K S RANGASAMY COLLEGE OF TECHNOLOGY

(Autonomous)

TIRUCHENGODE-637215

A MINI PROJECT REPORT

E-COMMERSE PRODUCT CATALOG

60 IT L04 – C# AND .NET FRAMEWORK

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

Submitted by

PRAVEENKUMAR V (73772214179)

K S RANGASAMY COLLEGE OF TECHNOLOGY

(Autonomous)

TIRUCHENGODE-637215

BONAFIDE CERTIFICATE

Certified that this project report titled "**E-COMMERSE PRODUCT CATALOG**" is the bonafide work of **P R A V E E N K U M A R V (73772214179)** who carried out the project under my guidance.

Dr. S. Madhavi M.E., Ph.D.,
Professor and Head of the Department
Department of Computer Science and Engineering
K.S. Rangasamy College of Technology
Tiruchengode-637 215.

Mrs. N. Sathiyapriya M.E.,
Assistant Professor
Department of Information Technology
K.S. Rangasamy College of Technology
Tiruchengode - 637 215.

ABSTRACT

The **E-commerce Product Catalog** is an online system designed to manage and display a wide range of products for e-commerce platforms. It allows users to browse through a catalog of products, view detailed information, and search for specific items based on criteria like name or category. The system features product details such as name, price, category, and unique product IDs, and is designed to provide an organized, searchable interface for users to easily navigate through available products. The catalog aims to enhance the shopping experience by presenting products in a structured and accessible manner.

This system also supports the addition of new products, where admins can input product information such as name, category, and price, with the system automatically assigning a unique ID to each product. Users can search for products using keywords and view relevant results, making it easier to find desired items. The E-commerce Product Catalog is a scalable solution, suitable for businesses looking to manage a diverse product range, and can be extended with additional features such as inventory management, user reviews, or integration with shopping carts for a complete online shopping experience.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The E-commerce Product Catalog is an online platform that organizes and presents a wide range of products for users to browse, search, and purchase. It displays product information, including names, categories, prices, and unique product IDs, making it easy for customers to find and explore products of interest. The catalog serves as the foundation for e-commerce websites, providing a seamless browsing experience for customers while also enabling businesses to efficiently manage their product inventory and offerings. The system is designed to ensure that users can search for products based on categories or keywords, improving the shopping experience by making it simple to discover desired items.

Beyond simply displaying products, the catalog allows administrators to add new items, update existing product details, and remove outdated products. This functionality makes it ideal for businesses that frequently add new products or need to maintain up-to-date listings. With its user-friendly interface, the E-commerce Product Catalog enhances both customer satisfaction and business operations by streamlining product management, and can be extended with additional features like customer reviews, payment processing, or inventory tracking to create a complete e-commerce solution.

CHAPTER 2

REQUIREMENT ANALYSIS

Functional Requirements:

1. **Add a Product**: Users can create new product entries with details such as name, description, price, and category.
2. **Edit a Product**: Users can modify existing product details.
3. **Delete a Product**: Users can remove outdated or unwanted product entries.
4. **Display Products**: Products are displayed in a **DataGridView** for easy viewing and management.
5. **Clear Fields**: The application allows users to reset input fields to quickly add or update products.

Non-Functional Requirements:

1. **User-Friendly Interface**: The application provides a clean, intuitive, and easy-to-navigate interface for users.
2. **Responsiveness**: Actions like adding, editing, and deleting products are processed in real-time, reflecting changes immediately.
3. **Scalability**: The application is designed to support future features such as product filtering, sorting, and stock management.

Tools and Technologies:

- **Programming Language**: C#
- **Framework**: .NET Framework (Windows Forms)
- **Integrated Development Environment (IDE)**: Visual Studio

CHAPTER 3

DESIGN AND IMPLEMENTATION

System Overview:

Graphical User Interface (GUI):

1. A **DataGridView** control displays the list of products in a tabular format, showing details like name, category, price, and description.
2. **Text fields (TextBox)** are used for inputting product details such as name, price, and description.
3. **Dropdowns (ComboBox)** allow users to select categories.
4. **Buttons (Button)** perform actions such as adding, saving, editing, and deleting product entries.

