

# MiniMart - ITI .NET Final Project

**Training Name:** ITI Summer Training 2025 Web Development Using .Net G1

**Instructor:** Eng. Mohammed Abdullatif

---

## Personal Information:

**Name:** Malak Hussein Ahmed Mohamed

**University:** Fayoum University

**Faculty:** Faculty of Computers and Artificial Intelligence

**Academic Level:** Up to Level 4

---

## Project Summary:

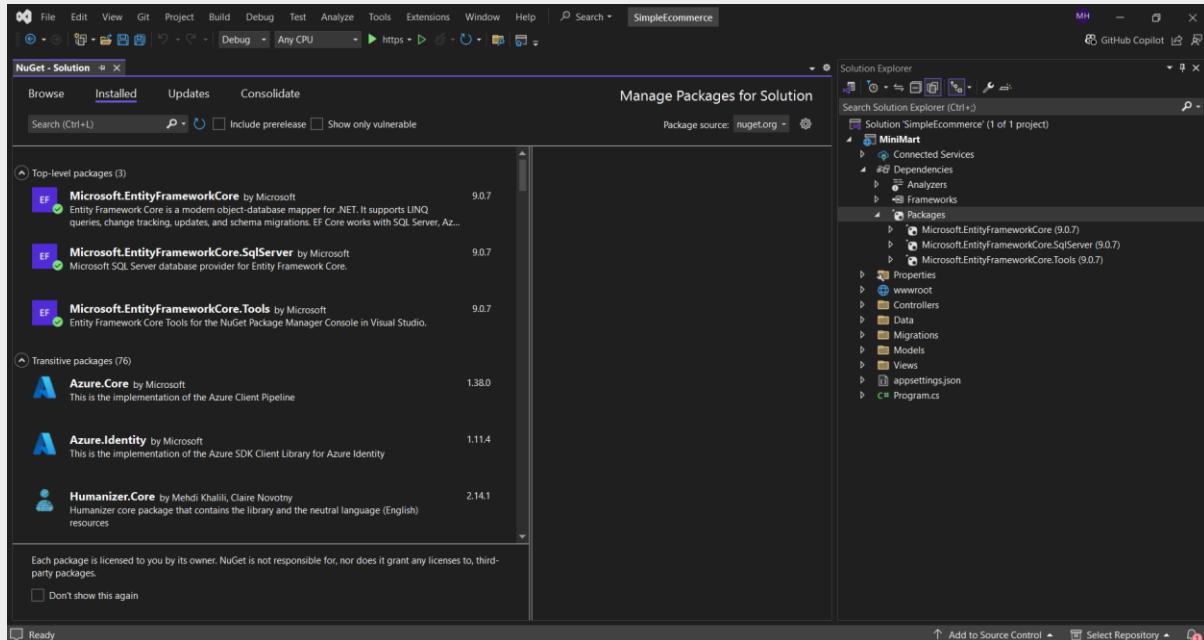
**MiniMart** is a mini e-commerce web application developed using ASP.NET Core MVC.

The application allows users to:

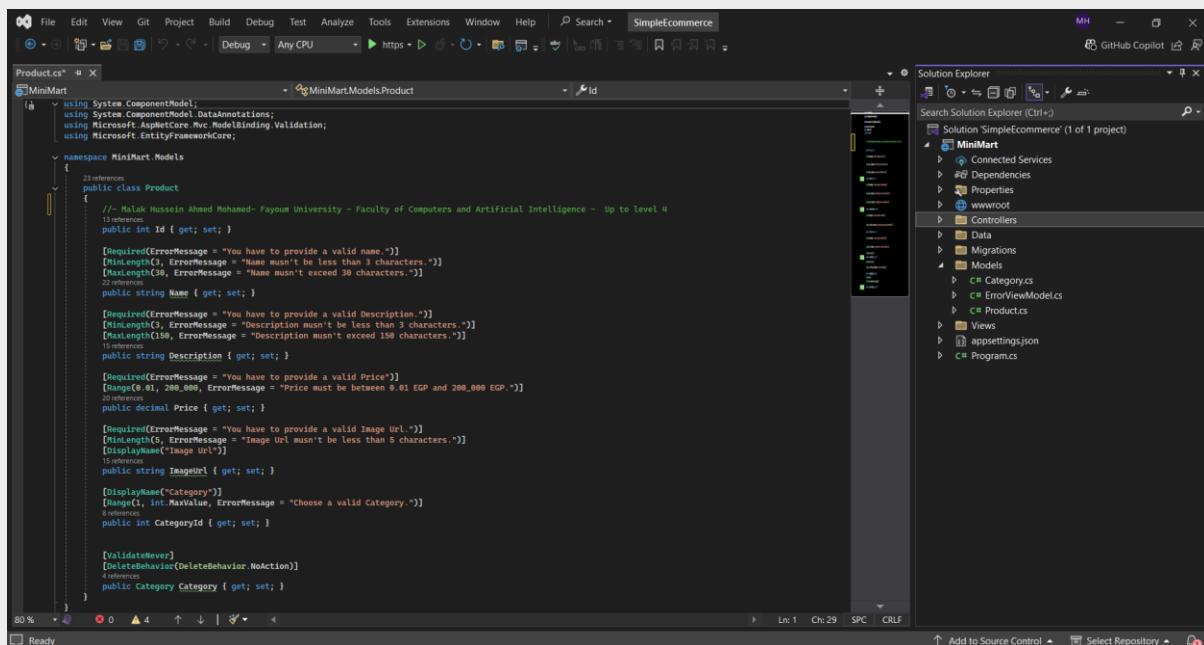
- Create new categories
  - View category details and their associated products
  - Edit and delete existing categories
  - Create, view, edit, and delete products
- 

**In the following pages, the project will be explained step by step, including screenshots of the code and screenshots of the web interface, to clarify each part in detail.**

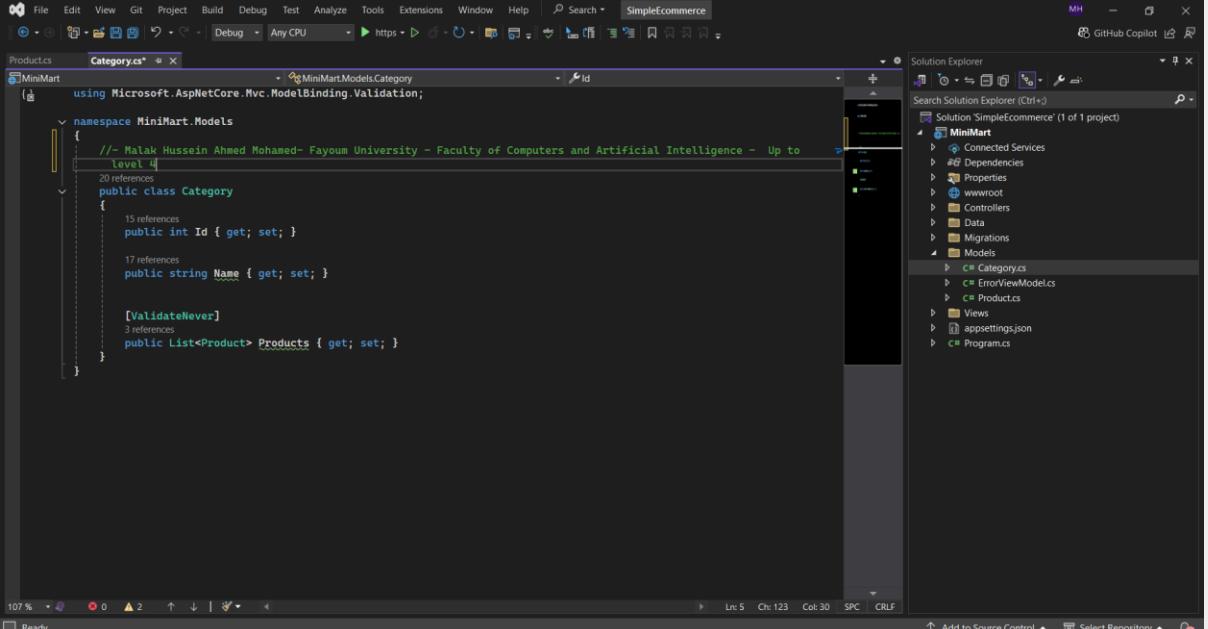
# 1. Installed all required packages:



## 2. Created Model folder and then created two classes: a. Product Class:



## b. Category Class:



The screenshot shows the Visual Studio IDE interface. The title bar says "SimpleCommerce". The main window displays the code for "Category.cs" in the "MiniMart" project. The code defines a class "Category" with properties "Id" and "Name", and a list of "Products". The Solution Explorer on the right shows the project structure, including "Models", "Views", and "Program.cs".

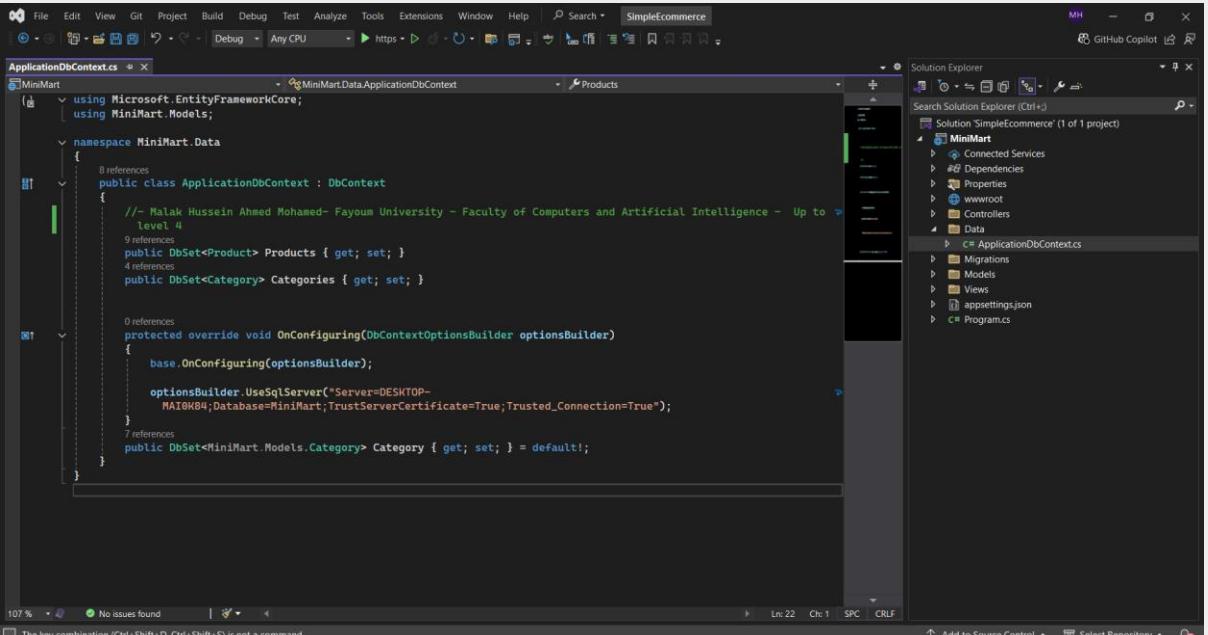
```
using Microsoft.AspNetCore.Mvc.ModelBinding.Validation;
using System.ComponentModel.DataAnnotations;
using System.ComponentModel.DataAnnotations.Schema;

namespace MiniMart.Models
{
    // Malak Hussein Ahmed Mohamed- Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4
    [Table("Category")]
    public class Category
    {
        [Required]
        public int Id { get; set; }

        [Required]
        public string Name { get; set; }

        [ValidateNever]
        public List<Product> Products { get; set; }
    }
}
```

## 3. Created Data folder and then ApplicationDbContext:



The screenshot shows the Visual Studio IDE interface. The title bar says "SimpleCommerce". The main window displays the code for "ApplicationDbContext.cs" in the "MiniMart" project. The code defines a class "ApplicationDbContext" that inherits from "DbContext" and contains DbSet properties for "Products" and "Categories". It also includes an override for "OnConfiguring" to set the database connection. The Solution Explorer on the right shows the project structure, including "Data", "Migrations", and "Program.cs".

```
using Microsoft.EntityFrameworkCore;
using MiniMart.Models;

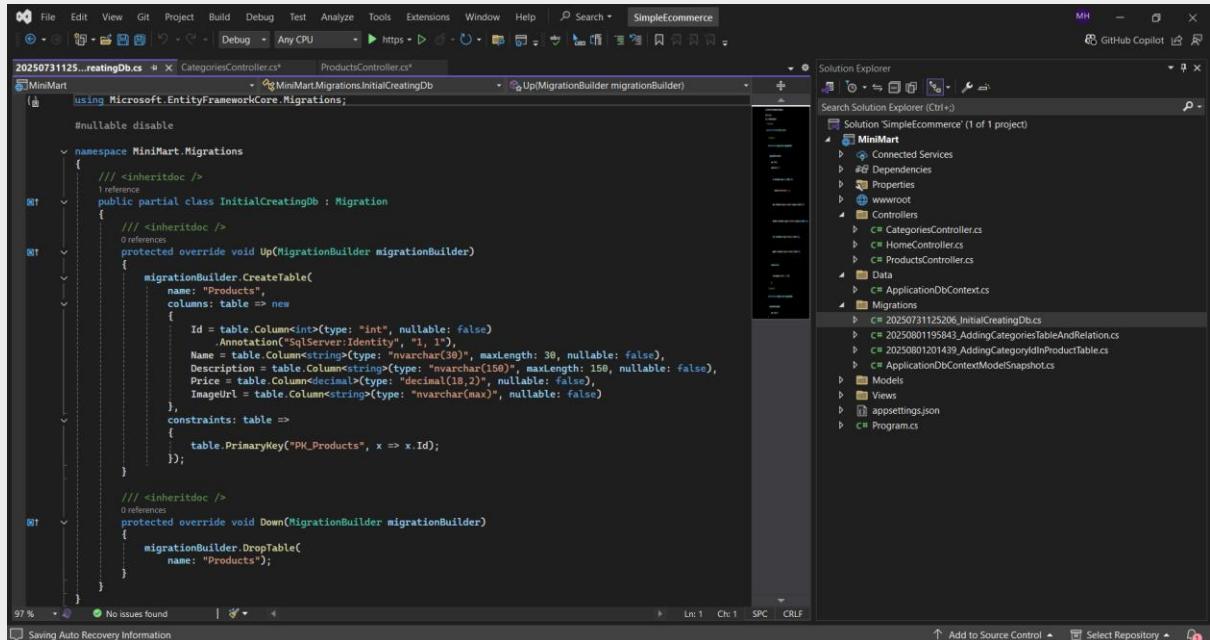
namespace MiniMart.Data
{
    public class ApplicationDbContext : DbContext
    {
        // Malak Hussein Ahmed Mohamed- Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4
        public DbSet<Product> Products { get; set; }
        public DbSet<Category> Categories { get; set; }

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            base.OnConfiguring(optionsBuilder);

            optionsBuilder.UseSqlServer("Server=DESKTOP-MA1OK84;Database=MiniMart;TrustServerCertificate=True;Trusted_Connection=True");
        }

        public DbSet<MiniMart.Models.Category> Category { get; set; } = default!;
    }
}
```

