**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, <Month (mm), Year (yyyy)**

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, M.Sc.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](#_Toc161133453)

[ACKNOWLEDGEMENT ii](#_Toc161133454)

[TABLE OF CONTENTS iii](#_Toc161133455)

[LIST OF FIGURES iv](#_Toc161133456)

[LIST OF TABLES v](#_Toc161133457)

[LIST OF ABBREVIATIONS vi](#_Toc161133458)

[CHAPTER 1. OVERVIEW 1](#_Toc161133459)

[1.1. Introduction 1](#_Toc161133460)

[1.2. Project objectives 1](#_Toc161133461)

[1.3. Project structure 1](#_Toc161133462)

[CHAPTER 2. INTRODUCTION TO TECHNOLOGIES 2](#_Toc161133463)

[2.1. Introduction to HTML 5, CSS 3, JavaScript 2](#_Toc161133464)

[2.1.1. HTML 2](#_Toc161133465)

[2.1.2. CSS 2](#_Toc161133466)

[2.2. Introduction to Java Spring 2](#_Toc161133467)

[CHAPTER 3. APPLICATION ANALYSIS, DESIGN AND IMPLEMENTATION 3](#_Toc161133468)

[3.1. Requirements 3](#_Toc161133469)

[3.2. Introduction to Java Spring 3](#_Toc161133470)

[3.2.1. Installation 3](#_Toc161133471)

[3.2.2. Java Spring Project Structure 3](#_Toc161133472)

[3.3. Database 3](#_Toc161133473)

[CHAPTER 4. EXPERIMENT AND DISCUSSION 4](#_Toc161133474)

[4.1. Installation Environment 4](#_Toc161133475)

[4.2. Results and Discussion 4](#_Toc161133476)

[CHAPTER 5. CONCLUSION AND FUTURE WORKS 5](#_Toc161133477)

[5.1. Conclusion 5](#_Toc161133478)

[5.2. Future works 5](#_Toc161133479)

[REFERENCES 6](#_Toc161133480)

[APPENDIX 7](#_Toc161133481)

LIST OF FIGURES

[Figure 1. HTML Logo 4](#_Toc185941967)

[Figure 2. CSS Logo 5](#_Toc185941968)

[Figure 3. JavaScript Logo 7](#_Toc185941969)

[Figure 4. ASP.NET Core Logo 8](#_Toc185941970)

[Figure 5. Admin use case diagram 13](#_Toc185941971)

[Figure 6. Customer use case diagram 14](#_Toc185941972)

[Figure 7. ERD diagram 18](#_Toc185941973)

[Figure 8. ERM diagram 18](#_Toc185941974)

[Figure 9. Admin create page 21](#_Toc185941975)

[Figure 10. Admin read page (Home page) 23](#_Toc185941976)

[Figure 11. Admin update page (edit page) 24](#_Toc185941977)

[Figure 12. Admin delete page 24](#_Toc185941978)

[Figure 13. Admin delete successfully page 25](#_Toc185941979)

[Figure 14. Home page 27](#_Toc185941980)

[Figure 15. Sign up page 28](#_Toc185941981)

[Figure 16. Sign in page 29](#_Toc185941982)

[Figure 17. Shopping cart page 30](#_Toc185941983)

LIST OF TABLES

[Table 1. Table Customer 14](#_Toc185937854)

[Table 2. Table Product 15](#_Toc185937855)

[Table 3. Table Order 15](#_Toc185937856)

[Table 4. Table OrderDetail 15](#_Toc185937857)

[Table 5. Table Admin 16](#_Toc185937858)

[Table 6. Table Product 16](#_Toc185937859)

[Table 7. Table Customer 16](#_Toc185937860)

[Table 8. Table Order 16](#_Toc185937861)

[Table 9. Table OrderDetail 17](#_Toc185937862)

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 |

# 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 main goals of the ANNA Cosmetic Website project are to construct an informative and user-friendly e-commerce site for customers looking to purchase cosmetics.

The primary objective is to develop a site where customers can comfortably navigate, search for, and buy cosmetic products. In addition, the website will help users make purchases by providing information about skincare and makeup trends to help users decide what to buy.

Making sure that online payments are safe is also important and integration of trustworthy, dependable, and secure payment gateways is out to be done. In the same way, there will be an effective inventory system that will enable businesses to capture and update product details instantaneously.

Finally, the website will provide administrators with a data warehouse, and integrated information storage which will assist them in understanding the customers and market demand for the products offered on the website. The website shall be available at any time and will be fully responsive. Ensuring compatibility for users on desktop computers, tablets, and mobile phones.

## 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

### Admin Panel Interface

Developed a clean and responsive admin interface to manage the website effectively. The design focuses on simplicity and functionality, allowing administrators to navigate easily. Backend CRUD operations are under development, starting with essential functions like managing products, orders, and customer data.

### Customer-Side Theme

Designed and implemented a modern, visually appealing, and fully responsive theme for the customer-facing website. The theme ensures a consistent user experience across devices, including desktops, tablets, and mobile phones. Basic UI elements, such as navigation menus, product listings, and a homepage layout, have been set up to create a foundation for future functionality.

### Responsive Design

Ensured both admin and customer interfaces adapted seamlessly to various screen sizes. Followed best practices in UI/UX design to provide an intuitive experience for all users.

### Preparation for Backend Integration

Organized the structure of both admin and customer interfaces to support future backend functionalities. Focused on scalability to allow easy addition of advanced features like product search, filtering, and analytics. This stage of the project lays a strong foundation for adding dynamic features and backend logic in the next phase of development.

# INTRODUCTION TO TECHNOLOGIES

This chapter introduces the core technologies used to develop the project, including HTML5, CSS3, JavaScript, and ASP.NET Core. 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.

## Introduction to HTML 5, CSS 3, ASP.NET Core

### 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]](#Book1)

**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 powerful styling language used in web development to enhance the visual presentation and layout of HTML documents. It allows developers to define the appearance of web pages, including colors, fonts, spacing, and positioning.[[1]](#Book1)

**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

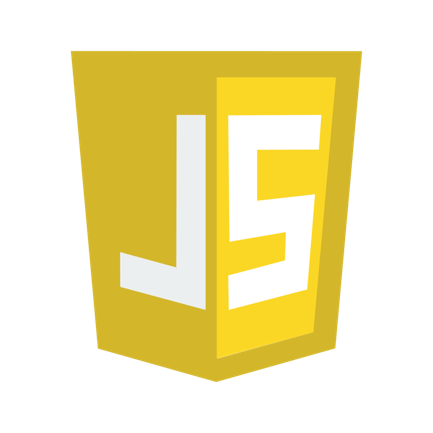
****

Figure . JavaScript Logo

JavaScript, created by Brendan Eich in 1995, is essential for dynamic web applications. It enables interactive user experiences without constant server interaction, with intuitive syntax and dynamic typing. Evolving through the ECMAScript standard, it's used not only for front-end but also back-end and mobile development. Its asynchronous nature ensures smooth performance, making it indispensable in modern web development.[[1]](#Book1)

**Customer-Side Interactivity**: JavaScript allows for dynamic and interactive elements on web pages without needing to reload the entire page. This enhances user experience by providing immediate feedback and responsiveness.

**Cross-Platform Compatibility**: JavaScript runs on all major browsers and platforms, making it a versatile choice for web development. It ensures consistent behavior across different devices and operating systems.

**Rich Ecosystem**: JavaScript has a vast ecosystem of libraries, frameworks, and tools that streamline development processes. These resources enable developers to build complex applications efficiently and with minimal effort.

**Asynchronous Programming**: JavaScript's asynchronous nature allows non-blocking operations, such as fetching data from servers or handling user input, without freezing the user interface. This enhances performance and responsiveness.

**Versatility**: JavaScript is not limited to front-end development. With frameworks like Node.js, it can also be used for server-side development, enabling full-stack development with a single programming language.

**Easy to Learn**: JavaScript has a relatively simple syntax compared to other programming languages, making it accessible to beginners. Additionally, its dynamic typing eliminates the need for explicit data type declarations, reducing development time and complexity.

**Community Support**: JavaScript has a large and active community of developers who contribute to open-source projects, share knowledge, and provide support. This vibrant community ensures that developers have access to resources and assistance when needed.

## Introduction to Java Spring



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, as shown in figure 1.1. 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.[[2]](#Book2)

**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.

# 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

* Operating System: Windows, macOS, or Linux.
* Development Environment: Visual Studio 2022(Window/macOS) or Visual Studio Code.

## 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.

**For Visual Studio Code Users:**

* Download and install Visual Studio Code.
* Install the following extensions:
* **C# Extension** by Microsoft.
* **ASP.NET Core Snippets** (optional).
* Use the terminal to create a new project (details in the next step).

