy to Learn**: HTML has a straightforward syntax and requires minimal setup, making it easy for beginners to learn and understand. Its simplicity allows for rapid development of basic web pages.
* **Semantics**: HTML provides semantic elements that describe the meaning and structure of content, making web pages more accessible to users and search engines. Semantic HTML elements, such as **<header>**, **<nav>**, **<section>**, and **<footer>**, enhance the clarity and organization of web documents.
* **SEO (Search Engine Optimization)**: Semantic HTML helps improve the search engine ranking of web pages by providing search engines with clear and structured content to index. Properly structured HTML documents with meaningful tags and attributes make it easier for search engines to understand and rank the content.
* **Flexibility**: HTML can be combined with other web technologies, such as CSS for styling and JavaScript for interactivity, to create rich and dynamic web experiences. This flexibility allows developers to build a wide range of web applications, from simple static websites to complex web applications.
* **Scalability**: HTML is scalable and can accommodate the needs of various types of websites and web applications. Whether building a small personal blog or a large e-commerce platform, HTML provides the foundation for creating scalable and robust web solutions.
* **Cross-Platform Compatibility**: HTML-based web pages can be accessed on various platforms, including desktop computers, laptops, tablets, and smartphones. This cross-platform compatibility ensures that web content can reach a diverse audience across different devices and operating systems.
* **Cost-Effectiveness**: HTML development is cost-effective compared to other web development technologies. Since HTML is an open standard and requires no licensing fees, businesses and developers can create and deploy web content without incurring additional costs.

## CSS



Figure . CSS Logo

**CSS (Cascading Style Sheets)** is a language used for describing the presentation and formatting of documents written in HTML or XML. It allows the separation of content (structured with HTML) from presentation (layout and design), making it easier and more efficient to manage and modify web pages' appearance.[2]

**Key points**

* **Separation of Concerns**: CSS allows for the separation of content (HTML) from presentation (styling). This separation makes code more maintainable, improves readability, and facilitates collaboration among developers, designers, and content creators.
* **Consistent Styling**: CSS enables developers to apply consistent styling across multiple web pages by defining styles once and applying them universally. This consistency enhances the user experience and strengthens branding and identity across a website or web application.
* **Flexibility and Control**: CSS provides granular control over the styling of HTML elements, allowing developers to customize the appearance of elements precisely according to design requirements. Properties like color, font, size, spacing, and layout can be adjusted with ease.
* **Responsive Design**: CSS supports responsive web design techniques, allowing developers to create layouts that adapt to different screen sizes and devices. Media queries, flexible layout options (e.g., flexbox, grid), and viewport settings enable developers to create designs that look and function well across desktops, tablets, and smartphones.
* **Fast Loading Times**: Separating styling into external CSS files allows browsers to cache stylesheets, resulting in faster loading times for subsequent page visits. This optimization reduces bandwidth usage and improves overall website performance.
* **Accessibility**: CSS supports accessibility by allowing developers to define semantic HTML elements and apply appropriate styling to enhance readability and usability for users with disabilities. Accessible styling practices, such as high contrast, proper text sizing, and keyboard navigation support, ensure that websites are inclusive and accessible to all users.
* **Modularity and Reusability**: CSS promotes modularity and reusability through the use of classes, IDs, and reusable style rules. Developers can create a library of reusable styles that can be applied to multiple elements throughout a website, reducing redundancy and improving code maintainability.
* **Ease of Maintenance**: Centralizing styling in external CSS files makes it easier to update and maintain styles across an entire website or web application. Changes made to CSS styles are automatically applied to all HTML elements associated with those styles, simplifying the maintenance process and reducing the risk of errors.

## JavaScript

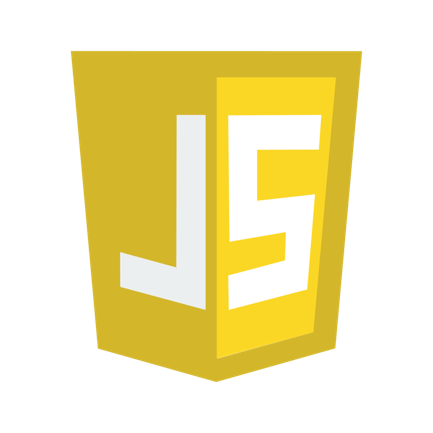
****

Figure . JavaScript Logo

JavaScript is a high-level, versatile programming language commonly used to create interactive effects within web browsers. It is an essential part of web development, alongside HTML and CSS. JavaScript allows developers to implement complex features on web pages, making them interactive and dynamic.[3]

**Key points**

* **Interpreted Language:** JavaScript is executed directly by the web browser without the need for prior compilation.
* **Event-driven:** JavaScript can respond to user actions, such as clicks, form submissions, and mouse movements.
* **Object-Oriented:** Supports object-oriented programming principles, including objects, inheritance, and polymorphism.
* **Client-Side and Server-Side:** While primarily used on the client side, JavaScript can also be executed on the server side with environments like Node.js.
* **Cross-Platform:** Runs on various devices and operating systems, making it highly adaptable.
* **Rich Ecosystem:** Extensive libraries and frameworks enhance development efficiency and capabilities.

## ASP.NET Core



Figure . ASP.NET Core Logo

ASP.NET Core is a cross-platform, open-source application framework that you can use to build dynamic web applications quickly. You can use ASP.NET Core to build server-rendered web applications, backend server applications, HTTP APIs that can be consumed by mobile applications, and much more. ASP.NET Core runs on .NET 7, which is the latest version of .NET Core—a high-performance, cross-platform, open-source runtime.

ASP.NET Core provides structure, helper functions, and a framework for building applications, which saves you from having to write a lot of this code yourself. Then the ASP.NET Core framework code calls in to your handlers, which in turn call methods in your application’s business logic. This business logic is the core of your application. You can interact with other services here, such as databases or remote APIs, but your business logic typically doesn’t depend *directly* on ASP.NET Core.[4]

**Key points**

* **Modern Web Framework:** It’s a modern, high-performance, open-source web framework.
* **Familiar Design Patterns:** It uses familiar design patterns and paradigms.
* **Language Flexibility:** C# is a great language (but you can use VB.NET or F# if you prefer). You can build and run on any platform.
* **Cross-Platform Support:** Cross-platform development and deployment focus on performance as a feature.
* **Simplified Hosting:** A simplified hosting model.
* **Frequent Updates:** Regular releases with a shorter release cycle.
* **Open-Source:** Open-source for community collaboration and transparency.
* **Modular Design:** Modular features for tailored application development.
* **Flexible Paradigms:** More application paradigm options.
* **Standalone Deployment:** The option to package .NET with an app when publishing for standalone deployments.

## MySQL



Figure . MySQL Logo

MySQL Community Edition, also known as the *upstream* or *vanilla* version of MySQL, is the open source version distributed by Oracle. This version drives the development of the InnoDB engine and new features, and it is the first one to receive updates, new features, and bug fixes.[5]

**Key points**

* **Open Source:** Free to use with a large community and extensive documentation.
* **High Performance:** Optimized for speed and efficiency, capable of handling large-scale databases.
* **Scalability:** Supports large databases, making it suitable for both small and large applications.
* **Cross-Platform:** Runs on various operating systems including Windows, Linux, and MacOS.
* **ACID Compliance:** Ensures reliable transactions with Atomicity, Consistency, Isolation, and Durability.
* **Replication:** Supports master-slave replication for high availability and load balancing.
* **External, Internal, and Inline Styles:** CSS can be applied externally through separate style sheets, internally within HTML documents using the <style> element, or inline directly within HTML tags.
* **Security:** Provides robust security features like user authentication, SSL support, and data encryption.
* **Integration:** Easily integrates with various programming languages and web technologies.

