Module 12: Performance and Communication

Lab: Performance and Communication

Scenario

You have been asked to create a web-based electrical store application for your organization's customers. The application should have a page showing the sale of the day, products sorted by category, the ability to add products to your shopping list, and a chat page that lets users talk online. The application should contain a cache tag helper to cache content in a view, a memory cache in a controller, and session state configuration. Finally, you will write a chat room app by using SignalR.

Exercise 1: Implementing a Caching Strategy

Scenario

Caching can be used to greatly improve the performance of a web application. In this exercise, you will first add a cache tag helper to a view. After that, you will use the memory cache to store and retrieve items.

The main tasks for this exercise are as follows:

- Add a cache tag helper to a view
- Insert data to be cached by the cache tag helper
- Run the application
- Insert items to a memory cache
- Run the application

Task 1: Add a cache tag helper to a view

☐ 1. In File Explorer, navigate to **D:\Allfiles\Mod12\Labfiles\01_ElectricStore_begin**, and then double-click **ElectricStore.sln**.

> ⚠ **Note**: If a **Security Warning for Library** dialog box appears, verify that the **Ask me for every project in this solution** check box is cleared, and then click OK.

☐ 2. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, expand **Views**, expand

□ 3. In the **_Layout.cshtml** code window, locate the following code:

```
<ul class="navbar-nav" id="nav-content">
</ul>
```

□ 4. Place the cursor before the **<** (less than) sign of the **</ul>** tag, press Enter, press the up arrow key, and then type the following code:

```
<cache vary-by-route="RefreshCache">
    @await Component.InvokeAsync("NavbarMenu")
</cache>
```

Task 2: Insert data to be cached by the cache tag helper

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, right-click **ElectricStore**, point to **Add**, and then click **New Folder**.

□ 2. In the **NewFolder** box, type ViewComponents, and then press Enter.

□ 3. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, right-click **ViewComponents**, point to **Add**, and then click **Class...**.

□ 4. In the **Add New Item -- ElectricStore** dialog box, in the **Name** box, type NavbarMenuViewComponent, and then click **Add**.

□ 5. In the **NavbarMenuViewComponent.cs** code window, add the following code to the beginning of the file:

```
using ElectricStore.Data;
using Microsoft.AspNetCore.Mvc;
```

□ 6. In the **NavbarMenuViewComponent.cs** code window, locate the following code:

```
public class NavbarMenuViewComponent
```

□ 7. Append the following code to the existing line of code.

```
: ViewComponent
```

□ 8. In the **NavbarMenuViewComponent.cs** code window, place the cursor within the **NavbarMenuViewComponent** code block, and then type the following code:

```
    public NavbarMenuViewComponent(StoreContext context)
    {
        _context = context;
    }

    public IViewComponentResult Invoke()
    {
        var categories = _context.menuCategories.OrderBy(c => c.Name).ToList();
        return View("MenuCategories", categories);
    }
```

□ 9. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, under **Views**, right-click **Shared**, point to **Add**, and then click **New Folder**.

□ 10. In the **NewFolder** box, type Components, and then press Enter.

□ 11. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, right-click **Components**, point to **Add**, and then click **New Folder**.

□ 12. In the **NewFolder** box, type NavbarMenu, and then press Enter.

□ 13. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, right-click **NavbarMenu**, point to **Add**, and then click **New Item...**.

□ 14. In the **Add New Item - ElectricStore** dialog box, click **Web** in the navigation tree, and then, in the result pane, click **Razor View - Empty**.

□ 15. In the **Add New Item - ElectricStore** dialog box, in the **Name** box, type MenuCategories, and then click **Add**.