### 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;"

}

## Use case diagram

### Customer use case diagram

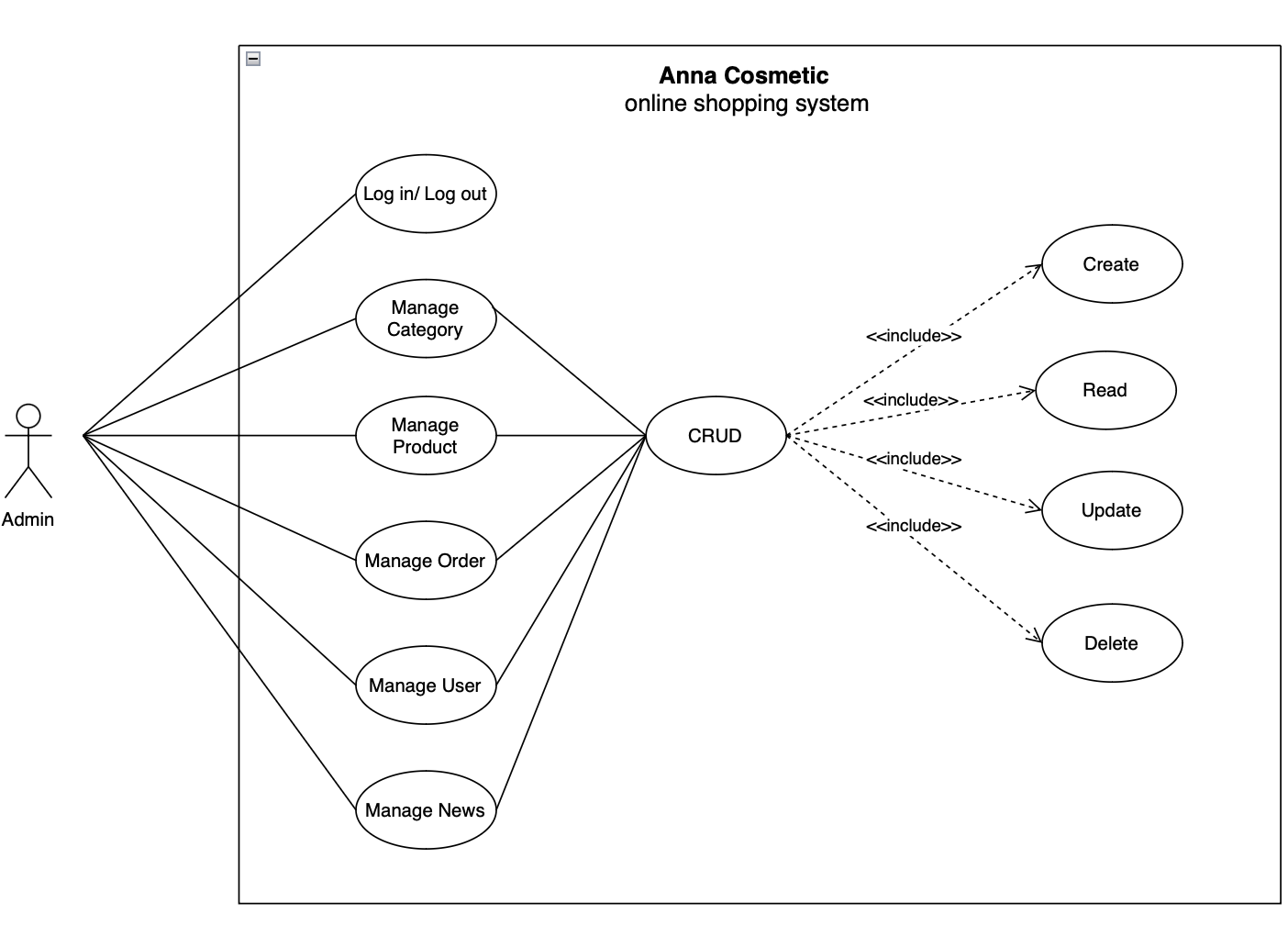


Figure . Admin use case diagram

### Admin use case diagram

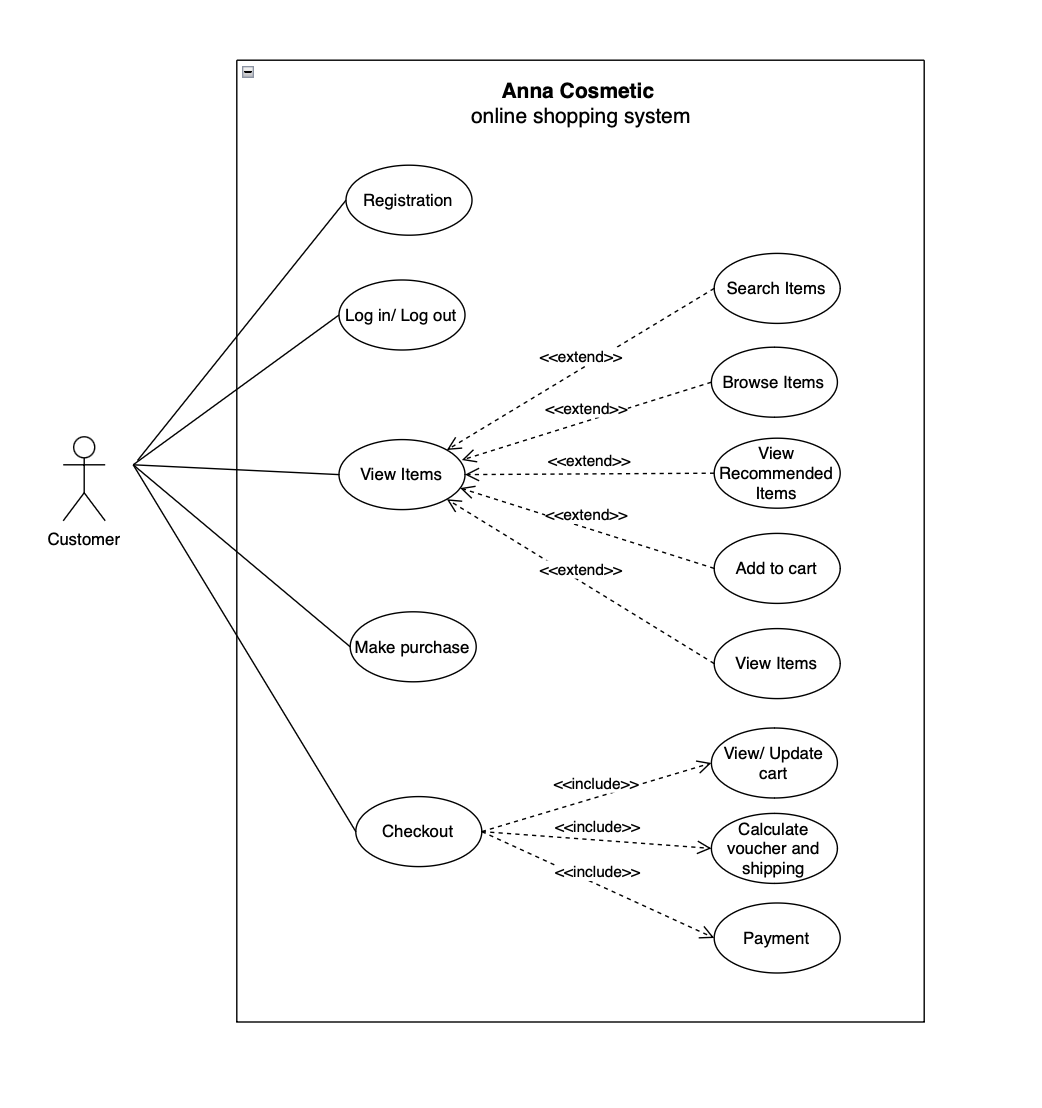


Figure . Customer use case diagram

## Database

### Database for customer

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 | Varchar(100) | Customer password |
| 5 | Address | Varchar(100) | Customer address |
| 6 | PhoneNumber | Varchar(15) | 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 | Int | 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 | Int | 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 | Int | Primary key |
| 2 | OrderID | Int | Foreign key |
| 3 | ItemID | Int | Foreign key |
| 4 | Quantity | Int | Total Price |
| 5 | Price | Decimal(10,2) |  |

### Database for admin

Table . Table Admin

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Type** | **Description** |
| 1 | AdminID | Int | Primary key |
| 2 | AdminName | Varchar(50) | Admin name |
| 3 | Password | Varchar(100) | Admin password |
| 4 | Email | Varchar(100), unique | Admin email |
| 5 | PhoneNumber | Varchar(15) | Admin phone number |
| 6 | Status | Enum(‘active’, ‘locked’) | Admin status |

Table . Table Product

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Type** | **Description** |
| 1 | ProductID | Int | Primary key |
| 2 | ProductName | Varchar(100) | Product name |
| 3 | Description | Text | Product description |
| 4 | Price | Decimal(10,2) | Product price |
| 5 | ImageURL | Varchar(255) | Product image URL |
| 6 | InStock | Int | Product in stock |
| 7 | Status | Enum(‘available’, ‘out of stock’) | Product status |

Table . Table Customer

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Field Name** | **Type** | **Description** |
| 1 | CustomerID | Int | Primary key |
| 2 | CustomerName | Varchar(50) | Customer name |
| 3 | Email | Varchar(100) | Customer email |
| 4 | Password | Varchar(100) | Customer password |
| 5 | Address | Varchar(255) | Customer address |
| 6 | PhoneNumber | Varchar(15) | Customer phone number |
| 7 | Status | Enum(‘active’, ‘locked’) | Customer status |

Table . Table Order

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

Table . Table OrderDetail

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

### ERD diagram

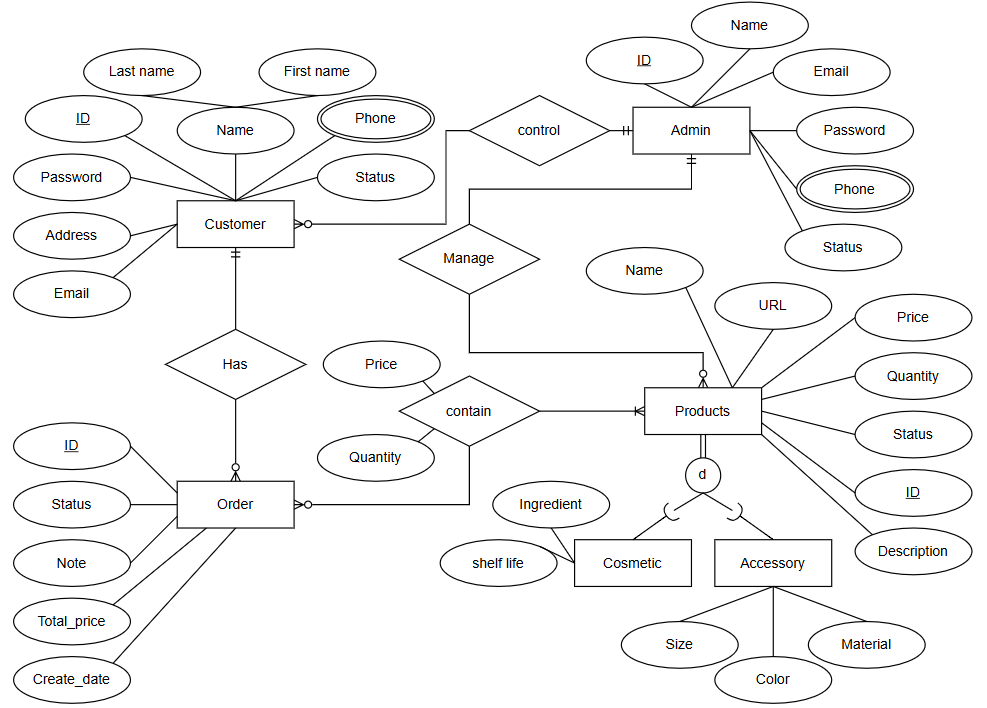


Figure . ERD diagram

The database for “Anna Cosmetic” is structured to efficiently manage the various entities involved in the e-commerce operation. The key components of the database are as follows:

**1. Customer Table**

**Attributes:**

* **ID:** A unique identifier for each customer.
* **Name:** Divided into first name and last name for better personalization.
* **Phone:** Contact number of the customer.
* **Email:** Email address for communication and account management.
* **Address:** Residential or shipping address of the customer.
* **Password:** Securely stored password for account login.
* **Status:** Current activity or account status of the customer.