## 4. Added migration and updated database.



```
20250731125...reatingDb.cs  CategoriesController.cs  ProductsController.cs
[...]
using Microsoft.EntityFrameworkCore.Migrations;

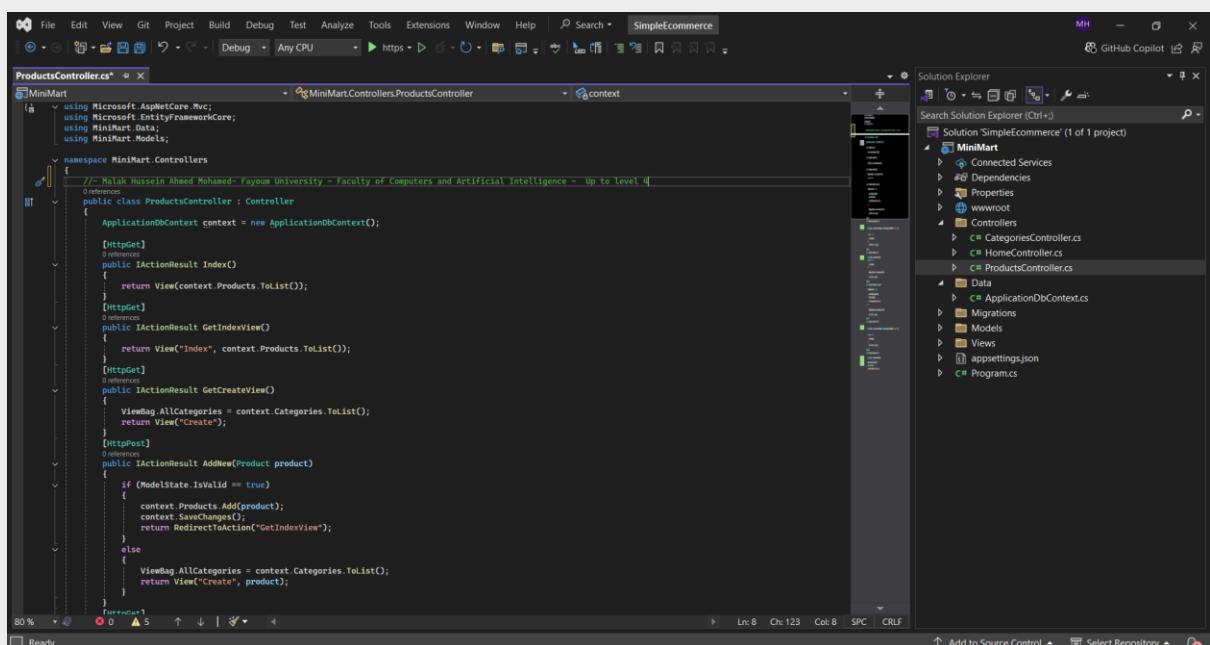
namespace MiniMart.Migrations
{
    [System.ComponentModel.InheritanceAttribute("Initial")]
    public partial class InitialCreatingDb : Migration
    {
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "Products",
                columns: table =>
                {
                    table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1");
                    table.Column<string>(type: "nvarchar(max)", maxLength: 30, nullable: false);
                    table.Column<string>(type: "nvarchar(150)", maxLength: 150, nullable: false);
                    table.Column<decimal>(type: "decimal(18,2)", nullable: false);
                    table.Column<string>(type: "nvarchar(max)", nullable: false);
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_Products", x => x.Id);
                }
            );
        }

        protected override void Down(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.DropTable(
                name: "Products");
        }
    }
}

[...]
```

Solution Explorer shows the project structure with files like CategoriesController.cs, HomeController.cs, ProductsController.cs, ApplicationDbContext.cs, and Program.cs.

## 5. Created ProductsController :



```
ProductsController.cs
[...]
using Microsoft.AspNetCore.Mvc;
using Microsoft.EntityFrameworkCore;
using MiniMart.Data;
using MiniMart.Models;

namespace MiniMart.Controllers
{
    // Malak Hussien Ahmed Mohamed - Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4
    public class ProductsController : Controller
    {
        private readonly ApplicationDbContext context = new ApplicationDbContext();

        [HttpGet]
        public IActionResult Index()
        {
            return View(context.Products.ToList());
        }

        [HttpGet]
        public IActionResult GetIndexView()
        {
            return View("Index", context.Products.ToList());
        }

        [HttpGet]
        public IActionResult GetCreateView()
        {
            ViewBag.AllCategories = context.Categories.ToList();
            return View("Create");
        }

        [HttpPost]
        public IActionResult AddNew(Product product)
        {
            if (ModelState.IsValid)
            {
                context.Products.Add(product);
                context.SaveChanges();
                return RedirectToAction("GetIndexView");
            }
            else
            {
                ViewBag.AllCategories = context.Categories.ToList();
                return View("Create", product);
            }
        }
    }
}
```

Solution Explorer shows the project structure with files like CategoriesController.cs, HomeController.cs, ProductsController.cs, ApplicationDbContext.cs, and Program.cs.

This screenshot shows the Visual Studio IDE interface with the ProductsController.cs file open in the main editor window. The code implements a controller for managing products in a simple e-commerce application. It includes methods for getting product details, editing products, and deleting products. The controller uses Entity Framework to interact with the database. The Solution Explorer on the right shows the project structure for 'SimpleCommerce'.

```
ProductsController.cs
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using SimpleCommerce.Models;
using Microsoft.EntityFrameworkCore;
using System.Threading.Tasks;

namespace SimpleCommerce.Controllers
{
    public class ProductsController : Controller
    {
        // ... (other code)

        [HttpGet]
        public ActionResult GetDetailsView(int id)
        {
            Product product = context.Products.Include(product => product.Category).FirstOrDefault(x => x.Id == id);
            if (product == null)
            {
                return NotFound();
            }
            else
            {
                return View("Details", product);
            }
        }

        [HttpGet]
        public ActionResult GetEditView(int id)
        {
            Product product = context.Products.Find(id);
            if (product == null)
            {
                return NotFound();
            }
            else
            {
                ViewBag.AllCategories = context.Categories.ToList();
                return View("Edit", product);
            }
        }

        [HttpPost]
        public ActionResult EditCurrent(Product product)
        {
            if (ModelState.IsValid == true)
            {
                context.Products.Update(product);
                context.SaveChanges();
                return RedirectToAction("GetIndexView");
            }
            else
            {
                ViewBag.AllCategories = context.Categories.ToList();
                return View("Edit", product);
            }
        }

        [HttpGet]
        public ActionResult GetDeleteView(int id)
        {
            Product product = context.Products.Include(product => product.Category).FirstOrDefault(x => x.Id == id);
            if (product == null)
            {
                return NotFound();
            }
            else
            {
                return View("Delete", product);
            }
        }

        [HttpPost]
        public ActionResult DeleteCurrent(int id)
        {
            Product product = context.Products.Find(id);
            context.Products.Remove(product);
            context.SaveChanges();
            return RedirectToAction("GetIndexView");
        }
    }
}
```

This screenshot shows the Visual Studio IDE interface with the ProductsController.cs file open in the main editor window. The code is identical to the one in the previous screenshot, implementing a controller for managing products. The Solution Explorer on the right shows the project structure for 'SimpleCommerce'.

```
ProductsController.cs
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using SimpleCommerce.Models;
using Microsoft.EntityFrameworkCore;
using System.Threading.Tasks;

namespace SimpleCommerce.Controllers
{
    public class ProductsController : Controller
    {
        // ... (other code)

        [HttpGet]
        public ActionResult GetEditView(int id)
        {
            Product product = context.Products.Find(id);
            if (product == null)
            {
                return NotFound();
            }
            else
            {
                ViewBag.AllCategories = context.Categories.ToList();
                return View("Edit", product);
            }
        }

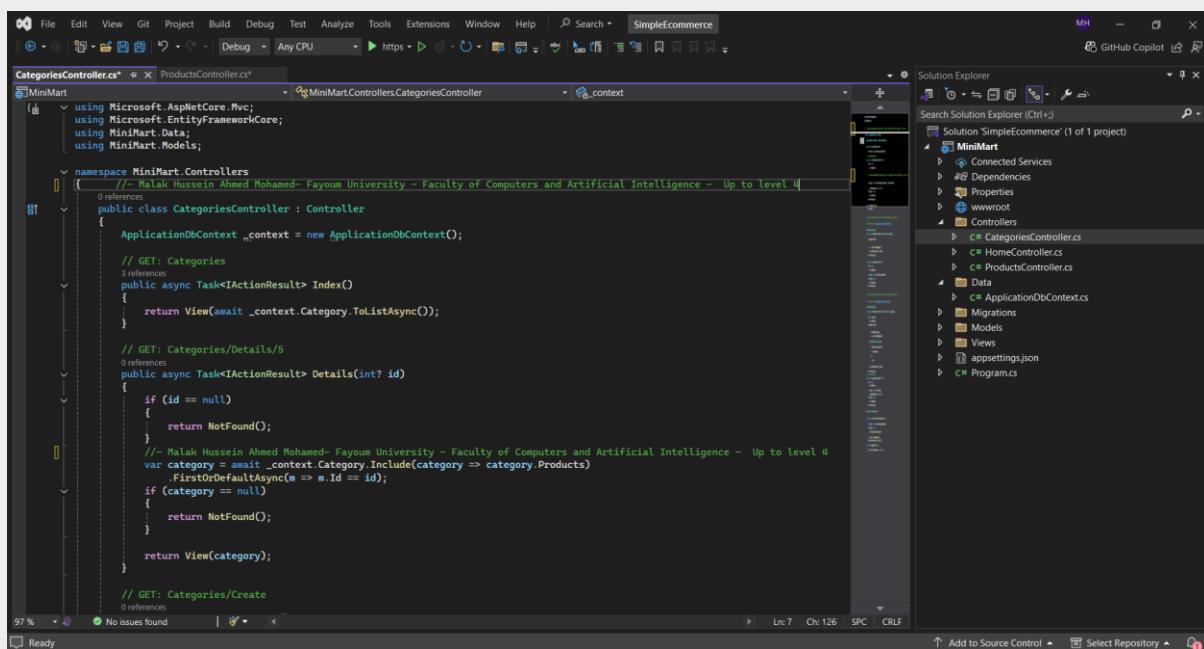
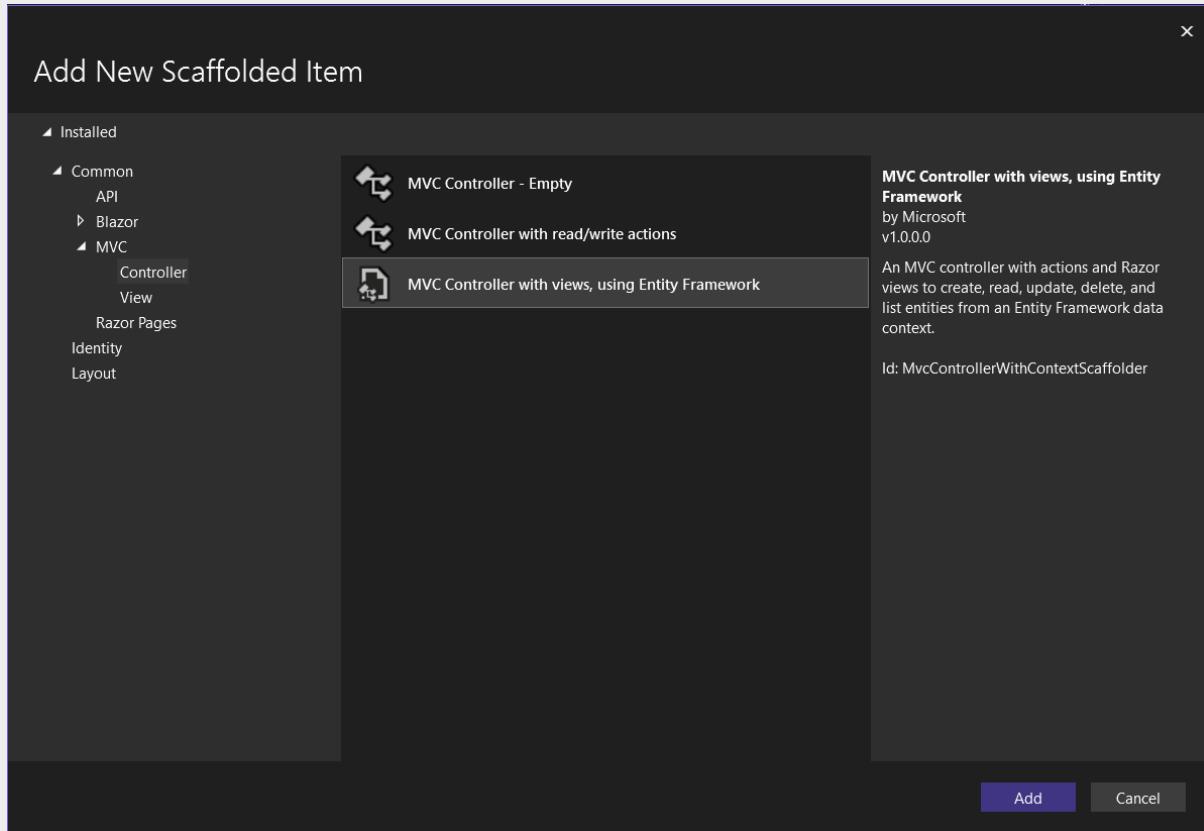
        [HttpPost]
        public ActionResult EditCurrent(Product product)
        {
            if (ModelState.IsValid == true)
            {
                context.Products.Update(product);
                context.SaveChanges();
                return RedirectToAction("GetIndexView");
            }
            else
            {
                ViewBag.AllCategories = context.Categories.ToList();
                return View("Edit", product);
            }
        }

        [HttpGet]
        public ActionResult GetDeleteView(int id)
        {
            Product product = context.Products.Include(product => product.Category).FirstOrDefault(x => x.Id == id);
            if (product == null)
            {
                return NotFound();
            }
            else
            {
                return View("Delete", product);
            }
        }

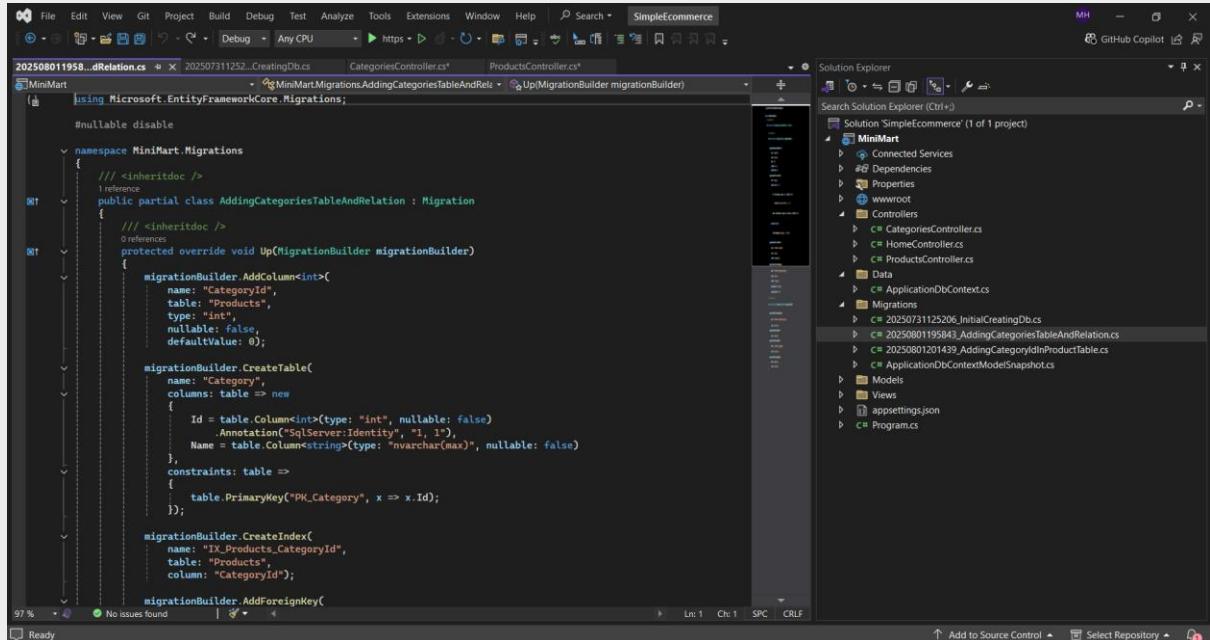
        [HttpPost]
        public ActionResult DeleteCurrent(int id)
        {
            Product product = context.Products.Find(id);
            context.Products.Remove(product);
            context.SaveChanges();
            return RedirectToAction("GetIndexView");
        }
    }
}
```

