University of Sialkot
Department of software engineering

Assignment no 2

Session BS(SE) fall 2019-2023

Submitted to:

Sir Museb

Submitted by:

Name	**Hassan Ahmed**
Roll #	**19101001-103**
Batch	**2019(fall)**
Course	**Web Engineering**

Class: **BS(SE)yellow-2019**

Date **08 Sep, 2024.**

Department of software engineering

University of Sialkot

code:

```
 using System;
 using System.ComponentModel.DataAnnotations;
 namespace BlazorFormApp.Models
 {
    public class ProductReview
    {
        [Required(ErrorMessage = "Product Name is required.")]
        public string ProductName { get; set; }
        [Required(ErrorMessage = "Rating is required.")]
        [Range(1, 5, ErrorMessage = "Rating must be between 1 and 5.")]
        public int Rating { get; set; }
        [MinLength(20, ErrorMessage = "Review Description must be at least 20 characters long.")]
        public string ReviewDescription { get; set; }
        [Required(ErrorMessage = "Purchase Date is required.")]
        [DataType(DataType.Date)]
        [FutureDate(ErrorMessage = "Purchase Date cannot be in the future.")]
        public DateTime PurchaseDate { get; set; }
        [Required(ErrorMessage = "Please indicate if you would recommend this product.")]
        public bool WouldRecommend { get; set; }
        [Required(ErrorMessage = "You must agree to the terms.")]
        public bool AgreeToTerms { get; set; }
    }
 }
```

Create a custom validation attribute for future dates. Add a new file Validation/FutureDateAttribute.cs:

code:

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

```
namespace BlazorFormApp.Validation
{
  public class FutureDateAttribute : ValidationAttribute
  {
    public override bool IsValid(object value)
    {
      if (value is DateTime date)
      {
        return date.Date <= DateTime.Now.Date;
      }
      return false;
    }
  }
}
```

Create the Form Component

Add a new Razor component named ProductReviewForm.razor under the Pages folder:

code:

```
@page "/product-review"
@using BlazorFormApp.Models
@using System.ComponentModel.DataAnnotations
@using Microsoft.AspNetCore.Components.Forms
<h3>Product Review Form</h3>
<EditForm Model="@productReview" OnValidSubmit="@HandleValidSubmit">
  <DataAnnotationsValidator />
  <ValidationSummary />
  <div>
    <label>Product Name:</label>
    <InputText @bind-Value="productReview.ProductName" />
    <ValidationMessage For="@(() => productReview.ProductName)" />
```

```
</div>
<div>
    <label>Rating:</label>
    <select @bind="productReview.Rating">
        <option value="">Select Rating</option>
        @for (int i = 1; i <= 5; i++)
        {
        }
            <option value="@i">@i</option>
    </select>
    <ValidationMessage For="@(() => productReview.Rating)" />
</div>
<div>
    <label>Review Description:</label>
    <InputTextArea @bind-Value="productReview.ReviewDescription" />
    <ValidationMessage For="@(() => productReview.ReviewDescription)" />
</div>
<div>
    <label>Purchase Date:</label>
    <InputDate @bind-Value="productReview.PurchaseDate" />
    <ValidationMessage For="@(() => productReview.PurchaseDate)" />
</div>
<div>
    <label>Would You Recommend?</label>
    <InputRadio @bind-Value="productReview.WouldRecommend" Value="true" /> Yes
    <InputRadio @bind-Value="productReview.WouldRecommend" Value="false" /> No
    <ValidationMessage For="@(() => productReview.WouldRecommend)" />
</div>
<div>
```

```
    <InputCheckbox @bind-Value="productReview.AgreeToTerms" />
    <label>I agree to the terms and conditions</label>
    <ValidationMessage For="@(() => productReview.AgreeToTerms)" />
  </div>
  <button type="submit">Submit</button>
</EditForm>
@if (isSubmitted)
{
  <h4>Thank you for your review!</h4>
  <p><strong>Product Name:</strong> @productReview.ProductName</p>
  <p><strong>Rating:</strong> @productReview.Rating</p>
  <p><strong>Review Description:</strong> @productReview.ReviewDescription</p>
  <p><strong>Purchase Date:</strong>
@productReview.PurchaseDate.ToShortDateString()</p>
  <p><strong>Would Recommend:</strong> @(productReview.WouldRecommend ? "Yes" :
"No")</p>
}
@code {
  private ProductReview productReview = new ProductReview();
  private bool isSubmitted = false;
  private void HandleValidSubmit()
  {
    isSubmitted = true;
  }
}

<CascadingAuthenticationState>
  <Router AppAssembly="@typeof(Program).Assembly">
    <Found Context="routeData">
```

```
            <RouteView RouteData="@routeData"
DefaultLayout="@typeof(MainLayout)" />
            <FocusOnNavigate RouteData="@routeData" Selector="h1" />
        </Found>
        <NotFound>
            <LayoutView Layout="@typeof(MainLayout)">
                <p>Sorry, there's nothing at this address.</p>
            </LayoutView>
        </NotFound>
    </Router>
 </CascadingAuthenticationState>
```