**Relationships:**

Customers place **orders** in the system, represented by the "Has" relationship.

**2. Admin Table**

**Attributes:**

* **ID:** A unique identifier for each admin.
* **Name:** Name of the admin managing the platform.
* **Email:** Official email for platform-related communication.
* **Phone:** Contact number for the admin.
* **Password:** Securely stored password for admin access.
* **Status:** Current activity or role status of the admin.

**Relationships:**

Admins **control** the customers and **manage** the products listed on the website.

**3. Products Table**

**Attributes:**

* **ID:** A unique identifier for each product.
* **Name:** Name of the product.
* **URL:** The URL of the product page on the website.
* **Price:** Cost of the product.
* **Quantity:** The available stock of the product.
* **Status:** Current availability status (e.g., In stock, Out of stock).
* **Description:** Detailed information about the product.

**Specialization:**

Products are divided into two categories:

* **Cosmetic:** Includes attributes like ingredients and shelf life.
* **Accessory:** Includes attributes like size, color, and material.

**Relationships:**

* Products are part of **orders** (via the "Contain" relationship).
* Managed by admins through the "Manage" relationship.

**4. Order Table**

**Attributes:**

* **ID:** A unique identifier for each order.
* **Status:** Current state of the order (e.g., Pending, Shipped, Delivered).
* **Note:** Additional notes or instructions related to the order.
* **Total Price:** The total cost of the order.
* **Create Date:** The date the order was placed.

**Relationships:**

* Customers **have** orders.
* Orders **contain** one or more products, with each product having a price and quantity.

This database structure ensures smooth management of customers, products, and orders while allowing admins to maintain control over the platform. It is designed for scalability and can handle various business operations, such as inventory tracking, order processing, and customer management efficiently.

### ERM diagram

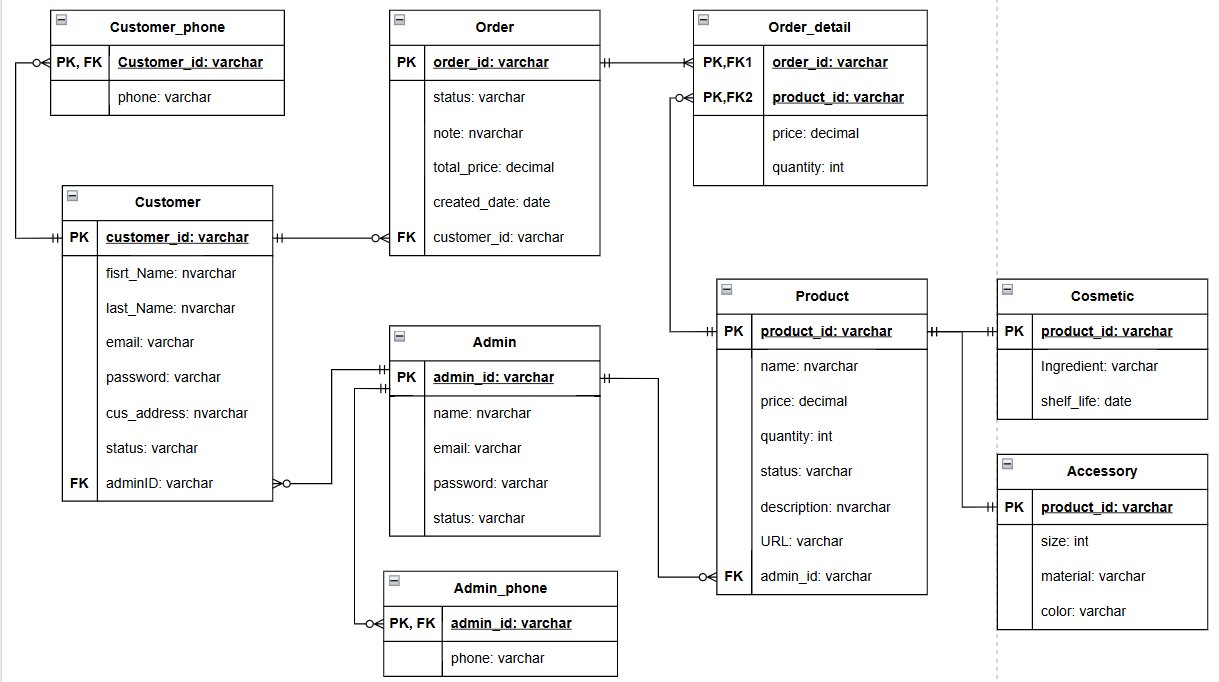


Figure . ERM diagram

The above ER model represents the database design for an e-commerce platform. It is structured to efficiently manage customers, administrators, products, orders, and product categories, ensuring data integrity and proper relationships among entities. Below is a detailed description of the entities and their relationships:

**1. Customer**

**Attributes:**

* **customer\_id (Primary Key):** Uniquely identifies a customer.
* **first\_name, last\_name:** Store the customer's full name.
* **email:** The customer's email address.
* **password:** Encrypted password for account access.
* **cus\_address:** The address of the customer.
* **status:** Indicates the account status (active/inactive).
* **adminID (Foreign Key):** Links to the Admin entity.

**Relationships:**

* Linked to the Customer\_phone table to manage multiple phone numbers for a customer.
* Connected to the Order table for tracking orders placed by customers.

**2. Admin**

**Attributes:**

* **admin\_id (Primary Key):** Uniquely identifies an administrator.
* **name:** Administrator's name.
* **email:** Contact email for the administrator.
* **password:** Encrypted password for system access.
* **status:** Indicates the status of the admin account.

**Relationships:**

* Connected to the Admin\_phone table for managing admin phone numbers.
* Related to the Product table, as administrators manage product listings.

**3. Product**

**Attributes:**

* **product\_id (Primary Key):** Uniquely identifies a product.
* **name:** The name of the product.
* **price:** The price of the product.
* **quantity:** Available stock for the product.
* **status:** Indicates whether the product is active or inactive.
* **description:** A detailed description of the product.
* **URL:** Link to the product's image or details.
* **admin\_id (Foreign Key):** Links to the Admin entity, showing which admin manages the product.

**Relationships:**

* Connected to Order\_detail for tracking products in orders.
* Inherits relationships with the Cosmetic and Accessory tables to handle product categories.

**4. Cosmetic**

**Attributes:**

* **product\_id (Primary Key):** Foreign key linking to the Product table.
* **ingredient:** Lists the ingredients of the cosmetic product.
* **shelf\_life:** Indicates the expiration date of the product.

**Relationships:**

Subclass of the Product table, providing additional attributes specific to cosmetic products.

**5. Accessory**

**Attributes:**

* **product\_id (Primary Key):** Foreign key linking to the Product table.
* **size:** Indicates the size of the accessory.
* **material:** Material used in manufacturing the accessory.
* **color:** The color of the accessory.

**Relationships:**

Subclass of the Product table, providing attributes specific to accessories.

**6. Order**

**Attributes:**

* **order\_id (Primary Key):** Uniquely identifies an order.
* **status:** Indicates the order's current status (e.g., pending, shipped, completed).
* **note:** Additional notes for the order.
* **total\_price:** Total cost of the order.
* **created\_date:** The date the order was created.
* **customer\_id (Foreign Key):** Links to the Customer entity.

**Relationships:**

Connected to Order\_detail to provide details of the products in each order.

**7. Order\_detail**

**Attributes:**

* **order\_id (Primary Key, Foreign Key):** Links to the Order table.
* **product\_id (Primary Key, Foreign Key):** Links to the Product table.
* **price:** The price of the product at the time of order.
* **quantity:** The quantity of the product ordered.

**Relationships:**

Acts as a junction table between Order and Product, allowing many-to-many relationships.

**8. Customer\_phone**

**Attributes:**

* **customer\_id (Primary Key, Foreign Key):** Links to the Customer table.
* **phone:** Stores customer phone numbers.

**Relationships:**

Allows customers to register multiple phone numbers.

**9. Admin\_phone**

**Attributes:**

* **admin\_id (Primary Key, Foreign Key):** Links to the Admin table.
* **phone:** Stores administrator phone numbers.

**Relationships:**

Allows administrators to register multiple phone numbers.

**Relationships Overview**

* The model uses one-to-many and many-to-many relationships to ensure data normalization.
* Order\_detail acts as a junction table to link Order and Product.
* Subclass entities Cosmetic and Accessory extend the Product table for specific categories.

This structure ensures modularity, scalability, and maintainability in managing customer, product, and order information.

## CRUD for Admin

### Create new category

The "Create New Category" feature in the admin panel of the Anna Cosmetic website is a dedicated functionality that allows administrators to add new cosmetic categories to the platform's inventory efficiently. Designed with user-friendliness and functionality in mind, this feature provides a seamless interface for category management, ensuring that new entries are accurately recorded and easily accessible on the customer-facing side of the website.

**Key Features and Workflow:**

**1. Intuitive Interface:**

The "Create New Category" page features a clean and minimalistic design, ensuring that administrators can focus on entering category details without unnecessary distractions. The fields are clearly labeled, making the process straightforward even for new users.

**2. Input Fields:**

* **Category Name:** A dropdown or text input field allows administrators to specify or select the category under which the new category falls, ensuring proper classification for customer searches and filters.
* **Description:** A text area is provided for administrators to add a detailed description of the category, including key features, benefits, and usage instructions. This field supports multiline input, allowing for comprehensive details.
* **Price:** An input field dedicated to entering the price of the category. The system may validate this field to ensure only numerical values are accepted, preventing data entry errors.
* **In Stock:** Administrators can specify the current stock quantity of the category, ensuring accurate inventory management and avoiding overselling.
* **Status:** A dropdown or text field to define the category’s availability status, such as "Available," "Out of Stock," or any other predefined status options. This helps in managing category visibility on the website.

