Setting Up a Full-Stack Development Environment

Objective: By the end of this activity, you will be able to set up a full-stack development environment using Visual Studio Code, create front-end and back-end projects using Blazor and Minimal APIs, and configure these projects to work seamlessly together.

Step 1: Prepare for the Application

You'll create a small application using the Visual Studio Code console you configured earlier. Your application will consist of a front-end Blazor project and a back-end Minimal API project.

Instructions:

1. Open Visual Studio Code and navigate to a folder where you want to set up your projects.
2. Open the terminal in VS Code using Ctrl + ~.
3. Create a new front-end project by running: dotnet new blazorwasm -o frontend This creates a Blazor WebAssembly project in a folder named "frontend."
4. Create a new back-end project by running: dotnet new webapi -o backend This sets up a Minimal API project in a folder named "backend."
5. Open both folders in separate VS Code windows using: code frontend code backend

Step 2: Install and Verify Dependencies

Set up and verify the dependencies required for your projects.

Instructions:

1. Ensure you have the .NET SDK installed by running: dotnet --version If the version number doesn't display, revisit the .NET SDK installation process.
2. Install the required VS Code extensions:
    - C# Dev Kit: For enhanced C# support.
    - REST Client: For making HTTP requests directly in VS Code.
    - CSharpier: For automatic code formatting.
3. Verify that the Blazor and Minimal API projects were created successfully:
    - Navigate to each project folder in the terminal.

- Run dotnet build to confirm that the projects compile without errors.

Step 3: Configure the Front-End Project

Prepare the front-end Blazor project to display data retrieved from the back-end API.

Instructions:

1. Open the frontend project in VS Code.
2. Locate the Program.cs file and prepare it for later integration with the back-end API.
3. Start the front-end application using: dotnet watch This command serves the application and opens it in a browser. Verify that the default Blazor app loads correctly.

Step 4: Configure the Back-End Project

Set up the back-end project to handle API requests.

Instructions:

1. Open the backend project in VS Code.
2. Locate the Program.cs file.
3. Define a simple API endpoint in Program.cs to return sample data, such as a list of products.
4. Start the back-end application using: dotnet watch Verify that the application runs and note the URL where the API is served.

Step 5: Integrate the Front-End and Back-End Projects

Connect the Blazor front end to the Minimal API back end.

Instructions:

1. Update the front-end Blazor application to make an HTTP request to the back-end API.
2. Use the HttpClient object in Blazor to call the API endpoint and retrieve the sample data.
3. Display the retrieved data in a simple Blazor component.