## GitHub

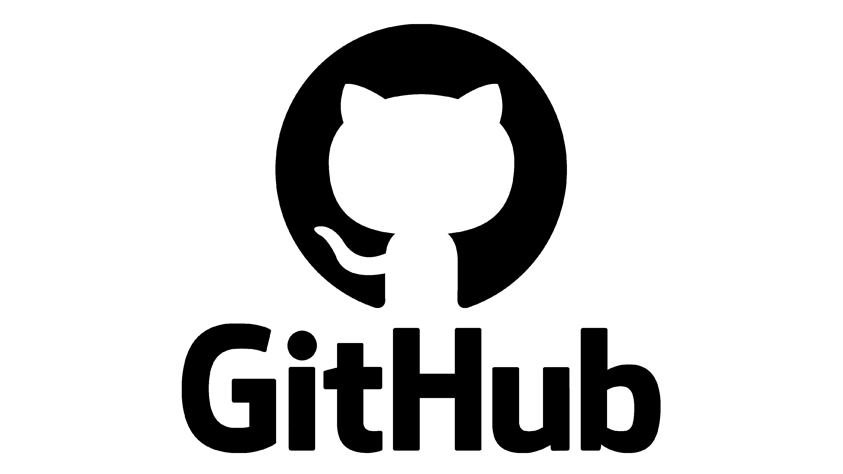


Figure . GitHub Logo

GitHub is the single largest host for Git repositories, and is the central point of collaboration for millions of developers and projects. A large percentage of all Git repositories are hosted on GitHub, and many open-source projects use it for Git hosting, issue tracking, code review, and other things. So while it’s not a direct part of the Git open source project, there’s a good chance that you’ll want or need to interact with GitHub at some point while using Git professionally. [7]

**Key points**

* **Simplified Version Control:** GitHub streamlines the use of Git, making version control accessible even for new developers.
* **Team Collaboration:** With features like pull requests and branch management, GitHub allows teams to work on the same codebase efficiently.
* **Open Source Contributions:** It acts as a central hub for open-source projects, enabling global developers to share and enhance code collaboratively.
* **Workflow Automation:** GitHub supports CI/CD pipelines, custom workflows, and API integrations to automate repetitive tasks.
* **Code Quality Assurance:** Tools like inline code comments and automated status checks ensure high-quality, bug-free software development.
* **Comprehensive Project Management:** Features like Kanban boards, milestones, and labels allow developers to track tasks and organize projects effectively.
* **Educational Platform:** GitHub provides opportunities for learning through real-world examples, tutorials, and community-shared codebases.
* **Robust Security:** Built-in tools like Dependabot and vulnerability scanning protect code from potential threats.
* **Global Collaboration:** GitHub connects developers worldwide, enabling seamless, real-time cooperation across different time zones.
* **Community Engagement:** It promotes active project support with features like wikis, discussions, and project-specific web pages.

# APPLICATION ANALYSIS, DESIGN AND IMPLEMENTATION

This chapter delves into the analysis, design, and implementation of the application, focusing on system requirements, ASP.NET Core setup, database design, and the creation of key functionalities for both customers and administrators.

## Requirements

The Anna Cosmetics web application is designed to cater to the needs of two primary user groups while implementing essential functionalities. This section outlines the requirements for the system, including user groups, core features, and project scope.

### ****User Groups****

The system identifies and supports two distinct groups of users:

1. **Customers**

* General users who browse and purchase cosmetics.
* Require an intuitive, user-friendly interface to interact with the website.

1. **Administrators**

* Internal team members responsible for managing products, orders, and customer data.
* Require a secure dashboard to oversee and update system content effectively.

### ****Core Features****

The application is built to provide fundamental functionalities for each user group:

1. **Customer Features**

* **Product Browsing:** Ability to view cosmetics organized by categories.
* **Product Details:** Display detailed information about each product, including images, descriptions, and prices.
* **Search Functionality:** Allow users to find products easily by name or category.
* **Basic Account Management:** Enable customers to create, update, or delete accounts.

1. **Admin Features**

* **CRUD Operations on Products:** Manage (Create, Read, Update, Delete) product details such as names, prices, and availability.
* **Category Management:** Organize products into categories for efficient browsing.
* **User Management:** View and manage customer accounts (basic operations only).

### ****Use Case Diagram****

Before diving into development, a **Use Case Diagram** is created to visualize how the user groups interact with the system. This diagram illustrates:

* The interactions between customers and core functionalities like product browsing and account management.
* The administrative operations such as managing products and categories.

### Scope

The scope of this project is limited to designing the interface and implementing essential CRUD functionalities:

* User Interface: Develop responsive and accessible interfaces for both customers and administrators.
* Basic CRUD Features: Enable product and category management for administrators and account management for customers.
* Advanced features like payment integration, order tracking, and analytics will be addressed in the next phase.

## Use case diagram

### Admin use case diagram

Figure . Admin use case diagram

The Admin Use Case Diagram illustrates the functionalities available to administrators in the Anna Cosmetics online shopping system. Admins are the primary actors, responsible for managing key aspects of the platform to ensure seamless operation and data integrity. Below are the descriptions of the main use cases:

* **Log in/Log out:** Admins authenticate their credentials to access the system securely. Logging out ensures account security after use.
* **Manage Category:** Admins create, read, update, or delete product categories to organize the cosmetics catalog effectively.
* **Manage Product:** Admins handle CRUD operations for individual products, including adding new items, updating details, or removing discontinued products.
* **Manage Order:** Admins review and update order statuses, including processing, shipping, or canceling customer orders.
* **Manage User:** Admins oversee user accounts, ensuring proper access control and handling issues like account activation or deactivation.
* **Manage News:** Admins publish, edit, or delete news articles or announcements to keep users informed about updates, promotions, or new arrivals.
* **CRUD Operations:** The core functionality that supports creating, reading, updating, and deleting data is applied across all management tasks (categories, products, orders, users, and news).

This diagram and its accompanying description ensure clarity on the admin’s role and their interaction with the system to maintain platform efficiency.

### Customer use case diagram

Figure . Customer use case diagram

The Customer Use Case Diagram illustrates the actions customers can perform on the Anna Cosmetics online shopping system. Customers are the primary users interacting with the platform to browse, select, and purchase products. Below is a summary of the key use cases:

* **Registration:** Customers create a new account by providing personal details to gain access to the system’s features.
* **Log in/Log out:** Registered users log in to their accounts to personalize their shopping experience and log out after completing their activities to secure their accounts.
* **View Items:** Customers can explore available products, including:
* **Search Items:** Find products by keywords.
* **Browse Items:** Navigate through categories to discover products.
* **View Recommended Items:** Receive suggestions based on preferences or popular products.
* **Add to Cart:** Allows customers to select desired products and save them for future checkout.
* **Make Purchase:** Customers proceed to buy selected items in their cart.
* **Checkout:** The final step of purchasing, which includes:
* **View/Update Cart:** Review and modify items before payment.
* **Calculate Voucher and Shipping:** Apply discounts and calculate shipping fees.
* **Payment:** Complete the purchase securely using available payment options.

This diagram highlights the essential features designed to enhance the customer experience, focusing on ease of use and seamless navigation throughout the platform.