□ 16. In the **MenuCategories.cshtml** code window, replace the entire contents of the file with the following code:

```
        <li class="nav-item">
            <a class="nav-link" href="@Url.Action("Index", "Products")">Home<span class="sr
        </li>

        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" data-t
                Electronics
            </a>
            <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
                @foreach (var item in Model)
                {
                    <a class="dropdown-item" asp-action="GetByCategory" asp-controller="Pro
                }
            </div>
        </li>

        <li class="nav-item">
            <a class="nav-link" href="@Url.Action("AddToShoppingList", "Products")">Add to
        </li>
        <li class="nav-item">
            <a class="nav-link" href="@Url.Action("Index", "ShoppingCart")">My Shopping Car
        </li>
        <li class="nav-item">
            <div class="display-cached-time">The content cached Since: @DateTime.Now</div>
        </li>
```

Task 3: Run the application

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.

□ 2. In the **ElectricStore - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Debugging**.

□ 3. On the taskbar, right-click the **Microsoft Edge** icon, and then click **New Window**.

□ 4. In the second **Microsoft Edge** window, in the address bar, type http://localhost:50672, and then press Enter.

> **Note**: Examine the browser content. The menu bar of the application is cached since the time displayed in the menu bar; the time is the same in both **Microsoft Edge** pages.

□ 5. On the taskbar, right-click the **Microsoft Edge** icon, and then click **New Window**.

□ 6. In the third **Microsoft Edge** window, in the address bar, type http://localhost:50672/products/index/1/RefreshCache, and then press Enter.

> the time is not the same as the other two **Microsoft Edge** pages because the **RefreshCache** in the route triggers a reload of the view component.

□ 7. Close all **Microsoft Edge** windows.

Task 4: Insert items to a memory cache

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, expand **Controllers**, and then click **ProductsController.cs**.

□ 2. In the **ProductsController.cs** code window, locate the following code:

```
using Microsoft.EntityFrameworkCore;
```

□ 3. Ensure that the cursor is at the end of the **Microsoft.EntityFrameworkCore** namespace, press Enter, and then type the following code:

```
using Microsoft.Extensions.Caching.Memory;
```

□ 4. In the **ProductsController.cs** code block, locate following code:

```
private StoreContext _context;
private IWebHostEnvironment _environment;
```

□ 5. Place the cursor at the end of the located code, press Enter, and then type the following code:

```
private IMemoryCache _memoryCache;
private const string PRODUCT_KEY = "Products";
```

□ 6. In the **ProductsController.cs** code block, select the following code:

```
public ProductsController(StoreContext context, IWebHostEnvironment environment)
{
    _context = context;
    _environment = environment;
}
```

□ 7. Replace the selected code with the following code:

```
public ProductsController(StoreContext context, IWebHostEnvironment environment, IM
{
    _context = context;
    _environment = environment;
    _memoryCache = memoryCache;
}
```

```
public IActionResult Index()
{
    return View();
}
```

□ 9. Replace the selected code with the following code:

```
public IActionResult Index()
{
    List<Product> products;
    if (!_memoryCache.TryGetValue(PRODUCT_KEY, out products))
    {
        products = _context.Products.ToList();
        products.Select(c => { c.LoadedFromDatabase = DateTime.Now; return c; }).To
        MemoryCacheEntryOptions cacheOptions = new MemoryCacheEntryOptions();
        cacheOptions.SetPriority(CacheItemPriority.High);
        _memoryCache.Set(PRODUCT_KEY, products, cacheOptions);
    }
    return View(products);
}
```

□ 10. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, under **Views**, expand **Products**, and then click **Index.cshtml**.

□ 11. In the **Index.cshtml** code window, locate the following code:

```
<h2 class="title">
    <img src="~/images/sale-banner.jpg" />
    <span class="span-col-s">Sale of The Day</span>
</h2>
```

□ 12. Place the cursor after the **>** (greater than) sign of the **</h2>** tag, press Enter twice, and then type the following code:

```
            {
                <div class="item">
                    <h3>
                        @Html.DisplayFor(modelItem => item.ProductName)
                    </h3>
                    @if (item.PhotoFileName != null)
                    {
                        <div>
                            <img src="@Url.Action("GetImage", "Products", new { productId =
                        </div>
                    }
                        <div>
                            <p>
                                @Html.DisplayFor(model => item.Description)
                            </p>
                            <p>
                                @Html.DisplayFor(model => item.Price)
                            </p>
                        </div>
                        <div>
                            <p>
                                @Html.DisplayNameFor(model => item.LoadedFromDatabase)
                            </p>
                            <p>
                                @Html.DisplayFor(model => item.LoadedFromDatabase)
                            </p>
                        </div>
                </div>
            }
        </div>
```

Task 5: Run the application

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.

