BÀI 5: XÂY DỰNG ỨNG DỤNG WEBSITE BÁN HÀNG VỚI ASP.NET CORE MVC (PHẦN 3)

Sau khi học xong bài này, sinh viên có thể nắm được:

- Xây dựng chức năng giỏ hàng và đặt hàng trong ứng dụng ASP.NET Core MVC

5.1 Mục tiêu của bài thực hành

5.1.1 Yêu cầu

Xây dựng chức năng giỏ hàng và đặt hàng trong ứng dụng ASP.NET Core MVC

5.2 Hướng dẫn thực hiện

5.2.1 Code mẫu thực hiện chức năng giỏ hàng

Để xây dựng một chức năng giỏ hàng sử dụng session trong ASP.NET Core, bạn cần thực hiện các bước sau:

- *Cấu Hình Session trong `Program.cs`*

```
var builder = WebApplication.CreateBuilder(args);

builder.Services.AddDistributedMemoryCache();
builder.Services.AddSession(options =>
{
    options.IdleTimeout = TimeSpan.FromMinutes(30);
    options.Cookie.HttpOnly = true;
    options.Cookie.IsEssential = true;
```

```
});

builder.Services.AddControllersWithViews();

var app = builder.Build();

// Đặt trước UseRouting
app.UseSession();

// Các middleware khác...
app.UseRouting();
app.UseEndpoints(endpoints =>
{
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller=Home}/{action=Index}/{id?}");
});

app.Run();
```

- *Tạo Helper Class để Làm Việc với Session*

```
public static class SessionExtensions
{
    public static void SetObjectAsJson(this ISession session, string key,
object value)
    {
        session.SetString(key, JsonSerializer.Serialize(value));
    }

    public static T GetObjectFromJson<T>(this ISession session, string
key)
    {
        var value = session.GetString(key);
        return value == null ? default :
JsonSerializer.Deserialize<T>(value);
    }
}
```

- *CartItem.cs*

```
public class CartItem
{
    public int ProductId { get; set; }
    public string Name { get; set; }
    public decimal Price { get; set; }
    public int Quantity { get; set; }
}
```

- *ShoppingCart.cs*

```
public class ShoppingCart
{
    public List<CartItem> Items { get; set; } = new List<CartItem>();

    public void AddItem(CartItem item)
    {
        var existingItem = Items.FirstOrDefault(i => i.ProductId ==
item.ProductId);
        if (existingItem != null)
        {
            existingItem.Quantity += item.Quantity;
        }
        else
        {
            Items.Add(item);
        }
    }

    public void RemoveItem(int productId)
    {
        Items.RemoveAll(i => i.ProductId == productId);
    }

    // Các phương thức khác...
}
```

- *Tạo `ShoppingCartController`*

```
public class ShoppingCartController : Controller
{
    public IActionResult AddToCart(int productId, int quantity)
    {
        // Giả sử bạn có phương thức lấy thông tin sản phẩm từ productId
        var product = GetProductFromDatabase(productId);

        var cartItem = new CartItem
```

```
            {
                ProductId = productId,
                Name = product.Name,
                Price = product.Price,
                Quantity = quantity
            };

            var cart =
HttpContext.Session.GetObjectFromJson<ShoppingCart>("Cart") ?? new
ShoppingCart();
            cart.AddItem(cartItem);

            HttpContext.Session.SetObjectAsJson("Cart", cart);

            return RedirectToAction("Index");
        }

        public IActionResult Index()
        {
            var cart =
HttpContext.Session.GetObjectFromJson<ShoppingCart>("Cart") ?? new
ShoppingCart();
            return View(cart);
        }

        // Các actions khác...

        private Product GetProductFromDatabase(int productId)
        {
            // Truy vấn cơ sở dữ liệu để lấy thông tin sản phẩm
        }
    }
```

- *Index.cshtml (Trong Views/ShoppingCart)*

```
@model ShoppingCart

<h2>Your Cart</h2>

<table>
    <tr>
        <th>Product</th>
        <th>Quantity</th>
        <th>Price</th>
        <th>Total</th>
    </tr>
```

```
        @foreach (var item in Model.Items)
        {
            <tr>
                <td>@item.Name</td>
                <td>@item.Quantity</td>
                <td>@item.Price</td>
                <td>@(item.Price * item.Quantity)</td>
                <td>
                    <a asp-action="RemoveFromCart" asp-route-
productId="@item.ProductId">Remove</a>
                </td>
            </tr>
        }
    </table>
```

Lưu ý:

- Đảm bảo rằng bạn đã xử lý việc lấy thông tin sản phẩm từ cơ sở dữ liệu trong phương thức `GetProductFromDatabase`.

