CST8259 Web Programming Language II

Lab 3

Objective

- Use .NET utility XSD.EXE and .NET XML serialization classes to process XML files.

Due Date

See Brightspace posting for the due date of this lab. To earn 5 points, you are required:

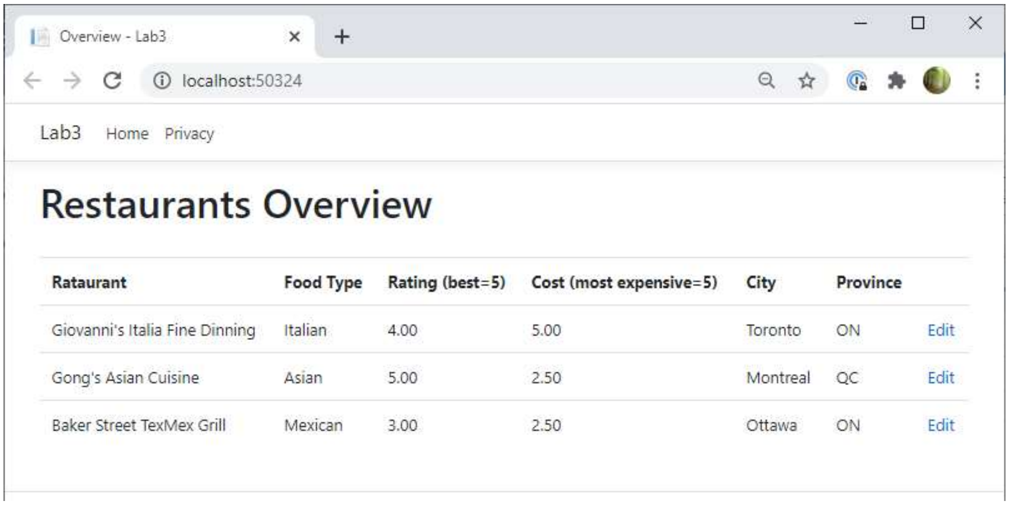
1. Complete the lab as required.
2. Submit your lab work to the Brightspace before the due date.
3. Demo your lab work during the lab session after the due date.

Requirements

- Use XSD.EXE to generate classes from the XML schema you created in Lab 2, **restaurant_review.xsd** file. Copy the generated C# file **restaurant_review.cs** into a folder, say, **Data**. of the ASP.NET MVC Core web application project created below.

- Create an ASP.NET MVC Core web application which has the following functionality

    1) On start the page displays the overviews of all restaurants in the XML file **restaurant_review.xml**:



Restaurants Overview

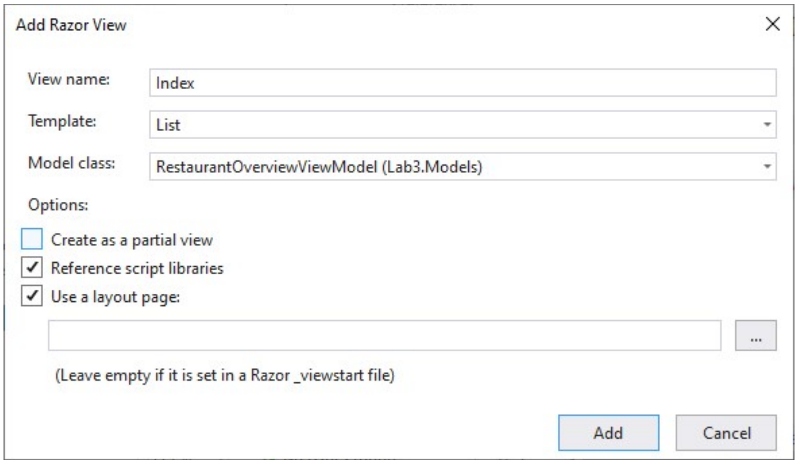
Rataurant	Food Type	Rating (best=5)	Cost (most expensive=5)	City	Province	
Giovanni's Italia Fine Dinning	Italian	4.00	5.00	Toronto	ON	Edit
Gong's Asian Cuisine	Asian	5.00	2.50	Montreal	QC	Edit
Baker Street TexMex Grill	Mexican	3.00	2.50	Ottawa	ON	Edit