□ 2. In the **ElectricStore - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Debugging**.

> **Note**: Examine the products **Last retrieved on** data.

□ 3. On the taskbar, right-click the **Microsoft Edge** icon, and then click **New Window**.

□ 4. In the second **Microsoft Edge** window, in the address bar, type http://localhost:50672/, and then press Enter.

> **Note**: On examining the browser contents, you can see that products in the **Sale of The**

□ 5. Close all **Microsoft Edge** windows.

✓ **Results**: After completing this exercise, you have implemented a caching strategy by adding a cache tag helper to a view, and inserted items into a memory cache.

Exercise 2: Managing State

Scenario

HTTP is a stateless protocol, but in web applications it is frequently necessary to retain information across requests. This is referred to as application state, and is typically associated with a user's session. In this exercise, you will use server-side session state in a web application.

The main tasks for this exercise are as follows:

- Enable working with sessions
- Use session to store values
- Retrieve values from a session
- Run the application and navigate from view to view

Task 1: Enable working with sessions

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, click **Program.cs**.

□ 2. In the **Program.cs** code window, locate the following code:

```
builder.Services.AddDbContext<StoreContext>(options =>
    options.UseSqlite("Data Source=electricStore.db"))
```

□ 3. Ensure that the cursor is at the end of the located code, press Enter twice, and then type the following code:

```
builder.Services.AddSession(
    options => options.IdleTimeout = TimeSpan.FromSeconds(60));
```

□ 4. In the **Program.cs** code window, locate the following code:

```
app.UseStaticFiles();
```

□ 5. Ensure that the cursor is at the end of the located code, press Enter, and then type the following code:

go deploy ☰

Task 2: Use session to store values

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, under **Controllers**, click **ProductsController.cs**.

□ 2. In the **ProductsController.cs** code window, locate the following code:

```
using Microsoft.Extensions.Caching.Memory;
```

□ 3. Ensure that the cursor is at the end of the **Microsoft.Extensions.Caching.Memory** namespace, press Enter, and then type the following code:

```
using System.Text.Json;
```

□ 4. In the **ProductsController.cs** code block, select the following code:

```
[HttpPost, ActionName("AddToShoppingList")]
public IActionResult AddToShoppingListPost(Customer customer)
{
    if (ModelState.IsValid)
    {
        _context.Customers.Add(customer);
        _context.SaveChanges();
        return RedirectToAction(nameof(Index));
    }
    PopulateProductsList(customer.SelectedProductsList);
    return View(customer);
}
```

□ 5. Replace the selected code with the following code:

```
        {
            if (ModelState.IsValid)
            {
                HttpContext.Session.SetString("CustomerFirstName", customer.FirstName);
                HttpContext.Session.SetString("CustomerLastName", customer.LastName);
                HttpContext.Session.SetString("CustomerEmail", customer.Email);
                HttpContext.Session.SetString("CustomerAddress", customer.Address);
                HttpContext.Session.SetInt32("CustomerPhoneNumber", customer.PhoneNumber);
                if (HttpContext.Session.GetString("CustomerProducts") != null)
                {
                    List<int> productsListId = JsonSerializer.Deserialize<List<int>>(HttpCo
                    customer.SelectedProductsList.AddRange(productsListId);
                }
                var serialisedDate = JsonSerializer.Serialize(customer.SelectedProductsList
                HttpContext.Session.SetString("CustomerProducts", serialisedDate);
                return RedirectToAction(nameof(Index));
            }
            PopulateProductsList(customer.SelectedProductsList);
            return View(customer);
        }
```

Task 3: Retrieve values from a session

☐ 1. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, under **Controllers**, click **ProductsController.cs**.