## 6. Created CategoriesController :

Using:



## 7. Added migration and updated database again.



```
202508011958...dRelation.cs  x  202507311252_CreatingDb.cs  CategoriesController.cs*  ProductsController.cs*
```

```
using Microsoft.EntityFrameworkCore.Migrations;

namespace MiniMart.Migrations
{
    public partial class AddingCategoriesTableAndRelation : Migration
    {
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.AddColumn<int>(
                name: "CategoryId",
                table: "Products",
                type: "int",
                nullable: false,
                defaultValue: 0);

            migrationBuilder.CreateTable(
                name: "Category",
                columns: table =>
                {
                    Id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    Name = table.Column<string>(type: "nvarchar(max)", nullable: false),
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_Category", x => x.Id);
                });

            migrationBuilder.CreateIndex(
                name: "IX_Products_CategoryId",
                table: "Products",
                column: "CategoryId");

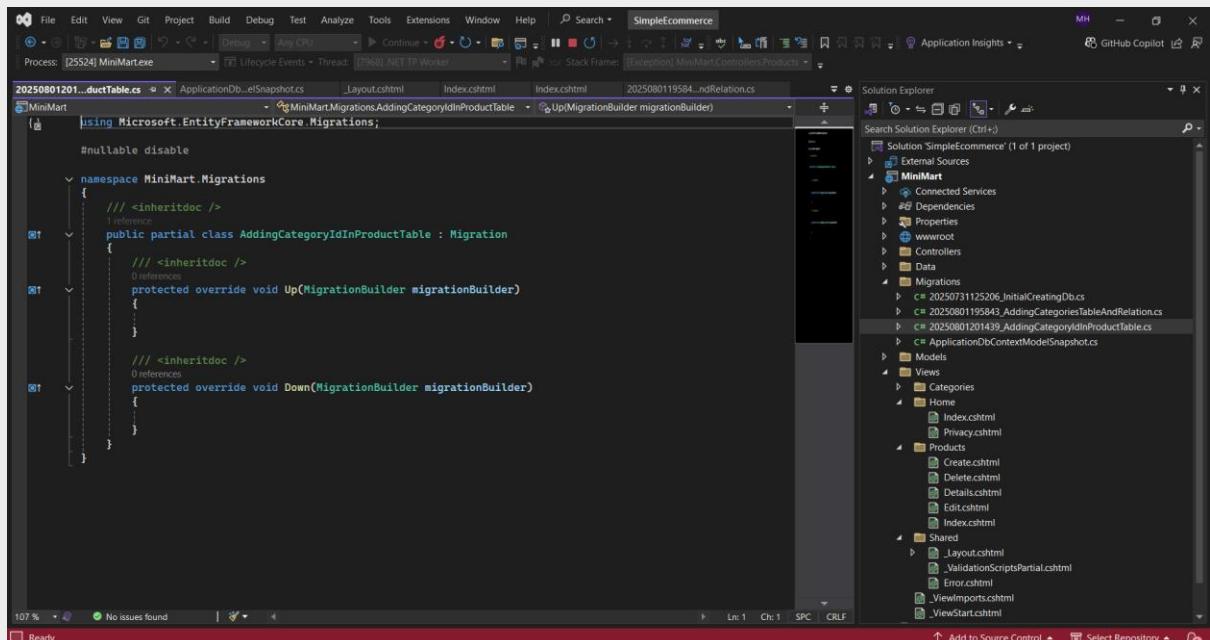
            migrationBuilder.AddForeignKey(

```

Solution Explorer

```
Search Solution Explorer (Ctrl+F)
Solution 'SimpleCommerce' (1 of 1 project)
  MiniMart
    Connected Services
    Dependencies
    Properties
    wwwroot
    Controllers
      CategoriesController.cs
      HomeController.cs
      ProductsController.cs
    Data
      ApplicationDbContext.cs
    Migrations
      20250731125206_InitialCreatingDb.cs
      20250801195843.AddingCategoriesTableAndRelation.cs
      20250801201439.AddingCategoryInProductTable.cs
      ApplicationDbContextModelSnapshot.cs
    Models
    Views
    appsettings.json
    Program.cs
```

## 8. Added migration and updated database again because forgot to add CategoryId in Product table.



```
20250801201...noRelation.cs  x  ApplicationDbContext.cs  _Layout.cshtml  Index.cshtml  Index.cshtml  20250801195843...dRelation.cs
```

```
using Microsoft.EntityFrameworkCore.Migrations;

namespace MiniMart.Migrations
{
    public partial class AddingCategoryInProductTable : Migration
    {
        protected override void Up(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.CreateTable(
                name: "Category",
                columns: table =>
                {
                    Id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    Name = table.Column<string>(type: "nvarchar(max)", nullable: false),
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_Category", x => x.Id);
                });

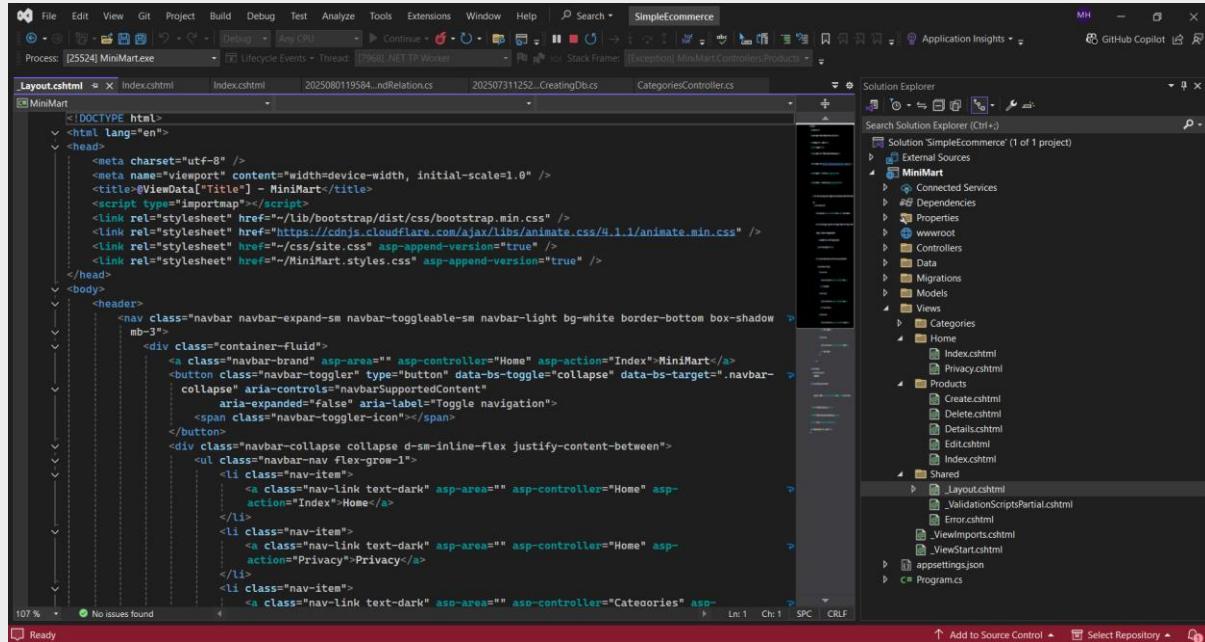
            migrationBuilder.CreateTable(
                name: "Products",
                columns: table =>
                {
                    Id = table.Column<int>(type: "int", nullable: false)
                        .Annotation("SqlServer:Identity", "1, 1"),
                    Name = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    Description = table.Column<string>(type: "nvarchar(max)", nullable: false),
                    Price = table.Column<decimal>(type: "decimal(18, 2)", nullable: false),
                    CategoryId = table.Column<int>(type: "int", nullable: false),
                },
                constraints: table =>
                {
                    table.PrimaryKey("PK_Products", x => x.Id);
                    table.ForeignKey(
                        name: "FK_Products_Categories",
                        column: x => x.CategoryId,
                        principalTable: "Category",
                        principalColumn: "Id",
                        onDelete: ReferentialAction.Cascade);
                });
            migrationBuilder.CreateIndex(
                name: "IX_Products_CategoryId",
                table: "Products",
                column: "CategoryId");
        }

        protected override void Down(MigrationBuilder migrationBuilder)
        {
            migrationBuilder.DropTable(
                name: "Products");
            migrationBuilder.DropTable(
                name: "Category");
        }
    }
}
```

Solution Explorer

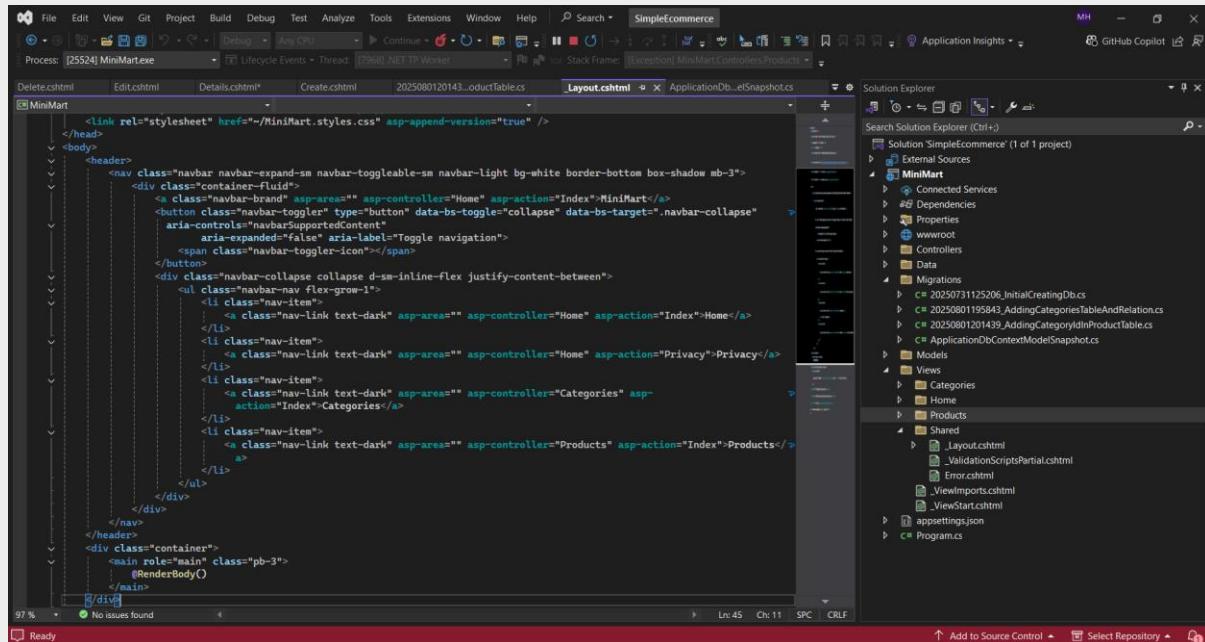
```
Search Solution Explorer (Ctrl+F)
Solution 'SimpleCommerce' (1 of 1 project)
  External Sources
  MiniMart
    Connected Services
    Dependencies
    Properties
    wwwroot
    Controllers
    Data
    Migrations
      20250731125206_InitialCreatingDb.cs
      20250801195843.AddingCategoriesTableAndRelation.cs
      20250801201439.AddingCategoryInProductTable.cs
      ApplicationDbContextModelSnapshot.cs
    Models
    Views
      Categories
        Home
          Index.cshtml
          Privacy.cshtml
        Products
          Create.cshtml
          Delete.cshtml
          Details.cshtml
          Edit.cshtml
          Index.cshtml
      Shared
        _Layout.cshtml
        _ValidationScriptsPartial.cshtml
        Error.cshtml
        ViewImports.cshtml
        ViewStart.cshtml
```

## 9. Added link for animation library in Layout View:



The screenshot shows the Visual Studio IDE with the SimpleCommerce project open. The Solution Explorer on the right shows the project structure with files like Index.cshtml, Create.cshtml, Delete.cshtml, Details.cshtml, Edit.cshtml, and Index.cshtml under the Views\Categories\Home folder. The Layout.cshtml file is currently selected in the left-hand list. The code in the editor includes a meta tag for charset, a viewport for mobile devices, a title, and several script and link tags. One of the link tags points to a CSS file for animation: `<link rel="stylesheet" href="~/lib/bootstrap/dist/css/bootstrap.min.css" />`, `<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css" />`, and `<link rel="stylesheet" href="~/css/site.css" asp-append-version="true" />`. The code also includes a reference to `~/Minimart.styles.css` with `asp-append-version="true"`.