**3. Validation and Error Handling:**

The system incorporates validation mechanisms to ensure that all mandatory fields are completed and contain appropriate data formats. For instance:

* Price must be numeric.
* Mandatory fields like Category Name and Price cannot be left empty. Error messages or alerts are displayed if validation fails, guiding the administrator to resolve issues promptly.

**4. Action Buttons:**

* **Create:** Once all the details are filled in, clicking the "Create" button submits the information to the database. This action triggers backend processes that validate and save the new category while making it visible on the customer side.
* **Cancel:** The cancel button allows administrators to exit the page or discard any entered information without saving, providing flexibility and preventing unintended submissions.

**5. Responsive Design:**

The feature is fully responsive, ensuring that administrators can add new categories from various devices, including desktops, tablets, or mobile phones, without compromising functionality or user experience.

**6. Integration with Backend:**

Upon submission, the entered data is processed and stored in the website's database. This seamless integration ensures that the new category immediately reflects on the inventory list, ready to be displayed on the storefront for customers.

**Benefits:**

* **Efficient Category Management:** Administrators can easily expand the category catalog, keeping the inventory updated with the latest offerings.
* **Enhanced User Experience:** Accurate and detailed category entries improve customer satisfaction by providing comprehensive information.
* **Time-Saving:** The intuitive design and streamlined workflow minimize the time required for category entry, allowing administrators to focus on other tasks.

**Overall,** the "Create New Category " feature is a cornerstone of the Anna Cosmetic admin panel, enabling effective inventory management and ensuring that the e-commerce platform remains up-to-date with the latest category for its customers.

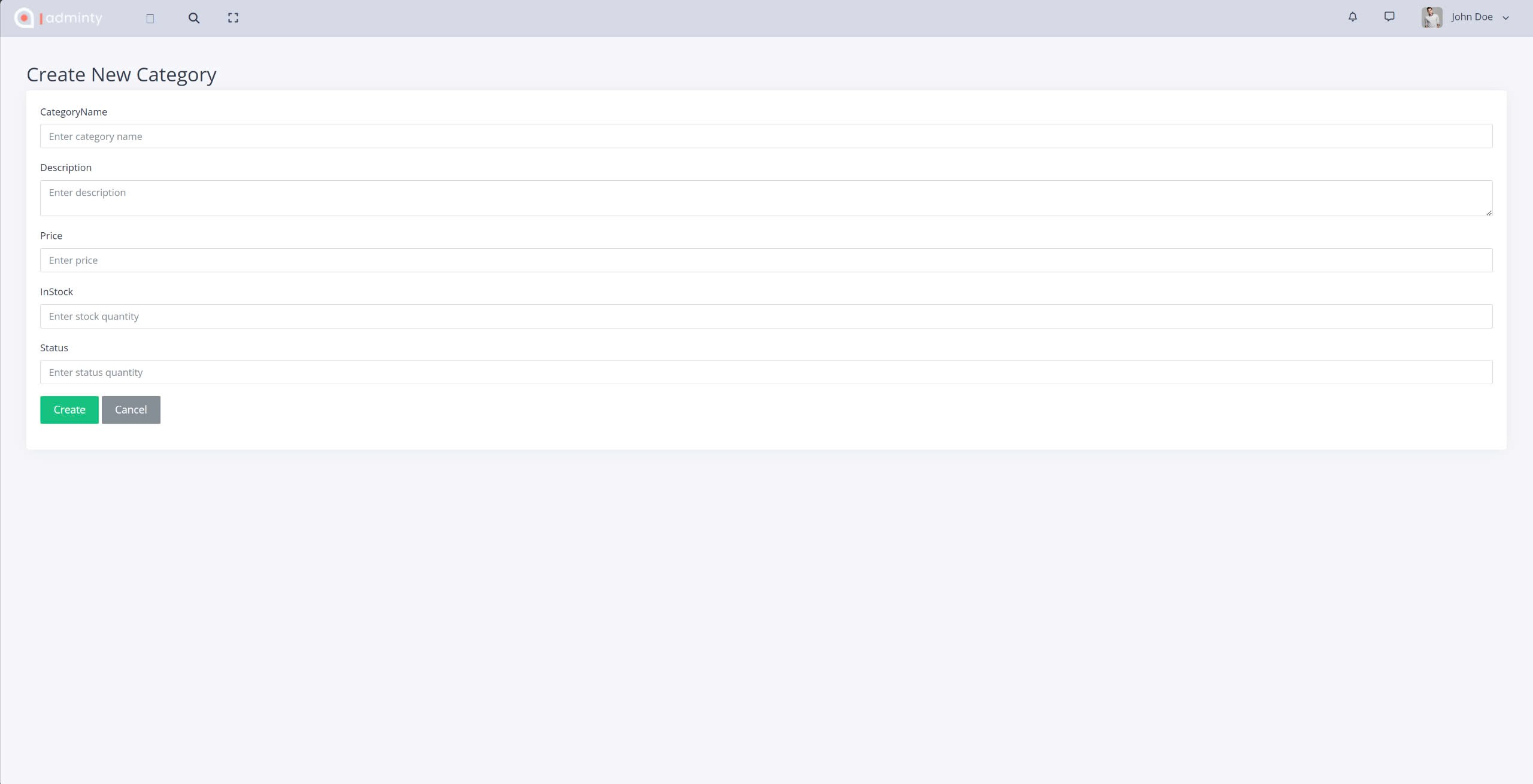


Figure . Admin create page

### Read

The "Read" feature within the admin panel of the Anna Cosmetic website serves as an essential tool for administrators to view, manage, and analyze the current categories and inventory. This functionality is designed with clarity and efficiency in mind, enabling users to monitor all category details in one comprehensive table. Its intuitive interface allows for seamless navigation and interaction, providing a central hub for category data management.

**Key Features and Workflow:**

**1. Centralized Category Overview:**

The "Read" page displays a detailed list of all category categories, offering administrators a clear view of the existing inventory. Each entry is organized into rows and columns within a structured table format for easy readability and access.

**2. Table Layout and Data Presentation:**

The table is neatly segmented into columns that display key attributes of each category:

* **ID:** A unique identifier for each category, ensuring precise reference during updates or deletions.
* **Name:** The name of the category category, making it easy to identify at a glance.
* **Description:** A brief explanation of the category or category, highlighting its features or purpose.
* **Price:** Displays the price of each category, allowing administrators to evaluate and compare prices across categories quickly.
* **In Stock:** Indicates the available quantity of each category, ensuring inventory levels are easy to track and manage.
* **Status:** Displays the availability status of each category (e.g., active, out of stock), providing real-time insights into category readiness for sale.

**3. Action Buttons:**

Each row includes actionable buttons for further management of individual categories:

* **Edit:** Clicking this button navigates administrators to a dedicated page or modal where they can modify the category details, such as updating prices, descriptions, or stock levels.
* **Delete:** Allows administrators to remove a category or category entirely from the database. A confirmation prompt may appear to prevent accidental deletions.

**4. Search and Navigation Options:**

* **Search Functionality:** Administrators can use a search bar (if available) to filter and locate specific categories quickly, saving time and effort when working with a large inventory.
* **Pagination:** For extensive inventories, the table may include pagination controls, allowing administrators to navigate through multiple pages of category data efficiently.

**5. Real-Time Data Synchronization:**

The table dynamically updates with any changes made to the category details, ensuring that the displayed data remains accurate and up-to-date. This is crucial for inventory accuracy and operational efficiency.

**6. "Create New" Button:**

Positioned prominently above the table, the "Create New" button provides a quick pathway for adding new category categories. This integration ensures administrators can seamlessly switch between viewing and creating categories without navigating to a separate section of the admin panel.

**Benefits of the "Read" Feature:**

* **Efficient Inventory Management:** By providing a clear and concise overview of all categories, this feature enables administrators to monitor inventory levels and ensure category availability without manual tracking.
* **Streamlined Decision-Making:** The accessible presentation of category details aids in making informed decisions regarding stock replenishment, pricing adjustments, or discontinuations.
* **Enhanced Productivity:** The combination of quick access to category information and integrated editing/deleting options minimizes the time spent on managing inventory.

**User Experience Considerations:**

* **Responsive Design:** The feature is optimized for all devices, allowing administrators to manage inventory effectively whether they are on a desktop, tablet, or smartphone.
* **Error Handling:** Robust error handling ensures that issues such as data retrieval failures or invalid entries are promptly flagged, with clear instructions on how to resolve them.
* **Security:** Access to the "Read" feature and its related actions is secured through role-based permissions, ensuring that only authorized personnel can view or modify category information.

**In conclusion**, the "Read" feature in the Anna Cosmetic admin panel acts as a backbone for the platform's inventory management. By presenting all essential category information in a single view, it enhances efficiency, accuracy, and user experience, empowering administrators to maintain a well-organized and up-to-date catalog.