## Database

### ERM diagram

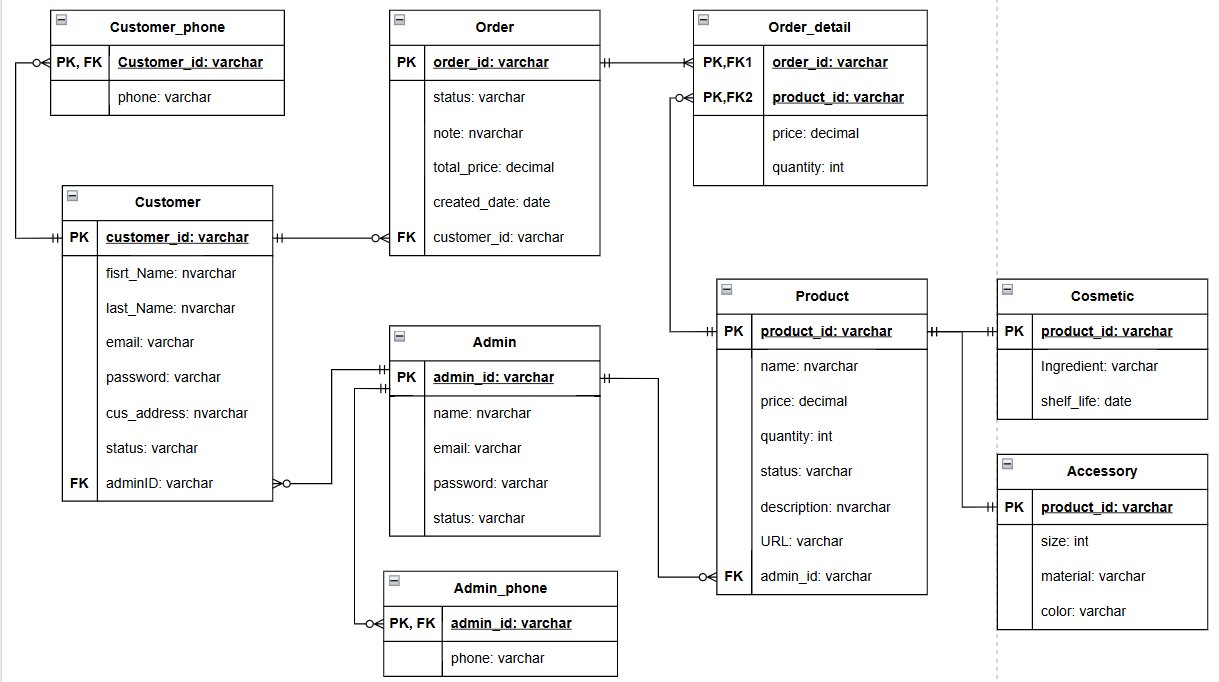


Figure . ERM diagram

### Table in database

Table . Table Customer

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Type** | **Description** |
| 1 | CustomerID | BigInt(20) | Primary key |
| 2 | CustomerName | Varchar(50) | Customer name |
| 3 | Email | Varchar(50), unique | Customer email |
| 4 | Password | Hash(255) | Customer password |
| 5 | Address | Varchar(100) | Customer address |
| 6 | PhoneNumber | Varchar(10) | Customer phone number |
| 7 | CustomerStatus | Enum(‘active’, ‘locked’) | Customer status |

Table . Table Product

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Type** | **Description** |
| 1 | ProductID | BigInt(20) | Primary key |
| 2 | ProductName | Varchar(50) | Product name |
| 3 | Description | Text | Product description |
| 4 | Price | Decimal(10,2) | Product price |
| 5 | ImageURL | Varchar(255) | Product image URL |
| 6 | InStock | BigInt(20) | Product in stock |
| 7 | ProductStatus | Enum(‘available’, ‘out of stock’) | Product status |

Table . Table Order

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Type** | **Description** |
| 1 | OrderID | BigInt(20) | Primary key |
| 2 | CustomerID | BigInt(20) | Foreign key |
| 3 | OrderDate | Datetime | Order date |
| 4 | TotalPrice | Decimal(10,2) | Total Price |
| 5 | Note | Text | Ex: Only delivery in office hour |
| 6 | OrderStatus | Enum(‘pending’, ‘shipped’, ‘delivered’, ‘cancelled’) | Order status |

Table . Table OrderDetail

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Type** | **Description** |
| 1 | OrderDetailID | BigInt(20) | Primary key |
| 2 | OrderID | BigInt(20) | Foreign key |
| 3 | ItemID | BigInt(20) | Foreign key |
| 4 | Quantity | BigInt(20) | Total Price |
| 5 | Price | Decimal(10,2) |  |

Table . Table Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Type** | **Description** |
| 1 | AdminID | BigInt(20) | Primary key |
| 2 | AdminName | Varchar(50) | Admin name |
| 3 | Password | Hash(255) | Admin password |
| 4 | Email | Varchar(100), unique | Admin email |
| 5 | PhoneNumber | Varchar(10) | Admin phone number |
| 6 | Status | Enum(‘active’, ‘locked’) | Admin status |

## Introduction to ASP.NET Core

### Installation

Visit the official [.NET download page](https://dotnet.microsoft.com/download).

* Select the appropriate SDK version for your operating system.
* Download and run the installer.
* Verify the installation by opening a terminal or command prompt and running: “dotnet –version”. The output should display the installed .NET SDK version.

**For Visual Studio Users:**

* Download and install Visual Studio 2022.
* During installation, select the ASP.NET and web development workload.
* Once installed, open Visual Studio and create a new ASP.NET Core project.

### ASP.NET Core Project Structure

#### Root Directory

The root directory contains essential files and folders, including:

* Program.cs / Startup.cs: Entry point for configuring the application’s services and middleware pipeline.
* appsettings.json: Primary configuration file for application settings.
* wwwroot/: Static files such as CSS, JavaScript, and images.

#### Folders

* Controllers/: Contains controller classes that handle HTTP requests and responses. For example:

public class HomeController : Controller

{

public IActionResult Index()

{

return View();

}

}

* Models/: Houses data models that define the structure of your application’s data.
* Views/: Stores Razor view files used for rendering HTML pages. Example structure:
* Views/Home/Index.cshtml
* Views/Shared/\_Layout.cshtml
* Data/: (Optional) Manages database-related operations, including context and migrations.

#### Key Files

* launchSettings.json: Configures application launch settings for different environments.
* Csproj File: Manages project dependencies and configurations.

#### Creating and Initializing the Project

* Open a terminal and navigate to your working directory.
* Generate a new project:

“dotnet new webapp -o MyProject

cd MyProject”

* Build and run the application:

“dotnet build

dotnet run”

Open the URL provided in your browser to test the application.

#### Configuring ASP.NET Core

**Application Settings**

* Edit appsettings.json to include custom configurations. Example:

{

“Logging”: {

“LogLevel”: {

“Default”: “Information”,

“Microsoft”: “Warning”,

“Microsoft.Hosting.Lifetime”: “Information”

}

},

“AllowedHosts”: “\*”

}

**Middleware Setup**

* Modify Program.cs or Startup.cs to define middleware components. Example:

app.UseRouting();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllers();

});

#### Running the Application

* Start the application: “dotnet run”
* Open the provided URL (e.g., https://localhost:5001) in a browser to verify successful deployment.