Hints:

a. Create a view model class for this view:

```
 public class RestaurantOverviewViewModel
 {
        public int Id { get; set; }

        [Display(Name = "Rataurant")]
        public string Name { get; set; }

        [Display(Name = "Food Type")]
        public string FoodType { get; set; }

        [Display(Name = "Rating (best=5)")]
        public decimal Rating { get; set; }

        [Display(Name = "Cost (most expensive=5)")]
        public decimal Cost { get; set; }

        [Display(Name = "City")]
        public string City { get; set; }

        [Display(Name = "Province")]
        public string ProvinceState { get; set; }
}
```

b. In Visual Studio, generate a view binding to list of above view model objects:



Add Razor View

View name: Index

Template: List

Model class: RestaurantOverviewViewModel (Lab3.Models)

Options:

□ Create as a partial view

☑ Reference script libraries

☑ Use a layout page:

(Leave empty if it is set in a Razor _viewstart file)

c. Create a list of RestaurantOverviewViewModel with the data retrieved from **restaurant_review.xml**. You can use the sequence of a restaurant in the XML file as the value of **Id** property or you can use the id attribute if the restaurants in your XML have this attribute.

You can use **Path.GetFullPath(…)** method to get a full path of your **restaurant_review.xml** file from its relative path to the root of the website, for example:

```
string xmlFilePath = Path.GetFullPath("Data/restaurant_review.xml");
```

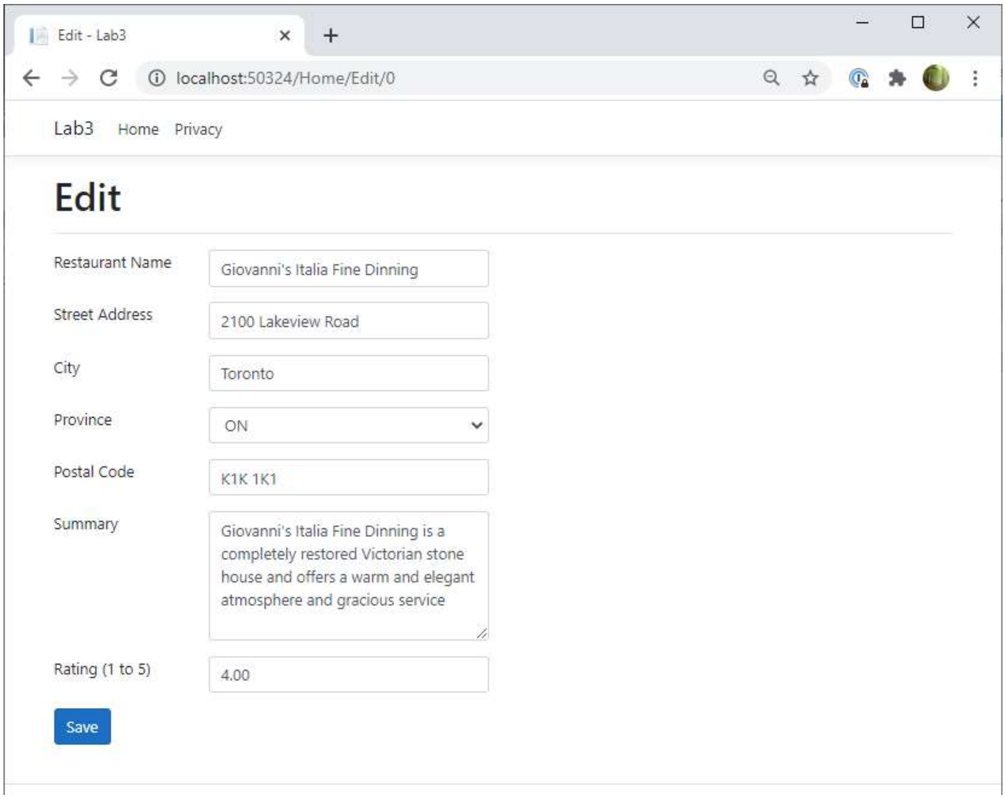
when **restaurant_review.xml** is inside the **Data** folder under application root.

d. Pass the list as the model object when showing the view

e. Edit link can have the restaurant’s Id as the value of attribute **asp-route-id**

```
<a asp-action="Edit" asp-route-id="@item.Id">Edit</a>
```

2) For each restaurant, there is an “**Edit**” link which leads the user to a view to edit the data of the restaurant (for simplicity, we only edit a few data items):



Edit

Restaurant Name	Giovanni's Italia Fine Dinning
Street Address	2100 Lakeview Road
City	Toronto
Province	ON
Postal Code	K1K 1K1
Summary	Giovanni's Italia Fine Dinning is a completely restored Victorian stone house and offers a warm and elegant atmosphere and gracious service
Rating (1 to 5)	4.00