## 10. Added Two Anchors in Layout View to navigate to Products and Categories:



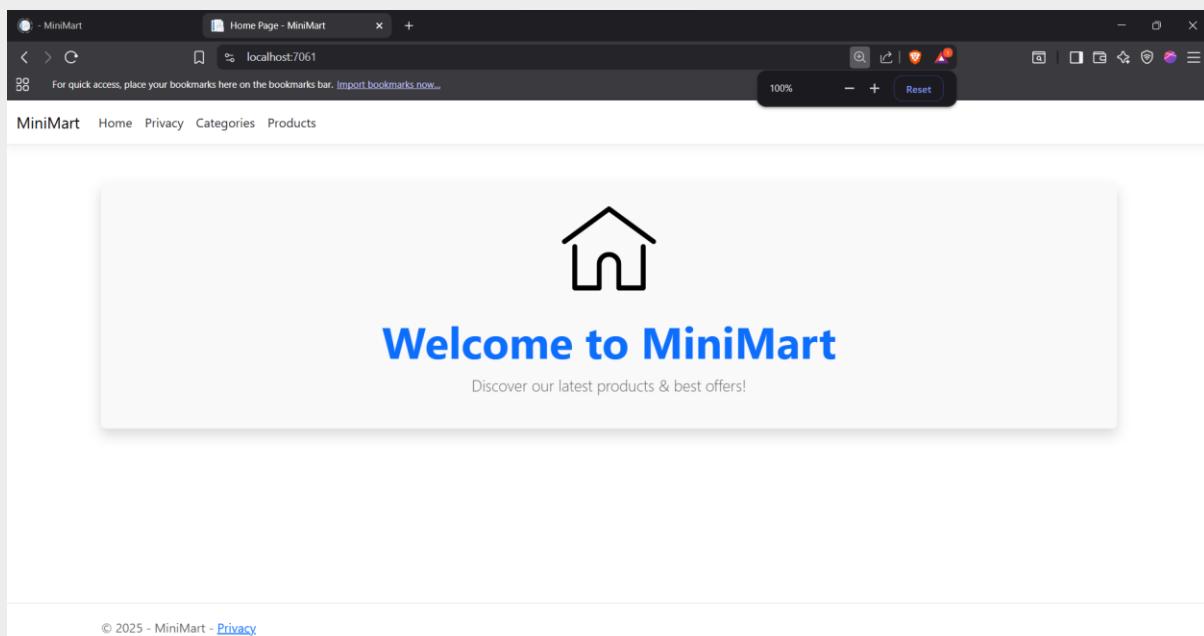
The screenshot shows the Visual Studio IDE with the SimpleCommerce project open. The Solution Explorer on the right shows the project structure with files like Index.cshtml, Create.cshtml, Delete.cshtml, Details.cshtml, Edit.cshtml, and Index.cshtml under the Views\Categories\Home folder. The Layout.cshtml file is currently selected in the left-hand list. The code in the editor includes a meta tag for charset, a viewport for mobile devices, a title, and several script and link tags. It also includes two new anchor links in the navigation bar: one for 'Products' and one for 'Categories'. The 'Products' link uses the controller 'Products' and action 'Index'. The 'Categories' link uses the controller 'Categories' and action 'Index'.

## 11. Updated Index View in Home View:

The screenshot shows the Visual Studio interface. The code editor displays the `Index.cshtml` file for the `MinMart` controller. The file contains HTML and C# code for a home page. The Solution Explorer on the right shows the project structure, including the `Views/Home/Index.cshtml` file selected.

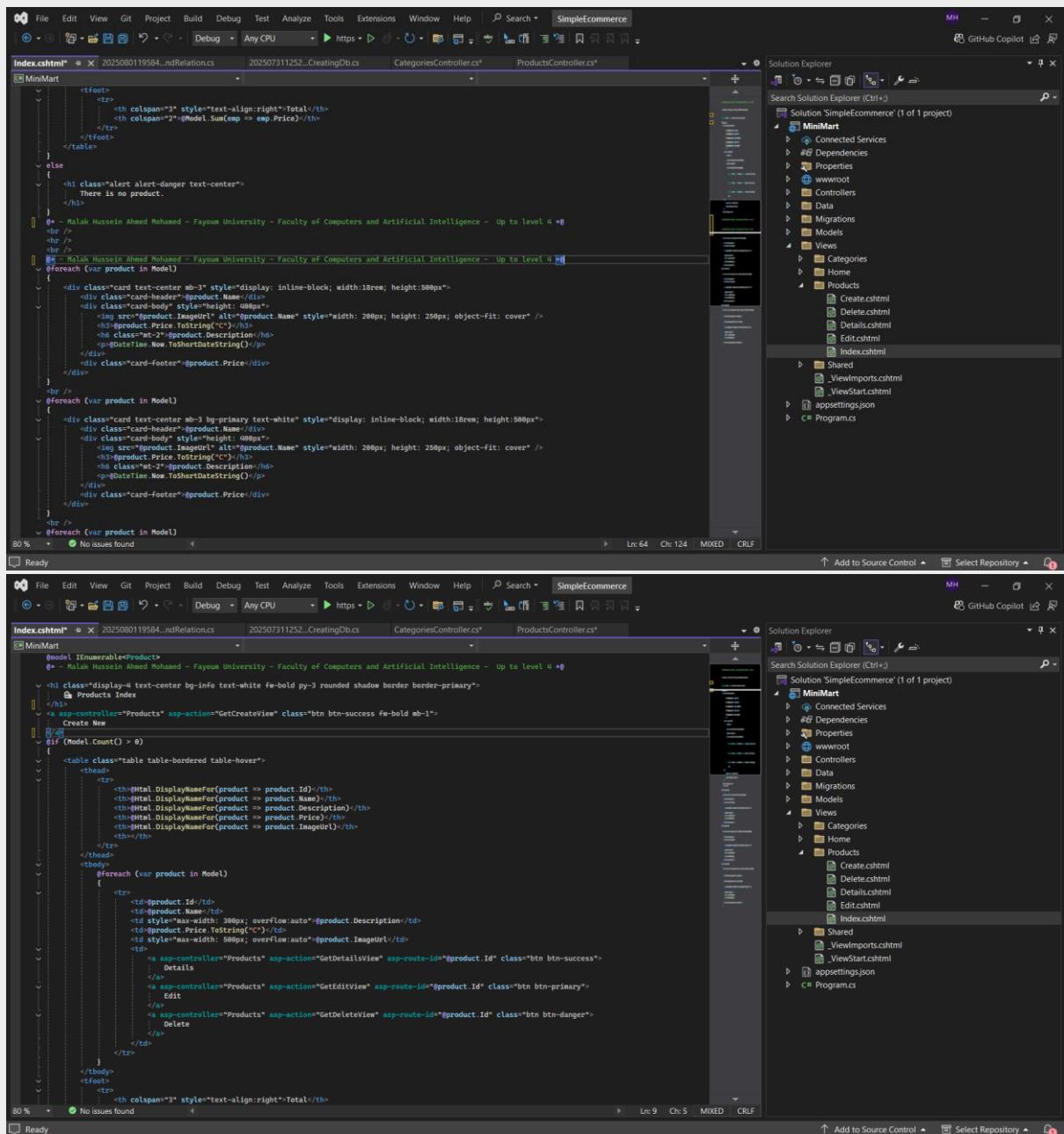
```
 ViewData["Title"] = "Home Page";
}
@* - Malak Hussein Ahmed Mohamed - Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4 *@
<div class="text-center my-5 p-4 rounded shadow" style="background: #F9F9F9;>

<h1 class="display-4 fw-bold text-primary animate__animated animate__fadeInUp">
    Welcome to MiniMart
</h1>
<p class="lead text-muted animate__animated animate__fadeInUp animate__delay-1s">
    Discover our latest products & best offers!
</p>
</div>
```



## 12. Created Products folder in Views folder and then created:

### a. Index View:



The screenshot shows two instances of the Visual Studio IDE side-by-side, comparing two versions of the Index.cshtml view. Both instances have the following tabs open: Index.cshtml\*, 2025080119584\_ndRelation.cs, 202507311252\_CreatingDb.cs, CategoriesController.cs\*, and ProductsController.cs\*. The Solution Explorer on the right shows a project named 'SimpleCommerce' with a 'Views' folder containing 'Categories' and 'Products' subfolders. The 'Products' folder contains files: Create.cshtml, Delete.cshtml, Details.cshtml, Edit.cshtml, and Index.cshtml.

**Top Window (Left):**

```
<tbody>
    <tr>
        <th colspan="3" style="text-align:right;">Total</th>
    <tr>
        <th colspan="2">@Model.Sum(emp => emp.Price)</th>
    </tr>
    </tbody>
</table>
}
else
{
    <h1 class="alert alert-danger text-center">
        There is no product.
    </h1>
}
@* - Malak Hussein Ahmed Mohamed - Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4 *@
<br />
<br />
@* - Malak Hussein Ahmed Mohamed - Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4 *@

@foreach (var product in Model)
{
    <div class="card text-center mb-3" style="display: inline-block; width:18rem; height:50px;">
        <div class="card-header" style="height: 40px">
            
            <h3>@product.Price.ToString("C")</h3>
            <div class="mt-2" style="font-size: small; color: #808080; margin-top: 5px; font-weight: bold; font-style: italic; opacity: 0.5; transition: opacity 0.5s ease-in-out; position: relative; z-index: 1;*>
                <p>@DateTime.Now.ToString("dd/MM/yyyy")</p>
            </div>
        <div class="card-footer" style="background-color: #f2f2f2; border-top: none; padding: 5px; position: relative; z-index: 0;*>
            <div style="display: flex; justify-content: space-between; align-items: center; gap: 10px; font-size: small; color: #808080; font-weight: bold; font-style: italic; opacity: 0.5; transition: opacity 0.5s ease-in-out; position: relative; z-index: 1;*>
                <span>#@product.Id</span>
                <span>@product.Name</span>
                <span>@product.Description</span>
                <span>@product.Price</span>
                <span>@product.ImageUrl</span>
            </div>
        </div>
    </div>
}
@endforeach
@foreach (var product in Model)
{
    <div class="card text-center mb-3 bg-primary text-white" style="display: inline-block; width:18rem; height:50px;">
        <div class="card-header" style="height: 40px">
            
            <h3>@product.Price.ToString("C")</h3>
            <div class="mt-2" style="font-size: small; color: #808080; margin-top: 5px; font-weight: bold; font-style: italic; opacity: 0.5; transition: opacity 0.5s ease-in-out; position: relative; z-index: 1;*>
                <p>@DateTime.Now.ToString("dd/MM/yyyy")</p>
            </div>
        <div class="card-footer" style="background-color: #f2f2f2; border-top: none; padding: 5px; position: relative; z-index: 0;*>
            <div style="display: flex; justify-content: space-between; align-items: center; gap: 10px; font-size: small; color: #808080; font-weight: bold; font-style: italic; opacity: 0.5; transition: opacity 0.5s ease-in-out; position: relative; z-index: 1;*>
                <span>#@product.Id</span>
                <span>@product.Name</span>
                <span>@product.Description</span>
                <span>@product.Price</span>
                <span>@product.ImageUrl</span>
            </div>
        </div>
    </div>
}
@endforeach

```

**Bottom Window (Right):**

```
<model Ienumerable<Product>>
@* - Malak Hussein Ahmed Mohamed - Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4 *@

<h1 class="display-4 text-center bg-info text-white fw-bold py-3 rounded shadow border border-primary">
    Products Index
</h1>
<a href="#" asp-controller="Products" asp-action="GetCreateView" class="btn btn-success fw-bold mb-1">
    Create New
</a>
@if (Model.Count() > 0)
{
    <table class="table table-bordered table-hover">
        <thead>
            <tr>
                <th>@Html.DisplayNameFor(product => product.Id)</th>
                <th>@Html.DisplayNameFor(product => product.Name)</th>
                <th>@Html.DisplayNameFor(product => product.Description)</th>
                <th>@Html.DisplayNameFor(product => product.Price)</th>
                <th>@Html.DisplayNameFor(product => product.ImageUrl)</th>
            </tr>
        </thead>
        <tbody>
            @foreach (var product in Model)
            {
                <tr>
                    <td>@product.Id</td>
                    <td>@product.Name</td>
                    <td style="max-width: 30px; overflow:auto;">@product.Description</td>
                    <td style="max-width: 50px; overflow:auto;">@product.Price.ToString("C")</td>
                    <td style="max-width: 50px; overflow:auto;">@product.ImageUrl</td>
                    <td>
                        <a href="#" asp-controller="Products" asp-action="GetDetailsView" asp-route-id="@product.Id" class="btn btn-success">
                            Details
                        </a>
                        <a href="#" asp-controller="Products" asp-action="GetEditView" asp-route-id="@product.Id" class="btn btn-primary">
                            Edit
                        </a>
                        <a href="#" asp-controller="Products" asp-action="GetDeleteView" asp-route-id="@product.Id" class="btn btn-danger">
                            Delete
                        </a>
                    </td>
                </tr>
            }
        </tbody>
        <tfoot>
            <tr>
                <th colspan="3" style="text-align:right;">Total</th>
            </tr>
        </tfoot>
    </table>
}
@* - Malak Hussein Ahmed Mohamed - Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4 *@


```

Screenshot of a Microsoft Visual Studio IDE showing the development environment for a .NET application named "SimpleCommerce".

The Solution Explorer shows a project named "MinMart" containing files such as "Index.cshtml", "2025080119584...ndRelation.cs", "202507311252...CreatingDb.cs", "CategoriesController.cs", and "ProductsController.cs".