#### Advanced Configurations

**Enabling HTTPS**

* Check launchSettings.json for HTTPS settings:

“profiles”: {

“MyProject”: {

“commandName”: “Project”,

“launchBrowser”: true,

“applicationUrl”: “https://localhost:5001;http://localhost:5000”,

“environmentVariables”: {

“ASPNETCORE\_ENVIRONMENT”: “Development”

}

}

}

**Database Integration**

* Install the necessary database package:

dotnet add package Microsoft.EntityFrameworkCore.SqlServer

* Update appsettings.json with a connection string:

“ConnectionStrings”: {

“DefaultConnection”: “Server=(localdb)\\mssqllocaldb;Database=MyDatabase;Trusted\_Connection=True;”

}

## CRUD for Admin

### Create new category

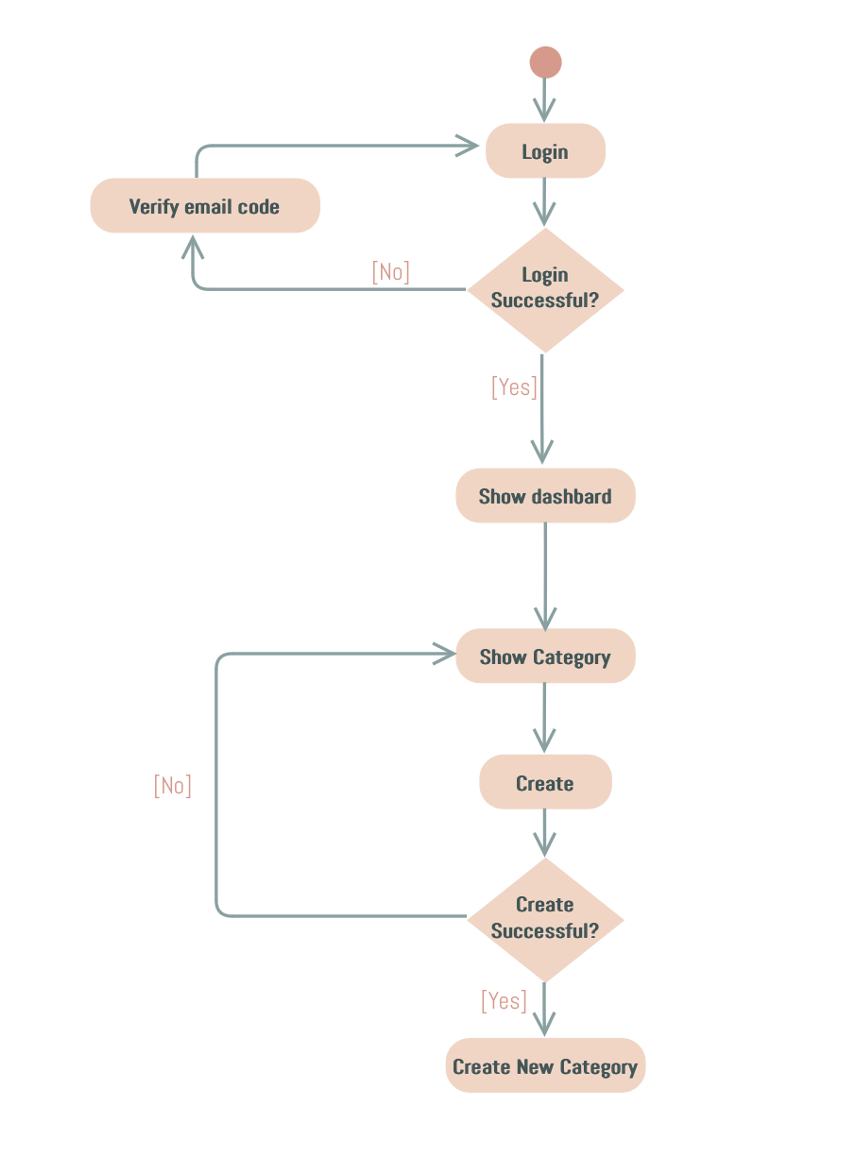


Figure .Create activity diagram

The activity diagram demonstrates the process for an admin to create a new category in the Anna Cosmetics system. The flow is as follows:

* **Login:** The admin starts by logging into the system.
* **Email Verification (if required):** If the login attempt is unsuccessful, the system may prompt the admin to verify their email code. Once verified, the login is retried.
* **Dashboard Access:** Upon successful login, the admin is directed to the dashboard where all management options are displayed.
* **Navigate to Categories:** The admin selects the category management feature to view existing categories.
* **Create Category:** The admin fills out the details for a new category and submits the form.
* **Validation:** The system verifies whether the category creation process is successful:
* If successful, a new category is added.
* If unsuccessful, the admin is prompted to reattempt the process.

### Read category

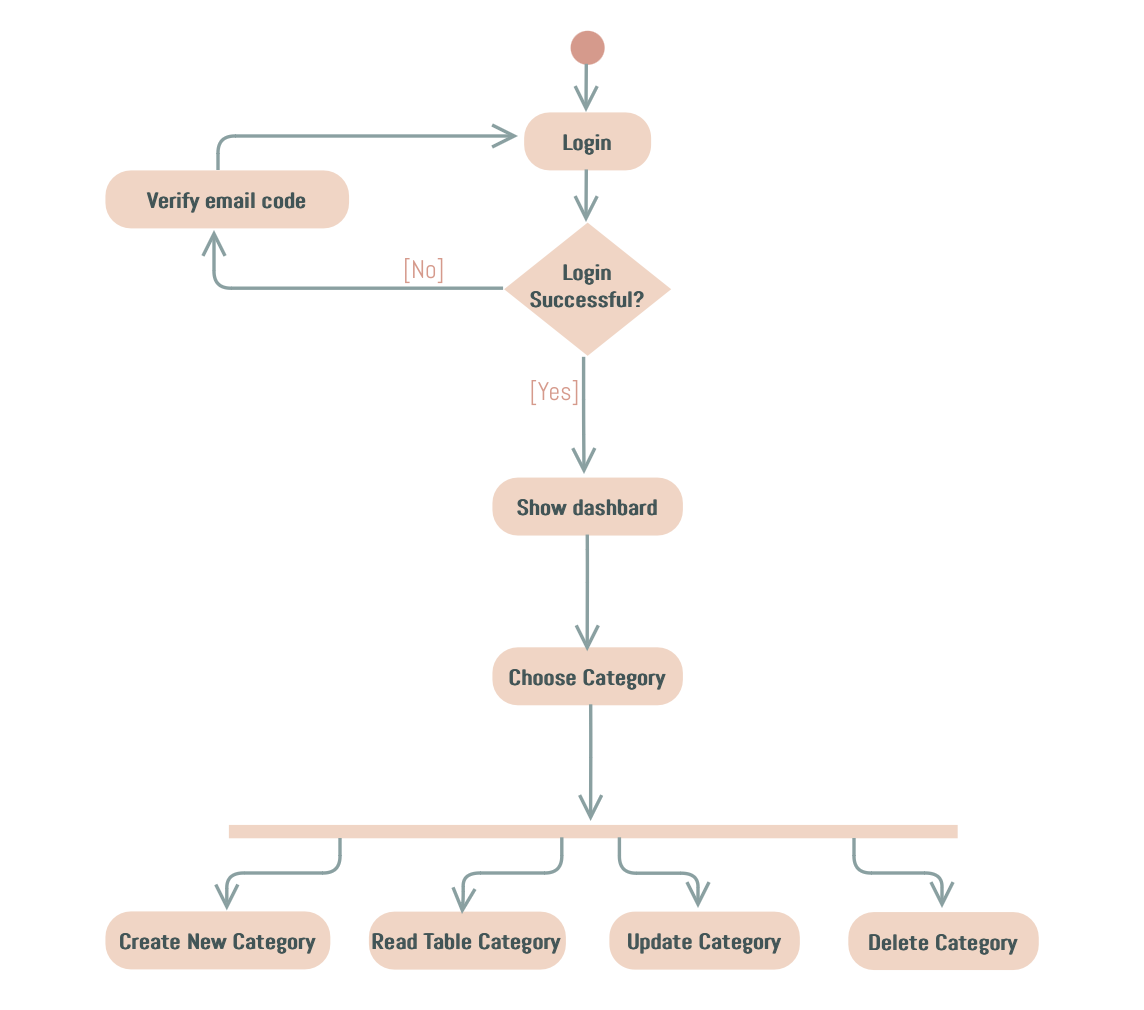


Figure .Read activity diagram

This activity diagram outlines the process for an admin to view category details in the Anna Cosmetics system. The flow proceeds as follows:

* **Login:** The admin begins by logging into their account.
* **Email Verification (if required):** If the login is unsuccessful, the system prompts the admin to verify their email code before attempting to log in again.
* **Access Dashboard:** Upon successful login, the admin is redirected to the dashboard where all management options are displayed.
* **Choose Category Management:** The admin selects the category management section to view the list of available categories.
* **Read Category Details:** The admin reviews the table containing all existing categories, including details such as category names, descriptions, and statuses.

### Update category

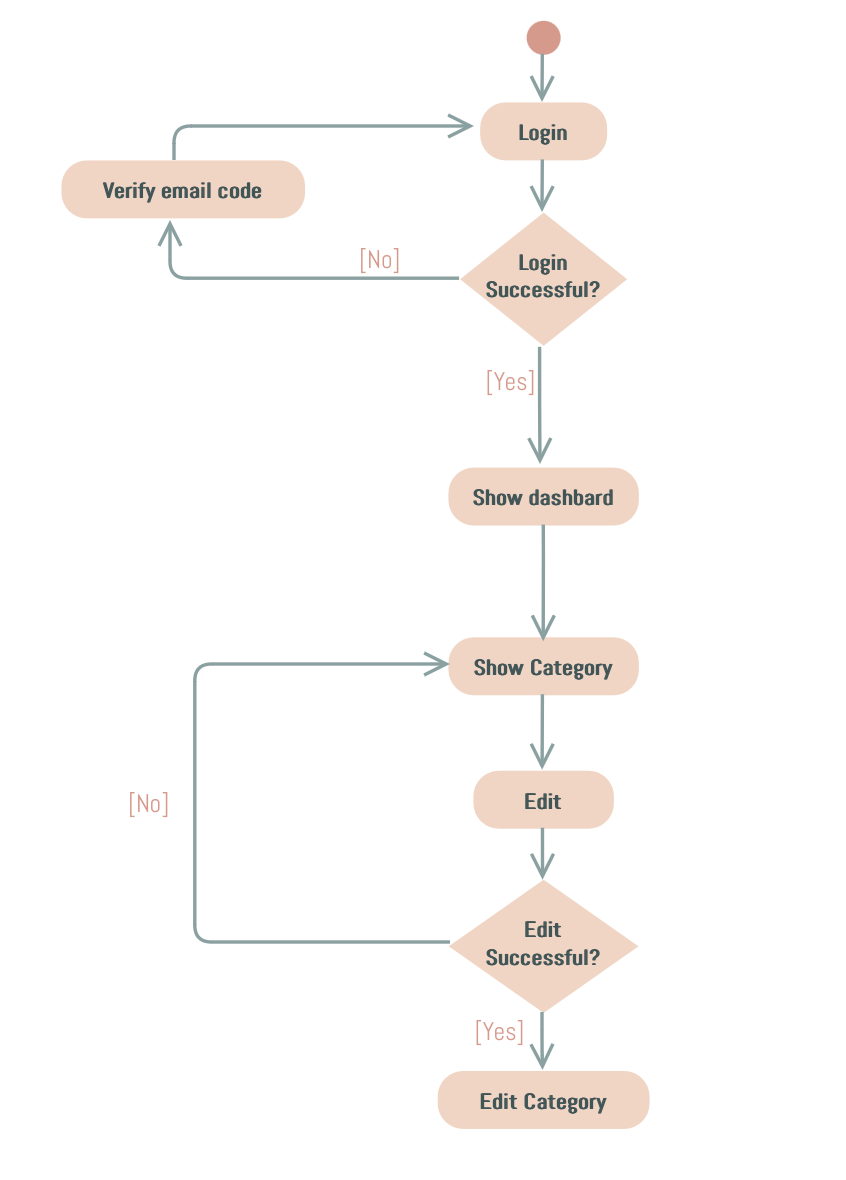


Figure .Edit activity diagram

This activity diagram outlines the steps an admin takes to edit a product category in the Anna Cosmetics system. The flow is described as follows:

* **Login:** The admin begins by logging into the system.
* **Email Verification (if required):** If the login attempt fails, the system prompts the admin to verify their email code before retrying the login process.
* **Access Dashboard:** Upon successful login, the admin is directed to the dashboard where all management options are accessible.
* **Navigate to Categories:** The admin selects the category management option to view the list of existing categories.
* **Edit Category:** The admin selects a specific category to modify its details and submits the changes.
* **Validation:** The system verifies the success of the edit operation:
* If the edit is successful, the changes are saved, and the updated category is displayed.
* If the edit fails, the admin is prompted to retry.