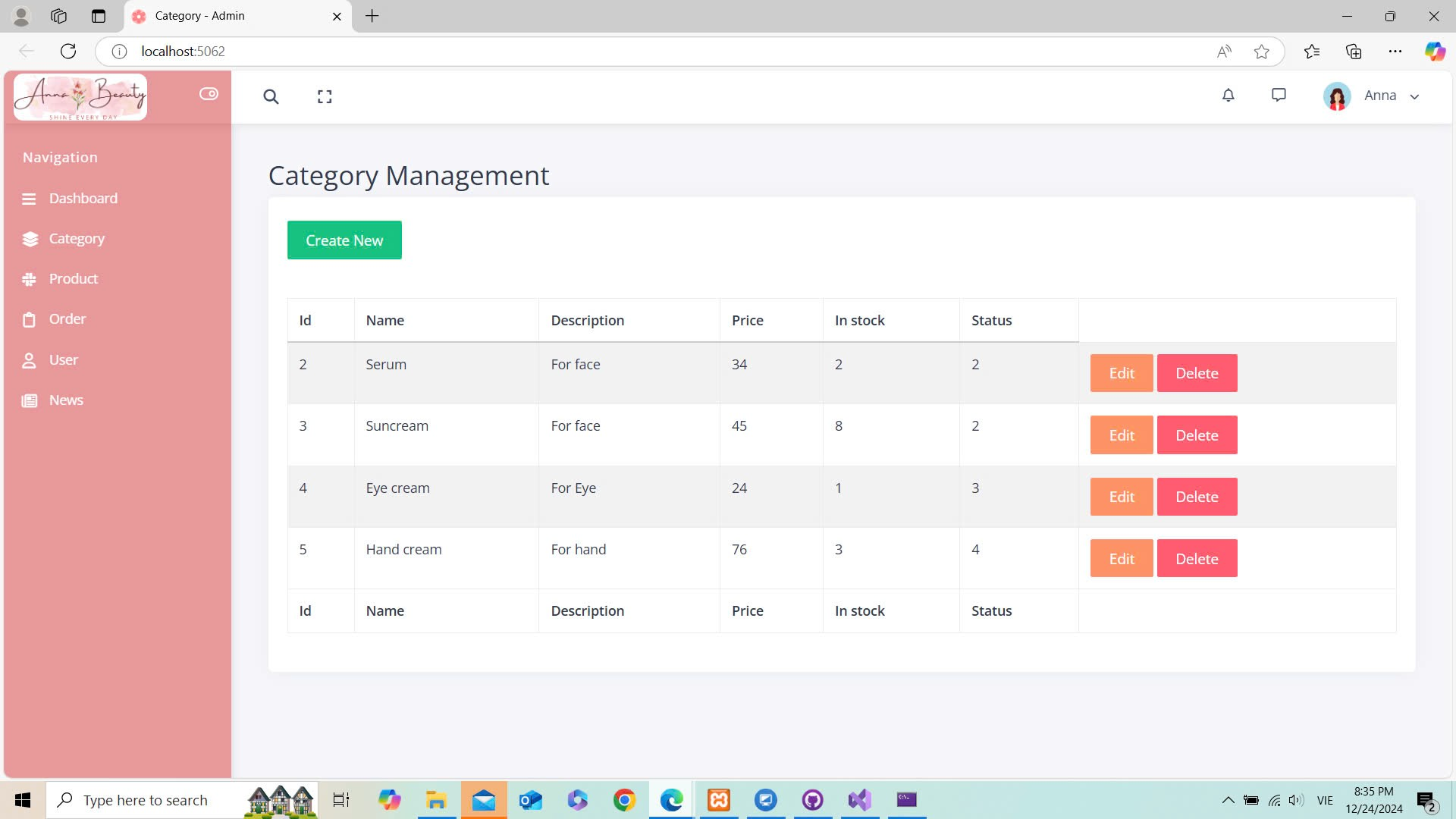


Figure . Admin read page (Home page)

### Update

The "Edit Category" feature is a key functionality in the Anna Cosmetic admin panel that allows administrators to update and modify existing product categories effectively. Designed to prioritize usability and efficiency, this feature ensures that category details remain accurate and up-to-date, which directly contributes to improved customer navigation and inventory management.

**Key Features and Workflow:**

**1. Intuitive Interface:**

The "Edit Category" page is designed with simplicity and clarity in mind. Each input field is clearly labeled, ensuring administrators can easily understand the required information and make precise updates without distractions.

**2. Input Fields:**

The page provides specific fields for updating category details, such as:

* **Category Name:** A text input field for editing the name of the category to ensure clarity and relevance.
* **Description:** A multiline text area for providing or updating a detailed description of the category. This description can include an overview of the types of products the category includes, enabling better organization.
* **Price Range (Optional):** If applicable, an input field for adjusting the price range or baseline price of items in the category.
* **In Stock:** A field to update the total number of products available in this category, helping to manage inventory effectively.
* **Status:** A dropdown or text field to update the status of the category, such as "Active," "Inactive," or other custom-defined states, to control its visibility on the customer-facing platform.

**3. Validation and Error Handling:**

The system includes robust validation checks to ensure data integrity. For example:

* Fields such as "Category Name" and "Description" must not be empty.
* Numerical fields, like "In Stock," accept only valid numbers.
* Any invalid entries trigger clear error messages, guiding administrators to make corrections.

**4. Action Buttons:**

* **Save:** The "Save" button submits the updated information to the system's backend. Upon successful validation, the changes are saved, and the updated category details are reflected in the database and on the website.
* **Cancel:** If the administrator decides not to proceed, they can exit or discard changes by navigating away from the page or using a cancel button (if available).

**5. Real-Time Feedback:**

After saving, the system provides immediate feedback, such as:

* A success message: “Category updated successfully.”
* Notifications in case of issues, such as a conflict with existing data or invalid input formats.

**6. Responsive Design:**

The feature is fully optimized for various devices, ensuring administrators can edit categories on desktops, tablets, or mobile phones without losing functionality or ease of use.

**7. Integration with Backend:**

When the updated details are saved, the backend system processes the data and updates the database in real-time. These changes are then propagated to the live platform, ensuring consistency between the admin panel and the customer-facing website.

**Benefits:**

* **Efficient Category Management:** Enables administrators to keep category details accurate and relevant, improving customer navigation and search functionality.
* **Error Reduction:** Validation mechanisms prevent incorrect or incomplete data entry, maintaining the integrity of the database.
* **Time-Saving:** The intuitive interface and seamless workflow minimize the effort required for category updates, allowing administrators to focus on other tasks.
* **Enhanced Customer Experience:** Updated and accurate category details ensure a smoother shopping experience for customers, improving overall satisfaction.

Conclusion, the "Edit Category" feature is an essential tool in the Anna Cosmetic admin panel, streamlining the process of updating category information while ensuring accuracy and reliability. By maintaining organized and up-to-date categories, the platform enhances both operational efficiency and the overall shopping experience for customers.

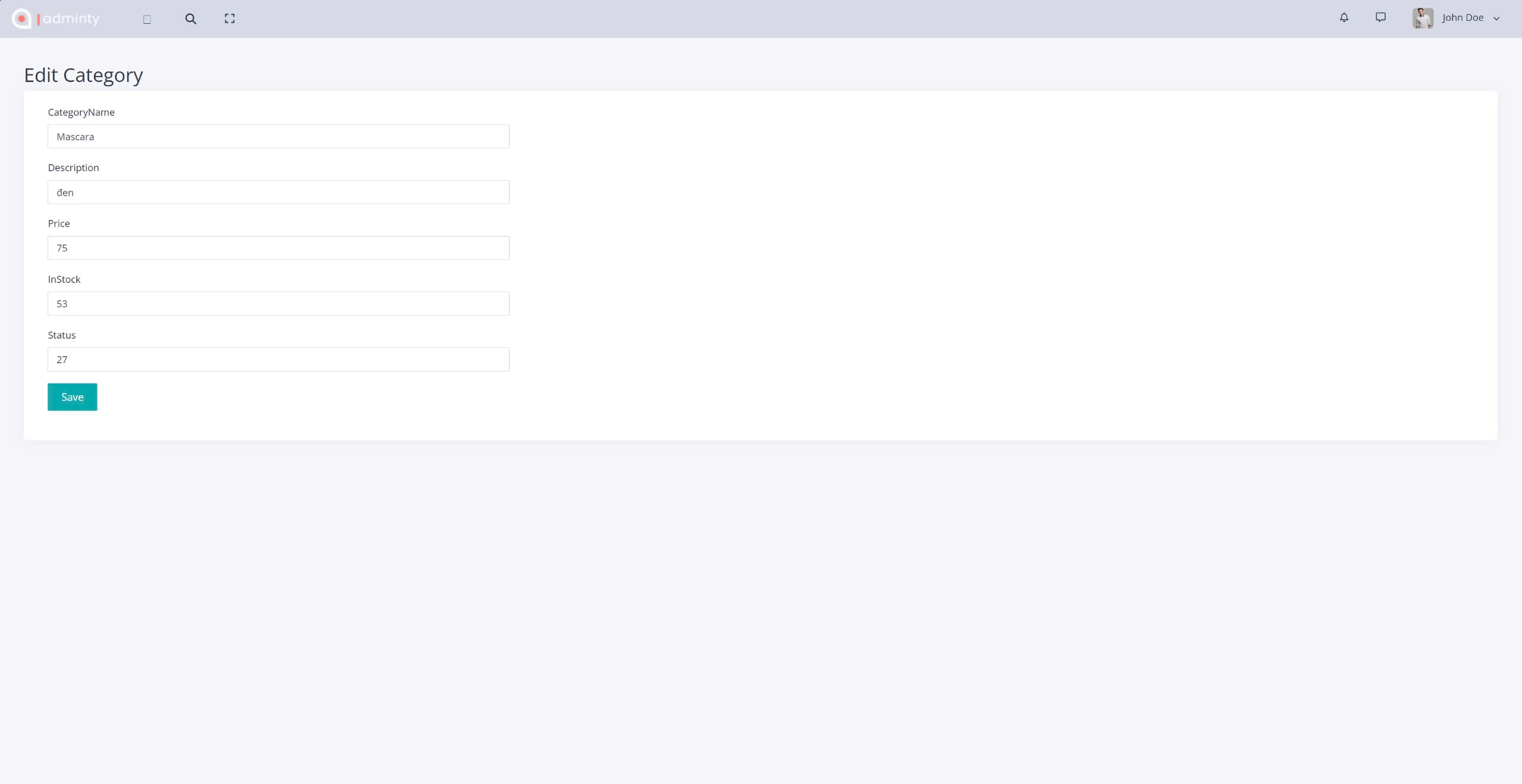


Figure . Admin update page (edit page)

### Delete

The "Delete Category" feature is a vital functionality within the Anna Cosmetic admin panel, designed to allow administrators to efficiently remove product categories from the platform's inventory. With an emphasis on user-friendliness and safety, this feature ensures seamless management of product classifications while preventing accidental data loss.

**Key Features and Workflow:**

**1. Intuitive Interface:**

The "Delete Category" feature is embedded in the product category management page. Each category row includes a clearly labeled "Delete" button in red, ensuring easy visibility and accessibility for administrators.

**2. Confirmation Modal:**

When the "Delete" button is clicked, a confirmation modal appears, displaying:

* A message: “Are you sure you want to delete the following category?”
* Details of the category, including its ID, name, and description.

