**PRACTICAL 10**

**Aim: Develop ASP.NET Core MVC application to demonstrate use of Response Caching and Caching Data using IMemoryCache.**

**Code:**

**HomeController.cs :** (Controllers / HomeController.cs)

using Microsoft.AspNetCore.Mvc;

using Microsoft.Extensions.Caching.Memory;

using PRACTICAL10.Models;

namespace PRACTICAL10.Controllers {

public class HomeController : Controller {

private readonly IMemoryCache \_memoryCache;

private const string CacheKey = "CachedProductData";

public HomeController(IMemoryCache memoryCache) {

\_memoryCache = memoryCache;

}

public IActionResult Index() {

return View();

}

[ResponseCache(Duration = 30, Location = ResponseCacheLocation.Client)]

public IActionResult CacheResponse() {

ViewBag.Message = $"Response cached at: {DateTime.Now}";

return View();

}

public IActionResult MemoryCacheData() {

List<ProductModel>? products;

if (!\_memoryCache.TryGetValue(CacheKey, out products) || products == null) {

products = new List<ProductModel> {

new ProductModel { Id = 1, Name = "Product A", Price = 100 },

new ProductModel { Id = 2, Name = "Product B", Price = 200 },

new ProductModel { Id = 3, Name = "Product C", Price = 300 }

};

*// Set cache options*

var cacheOptions = new MemoryCacheEntryOptions()

.SetSlidingExpiration(TimeSpan.FromSeconds(30))

.SetAbsoluteExpiration(TimeSpan.FromMinutes(2));

*// Save data in cache*

\_memoryCache.Set(CacheKey, products, cacheOptions);

ViewBag.Message = "Data is cached.";

}

else {

ViewBag.Message = "Data retrieved from cache.";

}

return View(products);

}

}

}

**ProductModel.cs :** (Models / ProductModel.cs)

namespace PRACTICAL10.Models {

public class ProductModel {

public int Id { get; set; }

public string? Name { get; set; }

public decimal Price { get; set; }

}

}

**Index.cshtml :** (Views / Home / Index.cshtml)

@{

ViewData["Title"] = "Index";

}

<h1>ASP.NET Core MVC Caching Demo</h1>

<ul>

<li><a href="/Home/CacheResponse">Response Caching Demo</a></li>

<li><a href="/Home/MemoryCacheData">Memory Cache Data Demo</a></li>

</ul>

**CacheResponse.cshtml :** (Views / Home / CatchResponse.cshtml)

@{

ViewData["Title"] = "Cache Response";

}

<h2>Response Caching Demo</h2>

<p>@ViewBag.Message</p>

**MemoryCacheData.cshtml :** (Views / Home / MemoryCacheData.cshtml)

@model List<PRACTICAL10.Models.ProductModel>

@{

ViewData["Title"] = "Memory Cache Data";

}

<h2>Memory Cache Data Demo</h2>

<p>@ViewBag.Message</p>

*<!-- Adding inline CSS to style the table -->*

<style>

table {

width: 50%;

border-collapse: collapse;

margin-top: 20px;

}

th, td {

border: 1px solid #ddd;

padding: 8px;

text-align: left;

}

th {

background-color: #f2f2f2;

}

tr:nth-child(even) {

background-color: #f9f9f9;

}

</style>

<table>

<thead>

<tr>

<th>ID</th>

<th>Name</th>

<th>Price</th>

</tr>

</thead>

<tbody>

@foreach (var product in Model)