### Delete category

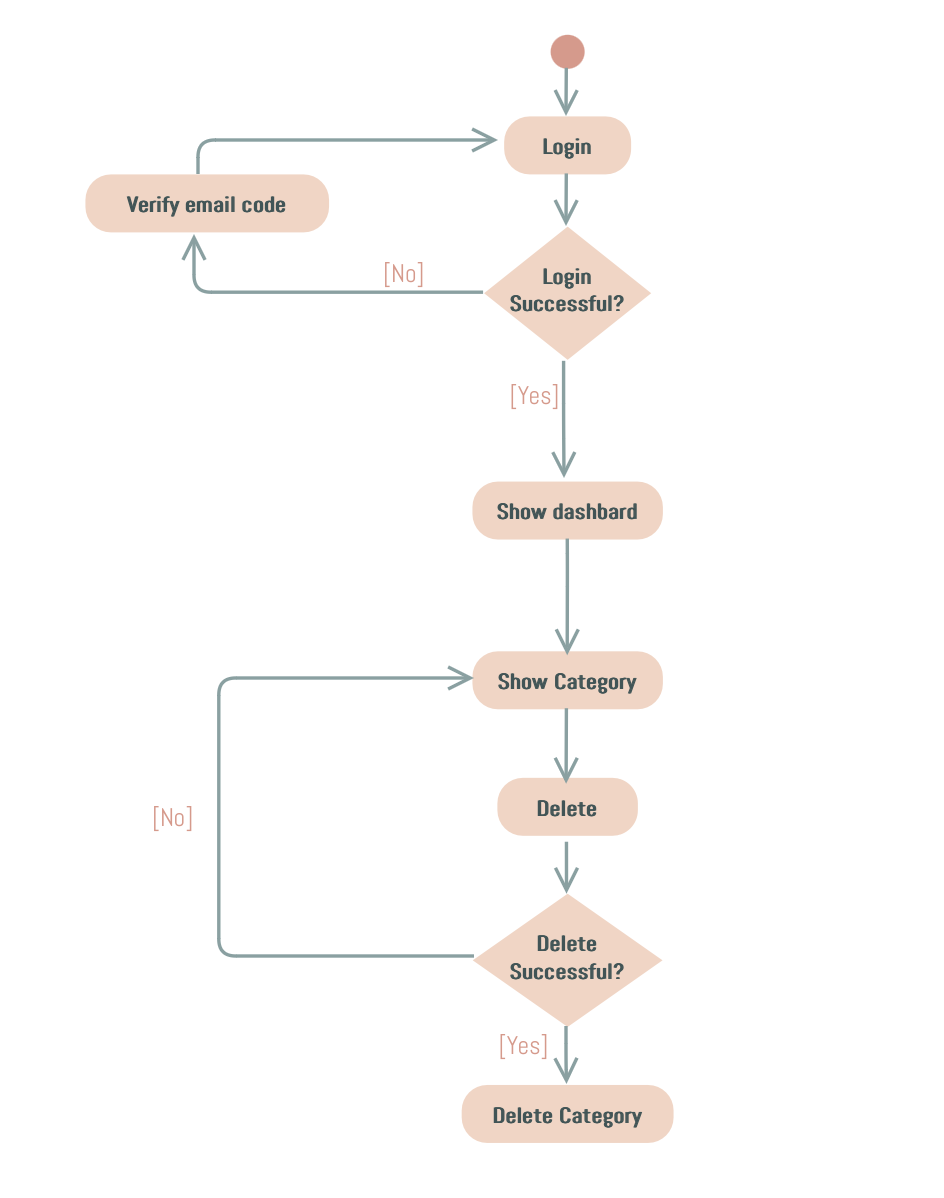


Figure .Delete activity diagram

This activity diagram represents the process an admin follows to delete a category in the Anna Cosmetics system. The flow is as follows:

* **Login:** The admin begins by logging into the system.
* **Email Verification (if required):** If the login fails, the system requests email verification before allowing the admin to retry logging in.
* **Access Dashboard:** Upon successful login, the admin is directed to the dashboard where management options are available.
* **Navigate to Categories:** The admin selects the category management section to view the list of available categories.
* **Delete Category:** The admin selects a category to delete and initiates the deletion process.
* **Validation:** The system checks if the category was successfully deleted:
* If successful, the category is permanently removed from the database.
* If unsuccessful, the admin is prompted to retry the deletion process.

# EXPERIMENT AND DISCUSSION

This chapter presents the practical implementation of the project, including the setup of the development environment, the outcomes of the system's functionalities, and an analysis of the results. It evaluates the effectiveness of the implemented features and discusses their performance based on real-world testing scenarios.

## Installation Environment

* Microsoft Visual Studio Community 2022 (64-bit) – Current Version: 17.12.3
* .NET SDK Version 9.0.101
* MySQL Version 8.0.40 (64 bits)
* GitHub Desktop Version 3.4.9 (arm64)
* ApacheFriends XAMPP Version 8.0.30

## Results and Discussion

### Home page for customer

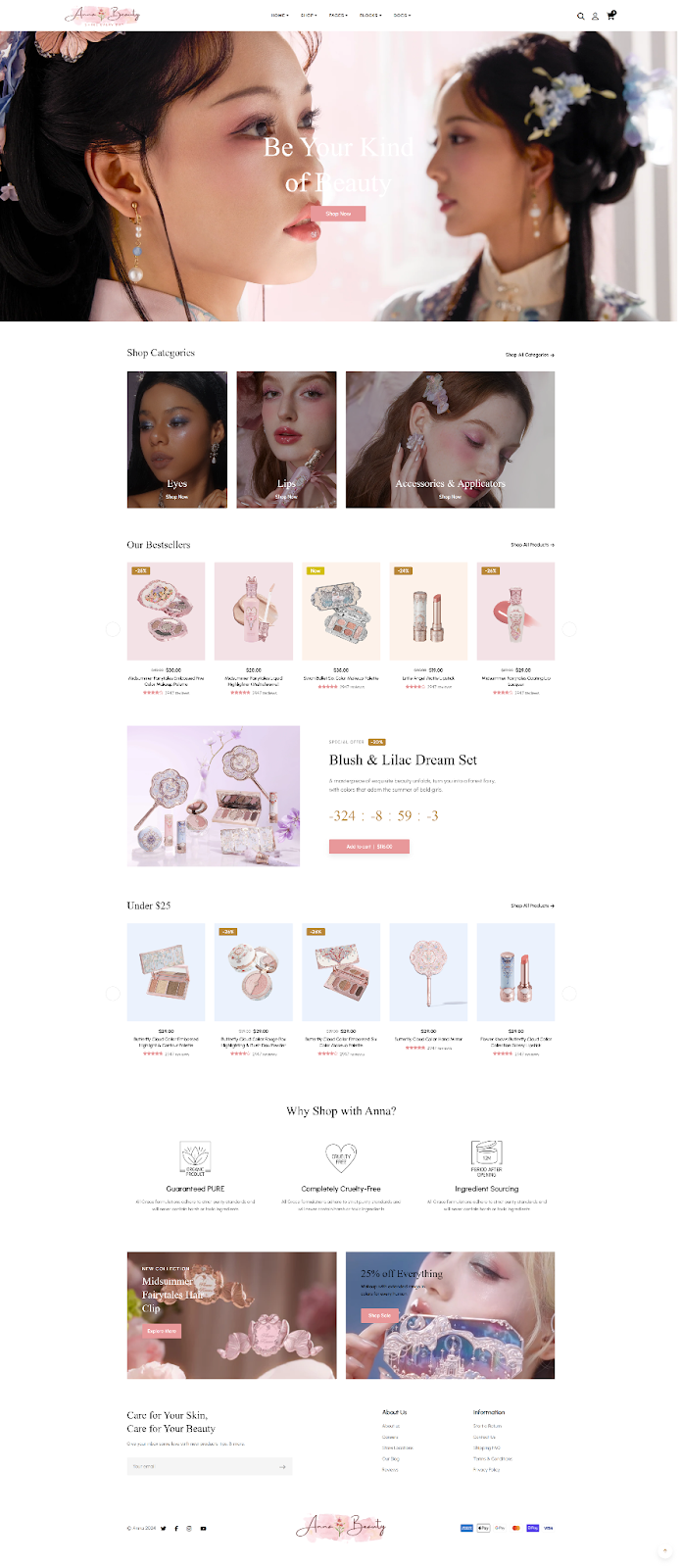


Figure .Home page

The home page interface of the Anna Cosmetics website is designed to provide users with a seamless and user-friendly shopping experience. Below is a detailed breakdown of the interface and features implemented, along with an evaluation of their functionality and performance.

#### Interface and Features Implemented

**Log In/Log Out Functionality**

* **Description:** The home page features a log-in system that allows users to access their accounts securely. The log-out option ensures users can safely exit their accounts after use.
* **Implemented Features:**
* Input validation for email and password fields.
* Real-time error messaging for incorrect login credentials.
* Auto-lock for repeated failed login attempts to enhance security.
* **Evaluation:** This feature ensures a secure user authentication process and protects user data. However, the error messages could include more specific hints (e.g., “Invalid email format”) for improved user guidance.

**Shopping Cart System**

* **Description:** Users can add, remove, or view items in their shopping cart. The cart is displayed prominently for easy access and provides a summary of selected items.
* **Implemented Features:**
* Dynamic price calculation, including taxes and discounts.
* “Save for later” option for items users may want to purchase later.
* Prevents adding out-of-stock items to the cart.
* **Evaluation:** The shopping cart is intuitive and provides a smooth shopping experience. The dynamic updates work efficiently, but additional feedback (e.g., confirmation for removed items) could enhance usability.