The code editor displays the "Index.cshtml" file, which contains a list of products. The template uses Bootstrap cards to display product details, including the product name, image, price, and description.

```

@model IEnumerable<MinMart.Models.Product>
@{
    ViewData["Title"] = "Products Index";
}



## Products Index



| ID | Name              | Description                            | Price      | Image URL                                                                                                                                                                                                                                                                                                                                         |
|----|-------------------|----------------------------------------|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6  | Smart Watch       | Stylish smartwatch with health tracker | \$8,999.99 | <a href="https://m.media-amazon.com/images/I/61plzNaNRWL_UF894,1000_QL80_.jpg">https://m.media-amazon.com/images/I/61plzNaNRWL_UF894,1000_QL80_.jpg</a>                                                                                                                                                                                           |
| 8  | Bluetooth Headset | Portable headset with deep bass        | \$5,500.00 | <a href="https://resource.logitech.com/w_800_c_lpad_ar_1:1,q_auto,f_auto,dpr_1.0,wireless-headset/gallery/zone-wireless-gallery-01.png?v=1">https://resource.logitech.com/w_800_c_lpad_ar_1:1,q_auto,f_auto,dpr_1.0,wireless-headset/gallery/zone-wireless-gallery-01.png?v=1</a>                                                                 |
| 9  | Denim Jacket      | Trendy blue denim jacket for all ages  | \$1,999.99 | <a href="https://www.harley-davidson.com/content/dam/h-d/images/product-images/merchandise/2022/july-replenishment/99040-23vw/99040-23VW_F.jpg?impolicy=mysize&amp;rws=700">https://www.harley-davidson.com/content/dam/h-d/images/product-images/merchandise/2022/july-replenishment/99040-23vw/99040-23VW_F.jpg?impolicy=mysize&amp;rws=700</a> |
| 10 | Table Lamp        | Decorative lamp with warm light        | \$1,000.00 | <a href="https://www.ks-licht.de/images/product_images/popup_images/STE-34562W_1.jpg">https://www.ks-licht.de/images/product_images/popup_images/STE-34562W_1.jpg</a>                                                                                                                                                                             |
| 11 | Fiction Book      | Exciting mystery novel                 | \$399.99   | <a href="https://tolstoytherapy.com/wp-content/uploads/2022/09/51QQRmnMwDL.jpeg">https://tolstoytherapy.com/wp-content/uploads/2022/09/51QQRmnMwDL.jpeg</a>                                                                                                                                                                                       |
| 12 | Teddy Bear        | Soft and cuddly teddy for kids         | \$150.00   | <a href="data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAD/2wCEAAk">data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAD/2wCEAAk</a>                                                                                                                                                                                                                 |


```

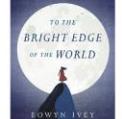
The browser preview shows the "Products Index" page with the heading "Products Index" and a "Create New" button. Below the heading is a table listing the products with their details and actions (Details, Edit, Delete).

ID	Name	Description	Price	Image URL
6	Smart Watch	Stylish smartwatch with health tracker	\$8,999.99	<a href="https://m.media-amazon.com/images/I/61plzNaNRWL_UF894,1000_QL80_.jpg">https://m.media-amazon.com/images/I/61plzNaNRWL_UF894,1000_QL80_.jpg</a>
8	Bluetooth Headset	Portable headset with deep bass	\$5,500.00	<a href="https://resource.logitech.com/w_800_c_lpad_ar_1:1,q_auto,f_auto,dpr_1.0,wireless-headset/gallery/zone-wireless-gallery-01.png?v=1">https://resource.logitech.com/w_800_c_lpad_ar_1:1,q_auto,f_auto,dpr_1.0,wireless-headset/gallery/zone-wireless-gallery-01.png?v=1</a>
9	Denim Jacket	Trendy blue denim jacket for all ages	\$1,999.99	<a href="https://www.harley-davidson.com/content/dam/h-d/images/product-images/merchandise/2022/july-replenishment/99040-23vw/99040-23VW_F.jpg?impolicy=mysize&amp;rws=700">https://www.harley-davidson.com/content/dam/h-d/images/product-images/merchandise/2022/july-replenishment/99040-23vw/99040-23VW_F.jpg?impolicy=mysize&amp;rws=700</a>
10	Table Lamp	Decorative lamp with warm light	\$1,000.00	<a href="https://www.ks-licht.de/images/product_images/popup_images/STE-34562W_1.jpg">https://www.ks-licht.de/images/product_images/popup_images/STE-34562W_1.jpg</a>
11	Fiction Book	Exciting mystery novel	\$399.99	<a href="https://tolstoytherapy.com/wp-content/uploads/2022/09/51QQRmnMwDL.jpeg">https://tolstoytherapy.com/wp-content/uploads/2022/09/51QQRmnMwDL.jpeg</a>
12	Teddy Bear	Soft and cuddly teddy for kids	\$150.00	<a href="data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAD/2wCEAAk">data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAAQABAD/2wCEAAk</a>

- MiniMart - MiniMart +

localhost:7061/Products

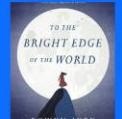
For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

Smart Watch	Bluetooth Headset	Denim Jacket	Table Lamp
			
<b>\$8,999.99</b> Stylish smartwatch with health tracker 8/2/2025 8999.99	<b>\$5,500.00</b> Portable headset with deep bass 8/2/2025 5500.00	<b>\$1,999.99</b> Trendy blue denim jacket for all ages 8/2/2025 1999.99	<b>\$1,000.00</b> Decorative lamp with warm light 8/2/2025 1000.00
Fiction Book	Teddy Bear		
			
<b>\$399.99</b> Exciting mystery novel 8/2/2025 399.99	<b>\$150.00</b> Soft and cuddly teddy for kids 8/2/2025 150.00		

- MiniMart - MiniMart +

localhost:7061/Products

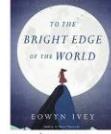
For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

Smart Watch	Bluetooth Headset	Denim Jacket	Table Lamp
			
<b>\$8,999.99</b> Stylish smartwatch with health tracker 8/2/2025 8999.99	<b>\$5,500.00</b> Portable headset with deep bass 8/2/2025 5500.00	<b>\$1,999.99</b> Trendy blue denim jacket for all ages 8/2/2025 1999.99	<b>\$1,000.00</b> Decorative lamp with warm light 8/2/2025 1000.00
Fiction Book	Teddy Bear		
			
<b>\$399.99</b> Exciting mystery novel 8/2/2025 399.99	<b>\$150.00</b> Soft and cuddly teddy for kids 8/2/2025 150.00		

- MiniMart - MiniMart +

localhost:7061/Products

For quick access, place your bookmarks here on the bookmarks bar. [Import bookmarks now...](#)

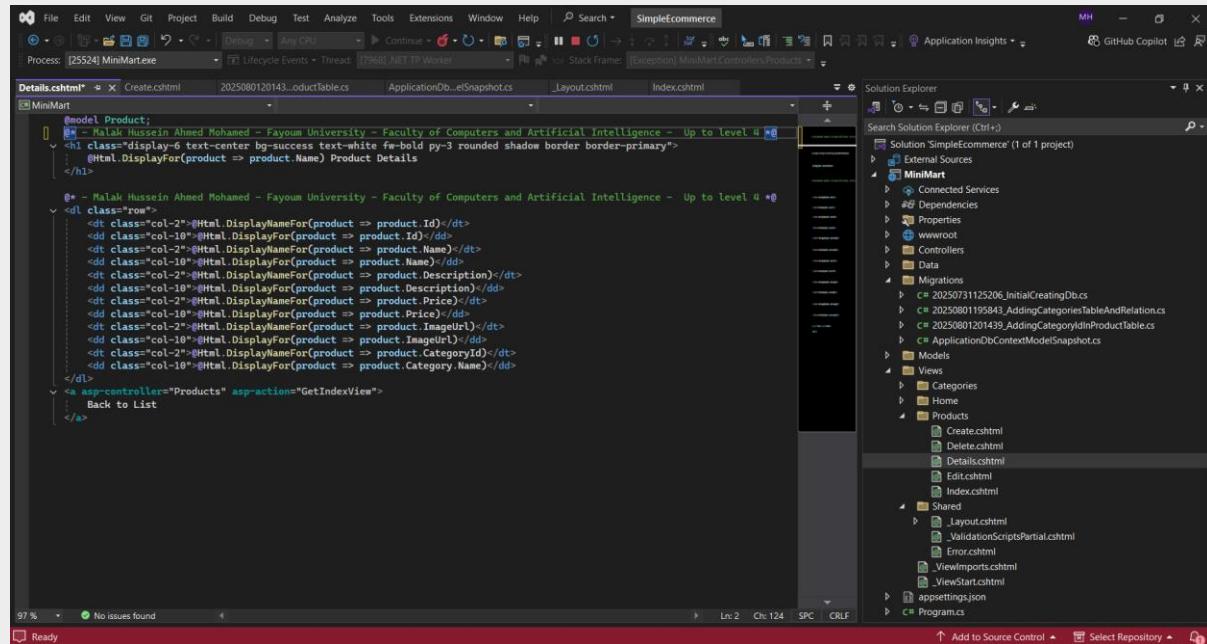
Smart Watch	Bluetooth Headset	Denim Jacket	Table Lamp
 \$8,999.99 Stylish smartwatch with health tracker 8/2/2025 8999.99	 \$5,500.00 Portable headset with deep bass 8/2/2025 5500.00	 \$1,999.99 Trendy blue denim jacket for all ages 8/2/2025 1999.99	 \$1,000.00 Decorative lamp with warm light 8/2/2025 1000.00
Fiction Book		Teddy Bear	
 \$399.99 Exciting mystery novel 8/2/2025 399.99		 \$150.00 Soft and cuddly teddy for kids 8/2/2025 150.00	

## b. Create View:

The screenshot shows the Visual Studio interface. The code editor displays the `Create.cshtml` view for adding a product. The view uses Bootstrap classes like `display-8`, `text-center`, and `py-3`. It includes fields for Name, Description, Price, and Image Url, each with validation messages. The `asp-validation-for` attributes point to their respective model properties. The `asp-action="AddNew"` attribute on the submit button indicates the controller and action to call. The Solution Explorer on the right shows the project structure for "SimpleCommerce" with "MinMart" selected, containing files like `Index.cshtml`, `Products.cshtml`, and `Category.cshtml`.

The screenshot shows a browser window titled "MiniMart" at the URL `localhost:7061/Products/AddNew`. The page has a yellow header bar with the title "Create Product". Below it is a list of validation errors: "You have to provide a valid Description." and "Choose a valid Category.". The form contains input fields for "Name" (with value "Smart Watch"), "Description" (empty), "Price" (with value "9999.99"), and "Image Url" (with value "https://www.ks-licht.de/images/product\_images/popup"). A dropdown menu for "Category" is open, showing "Select..." and "Choose a valid Category." buttons. At the bottom are two buttons: a green "Add" button and a blue "Back to List" button.

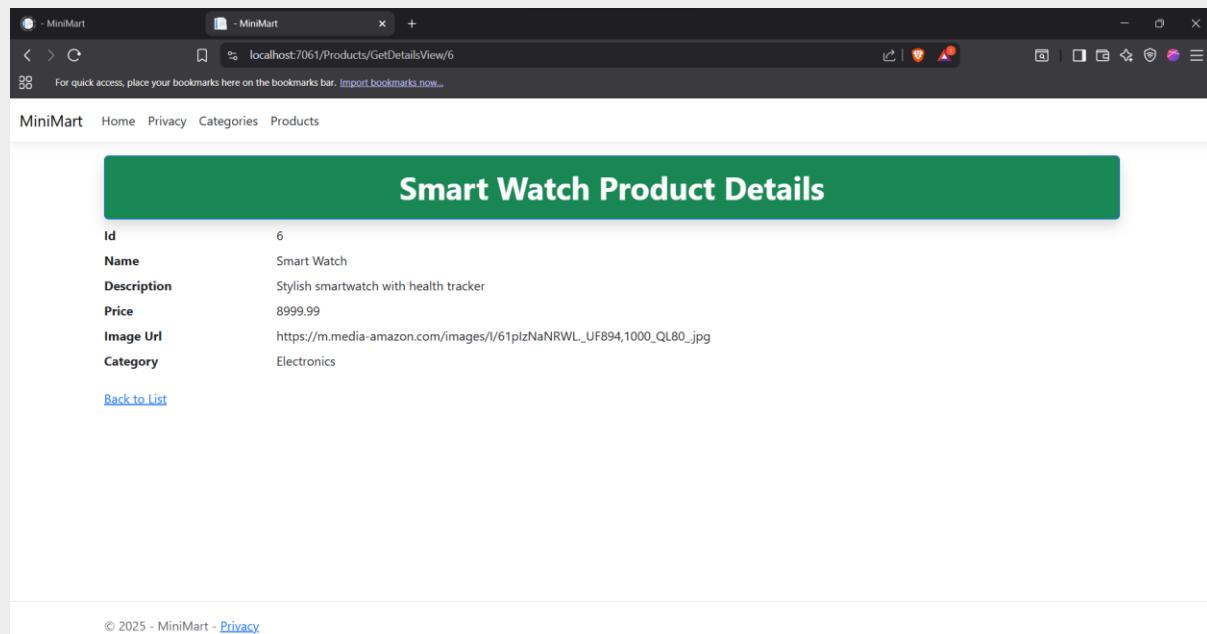
## c. Details View:



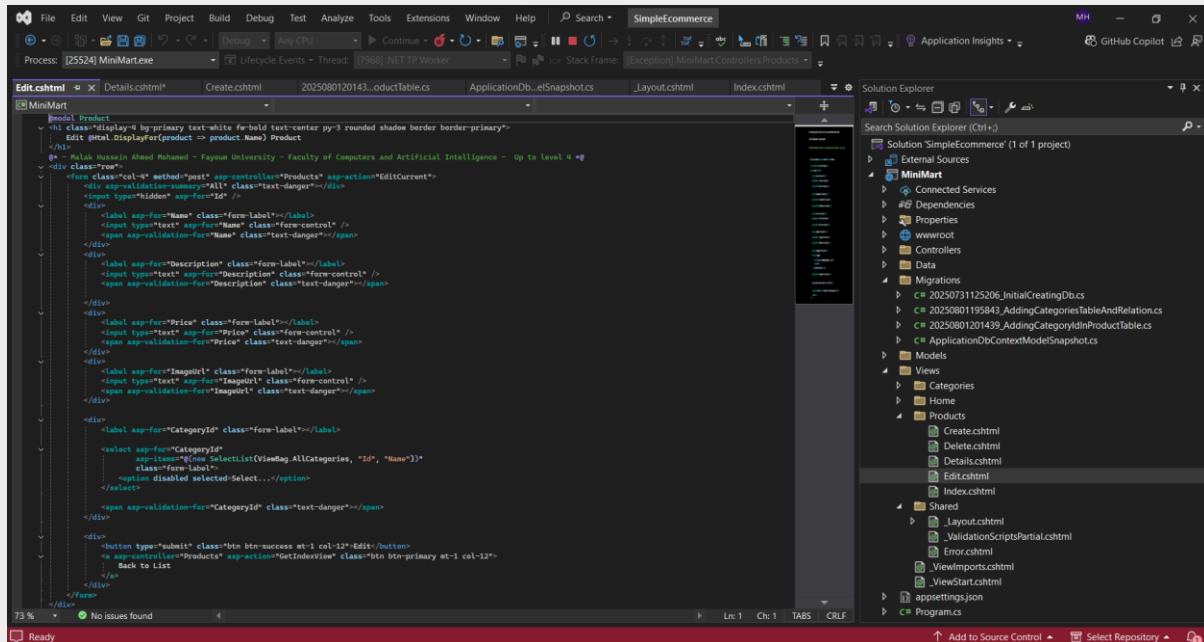
The screenshot shows the Visual Studio IDE with the 'SimpleCommerce' project open. The 'Details.cshtml' file is selected in the Solution Explorer. The code editor displays the following HTML and C# code:

```
<model Product=">
    <!-- Malak Hussein Ahmed Mohamed - Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4 -->
    <h1 class="display-6 text-center bg-success text-white fw-bold py-3 rounded shadow border border-primary">
        @Html.DisplayFor(product => product.Name) Product Details
    </h1>

    <!-- Malak Hussein Ahmed Mohamed - Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4 -->
    <dl class="row">
        <dt class="col-2">@Html.DisplayNameFor(product => product.Id)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.Id)</dd>
        <dt class="col-2">@Html.DisplayNameFor(product => product.Name)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.Name)</dd>
        <dt class="col-2">@Html.DisplayNameFor(product => product.Description)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.Description)</dd>
        <dt class="col-2">@Html.DisplayNameFor(product => product.Price)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.Price)</dd>
        <dt class="col-2">@Html.DisplayNameFor(product => product.ImageUrl)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.ImageUrl)</dd>
        <dt class="col-2">@Html.DisplayNameFor(product => product.CategoryId)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.Category.Name)</dd>
    </dl>
<a asp-controller="Products" asp-action="GetIndexView">
    Back to List
</a>
```