Hints:

a) Create an action method take the restaurant id as its parameter:

```
public IActionResult Edit(int? id)
{ … …}
```

Use **id** as the index to find the restaurant in the object deserialized from your **restaurant_review.xml** file. Set the property values of the view model object defined next with the data of the found restaurant.

b) Create a view model for this view:

```
public class RestaurantEditViewModel
{
    public int Id { get; set; }

    [Display(Name="Restaurant Name")]
    public string Name { get; set; }

    [Required]
    [Display(Name = "Street Address")]
    public string StreetAddress { get; set; }

    [Required]
    [Display(Name = "City")]
    public string City { get; set; }
    [Display(Name = "Province")]
    public StateProvinceType ProvinceState { get; set; }

    [Required]
    [RegularExpression(@"^[a-zA-Z]\d[a-zA-Z](\s)*\d[a-zA-Z]\d$",
                    ErrorMessage = "Must be in the form of A1A 1A1")]
    [Display(Name = "Postal Code")]
    public string PostalZipCode { get; set; }

    [Required]
    [Display (Name="Summary")]
    public string Summary { get; set; }

    [Required]
    [Range(1, 5)]
    [Display(Name = "Rating (1 to 5)")]
    public decimal Rating { get; set; }
}
```

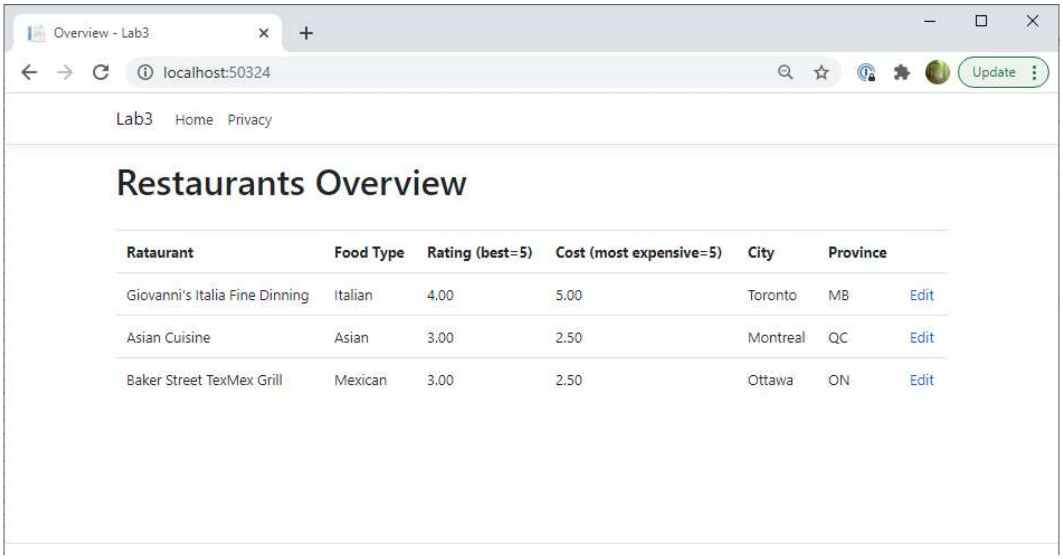
StateProvinceType above is an **enum** generated by XSD.EXE from my restaurant_review.xsd schema. Replace it with the **enum** for provinces generated from your **restaurant_review.xsd** schema.

c) Generate a view binding to the view model. Change the input field for the Id to hidden, the input field for Summary to text area and the input field for Province to dropdown.

d) For simplicity, the Province dropdown in the view only shows the Simples of the provinces, not the full names of the provinces. You can populate the dropdown from an **enum** and bind the selected provinces as:

```
<select asp-for="ProvinceState"
        asp-items="Html.GetEnumSelectList<StateProvinceType>()"
        class="form-control">
</select>
```

3) After click save button, the application saves the changes made to the restaurant and back to the overview page reflecting the modified data of the restaurant:



Restaurants Overview

Rataurant	Food Type	Rating (best=5)	Cost (most expensive=5)	City	Province	
Giovanni's Italia Fine Dinning	Italian	4.00	5.00	Toronto	MB	Edit
Asian Cuisine	Asian	3.00	2.50	Montreal	QC	Edit
Baker Street TexMex Grill	Mexican	3.00	2.50	Ottawa	ON	Edit