**Mobile-Friendly Design**

* **Description:** The interface is responsive and optimized for mobile devices, ensuring a consistent user experience across different screen sizes.
* **Implemented Features:**
* Adaptive layouts that adjust seamlessly to various screen resolutions.
* Touch-friendly navigation elements and buttons for mobile users.
* **Evaluation:** The mobile optimization improves accessibility and broadens the audience. The loading speed is commendable, though further testing on older devices may reveal additional optimization needs.

**Enhanced User Experience**

* **Description:** The interface includes several user-focused improvements aimed at increasing convenience and accessibility.
* **Implemented Features:**
* Restricts numeric fields (e.g., phone number input) from accepting non-numeric characters.
* Quick load times for images and product information due to lazy loading.
* Clear and organized navigation menu for easy access to categories and products.
* **Evaluation:** These enhancements make the home page highly user-friendly. The restricted inputs prevent common errors, and the lazy loading ensures faster performance without compromising content quality.

#### Overall Evaluation

The home page successfully achieves its goals of providing an intuitive, responsive, and secure interface for users. While the current features meet the basic requirements, future improvements could include:

* Personalization options, such as product recommendations based on user preferences.
* Integration of advanced filtering and sorting in the shopping cart.
* Expanded visual cues for better user engagement, such as animations for interactions.

With these features and evaluations, the home page of the Anna Cosmetics website stands as a strong foundation for further development and enhancement.

### CRUD for admin

The admin CRUD interface of the Anna Cosmetics website is designed to streamline the management of product categories, orders, and users. This section provides an overview of the implemented interface and its features, along with an evaluation of their effectiveness.

#### Create Category

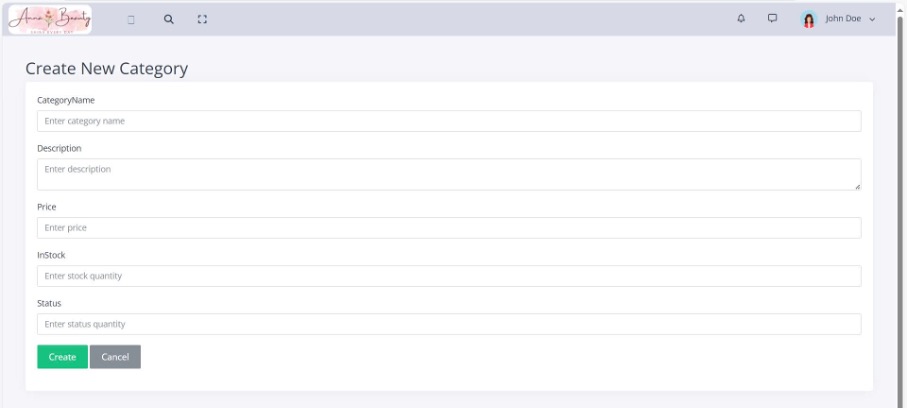


Figure .Create category interface

* **Description:** The create feature allows admins to add new categories, products, or users to the system. Admins are required to input relevant details into a form.
* **Implemented Features:**
* Input validation to ensure fields like price, phone number, and stock accept only valid data types.
* Error notifications for incomplete or invalid form submissions.
* Auto-reset of form fields after successful submission.
* **Evaluation:** The form is user-friendly and minimizes input errors with validation rules. The error messages are helpful, but the design could benefit from tooltips or placeholders to guide data entry.

#### Read

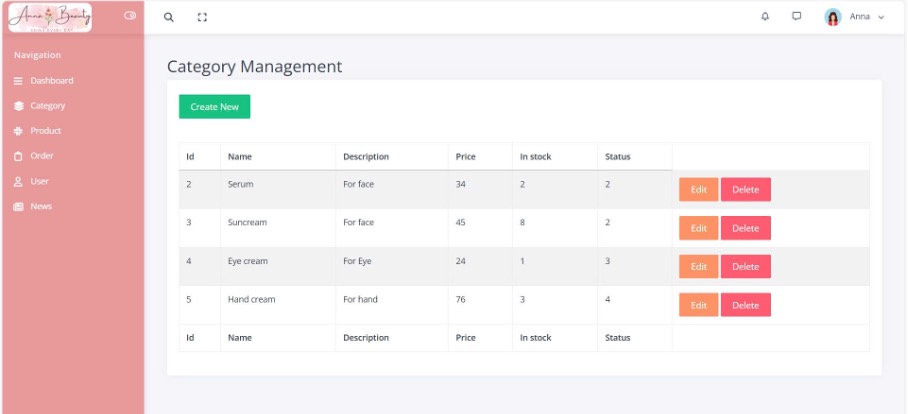


Figure .Read category interface

* **Description:** Admins can view a list of all existing records (e.g., categories, products, users) in a tabular format with pagination and search functionality.
* **Implemented Features:**
* Pagination for better performance with large datasets.
* Search bar with filters for fields like category, price, or user roles.
* Clickable rows to access detailed information about a specific record.
* **Evaluation:** The table is well-organized and responsive, making it easy to navigate and retrieve information. However, adding export options (e.g., CSV or Excel) could enhance usability for bulk data handling.

#### Update

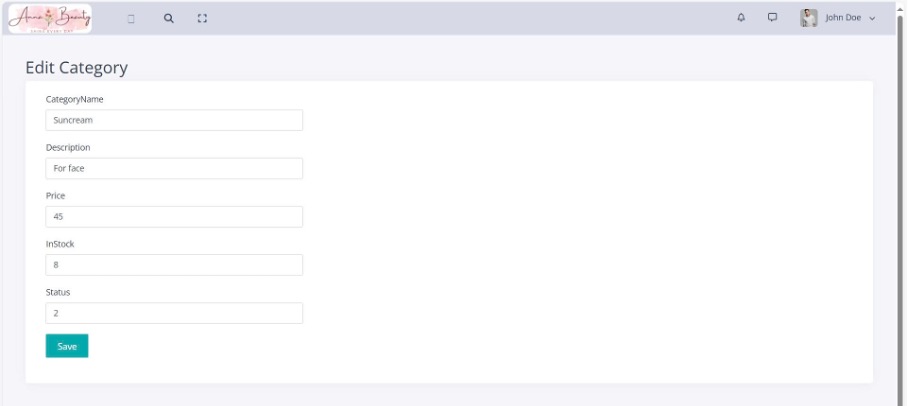


Figure .Update(Edit) category interface

* **Description:** Admins can update existing records by selecting them from the list and editing their details.
* **Implemented Features:**
* Pre-filled fields to display current data for easy modification.
* Input validation to prevent invalid data submissions.
* Confirmation dialogs before saving changes.
* **Evaluation:** The edit functionality works efficiently and reduces errors with pre-filled fields. Including an activity log to track changes made by admins could improve accountability.