Data Management:

1. A **DataTable** acts as an in-memory storage structure for product data.
2. **Binding** is used to connect the DataTable with the DataGridView, enabling real-time updates.

Code Summary:

1. **Form Load**: Initializes the DataTable and sets up the binding between the DataTable and the DataGridView to display products.
2. **New Product**: Clears all input fields to allow entry of new product details.
3. **Edit Product**: Loads the selected product's details into input fields for modification.
4. **Delete Product**: Removes the selected product entry from the DataTable.
5. **Save Product**: Adds a new product or updates existing product details in the DataTable.

CHAPTER 4

CODE

```
using System;

using System.Collections.Generic;



class Product

{

    public int Id { get; set; }

    public string Name { get; set; }

    public string Category { get; set; }

    public double Price { get; set; }



    public override string ToString()

    {

        return $"ID: {Id}, Name: {Name}, Category: {Category}, Price:
${Price:F2}";

    }

}
```

```
class ProductCatalog
{
    static List<Product> products = new List<Product>();
    static int nextId = 1; // Auto-incremented product ID

    static void Main()
    {
        Console.WriteLine("Welcome to the E-commerce Product Catalog!");

        // Add some sample products
        products.Add(new Product { Id = nextId++, Name = "Laptop", Category = "Electronics", Price = 1000 });
        products.Add(new Product { Id = nextId++, Name = "Smartphone", Category = "Electronics", Price = 800 });
        products.Add(new Product { Id = nextId++, Name = "Headphones", Category = "Accessories", Price = 50 });

        bool running = true;
```

```
while (running)
{
    Console.WriteLine("\nChoose an option:");
    Console.WriteLine("1. View Products");
    Console.WriteLine("2. Add Product");
    Console.WriteLine("3. Search Product");
    Console.WriteLine("4. Exit");
    Console.Write("Enter your choice: ");

    string choice = Console.ReadLine();
    switch (choice)
    {
        case "1":
            ViewProducts();
            break;
        case "2":
            AddProduct();
            break;
```

```
            case "3":
                SearchProduct();
                break;
            case "4":
                running = false;
                Console.WriteLine("Thank you for using the Product Catalog.
Goodbye!");
                break;
            default:
                Console.WriteLine("Invalid choice. Please try again.");
                break;
        }
    }
}

static void ViewProducts()
{
    Console.WriteLine("\nProduct Catalog:");
```

```
    if (products.Count == 0)
    {
        Console.WriteLine("No products available.");
    }
    else
    {
        foreach (var product in products)
        {
            Console.WriteLine(product);
        }
    }
}

static void AddProduct()
{
    Console.WriteLine("\nAdd a New Product:");

    Console.Write("Enter product name: ");
```