## d. Edit View:

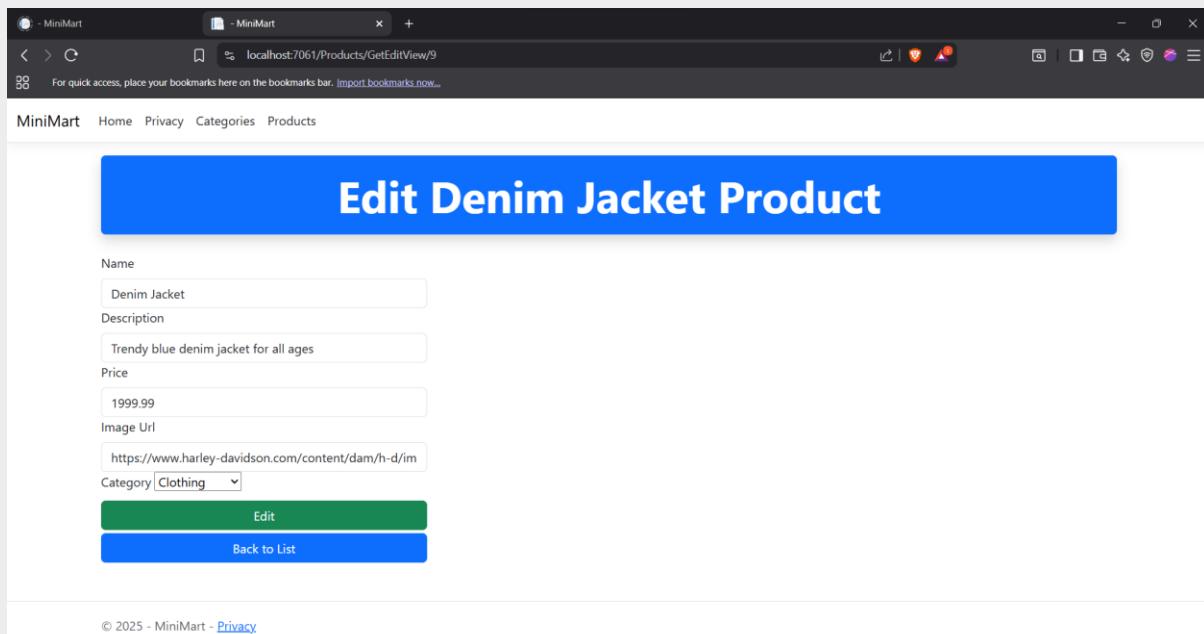


The screenshot shows the Visual Studio IDE with the code editor open to the `Edit.cshtml` file. The code is an ASP.NET MVC view for editing a product. It includes HTML structure for a form, with fields for Name, Description, Price, Image URL, and Category ID. Validation messages are displayed using `asp-validation-for` attributes and `text-danger` classes. The `Solution Explorer` on the right shows the project structure, including files like `Index.cshtml`, `Create.cshtml`, and `Details.cshtml`.

```
<h1 class="display-8 bg-primary text-white fw-bold text-center py-3 rounded shadow border border-primary">
    Edit (@Model.Name)
    Faculty of Computers and Artificial Intelligence - Up to level 4
</h1>


<div asp-validation-summary="All" class="text-danger" style="margin-bottom: 10px;"></div>
    <div asp-for="Name" class="col-12">
        <input type="hidden" asp-for="Id" />
        <label asp-for="Name" class="form-label"></label>
        <input type="text" asp-for="Name" class="form-control" />
        <span asp-validation-for="Name" class="text-danger"></span>
    </div>
    <div asp-for="Description" class="col-12">
        <label asp-for="Description" class="form-label"></label>
        <input type="text" asp-for="Description" class="form-control" />
        <span asp-validation-for="Description" class="text-danger"></span>
    </div>
    <div asp-for="Price" class="col-12">
        <label asp-for="Price" class="form-label"></label>
        <input type="text" asp-for="Price" class="form-control" />
        <span asp-validation-for="Price" class="text-danger"></span>
    </div>
    <div asp-for="ImageUrl" class="col-12">
        <label asp-for="ImageUrl" class="form-label"></label>
        <input type="text" asp-for="ImageUrl" class="form-control" />
        <span asp-validation-for="ImageUrl" class="text-danger"></span>
    </div>
    <div asp-for="CategoryId" class="col-12">
        <label asp-for="CategoryId" class="form-label"></label>
        <select asp-for="CategoryId" class="form-select" style="width: 100%; height: 35px;">
            <option value="">Select...</option>
            <option selected="selected" value="1">Clothing</option>
        </select>
        <span asp-validation-for="CategoryId" class="text-danger"></span>
    </div>


```



The screenshot shows a web browser displaying the `Edit.cshtml` view. The title bar says "MiniMart" and the address bar shows "localhost:7061/Products/EditView/9". The page has a blue header with the text "Edit Denim Jacket Product". Below the header is a form with fields for Name, Description, Price, Image Url, and Category. The category dropdown is set to "Clothing". At the bottom are two buttons: a green "Edit" button and a blue "Back to List" button.

MinMart Home Privacy Categories Products

## Edit Denim Jacket Product

Name  
Denim Jacket

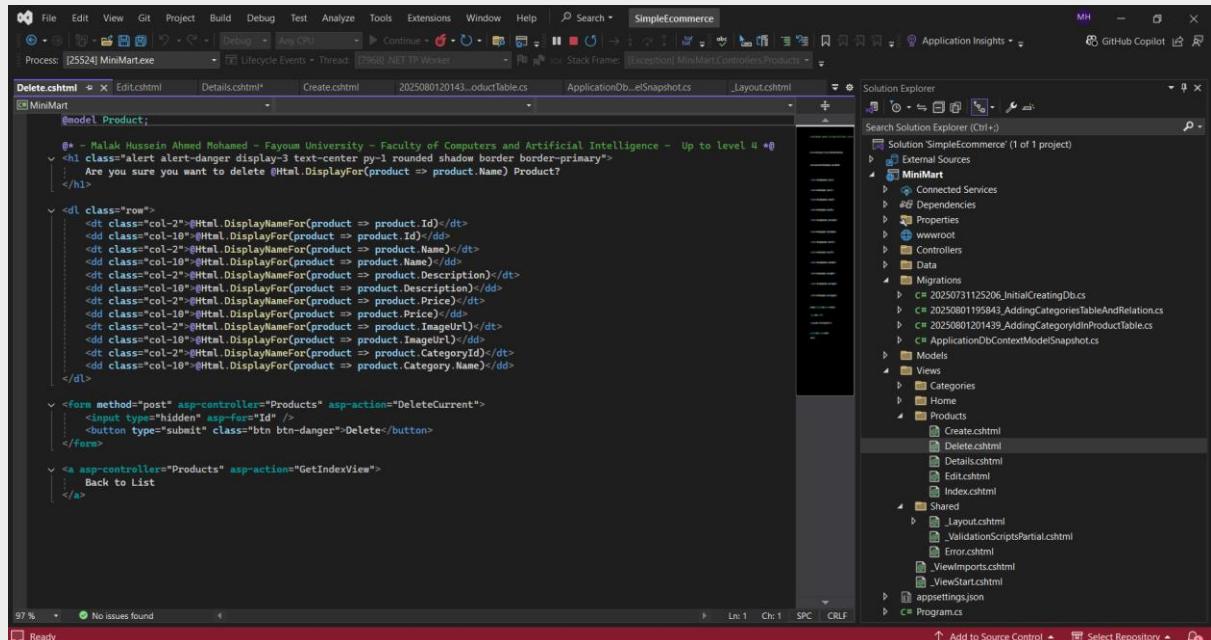
Description  
Trendy blue denim jacket for all ages

Price  
1999.99

Image Url  
<https://www.harley-davidson.com/content/dam/h-d/im>

Category

## e. Delete View:



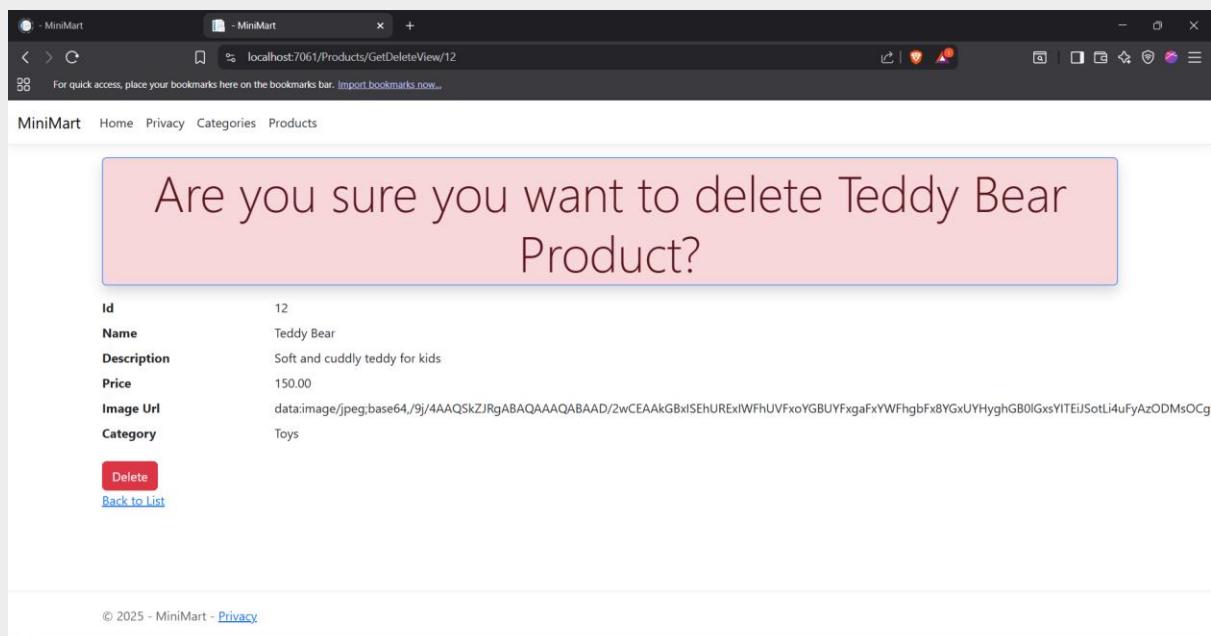
The screenshot shows the Visual Studio IDE with the code editor open to the Delete.cshtml file. The code is a standard ASP.NET MVC delete view. It includes a header with the user's name and a confirmation message. Below this is a table-like structure displaying product details such as Id, Name, Description, Price, Image Url, and Category. A form is present for deleting the product, and navigation links for Back to List and GetIndexView.

```
<model Product>
    <!-- Malak Hussein Ahmed - Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4 @#-->
    <h1 class="alert alert-danger display-3 text-center py-1 rounded shadow border border-primary">
        Are you sure you want to delete @Html.DisplayFor(product => product.Name) Product?
    </h1>

    <dl class="row">
        <dt class="col-2">@Html.DisplayNameFor(product => product.Id)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.Id)</dd>
        <dt class="col-2">@Html.DisplayNameFor(product => product.Name)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.Name)</dd>
        <dt class="col-2">@Html.DisplayNameFor(product => product.Description)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.Description)</dd>
        <dt class="col-2">@Html.DisplayNameFor(product => product.Price)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.Price)</dd>
        <dt class="col-2">@Html.DisplayNameFor(product => product.ImageUrl)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.ImageUrl)</dd>
        <dt class="col-2">@Html.DisplayNameFor(product => product.CategoryId)</dt>
        <dd class="col-10">@Html.DisplayFor(product => product.Category.Name)</dd>
    </dl>

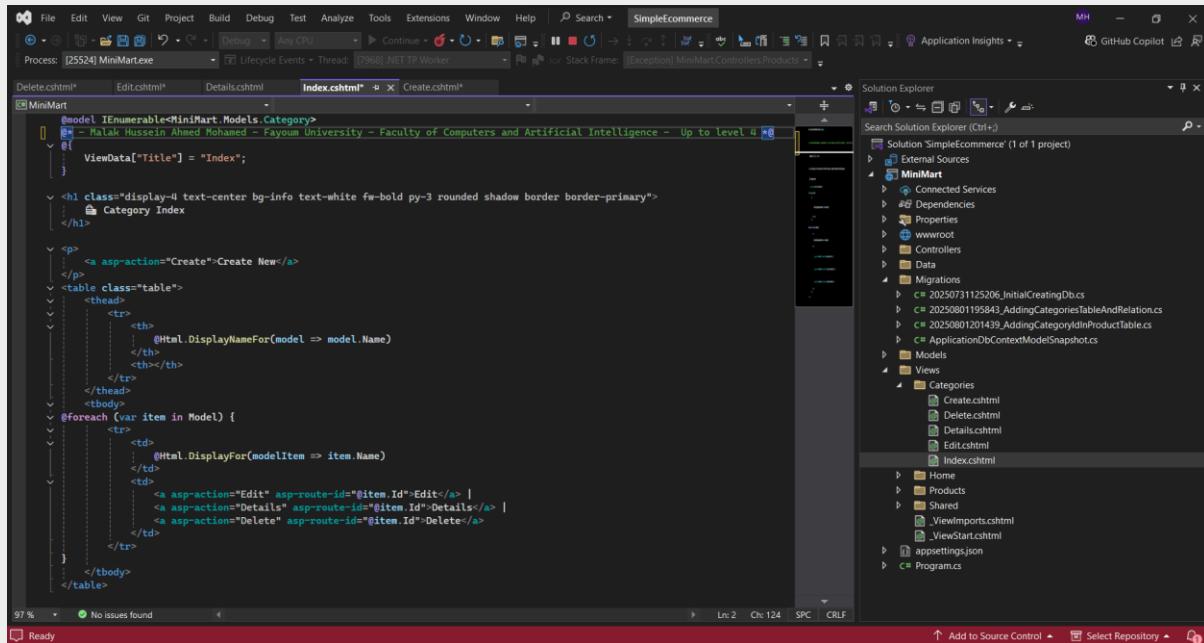
    <form method="post" asp-controller="Products" asp-action="DeleteCurrent">
        <input type="hidden" asp-for="Id" />
        <button type="submit" class="btn btn-danger">Delete</button>
    </form>

    <a asp-controller="Products" asp-action="GetIndexView">
        Back to List
    </a>
</model>
```