#### Delete

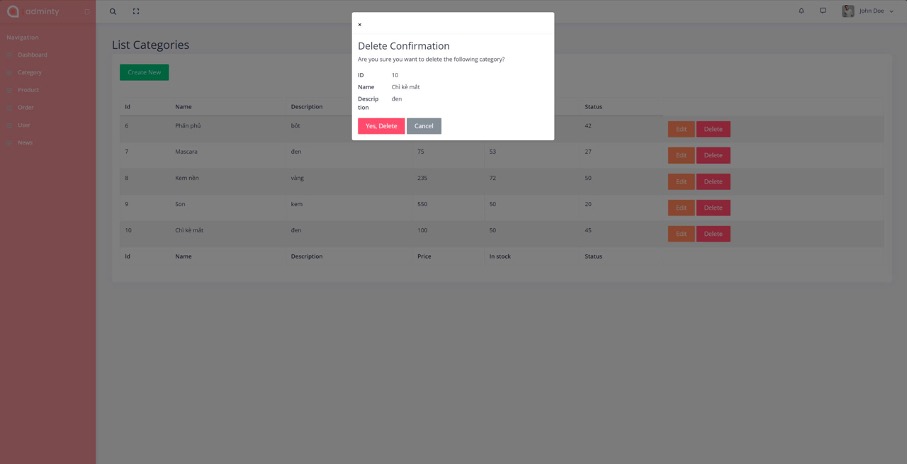


Figure .Delete category interface

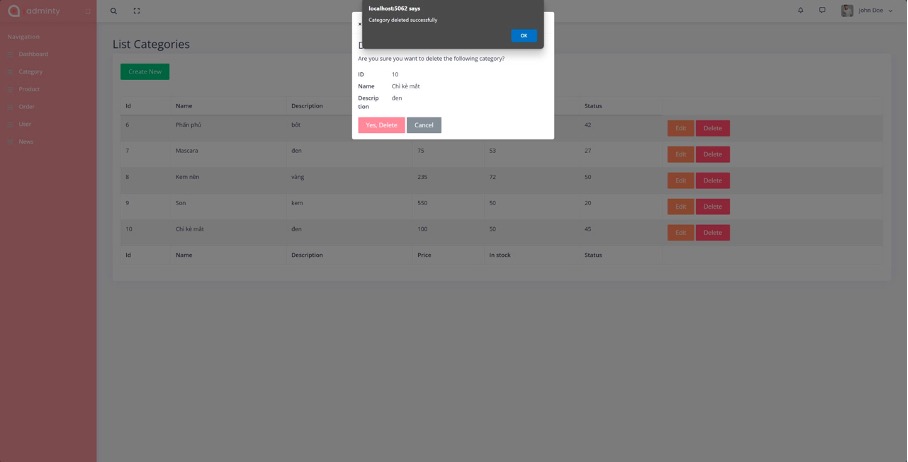


Figure .Delete category successfully interface

* **Description:** Admins can remove records from the system permanently using this feature.
* **Implemented Features:**
* Confirmation dialogs to prevent accidental deletions.
* Restriction to prevent the deletion of critical records (e.g., admin accounts).
* Real-time updates to the table after successful deletion.
* **Evaluation:** The delete feature is secure and straightforward. A “soft delete” option could be considered in the future to allow data recovery if needed.

#### Additional Features

* **Mobile Compatibility:** The interface is fully responsive, ensuring usability across devices of varying screen sizes. Buttons and input fields are touch-friendly for mobile users.
* **Optimized Performance:** The system implements lazy loading for large datasets to ensure fast load times, even with significant data volumes.
* **Error Handling and Notifications:** All operations include clear and concise error or success messages, providing feedback to the admin for every action performed.

#### Overall Evaluation

The CRUD interface is robust and effectively supports the admin's management tasks. The user-friendly design, combined with the implemented features, ensures that tasks can be completed efficiently and with minimal errors. Suggestions for future improvement include:

* Adding an undo feature for accidental deletions.
* Providing bulk actions for creating, editing, or deleting multiple records simultaneously.
* Incorporating real-time notifications for updates.

This admin CRUD interface serves as a strong foundation for managing the website’s content and will benefit from future enhancements as the system scales.

# CONCLUSION AND FUTURE WORKS

This chapter provides a summary of the knowledge, skills, and outcomes gained during the development of the project. It also outlines future improvements and enhancements that will be implemented to fully complete the system in the next phase.

## Conclusion

Through the successful completion of this project, I have acquired significant knowledge, practical experience, and essential skills that are highly applicable to real-world software development. This project has allowed me to gain a deeper understanding of both frontend and backend development, as well as the integration of modern tools and technologies.

### Knowledge Gained

We learned to utilize web development frameworks such as ASP.NET for backend operations and MySQL for managing relational databases. This expanded my understanding of how to build secure, scalable, and data-driven systems. Strengthened my skills in frontend development using HTML/CSS, JavaScript, and frameworks to create responsive and interactive user interfaces. Gained proficiency with version control systems like Git for tracking changes and collaborating efficiently on codebases.

### Product Outcomes

Through this project, we successfully developed a functional and practical web application for Anna Cosmetics, achieving the following outcomes:

**Home page for customer**

* The home page was designed to provide a seamless shopping experience for users, enabling them to log in, browse products, and add items to their shopping cart.
* This feature simplifies the shopping process, enhances user convenience, and improves the overall customer experience.

**CRUD features for admin**

* The CRUD interface empowers administrators to manage product categories, user accounts, and orders effectively.
* It supports efficient operations like adding, updating, viewing, and deleting records, ensuring smooth system maintenance and scalability.
* Impact of the product:
* The project demonstrates how a well-structured system can streamline e-commerce operations, benefiting both customers and administrators.
* It offers practical insights into creating user-friendly interfaces and implementing essential functionalities, contributing to the development of a complete, real-world application.

### Skills Developed

Enhanced technical skills in database management, including creating and managing tables, queries, and stored procedures in MySQL. Improved problem-solving and debugging abilities through troubleshooting frontend and backend issues. Developed project management and version control skills by utilizing Git for code organization, tracking progress, and collaboration. Strengthened soft skills such as teamwork, communication, and time management by working on multiple tasks and coordinating with team members.

### Team Growth

Through this project, we gained valuable skills in communication and teamwork:

* **Improved Communication:** We learned to listen actively, express ideas clearly, and resolve conflicts constructively to maintain a productive working environment.
* **Teamwork and Collaboration:** We effectively divided tasks based on strengths, adapted to challenges, and supported each other throughout the project.
* **Time Management:** Working together taught us to plan efficiently, meet deadlines, and stay focused under pressure.

## Future works

While this project achieves its core objectives, it serves as a foundation for further development in the next phase. In Project B, the goal is to complete the system and make it fully interactive and functional. The future improvements include:

* **Linking Pages:** Ensure seamless navigation by connecting all pages of the system, enabling a smooth user experience.
* **User Features:** Allow users to interact fully with the system, including searching for products, adding, updating, and removing products, and managing their accounts via the Sign Up/Sign In system.
* **Admin Panel:** Develop an Admin Dashboard for managing users and products, with functionality for adding, editing, and deleting product information.
* **System Completion:** Integrate all features to create a polished and fully operational web application, resembling a real-world e-commerce platform.

In conclusion, these enhancements will not only complete the current system but also ensure it is scalable, user-friendly, and ready for real-world applications. By achieving these goals in Project B, the system will deliver a fully interactive experience for both users and administrators, meeting the standards of a modern e-commerce solution.

REFERENCES

1. D. R. Brooks, An Introduction to HTML and JavaScript: for Scientists and Engineers, 2007
2. Meyer, Eric A., Cascading Style Sheets: The Definitive Guide, Third Edition, 2006
3. D. Flanagan and P. Ferguson, JavaScript: The Definitive Guide, 1998
4. Andrew Lock, ASP.NET Core in Action, Third Edition, 2023
5. Vinicius Grippa and Sergey Kuzmichev, O’Reilly Media, Learning MySQL: Get a Handle on Your Data, 2021
6. Michell Queiroz, Yasin N. Silva, Isadora Almeida, SQL: From Traditional Databases to Big Data, 2016
7. Sarah Guthals, Phil Haack, *GitHub For Dummies 1st Edition,* 2019
8. Ashima Mehta, Aditi Paul, Responsive Web Design and Its Impact on User Experience, 2023
9. Waseem I., Abdelaziz I. Hammouri, Responsive Web Design Techniques, 2016