{

<tr>

<td>@product.Id</td>

<td>@product.Name</td>

<td>@product.Price</td>

</tr>

}

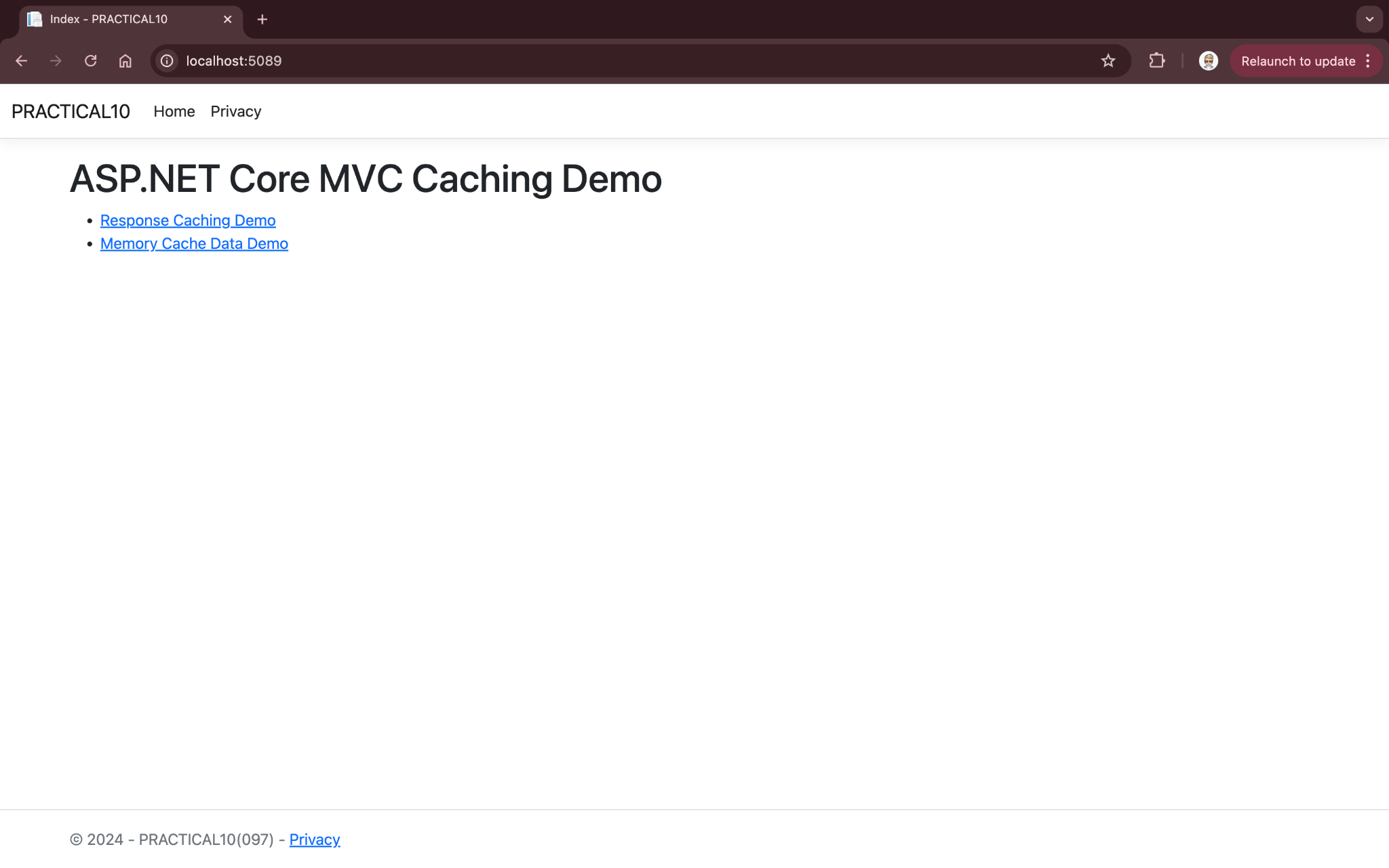
</tbody>

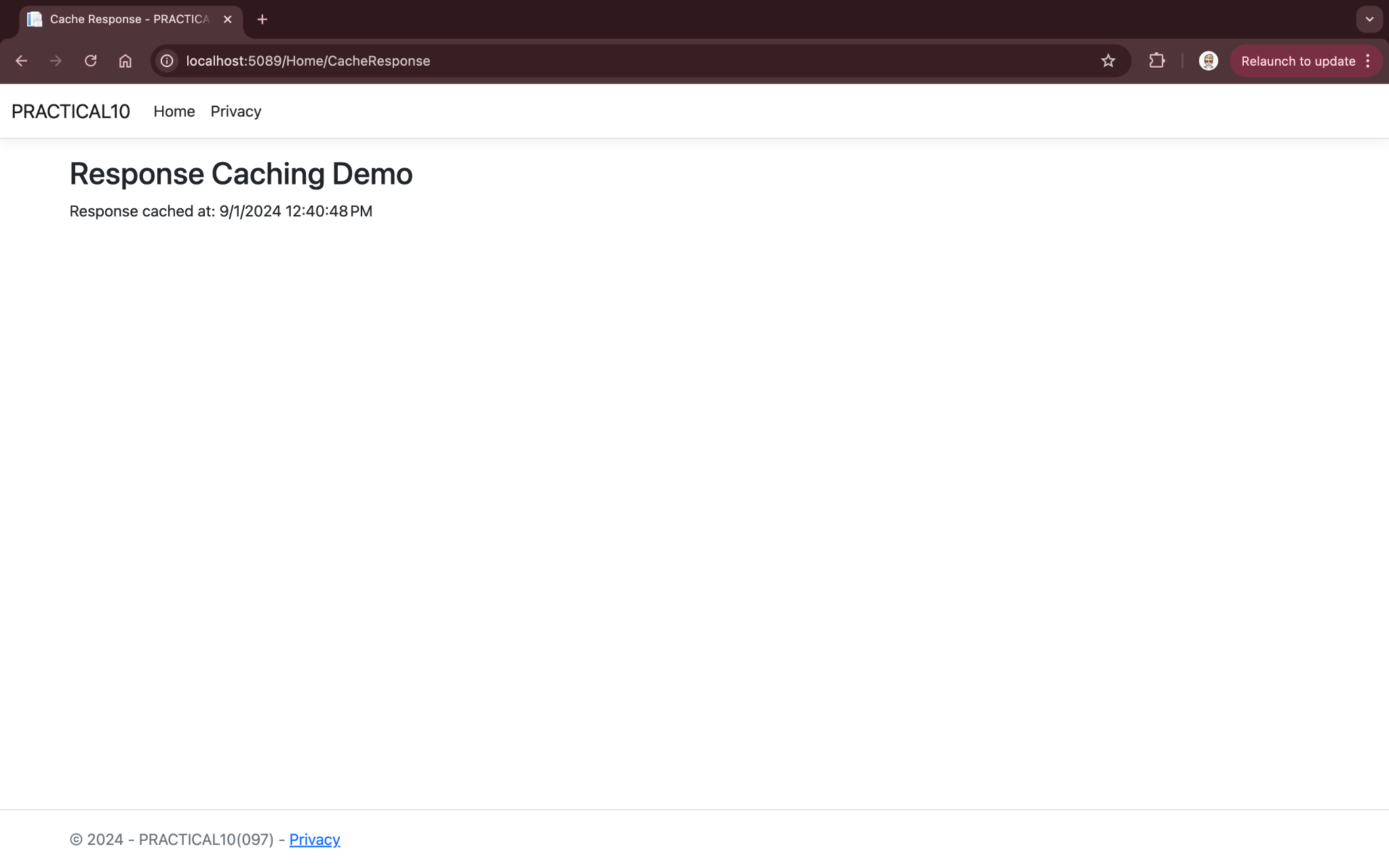