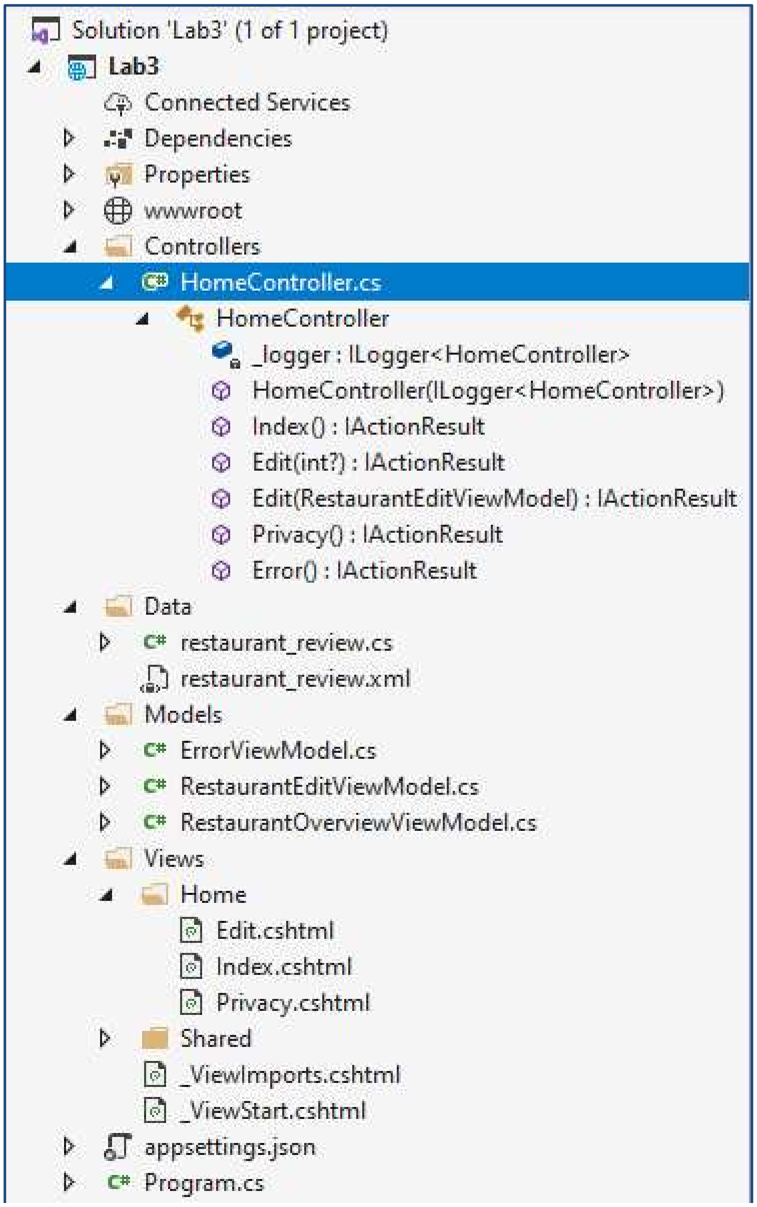
Hints

a) Create a [HttpPost] annotated action method which takes a `RestaurantEditViewModel` object as its parameter:

```
[HttpPost]
public IActionResult Edit(RestaurantEditViewModel rsVM)
{… …}
```

b) Use Id property of the view model object as the index to find the restaurant being edited.

c) After completion, my Visual Studio solution have the following structure, yours may be slightly different:



```
Solution 'Lab3' (1 of 1 project)
  Lab3
    Connected Services
    Dependencies
    Properties
    wwwroot
    Controllers
      HomeController.cs
        HomeController
          _logger : ILogger<HomeController>
          HomeController(ILogger<HomeController>)
          Index() : IActionResult
          Edit(int?) : IActionResult
          Edit(RestaurantEditViewModel) : IActionResult
          Privacy() : IActionResult
          Error() : IActionResult
    Data
      restaurant_review.cs
      restaurant_review.xml
    Models
      ErrorViewModel.cs
      RestaurantEditViewModel.cs
      RestaurantOverviewViewModel.cs
    Views
      Home
        Edit.cshtml
        Index.cshtml
        Privacy.cshtml
      Shared
      _ViewImports.cshtml
      _ViewStart.cshtml
    appsettings.json
    Program.cs
```