This step ensures administrators can review and verify the deletion request before proceeding.

**3. Validation and Error Handling:**

The system incorporates validation and error-checking mechanisms to handle potential issues, including:

* Preventing the deletion of categories linked to active products.
* Displaying clear error messages if an attempt to delete such a category is made, guiding the administrator on how to resolve the issue.

**4. Action Buttons:**

* **Yes, Delete:** Confirms the deletion, removes the category from the database, and displays a success message: “Category deleted successfully.”
* **Cancel:** Allows administrators to close the modal without performing any action, providing flexibility and minimizing unintended deletions.

**5. Responsive Design:**

The feature is fully responsive, ensuring smooth functionality across various devices, including desktops, tablets, and mobile phones.

**6. Integration with Backend:**

Once the administrator confirms the deletion, the system processes the request, updates the database, and ensures the category is no longer displayed in the inventory.

**Benefits:**

* **Efficient Category Management:** Streamlines the process of removing outdated or unnecessary categories, ensuring the platform remains organized and relevant.
* **Data Accuracy:** Prevents clutter and maintains a clean inventory for both administrators and customers.
* **Error Prevention:** The confirmation step and validation mechanisms reduce the risk of accidental deletions or data loss.
* **Time-Saving:** The intuitive interface and seamless workflow minimize administrative effort and allow users to focus on higher-priority tasks.

Conclusion, the "Delete Category" feature is an essential component of the Anna Cosmetic admin panel, enabling administrators to manage product categories effectively and maintain an up-to-date, user-friendly platform for customers. Through its robust design and error-handling capabilities, this feature enhances operational efficiency while ensuring data integrity.

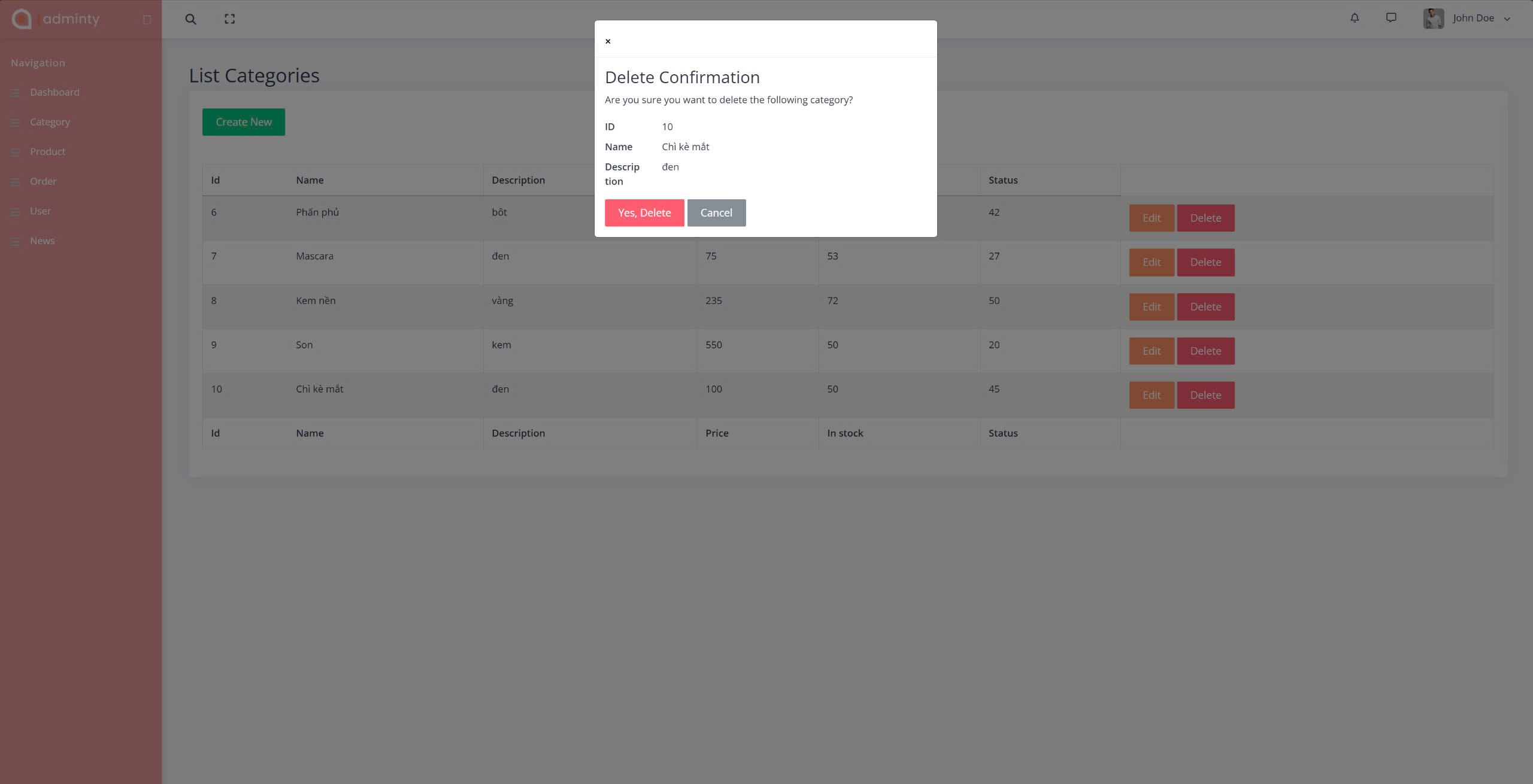


Figure . Admin delete page

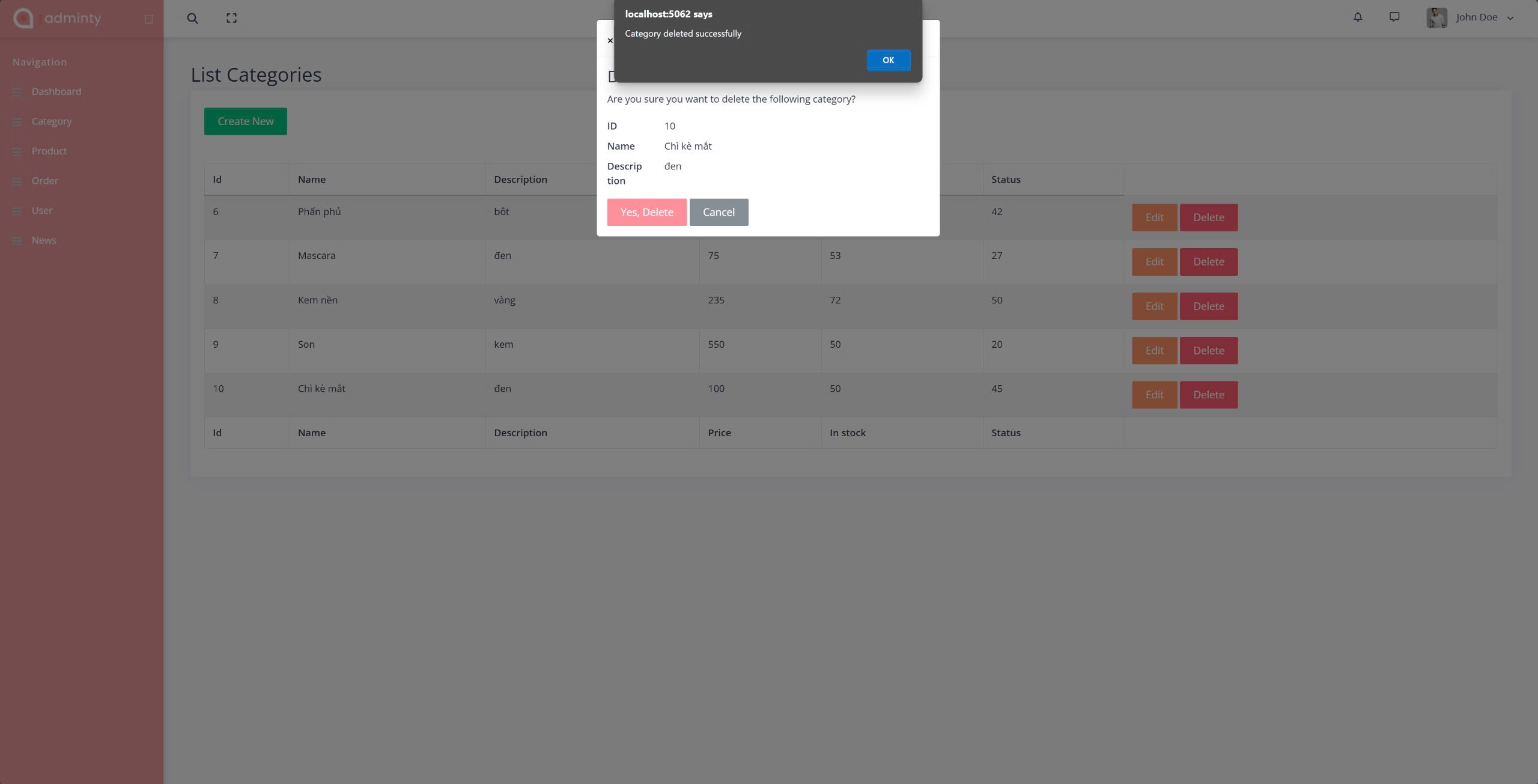


Figure . Admin delete successfully page

# EXPERIMENT AND DISCUSSION

This chapter presents the results of the implemented features and their evaluation. It includes the analysis of key interfaces such as the Home Page, Sign Up, Sign In, and Shopping Cart, followed by a discussion on the system’s design, usability, and performance.

## Installation Environment

Visual Studio Code Version: 1.93.1 (Universal)

## Results and Discussion

### Results

#### Home Page

The Home Page delivers a well-structured and visually appealing design, ensuring an intuitive and engaging user experience.

The header section includes a prominent search bar with suggested keywords for quick navigation, alongside a clear menu and icons for user account and shopping cart. The central hero banner features a high-quality image with the message "Be Your Kind of Beauty" and a prominent "Shop Now" button, effectively capturing user attention.

Key product sections such as Shop Categories, Bestsellers, and Special Offers highlight products with images, prices, discounts, and ratings. A dedicated countdown timer in the Special Offer section encourages urgency, while the Under $25 section caters to budget-conscious shoppers.