☐ 2. In the **ProductsController.cs** code block, select the following code:

```
[HttpGet]
public IActionResult AddToShoppingList()
{
    PopulateProductsList();
    return View();
}
```

☐ 3. Replace the selected code with the following code:

```
{
    if (HttpContext.Session.GetString("CustomerFirstName") != null)
    {
        Customer sessionCustomer = new Customer()
        {
            FirstName = HttpContext.Session.GetString("CustomerFirstName"),
            LastName = HttpContext.Session.GetString("CustomerLastName"),
            Email = HttpContext.Session.GetString("CustomerEmail"),
            Address = HttpContext.Session.GetString("CustomerAddress"),
            PhoneNumber = HttpContext.Session.GetInt32("CustomerPhoneNumber").Value
        };
        PopulateProductsList();
        return View(sessionCustomer);
    }
    PopulateProductsList();
    return View();
}
```

□ 4. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, under **Controllers**, click **ShoppingCartController.cs**.

□ 5. In the **ShoppingCartController.cs** code window, locate the following code:

```
using Microsoft.AspNetCore.Mvc;
```

□ 6. Ensure that the cursor is at the end of the **Microsoft.AspNetCore.Mvc** namespace, press Enter, and then type the following code:

```
using System.Text.Json;
```

□ 7. In the **ShoppingCartController.cs** code block, locate the following code:

```
    public IActionResult Index()
    {
        return View();
    }
```

□ 8. Place the cursor before the located code, press Enter, press the up arrow key, type the following code, and then press Enter.

```
        private SessionStateViewModel sessionModel;

        public ShoppingCartController(StoreContext context)
        {
            _context = context;
        }
```

□ 9. In the **ShoppingCartController.cs** code block, in the **Index** action code block, select the following code:

```
return View();
```

□ 10. Replace the selected code with the following code:

```
if (!string.IsNullOrEmpty(HttpContext.Session.GetString("CustomerFirstName")) && !s
{
    List<int> productsListId = JsonSerializer.Deserialize<List<int>>(HttpContext.Se
    products = new List<Product>();
    foreach (var item in productsListId)
    {
        var product = _context.Products.SingleOrDefault(p => p.Id == item);
        products.Add(product);
    }
    sessionModel = new SessionStateViewModel
    {
        CustomerName = HttpContext.Session.GetString("CustomerFirstName"),
        SelectedProducts = products
    };
    return View(sessionModel);
}
return View();
```

Task 4: Run the application

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.

□ 2. In the **ElectricStore - Microsoft Visual Studio** window, on the **Debug** menu, click **Start Debugging**.

□ 3. In the browser, on the menu bar, click **My Shopping Cart**.

> **Note**: Examine the browser contents - the shopping cart is empty.

□ 4. On the menu bar, click **Add to Shopping List**.

□ 5. On the **Add Products to Shopping List** page, in the **Products List** list, select **TV**.

□ 7. On the **Add Products to Shopping List** page, in the **Last Name** box, type your second name.

□ 8. On the **Add Products to Shopping List** page, in the **Address** box, type any string.

□ 9. On the **Add Products to Shopping List** page, in the **Email** box, type a plausible email address.

□ 10. On the **Add Products to Shopping List** page, in the **Phone** box, type some digits, and then click **Add to Shopping List**.

□ 11. In the menu bar, click **My Shopping Cart**.

> **Note**: Examine the browser contents. The browser displays the **ShoppingCart.cshtml** file content, rendered by **_Layout.cshtml**, and the cached contents are the products that were selected by you in the **Add Products to Shopping List** page.

□ 12. On the menu bar, click **Add to Shopping List**.