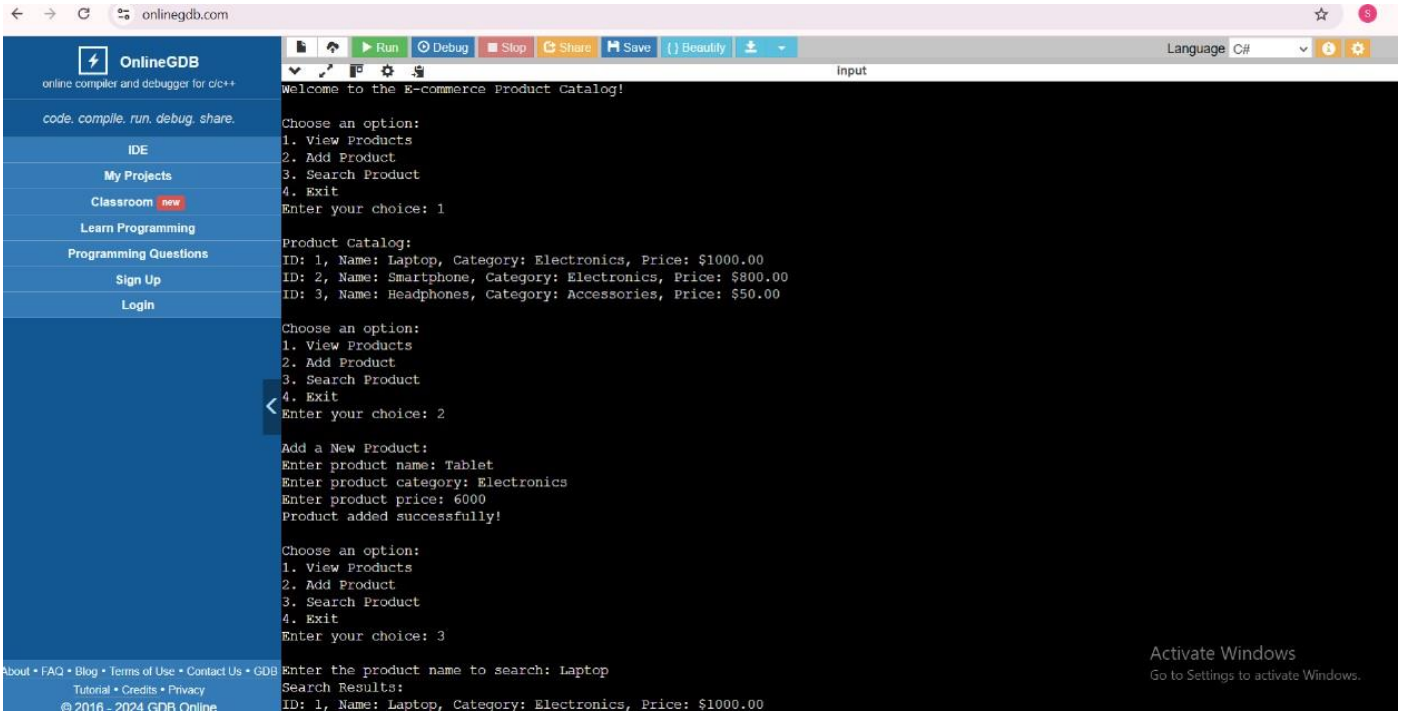
```
string name = Console.ReadLine();


Console.Write("Enter product category: ");
string category = Console.ReadLine();


Console.Write("Enter product price: ");
if (double.TryParse(Console.ReadLine(), out double price))
{
    Product newProduct = new Product
    {
        Id = nextId++,
        Name = name,
        Category = category,
        Price = price
    };
    products.Add(newProduct);
    Console.WriteLine("Product added successfully!");
}
```