The "Why Shop with Anna?" section builds user trust by emphasizing core selling points: Guaranteed PURE, Cruelty-Free, and Ingredient Sourcing. The footer includes promotional banners, a newsletter subscription, and useful links like About Us, FAQs, and Shipping Policies.

The responsive layout ensures a seamless experience across both desktop and mobile devices. Overall, the Home Page combines aesthetics and functionality, guiding users effectively through product discovery and enhancing engagement with features like search optimization, clear CTAs, and product highlights. Minor improvements, such as better text contrast on the banner and image loading optimization, could further refine the user experience.

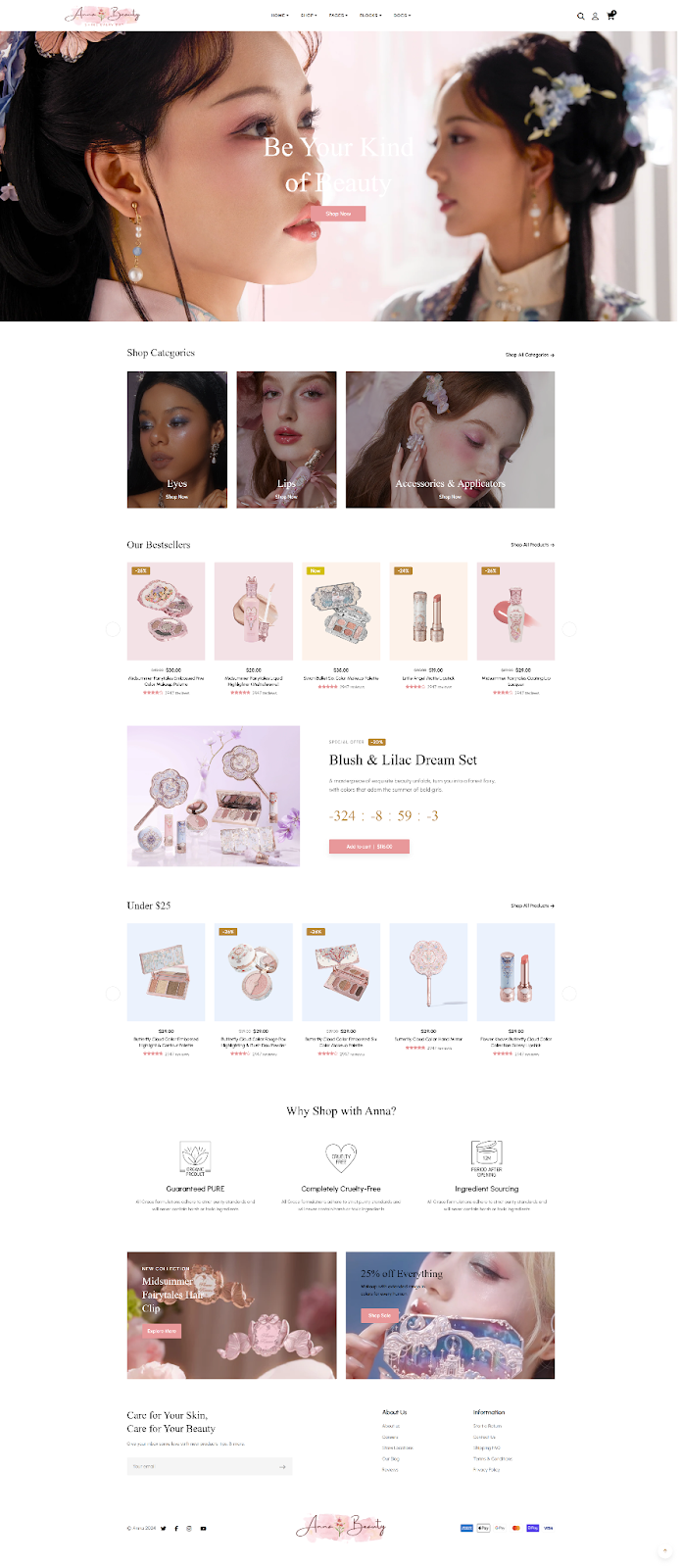


Figure . Home page

#### Sign Up

The Sign Up interface includes input fields for First Name, Last Name, Email, and Password, along with a checkbox to agree to the Privacy Policy and Terms of Use. A prominent Sign Up button is displayed, and users can also register quickly using Facebook or Google accounts. The form is presented in a popup modal, ensuring it overlays the current page without disrupting the user experience.

The design is clean, user-friendly, and responsive across multiple devices, including mobile. Input validation prevents invalid data, such as incorrectly formatted emails or passwords, while the quick sign-up options improve user convenience. However, the interface could benefit from additional features like error messages for invalid input and a show/hide password option to further enhance usability.

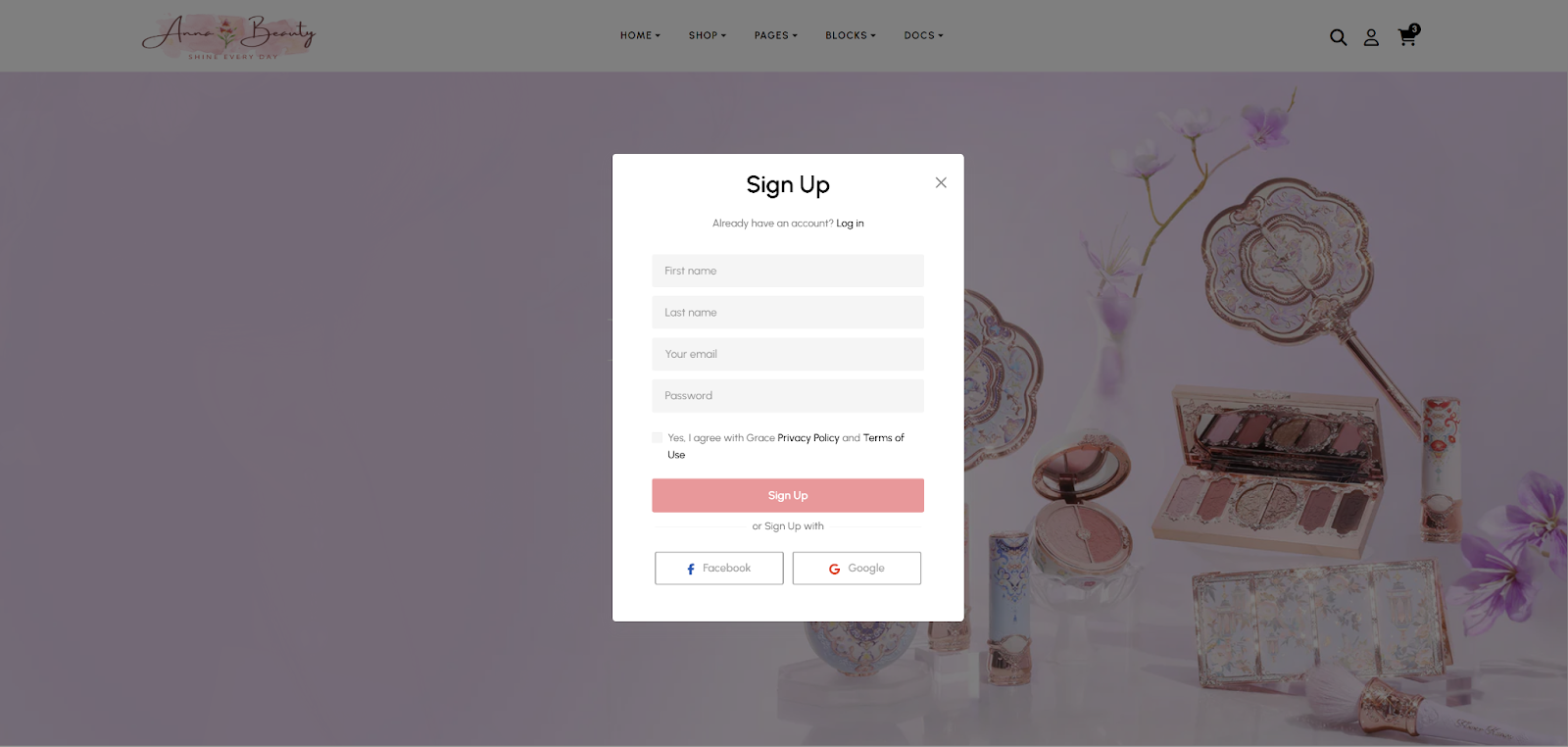


Figure . Sign up page

#### Sign In

Similar to the Sign Up form, the Sign In interface uses a popup modal design with input fields for Email and Password, a "Stay signed in" option, and a Forgot Password link. Quick login options via Facebook and Google enhance convenience. The interface is displayed as a popup modal, keeping users on the current page. The design is clean, user-friendly, and responsive across devices, ensuring a smooth experience on both desktop and mobile.

While the interface works efficiently, additional features like a password visibility toggle and inline error messages could further improve usability. Overall, the design provides a seamless and effective login process.

