Implementing Client-Side State Management in Blazor

Objective: by the end of this activity, you will be able to implement client-side state management techniques in a Blazor WebAssembly application. This includes using local storage, session storage, and clearing stored data programmatically.

Step 1: Prepare for the Application

You'll create a new Blazor WebAssembly application. Your application will include features that demonstrate client-side state management techniques.

Instructions:

1. Open Visual Studio Code and ensure you have .NET 6 or higher installed.
2. Open the terminal in VS Code.
3. Run the following command to create a new Blazor WebAssembly project: dotnet new blazorwasm -o ClientStateApp
4. Navigate to the project folder: cd ClientStateApp
5. Open the ClientStateApp folder in Visual Studio Code**.**

Step 2: Implementing Local Storage

You'll create a feature to save and retrieve the user's theme preference using local storage.

Instructions:

1. Install the Blazored.LocalStorage NuGet package: dotnet add package Blazored.LocalStorage
2. Open the Program.cs file and register the local storage service.
3. In the Pages folder, edit the Home.razor file to create a theme selector UI.
4. Add the necessary logic to save the user's theme preference in local storage and retrieve it on initialization.

Step 3: Using Session Storage

You’ll create a shopping cart page that uses session storage to manage session-specific data.

Instructions:

1. Install the Blazored.SessionStorage NuGet package: dotnet add package Blazored.SessionStorage
2. Register the session storage service in Program.cs.
3. Create a new Razor component named Cart.razor in the Pages folder.
4. Design a simple UI for the shopping cart with an input field to add items and a list to display the items.
5. Use session storage to save the cart’s contents and retrieve them on initialization.

Step 4: Clearing Storage

You’ll add functionality to clear all stored data from local and session storage.

Instructions:

1. Open Home.razor and add a "Clear All Storage" button.
2. Write the logic to clear both local and session storage when the button is clicked.
3. Test the functionality by interacting with the app and verifying that the data is cleared when you press the button.