## 13. Edited all Views in Categories folder in Views folder:

### a. Index View:

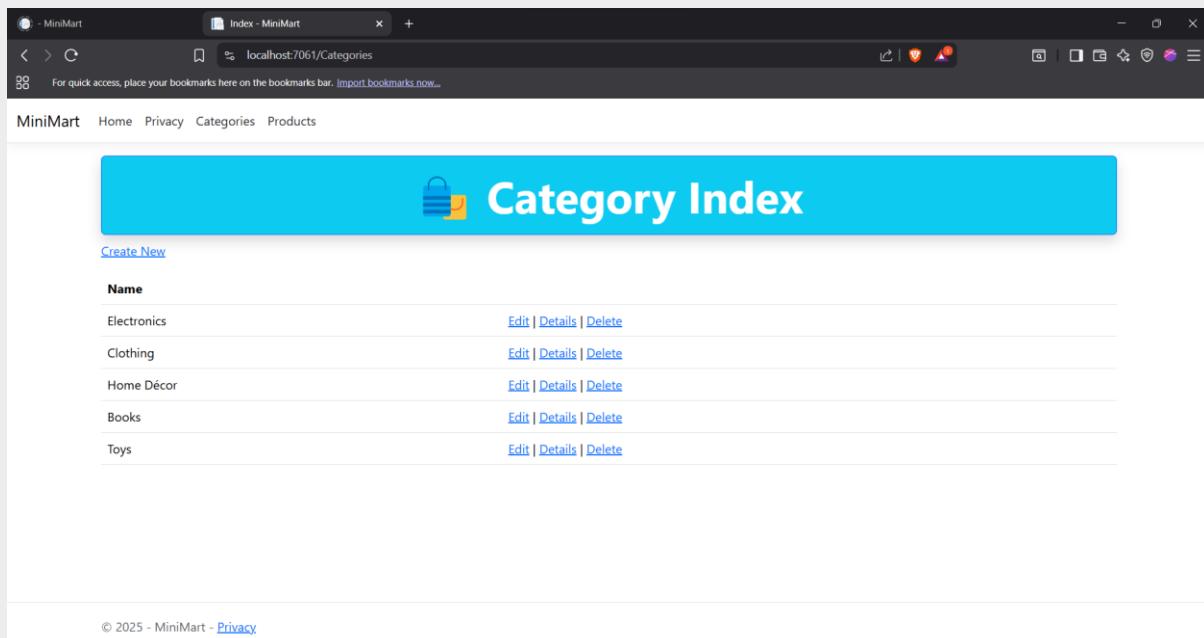


The screenshot shows the Visual Studio IDE with the 'Index.cshtml' file open in the code editor. The code is a C# Razor view for displaying a list of categories. It includes a title bar, a toolbar, a menu bar, and a Solution Explorer window on the right. The code itself starts with a model definition and then uses a foreach loop to iterate through the 'Model' (which is an IEnumerable<Category>) to generate a table with columns for Name, Edit, Details, and Delete actions.

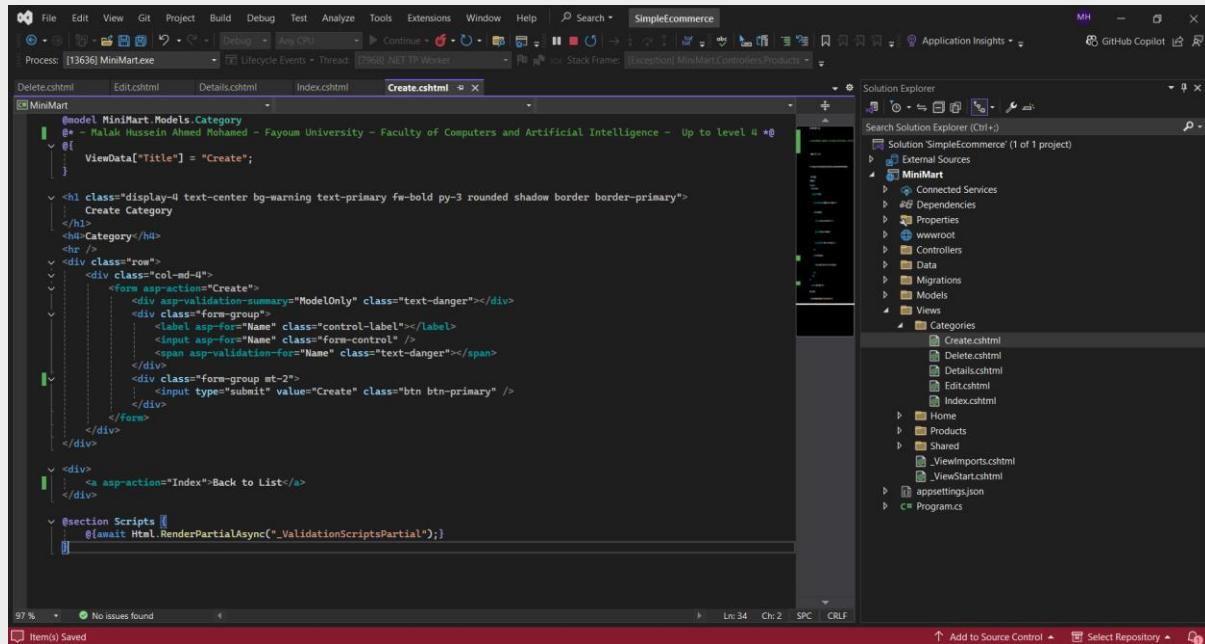
```
model IEnumerable<MiniMart.Models.Category>
{
    ViewData["Title"] = "Index";
}

<h1 class="display-4 text-center bg-info text-white fw-bold py-3 rounded shadow border border-primary">
    Category Index
</h1>

<p>
    <a href="#">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Name)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model) {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem => item.Name)
                </td>
                <td>
                    <a href="#">Edit</a> | 
                    <a href="#">Details</a> | 
                    <a href="#">Delete</a>
                </td>
            </tr>
        }
    </tbody>
</table>
```



## b. Create View:



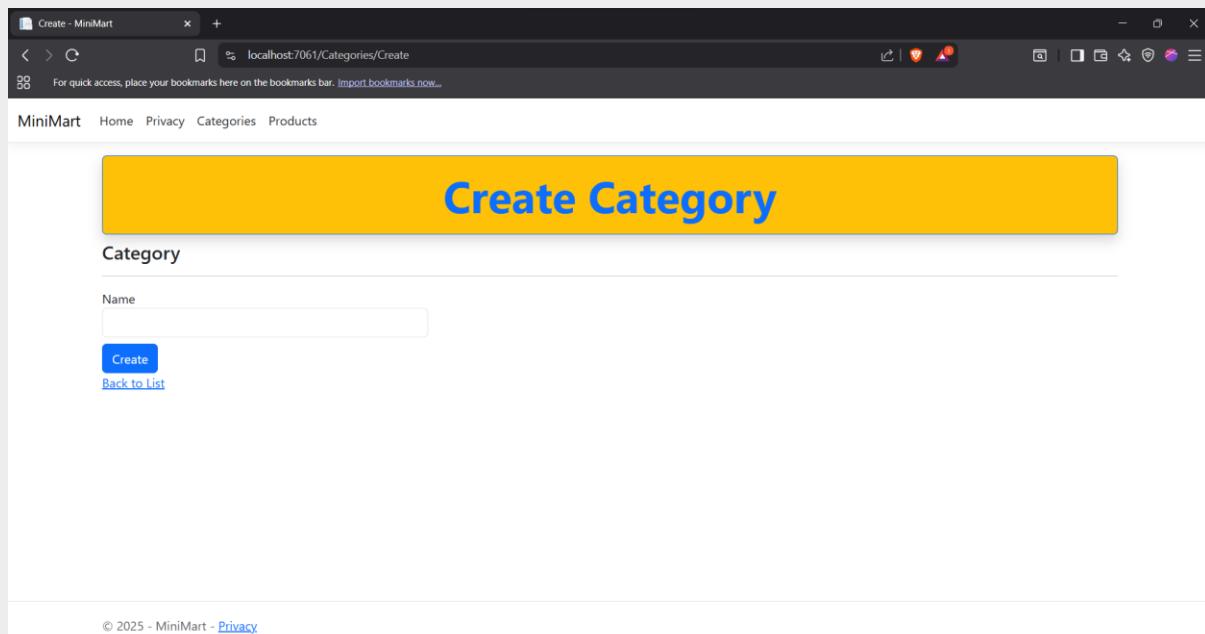
The screenshot shows the Visual Studio IDE interface. The top menu bar includes File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help, and Search. The title bar says "SimpleCommerce". The status bar at the bottom shows "97 %", "0 issues found", "Ln: 34 Ch: 2", and "SPC CRLF".

The main area displays the "Create.cshtml" view code for the "Categories" controller. The code includes a title, a form for creating a category, and a back-to-list link. The Solution Explorer on the right shows the project structure, including the "Views" folder and its "Categories" subfolder.

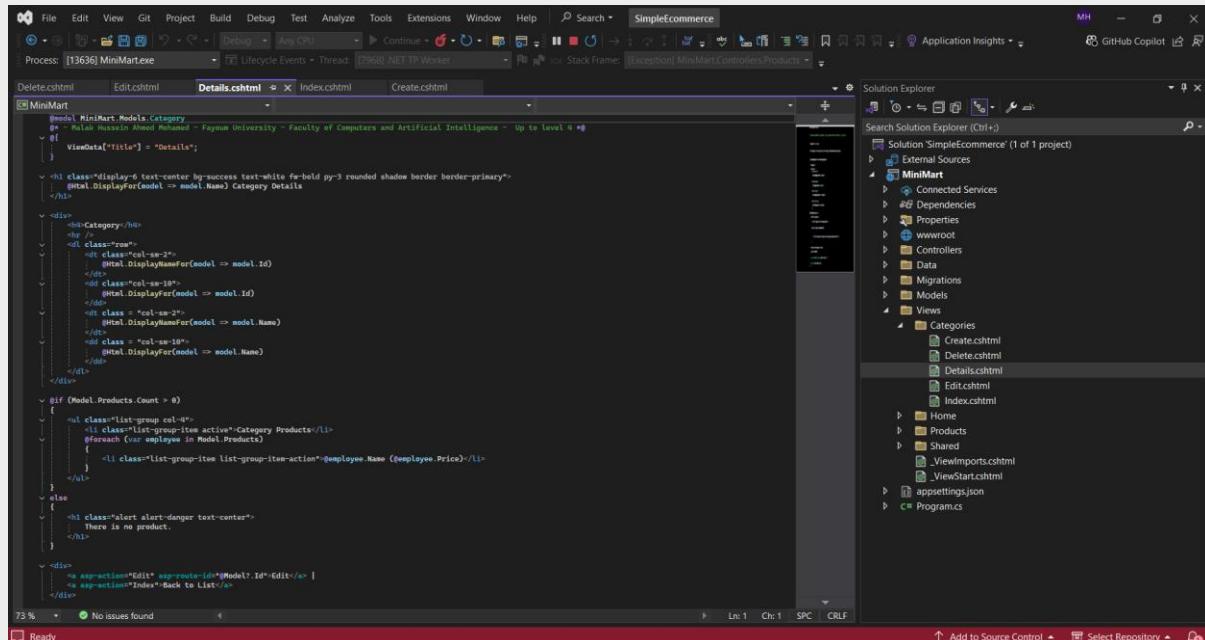
```
@model MiniMart.Models.Category
@{
    ViewData["Title"] = "Create";
}

<h1 class="display-4 text-center bg-warning text-primary fw-bold py-3 rounded shadow border border-primary">
    Create Category
</h1>
<hr />
<div class="row">
    <div class="col-md-4">
        <form action="Create">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="Name" class="control-label"></label>
                <input asp-for="Name" class="form-control" />
                <span asp-validation-for="Name" class="text-danger"></span>
            </div>
            <div class="form-group mt-2">
                <input type="submit" value="Create" class="btn btn-primary" />
            </div>
        </form>
    </div>
    <div>
        <a asp-action="Index">Back to List</a>
    </div>
</div>

@section Scripts {
    @await Html.RenderPartialAsync("_ValidationScriptsPartial");
}
```