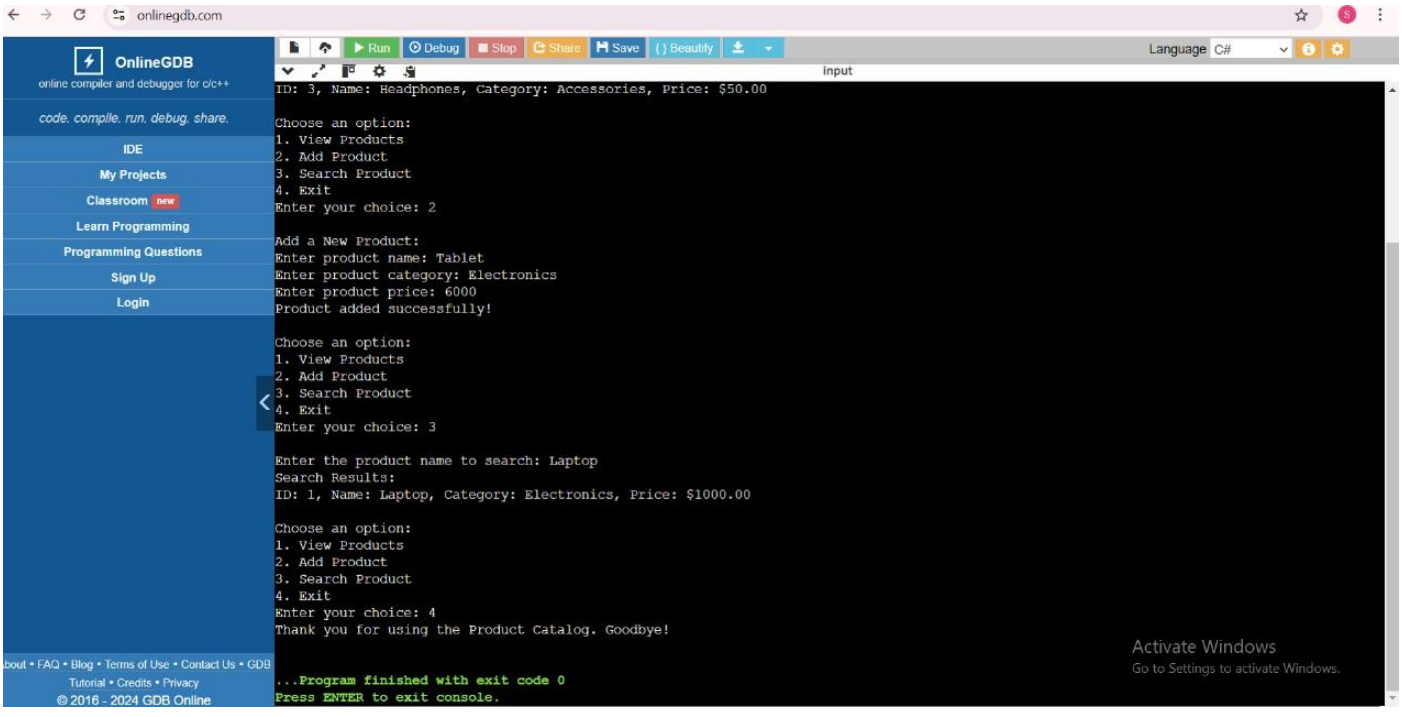
```
        else
        {
            Console.WriteLine("Invalid price. Product not added.");
        }
    }

    static void SearchProduct()
    {
        Console.Write("\nEnter the product name to search: ");
        string searchQuery = Console.ReadLine();

        var foundProducts = products.FindAll(p => p.Name.Contains(searchQuery,
StringComparison.OrdinalIgnoreCase));

        if (foundProducts.Count > 0)
        {
            Console.WriteLine("Search Results:");
            foreach (var product in foundProducts)
```

```
            {
                Console.WriteLine(product);
            }
        }
        else
        {
            Console.WriteLine("No products found matching your search.");
        }
    }
}
```

CHAPTER 5

OUTPUT





CHAPTER 6

CONCLUSION

The **E-commerce Product Catalog** project provides a solid foundation for managing and displaying products in an organized manner on e-commerce platforms. With the ability to add, search, and filter products, the system offers a smooth and efficient browsing experience for customers. The project focuses on easy product management, allowing businesses to keep their product listings up-to-date. The catalog's scalability allows for future integration of additional features such as customer reviews or inventory tracking. Ultimately, this project is an essential tool for businesses looking to enhance their online presence and create an effective and organized product catalog for their customers.

CHAPTER 7

REFERENCES

- "Building an E-Commerce Website with ASP.NET Core" Link: https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0
- "How to Build an E-Commerce Website with C#" Link: https://www.c-sharpcorner.com/
- "Creating an E-Commerce Product Catalog Using .NET Core" Link: https://medium.com/