5.2.2 Hướng dẫn thực hiện chức năng đặt hàng

- *Model `Order`*

```
    public class Order
    {
        public int Id { get; set; }
        public string UserId { get; set; }
        public DateTime OrderDate { get; set; }
        public decimal TotalPrice { get; set; }

        public string ShippingAddress { get; set; }
        public string Notes { get; set; }

        public IdentityUser User { get; set; }
        public List<OrderDetail> OrderDetails { get; set; }
    }
```

- *OrderDetail (Chi Tiết Đơn Hàng): Lưu thông tin chi tiết cho mỗi mặt hàng trong đơn.*

```
    public class OrderDetail
    {
        public int Id { get; set; }
        public int OrderId { get; set; }
        public int ProductId { get; set; }
        public int Quantity { get; set; }
```

```
        public decimal Price { get; set; }

        public Order Order { get; set; }
        public Product Product { get; set; }
    }
```

- *Cập Nhật ApplicationDbContext*

Thêm DbSet cho các lớp mới vào ApplicationDbContext.

```
    public class ApplicationDbContext : IdentityDbContext<IdentityUser>
    {
        // ... Các DbSet hiện có ...

        public DbSet<Order> Orders { get; set; }
        public DbSet<OrderDetail> OrderDetails { get; set; }
    }
```

- *`ShoppingCartController` để Xử Lý Đặt Hàng*

Trong `ShoppingCartController`, cập nhật action `Checkout` để xử lý thông tin địa chỉ giao hàng và ghi chú.

```
    [Authorize]
    public class ShoppingCartController : Controller
    {
        private readonly ApplicationDbContext _context;
        private readonly UserManager<IdentityUser> _userManager;
        public ShoppingCartController(ApplicationDbContext context,
UserManager<IdentityUser> userManager)
        {
            _context = context;
           _userManager = userManager;
        }

        public IActionResult Checkout()
        {
            return View(new Order());
        }

        [HttpPost]
        public async Task<IActionResult> Checkout(Order order)
        {
            var cart =
HttpContext.Session.GetObjectFromJson<ShoppingCart>("Cart");
            if (cart == null || !cart.Items.Any())
```

```
            {
                // Xử lý giỏ hàng trống...
                return RedirectToAction("Index");
            }

            var user = await _userManager.GetUserAsync(User);
            order.UserId = user.Id;
            order.OrderDate = DateTime.UtcNow;
            order.TotalPrice = cart.Items.Sum(i => i.Price * i.Quantity);
            order.OrderDetails = cart.Items.Select(i => new OrderDetail
            {
                ProductId = i.ProductId,
                Quantity = i.Quantity,
                Price = i.Price
            }).ToList();

            _context.Orders.Add(order);
            await _context.SaveChangesAsync();

            HttpContext.Session.Remove("Cart");

            return View("OrderCompleted", order.Id); // Trang xác nhận hoàn
thành đơn hàng
        }
    }
```

- *Tạo View `Checkout` Checkout.cshtml*

Tạo một view mới để nhập thông tin đặt hàng, bao gồm địa chỉ giao hàng và ghi chú.

```
@model Order

<h2>Checkout</h2>

<form asp-action="Checkout" method="post">
    <div class="form-group">
        <label asp-for="ShippingAddress">Shipping Address</label>
        <input asp-for="ShippingAddress" class="form-control" />
    </div>
    <div class="form-group">
        <label asp-for="Notes">Notes</label>
        <textarea asp-for="Notes" class="form-control"></textarea>
    </div>
    <button type="submit" class="btn btn-primary">Place Order</button>
</form>
```

- *Cập Nhật View `OrderCompleted` OrderCompleted.cshtml*

Cập nhật view `OrderCompleted` để hiển thị thông tin xác nhận đơn hàng.

```
@model int

<h2>Order Completed</h2>
<p>Your order with ID @Model has been placed successfully.</p>
```

5.2.3 Yêu cầu bổ sung

- Xây dựng giao diện thân thiện cho giỏ hàng.