## c. Details View:



The screenshot shows the Visual Studio IDE with the code editor open to the `Details.cshtml` file. The code is a Razor view for displaying a category. It includes a title bar with the project name "SimpleCommerce" and the file path "13636 MiniMart.Controllers.Products". The code uses Bootstrap classes for styling, including `text-center`, `text-white`, `py-3`, and `border border-primary`. It displays the category's ID and name, and lists its products if any exist. The Solution Explorer on the right shows the project structure, including the `Views/Categories` folder containing the `Details.cshtml` file.

```
Model: MiniMart.Models.Category
#* - Matlab Hussain Ahmed Mohamed - Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level II #*
@{
    ViewData["Title"] = "Details";
}

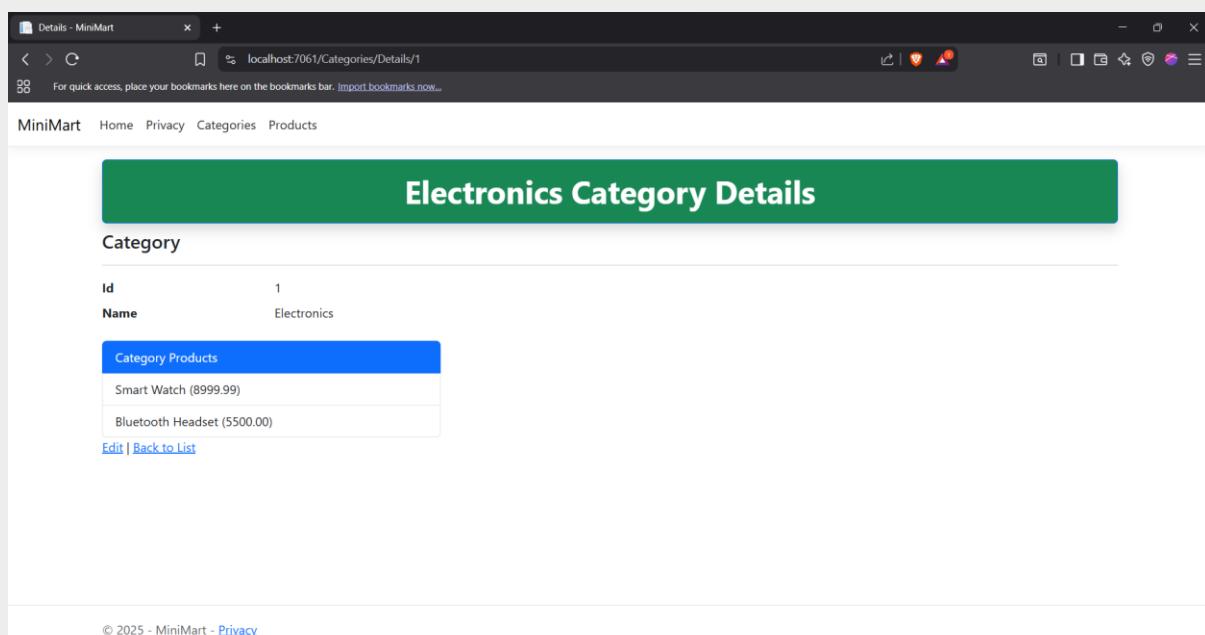


# Category Details



<h3>@Category.Name</h3>
    <hr />
    <div class="row">
        <div class="col-sm-2">
            @Html.DisplayFor(model => model.Id)
        </div>
        <div class="col-sm-10">
            @Html.DisplayFor(model => model.Name)
        </div>
    </div>
    <div class="row">
        <div class="col-2">
            @Html.DisplayFor(model => model.Name)
        </div>
        <div class="col-10">
            @Html.DisplayFor(model => model.Name)
        </div>
    </div>


@if (Model.Products.Count > 0)
{
    <ul class="list-group col-4">
        <li class="list-group-item active">Category Products:</li>
        @foreach (var employee in Model.Products)
        {
            <li class="list-group-item-action" href="#">@employee.Name (@employee.Price)</li>
        }
    </ul>
}
else
{
    <h3 class="alert alert-danger text-center">
        There is no product.
    </h3>
}
<div>
    <a href="#" asp-action="Edit" asp-route-id="@Model.Id">Edit</a> |
    <a href="#" asp-action="Index">Back to List</a>
</div>
```



The screenshot shows a web browser window titled "Details - MiniMart" with the URL `localhost:7061/Categories/Details/1`. The page has a green header bar with the text "Electronics Category Details". Below it, there is a section titled "Category" with two items: "Id" (1) and "Name" (Electronics). A table titled "Category Products" lists two items: "Smart Watch (8999.99)" and "Bluetooth Headset (5500.00)". At the bottom of the page, there are links for "Edit" and "Back to List". The footer of the page contains the copyright notice "© 2025 - MiniMart - Privacy".

## d. Edit View:

The screenshot shows the Visual Studio IDE interface. The code editor displays the `Edit.cshtml` file for the `Categories` view. The code includes a header with the author's name and title, followed by an `h1` heading, an `hr`, and a form for editing a category. The form contains a hidden input for the ID, a label and input for the name, and a save button. Below the form is a link to go back to the list. The Solution Explorer on the right shows the project structure, including the `Views/Categories` folder containing the `Edit.cshtml` file.

```
@model MiniMart.Models.Category
/* - Malak Hussein Ahmed Mohamed - Fayoum University - Faculty of Computers and Artificial Intelligence - Up to level 4 */
@{
    ViewData["Title"] = "Edit";
}

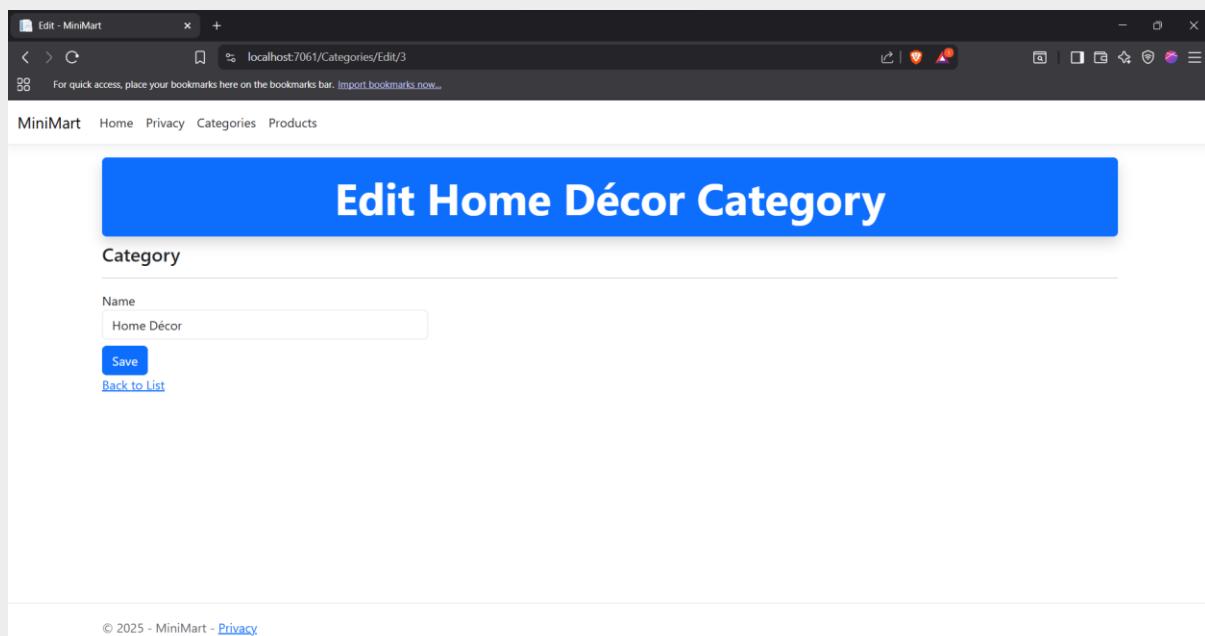
<h1 class="display-4 bg-primary text-white fw-bold text-center py-3 rounded shadow border border-primary">
    Edit @Html.DisplayFor(model => model.Name) Category
</h1>

<hr />

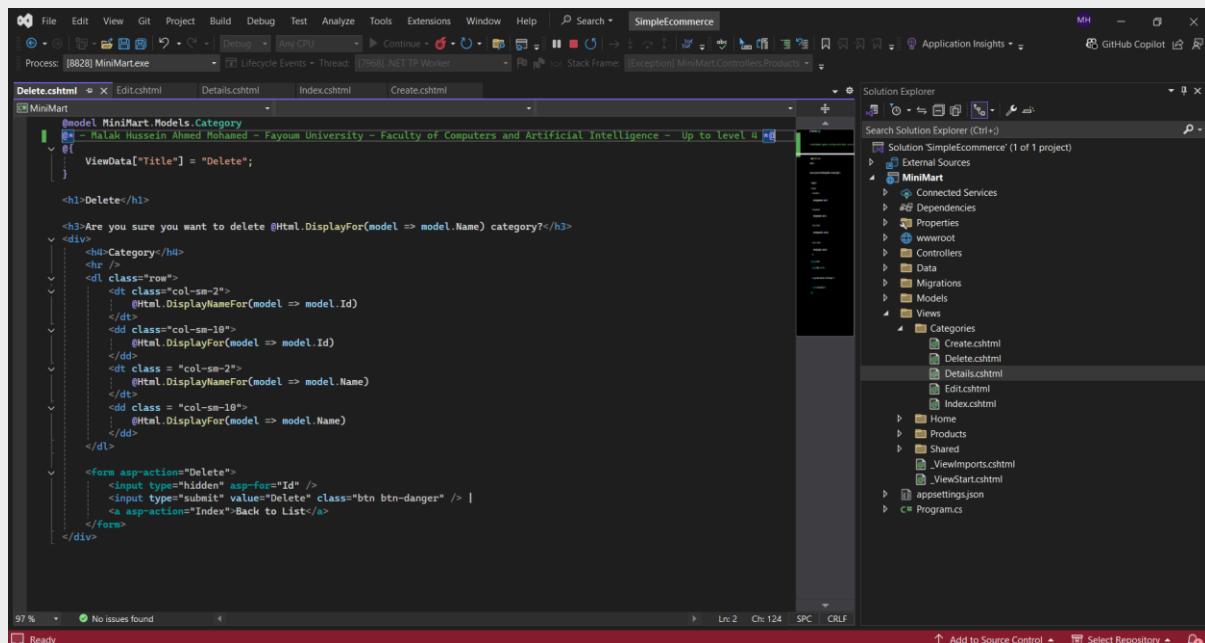

<div class="col-md-4">
        <form asp-action="Edit">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <input type="hidden" asp-for="Id" />
            <div class="form-group">
                <label asp-for="Name" class="control-label"></label>
                <input asp-for="Name" class="form-control" />
                <span asp-validation-for="Name" class="text-danger"></span>
            </div>
            <div class="form-group mt-2">
                <input type="submit" value="Save" class="btn btn-primary" />
            </div>
        </form>
    </div>


<div>
    <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
    @await Html.RenderPartialAsync("_ValidationScriptsPartial");
}
```



## e. Delete View:



The screenshot shows the Visual Studio IDE with the code editor open to the Delete.cshtml file. The code is a Razor view for deleting a category. It includes a title, a confirmation message, a table for displaying category details, and a form with a delete button. The Solution Explorer on the right shows the project structure, including the Views folder and its subfolders.

```
@model MiniMart.Models.Category
@{
    ViewData["Title"] = "Delete";
}



# Delete



Are you sure you want to delete @Html.DisplayFor(model => model.Name) category?

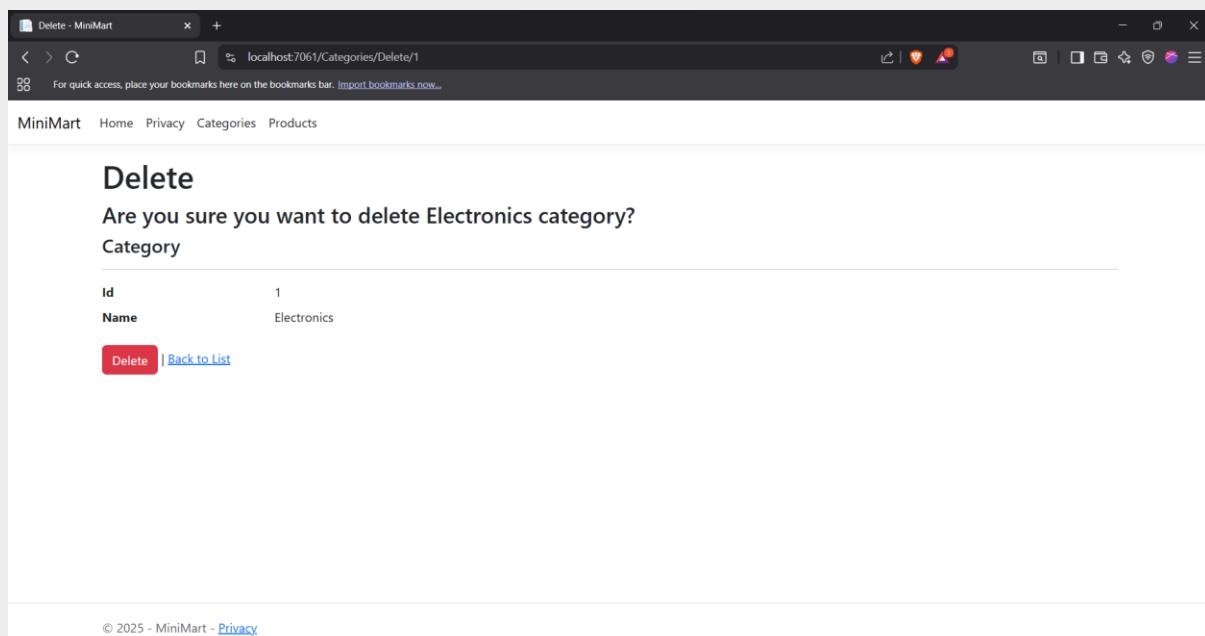


|                                           |                                       |
|-------------------------------------------|---------------------------------------|
| @Html.DisplayNameFor(model => model.Id)   | @Html.DisplayFor(model => model.Id)   |
| @Html.DisplayNameFor(model => model.Name) | @Html.DisplayFor(model => model.Name) |



Back to List


```



## 14. Show database and all tables in SSMS:

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. In the Object Explorer on the left, a tree view shows the database structure under 'DESKTOP-MAIOK84 (SQL Server 16.0.1140.6 - DESKTOP-MAIOK84\Abo\_Elalaa)'. The 'Tables' node is expanded, showing 'Category' and 'Products' tables. The 'Products' table has its columns listed below it. In the center, a query window titled 'SQLQuery3.sq\_Elalaa (67)\*' contains the following T-SQL code:

```
-- Malak Hussein Ahmed Mohamed - Fayoum University - Faculty of Computers and Artificial Intelligence
-- Up to level 4
Use MiniMart;
SELECT * FROM [dbo].[Category];
SELECT * FROM [dbo].[Products];
```

The 'Results' tab displays two result sets. The first result set shows the categories:

Id	Name
1	Electronics
2	Clothing
3	Home Décor
4	Books
5	Toys

The second result set shows the products:

Id	Name	Description	Price	ImageURL	CategoryId
1	Smart Watch	Stylish smartwatch with health tracker	8999.99	https://m.media-amazon.com/images/I/81plzHeN9NL_UF8...	1
2	Bluetooth Headset	Portable headset with deep bass	5500.00	https://resource.logitech.com/v_200_c_low_ar_1.1_q_80.../...	1
3	Denim Jacket	Trendy blue denim jacket for all ages	1999.99	https://www.herscheldivision.com/content/dam/hd/division-p...	2
4	Table Lamp	Decorative lamp with warm light	1000.00	https://www.lk-light.de/images/product_images/popup_im...	3
5	Fiction Book	Exciting mystery novel	399.99	https://hollytaylortherapy.com/wp-content/uploads/2022/09/51...	4
6	Teddy Bear	Soft and cuddly teddy for kids	150.00	data:image/png;base64,iVBORw0KGgoAAAQ...	6

At the bottom of the results grid, a message says 'Query executed successfully.'

Thank You