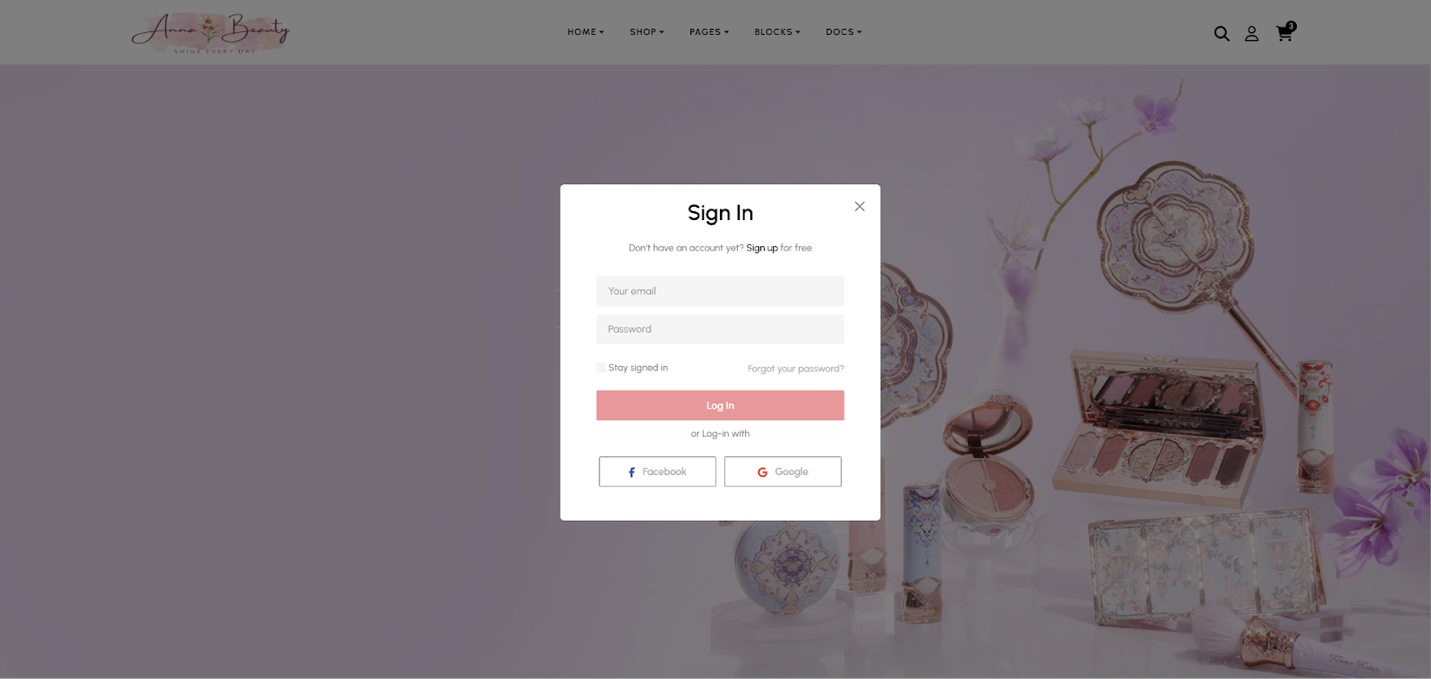


Figure . Sign in page

#### Shopping Cart

The Shopping Cart interface provides a seamless and intuitive experience for managing selected products while encouraging timely purchases. It displays product details such as thumbnail images, product names, prices (both original and discounted), and quantity controls with easy-to-use +/- buttons for adjustments. A dynamically updated total price is shown clearly at the bottom, allowing users to track the overall value of their cart.

A notable feature is the cart timer, which informs users their cart will be saved for a limited time (e.g., 4 minutes 34 seconds), creating a sense of urgency and encouraging faster decisions. Prominent action buttons like "Check Out" and "View Shopping Cart" enhance user convenience by allowing them to finalize or review purchases with ease.

The clean and responsive design ensures consistent usability across desktop and mobile devices, with product details neatly organized for readability. The quantity control feature simplifies real-time updates without disrupting the flow, while the cart layout keeps essential information visible and accessible.

Overall, the Shopping Cart combines functionality and aesthetics to deliver an efficient and user-friendly experience. The addition of a timer improves engagement, while features like clear pricing and accessible controls help guide users smoothly through the checkout process. Minor enhancements, such as bulk item removal or related product recommendations, could further elevate the shopping experience.

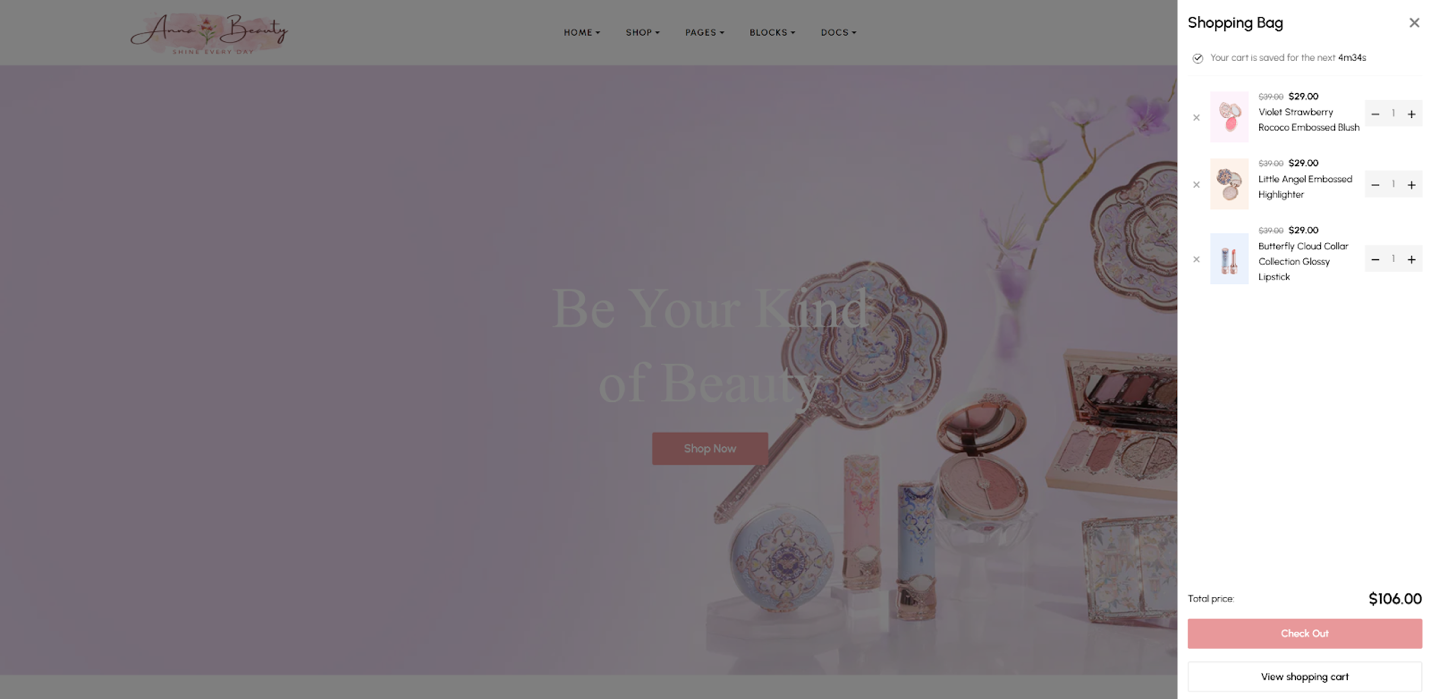


Figure . Shopping cart page

### Discussion

The implemented interfaces, including the Home Page, Sign Up, Sign In, and Shopping Cart, demonstrate a clean, intuitive, and responsive design, successfully delivering a user-friendly experience across devices. The Home Page effectively combines aesthetics and functionality with a prominent search bar, engaging product sections like Bestsellers and Special Offers, and a visually captivating hero banner. Features such as a countdown timer and organized product highlights encourage user interaction and improve navigation.

The Sign Up and Sign In interfaces utilize a popup modal design to enhance convenience without disrupting the current page view. Input validation ensures data accuracy, while quick login options via Facebook and Google improve accessibility. However, these forms could benefit from additional features, such as show/hide password toggles and inline error messages for invalid inputs, to further optimize usability.

The Shopping Cart interface stands out with its clean layout and functionality, displaying product details clearly along with real-time quantity controls and a dynamically updated total price. The addition of a cart timer creates urgency, encouraging users to complete purchases quickly. Despite its efficiency, the cart could be further improved by incorporating a bulk removal feature and personalized product recommendations to enhance user engagement and satisfaction.

Overall, the system meets its objectives of providing a visually appealing and functional user experience. The responsive design ensures consistent performance across desktop and mobile devices. Minor improvements, such as optimizing text contrast on the banner, improving image loading speeds, and adding advanced form features, would further refine the system and elevate its usability and performance.

# 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:**

I learned to utilize web development frameworks such as ASP.NET for backend operations and SQL Server 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:**

**Home Page:** Visually appealing design, search bar with suggested keywords, and product sections for easy navigation.

**Sign Up / Sign In:** User authentication with input validation and social login via Facebook and Google.

**Shopping Cart:** Product management with real-time quantity controls, dynamic pricing updates, and a countdown timer for creating urgency.

**Admin Management:** Gained additional skills in designing an Admin Dashboard using ASP.NET to manage products, categories. Connected the system to SQL Server for secure and efficient data storage.

The final product ensures a seamless and user-friendly experience while balancing aesthetics, performance, and functionality across desktop and mobile platforms.

**Skills Developed:**

Enhanced technical skills in database management, including creating and managing tables, queries, and stored procedures in SQL Server. 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.

**Personal Growth:**

This project provided an opportunity to apply theoretical knowledge to real-world development processes. I learned to analyze problems, find solutions, and develop user-centric features, preparing me for more advanced software development projects in the future.

## 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. ASP.NET Core in Action, Third Edition, Andrew Lock

APPENDIX

Your content.

**REPORT CONTENT AND STRUCTURE**

Abstract

Acknowledgement

Table of Content

List of Figures

List of Tables

List of Abbreviations

1. Introduction
2. Related Works
3. Proposed Approach, Analysis, Design, Implementation
4. Results and Discussion
5. Conclusion and Future Works

References  
Appendix

**GUIDELINES**

1. Report cover page is WHITE
2. Chapter TITLE: Times New roman, size 14 points, bold, CENTER
3. Section Title: Times New roman, size 13 points, bold
4. Sub-section Title: Times New roman, size 12 points, bold
5. Texts: Times New roman, size 12 points, Left and Right justified
6. Line spacing: 1.5
7. Printing of Page: Single sided with page number (Bottom, Center) of each page
8. Diagrams: Colored (using colored printer)
9. Introduction: Should include project statement, background and motivation.
10. Reference Format Example:
11. E. H. E. Bayoumi, M. N. F. Nashed, *A Fuzzy Predictive Sliding Mode Control for High Performance Induction Motor Position Drives*, Journal of Power Electronics, Vol. 5, No.1, pp.20-28, 2005.
12. B. K. Bose, *Power electronics and variable frequency drives: Technology and Applications,* IEEE Press, New York, 1996.
13. H. N. Duong, V. N. Nguyen, H. H. Lee. *Control of induction motor using IMC approach*. ICPE’07 Conference in Daegu, Korea, pp.15-21. 2007.