Implementing Data Binding, Event Handling, and Navigation in a Blazor Application

Instructions: By the end of this activity, you will create a Recipe Manager App in Blazor WebAssembly, implementing data binding, event handling, and routing/navigation.

Step 1: Prepare for the Application

You'll create a Blazor WebAssembly application to manage recipes. Users will view a list of recipes, add new recipes, and see detailed instructions for each recipe.

Instructions:

1. Open Visual Studio Code.
2. Open the terminal (Ctrl+` or View > Terminal) and create a new Blazor WebAssembly project: dotnet new blazorwasm -o RecipeManagerApp
3. Navigate to the project directory: cd RecipeManagerApp
4. Open the project in Visual Studio Code: code .
5. Run the application: dotnet run

Create a folder called Data in the root directory of the project.

Step 2: Modify the Existing Hore Component

The Home page will display the list of recipes and allow users to view details or navigate to the "Add Recipe" page.

Instructions:

1. Create a Recipe class in the Data folder with properties:
    a. Id: Unique identifier.
    b. Name: Recipe name.
    c. Description: Short description.
2. Open Home.razor in the Pages folder.
3. Add a List<Recipe> property to hold recipes.
4. Bind the recipe list to the UI to display the name and description of each recipe.
5. Add links to:

a. View the details of a recipe.

b. Navigate to the Add Recipe page.

Step 3: Create the Add Recipe Page

The Add Recipe page allows users to add a new recipe.

Instructions:

1. Create a new Razor component in the Pages folder named AddRecipe.razor.
2. Add a route directive (@page "/addrecipe") at the top.
3. Include:
    a. Input fields for recipe name and description.
    b. A submit button to add the recipe.
4. Use the NavigationManager service to navigate back to the Home page after adding a recipe.

Step 4: Add Dynamic Routing for Recipe Details

Enable users to navigate to a dynamically routed Recipe Details page.

Instructions:

1. Create a new Razor component named RecipeDetails.razor in the Pages folder.
2. Add a route directive like @page "/recipe/{id:int}" to accept recipe IDs.
3. Fetch the recipe details based on the provided ID and display them on the page.
4. Update the Home component to include clickable links for navigating to recipe details.

Step 5: Add Navigation Links

Add navigation links to switch between pages.

Instructions:

1. Open NavMenu.razor in the Layout folder.
2. Add a link to the Add Recipe page (/addrecipe).

Recipe.cs

```
using System.ComponentModel.DataAnnotations;
```

```
namespace RecipeManagerApp.Data;

public class Recipe
{
    public int Id { get; set; }

    [Required, StringLength(100)]
    public string Name { get; set; } = "";

    [Required, StringLength(500)]
    public string Description { get; set; } = "";
}
```

RecipeService.cs

```
namespace RecipeManagerApp.Data;

public class RecipeService
{
    private readonly List<Recipe> _recipes = new()
    {
        new Recipe { Id = 1, Name = "Pasta Carbonara", Description =
"Creamy sauce with bacon and parmesan." },
        new Recipe { Id = 2, Name = "Greek Salad", Description =
"Tomatoes, cucumbers, olives, feta." },
        new Recipe { Id = 3, Name = "Banana Bread", Description =
"Moist, sweet loaf with ripe bananas." }
    };

    public IReadOnlyList<Recipe> GetAll() => _recipes;

    public Recipe? GetById(int id) => _recipes.FirstOrDefault(r => r.Id
== id);

    public void Add(Recipe recipe)
    {
        recipe.Id = (_recipes.LastOrDefault()?.Id ?? 0) + 1;
        _recipes.Add(recipe);
    }
}
```

MainLayout.razor:

```
@inherits LayoutComponentBase
<div class="page">
    <div class="sidebar">
        <NavMenu />
    </div>

    <main>
        <div class="top-row px-4">
            <a href="https://learn.microsoft.com/aspnet/core/"
target="_blank">About</a>
        </div>
```

```
        <article class="content px-4">
            @Body
        </article>
    </main>
</div>
```

AddRecipe.razor

```
@page "/addrecipe"
@using RecipeManagerApp.Data
@inject RecipeService RecipeService
@inject NavigationManager Nav

<h3>Add Recipe</h3>

<EditForm Model="@model" OnValidSubmit="HandleSubmit">
    <DataAnnotationsValidator />
    <ValidationSummary />

    <div class="mb-3">
        <label class="form-label">Name</label>
        <InputText class="form-control" @bind-Value="model.Name" />
    </div>

    <div class="mb-3">
        <label class="form-label">Description</label>
        <InputTextArea class="form-control" @bind-
Value="model.Description" rows="4" />
    </div>

    <button type="submit" class="btn btn-primary">Add</button>
    <button type="button" class="btn btn-secondary ms-2"
@onclick="Cancel">Cancel</button>
</EditForm>

@code {
    private Recipe model = new();

    private void HandleSubmit()
    {
        RecipeService.Add(model);
        Nav.NavigateTo("/");
    }

    private void Cancel()
    {
        Nav.NavigateTo("/");
    }
}
```

RecipeDetails.razor:

```
@page "/recipe/{id:int}"
@using RecipeManagerApp.Data
@inject RecipeService RecipeService
@inject NavigationManager Nav
```

```
<h3>Recipe Details</h3>

@if (recipe is null)
{
    <div class="alert alert-warning">Recipe not found.</div>
    <button class="btn btn-secondary" @onclick="GoBack">Back</button>
}
else
{
    <div class="card">
        <div class="card-body">
            <h4 class="card-title">@recipe.Name</h4>
            <p class="card-text">@recipe.Description</p>
        </div>
    </div>

    <div class="mt-3">
        <button class="btn btn-secondary"
@onclick="GoBack">Back</button>
    </div>
}

@code {
    [Parameter] public int id { get; set; }
    private Recipe? recipe;

    protected override void OnParametersSet()
    {
        recipe = RecipeService.GetById(id);
    }

    private void GoBack() => Nav.NavigateTo("/");
}
```

Home.razor:

```
@page "/"
@using RecipeManagerApp.Data
@inject RecipeService RecipeService

<h1 class="mb-3">Recipes</h1>

<div class="mb-3">
    <a class="btn btn-primary" href="/addrecipe">Add Recipe</a>
</div>

@if (recipes.Count == 0)
{
    <p>No recipes yet. Click <strong>Add Recipe</strong> to create
one.</p>
}
else
{
    <ul class="list-group">
        @foreach (var r in recipes)
        {
```

```
            <li class="list-group-item d-flex justify-content-between
align-items-start">
                <div class="me-auto">
                    <div class="fw-bold">
                        <a href="@($"/recipe/{r.Id}")">@r.Name</a>
                    </div>
                    <small>@r.Description</small>
                </div>
                <a class="btn btn-sm btn-outline-secondary"
href="@($"/recipe/{r.Id}")">View details</a>
            </li>
        }
    </ul>
}

@code {
    private List<Recipe> recipes = new();

    protected override void OnInitialized()
    {
        recipes = RecipeService.GetAll().ToList();
    }
}
```