</table>

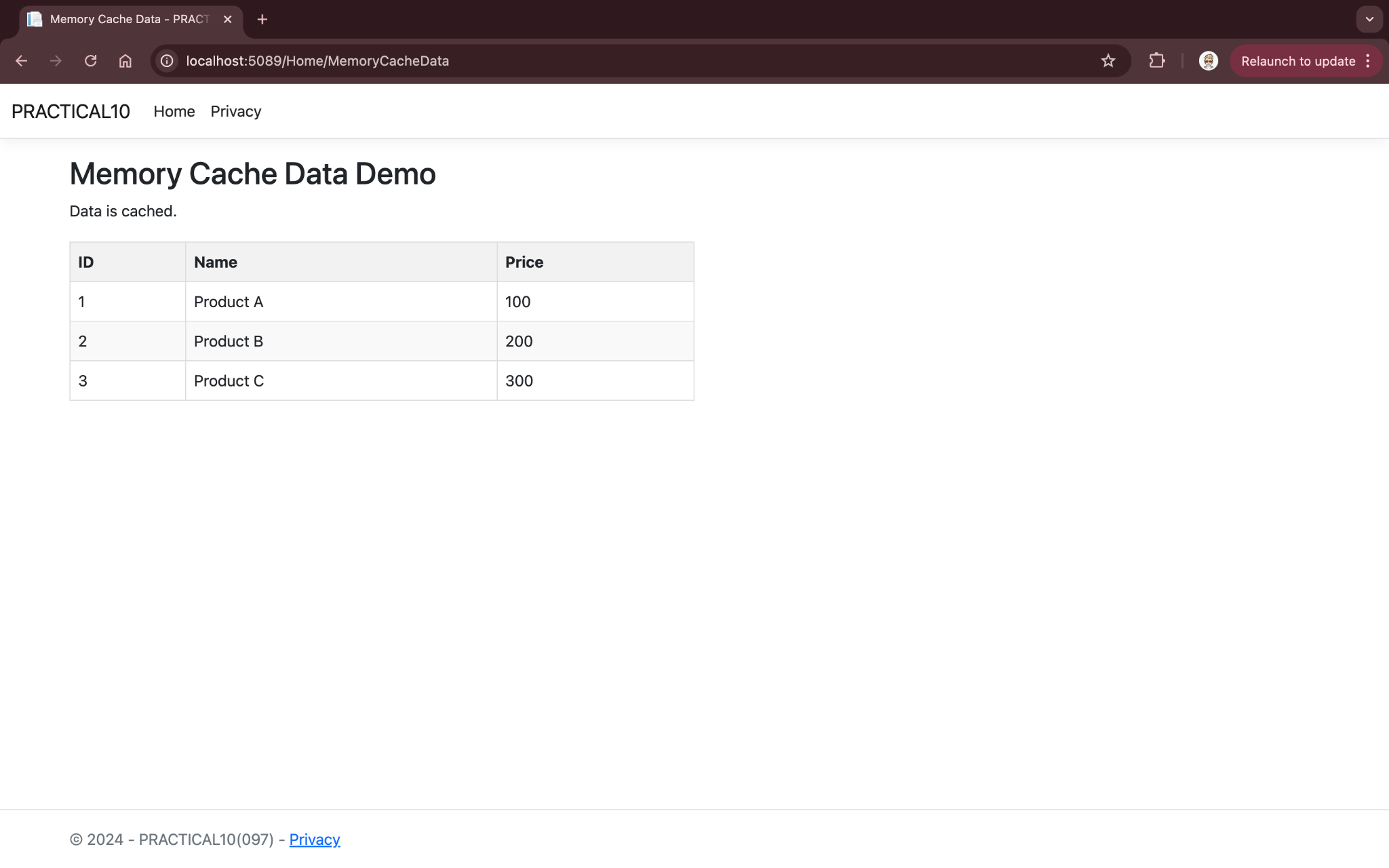
**Add This In Program.cs :**

builder.Services.AddMemoryCache();

**Output:**

****

****

****