> **Note**: Examine the browser content. The browser displays your cached personal information.

□ 13. On the **Add Products to Shopping List** page, in the **Products List** list, select **Coffee Machine**, and then click **Add to Shopping List**.

□ 14. On the menu bar, click **My Shopping Cart**.

> **Note**: Examine the browser contents. The product you selected in the second selection is displayed, in addition to the previous selection, in the cached list of products.

□ 15. On the taskbar, right-click the **Microsoft Edge** icon, and then click **New InPrivate window**.

□ 16. In the second **Microsoft Edge** window, in the address bar, type **http://localhost:50672**, and then press Enter.

□ 17. On the menu bar, click **My Shopping Cart**.

> **Note**: Examine the browser content. The list of products is not displayed because the shopping list is empty in this new session. The session is saved only in the original browser context.

□ 18. Close all the **Microsoft Edge** windows.

sessions.

Exercise 3: Two-Way Communication

Scenario

In this exercise, you will first add a SignalR Hub class named ChatHub, and register the ChatHub. Then, you will add a chat view. Finally, you will write the JavaScript code to connect to the server, and run the application and navigate from view to view.

The main tasks for this exercise are as follows:

- Add a SignalR Hub class named ChatHub
- Register the ChatHub in Program.cs
- Add a chat view
- Write the JavaScript code to connect to the server
- Run the application

Task 1: Add a SignalR Hub class named ChatHub

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, right-click **ElectricStore**, point to **Add**, and then click **New Folder**.

□ 2. In the **NewFolder** box, type **Hubs**, and then press Enter.

□ 3. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, right-click **Hubs**, point to **Add**, and then click **Class...**.

□ 4. In the **Add New Item - ElectricStore** dialog box, in the **Name** box, type **ChatHub**, and then click **Add**.

□ 5. In the **ChatHub.cs** code window, add the following code at the beginning of the file:

```
using Microsoft.AspNetCore.SignalR;
```

□ 6. In the **ChatHub.cs** code window, select the following code:

```
public class ChatHub
```

□ 7. Append the following code to the existing line of code.

```
: Hub
```

```
public async Task SendMessageAll(string user, string message)
{
    await Clients.All.SendAsync("NewMessage", user, message);
}
```

Task 2: Configure SignalR in Program.cs

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, click **Program.cs**.

□ 2. In the **Program.cs** code window, locate the following code:

```
using Microsoft.EntityFrameworkCore;
```

□ 3. After the located code, hit Enter, and then type the following code:

```
using ElectricStore.Hubs;
```

□ 4. In the **Program.cs** code window, locate the following code:

```
builder.Services.AddSession(
    options => options.IdleTimeout = TimeSpan.FromSeconds(60));
```

□ 5. Ensure that the cursor is at the end of the located code, press Enter twice, and then type the following code:

```
builder.Services.AddSignalR();
```

□ 6. In the **Program.cs** code window, locate the following code:

```
app.UseSession();
```

□ 7. Ensure that the cursor is at the end of the located code, press Enter, and then type the following code:

```
app.MapHub<ChatHub>("/chatHub");
```

Task 3: Add a chat view

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, under **Controllers**, click **ShoppingCartController.cs**.

□ 2. Ensure that the cursor is at the end of the **Index** action code block, press Enter twice, and then type the following code:

```
        return View();
    }
```

□ 3. In the **ShoppingCartController.cs** code window, right-click the following code, and then click **Add View...**.

```
public IActionResult Chat()
```

□ 4. In the **Add New Scaffolded Item** dialog box, ensure that **Razor View** (not the empty template) is selected, and click **Add**.

□ 5. In the **Add Razor View** dialog box, ensure that the value in the **View name** textbox is **Chat**.

□ 6. In the **Add Razor View** dialog box, ensure that the **Create as a partial view** check box is cleared and the **Use a layout page** check box is selected, and then click **Add**.

