Server-Side State Management in a Full-Stack Application

Objective: by the end of this activity, you will be able to implement session handling and caching techniques in a Blazor Server application to maintain server-side state efficiently.

Step 1: Prepare for the Application

You'll create a small Blazor Server app in Visual Studio Code. The application will demonstrate server-side state management using sessions and caching.

Instructions:

1. Open Visual Studio Code and ensure the terminal is ready.
2. Run the following command to create a new Blazor Server app: dotnet new blazor -o BlazorServerApp
3. Navigate to the new project directory: cd BlazorServerApp
4. Open the project in Visual Studio Code: code .
5. Delete any placeholder content in Program.cs.

Step 2: Configure Session Handling

Add middleware to the application to enable session handling.

Instructions:

1. Open the Program.cs file.
2. Add code to configure distributed memory cache and session handling.
3. Set the session timeout to 30 minutes and configure secure cookie options.
4. Ensure app.UseSession() is added to the application pipeline.

Step 3: Create a Caching Service

Implement a service for managing cached data.

Instructions:

1. In the root of the project, create a new folder called Services.
2. Add a file named CacheService.cs to the Services folder.

3. Define a caching service class using IMemoryCache to store and retrieve frequently accessed data.

Step 4: Register Services

Make the session handling and caching service available to the application.

Instructions:

1. Update Program.cs to include the caching service and memory cache in the dependency injection container.
2. Register CacheService and verify it is ready to be injected into components.

Step 5: Create a Component to Use Caching

Implement a Blazor component that uses the caching service to fetch and display data.

Instructions:

1. Create a Pages folder, and inside Pages, create a new file named FetchData.razor.
2. Inject the caching service into the component.
3. Add logic to fetch weather data and store it in the cache for 5 minutes.

Step 6: Enable Advanced Session Handling

Extend the session management to include persistent data handling in a user-friendly way.

Instructions:

1. Install the Blazored.SessionStorage NuGet package to simplify session storage: dotnet add package Blazored.SessionStorage
2. Update Program.cs to include the session storage service in the dependency injection container.
3. Create or update a Blazor component (e.g., Counter.razor) to store and retrieve data from session storage.

Step 7: Test Your Application

Run and test the application to verify that session and caching functionalities work correctly.

Instructions:

1. Use the terminal to run the application: dotnet run
2. Open the application in a browser and interact with the session and caching features to confirm they behave as expected.