□ 7. In the **Chat.cshtml** code window, select the following code:

```
<h1>Chat</h1>
```

□ 8. Replace the selected code with the following code:

```
<h2 class="title">
    <span>Chat with</span>
    <span class="chat-title">Online Site Visitors</span>
</h2>

<div class="chat">
    <div class="row justify-content-center">
        <div class="col-4 bcf">
            <div class="col-5">
                Name: <input type="text" id="inputUserName" />
            </div>
            <div class="col-sm-5">
                Message: <input type="text" id="inputMessage" />
            </div>
            <div class="col-sm-3">
                <input class="btn btn-info" type="submit" id="sendMessageBtn" value
            </div>
        </div>
        <div class="col-6 bcs">
            <ul id="messagesList"></ul>
        </div>
    </div>
</div>
```

□ 9. Place the cursor after the **>** (greater than) sign of the last **</div>** tag, press Enter twice, and then

```
@section Scripts {
    <script src="~/microsoft/signalr/dist/browser/signalr.js"></script>
}
```

□ 10. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, under **Views**, expand **ShoppingCart**, and then click **Index.cshtml**.

□ 11. In the **Index.cshtml** code window, append the following code:

```
<button type="button" id="btnChat" class="btn btn-success" onclick="location.href='
```

Task 4: Write the JavaScript code to connect to the server

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, right-click **wwwroot**, point to **Add**, and then click **New Folder**.

□ 2. In the **NewFolder** box, type js, and then press Enter.

□ 3. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, under **wwwroot**, right-click **js**, point to **Add**, and then click **Existing Item...**.

□ 4. In the **Add Existing Item - ElectricStore** dialog box, go to **D:\Allfiles\Mod12\Labfiles\ElectricStoreJS**, select **chat-hub.js** file, and then click **Add**.

□ 5. In Solution Explorer, click on the **chat-hub.js** file you just added to view its contents.

> **Note**: The client-side JavaScript uses jQuery to send and receive messages using SignalR.

□ 6. In the **ElectricStore - Microsoft Visual Studio** window, in Solution Explorer, under **Views**, under **ShoppingCart**, click **Chat.cshtml**.

□ 7. In the **Chat.cshtml** code window, find the following code:

```
@section Scripts {
    <script src="~/microsoft/signalr/dist/browser/signalr.js"></script>
```

□ 8. After the selected code add the following additional script tag to include the script you just added:

```
<script src="~/js/chat-hub.js"></script>
```

Task 5: Run the application

□ 1. In the **ElectricStore - Microsoft Visual Studio** window, on the **File** menu, click **Save All**.

□ 2. In the **ElectricStore - Microsoft Visual Studio** window, on the **Debug** menu, click **Start**

□ 3. In browser, on the menu bar, click **My Shopping Cart**.

□ 4. On the **My Shopping Cart** page, click **Chat with Online Visitors**.

□ 5. On the taskbar, right-click the **Microsoft Edge** icon, and then click **New InPrivate window**.

□ 6. In the second **Microsoft Edge** window, in the address bar, type **http://localhost:50672/ShoppingCart/Chat**, and then press Enter.

□ 7. In the second **Microsoft Edge** window, in the **Name** box, type **Bill**.

□ 8. In Microsoft Edge, in the **Message** box, type some text, and then click **Send Message** (you may find you need to click the button twice).

> **Note**: In both **Microsoft Edge** windows, the message appears immediately.

□ 9. In the first **Microsoft Edge** window, in the **Name** box, type the name **Mary**.

□ 10. In the first **Microsoft Edge** window, in the **Message** box, type some text, and then click **Send Message**.

> **Note**: In both the **Microsoft Edge** windows, the message appears.

□ 11. Close all the **Microsoft Edge** windows.

□ 12. In the **ElectricStore - Microsoft Visual Studio** window, on the **File** menu, click **Exit**.

> ✓ **Results**: After completing this exercise, you have created an electrical store in which users can view product details, order products, and chat with online site visitors in real time.