Lab: Applying Styles to MVC 5 Web Applications

# Scenario

You have created a good amount of the photo-handling functionality for the Photo Sharing web application. However, stakeholders are concerned about the basic black-and-white appearance of the application. In addition, titles and menus do not appear on every page.

To resolve these issues, your manager has asked you to implement the following user interface features:

* A layout for all webpages. The layout should include common elements, such as the main menu, which should appear on every page of the application.
* Icons on every button in order to improve the user experience.
* The web application should be accessible from mobile devices such as mobile phones and tablets. In particular, you need to ensure that devices with narrow screens can access photos easily.

# Objectives

After completing this lab, you will be able to:

* Use layouts to ensure common interface features, such as the menu, are consistent across the entire web application.
* Apply a consistent look and feel to the web application.
* Ensure that the web application renders smoothly on screens of different sizes and aspect ratios.

**Estimated Time**: 40 minutes

# Exercise 1: Creating and Applying Layouts

## Scenario

In this exercise, you will:

* Browse through the Photo Sharing web application without a layout applied.
* Create a new layout and link the application to the view by using a \_ViewStart.cshtml file.
* Modify the home index and photo display views to use the new layout.
* Browse through the resulting web application.

The main tasks for this exercise are as follows:

1. Open and browse through the Photo Sharing application.
2. Update the views to use the layout.
3. Browse through the web application.

### Task 1: Modify the layout menu items.

1. Open the layout view of the PhotoSharingApplication project by using the following information:

* File location: /Views/Shared
* Name: \_Layout.cshtml

1. Replace the menu items in the navigation toolbar with links to
   1. View All Photos (Index action of the Photos Controller)
   2. View Latest Photos (Index action of the Home Controller)
   3. Insert a New Photo (Create Action of the Photos Controller)
2. Save the layout.

### Task 2: Modify the Views to use the Layout

1. Open the Index View of the Photos Controller by using the following information:

* File location: /Views/Photos
* Name: Index.cshtml

1. Remove the C# block containing the Layout = null statement
2. Remove all the HTML from the beginning to the <body> tag (included)
3. Remove the last two closing tags </body> and </html>
4. Repeat steps 1 to 5 for
   1. /Views/Photos/Create.cshtml
   2. /Views/Photos/Delete.cshtml
   3. /Views/Photos/Details.cshml.

### Task 3: Browse through the web application.

1. Start the web application in debugging mode and verify the menu on the home page.
2. Stop debugging.

**Results**: After completing this exercise, you will be able to create an ASP.NET MVC Core web application that uses a single layout to display every page of the application.

# Exercise 2: Applying Styles to an MVC Web Application

## Scenario

In this exercise, you will

* Update the project to use bootstrap 4
* Use bootstrap to and unify the style of every view of the application
* Examine the changes to the user interface after the styles have been applied.

The main tasks for this exercise are as follows:

1. Update the project to use bootstrap 4
2. Update the element classes to use the styles.
3. Insert the icons on the buttons.
4. Browse the styled web application.

### Task 1: Update the project to use bootstrap 4

https://stackoverflow.com/questions/48481003/how-to-use-bootstrap-4-in-asp-net-core

1. Download bootstrap, jquery and popperjs
   1. On the Solution Explorer, right click on your project, select Add -> New Item
   2. Select “npm Configuration File”
   3. In package.json file
      1. Add a “dependencies” property
      2. Add “bootstrap”, “jquery” and “popper.js”. Select the versions presented by Visual Studio.  
         Your package.json may look like this (versions may vary)  
           
         {

"version": "1.0.0",

"name": "asp.net",

"private": true,

"devDependencies": {

},

"dependencies": {

"bootstrap": "~4.1.1",

"jquery": "~3.3.1",

"popper.js": "~1.14.3"

}

}

1. Remove old versions of bootstrap and jquery
   1. In Visual Studio, remove the content of the following existing folders:
      1. **PhotoSharingApplication/wwwroot/lib/bootstrap**
      2. **PhotoSharingApplication/wwwroot/lib/jquery**
2. Copy the new versions of bootstrap and jQuery into your project (usually done by tools like gulp / grunt / webpack and so on)
   1. In the Solution Explorer, click on “Show All Files”
   2. Copy the **node\_modules/bootstrap/dist** folder under **wwwroot/lib/bootstrap**
   3. Copy the **node\_modules/jquery/dist** folder under **wwwroot/lib/jquery**
   4. Create a **popper.js** folder under **wwwroot/lib**
   5. Copy the **node\_modules/popper.js/dist** folder under **wwwroot/lib/popper.js**
   6. Include the copied folders into your project
3. Update the \_Layout to use the new files
   1. At the top of the \_Layout view, in the <head> tag, locate the <environment exclude=”Development”> tag
   2. Replace the href value with “https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css”
   3. At the bottom of the \_Layout view, locate the <environment include=”Development”> tag and add the following script before bootstrap.js

<script src="~/lib/popper.js/dist/popper.js"></script>

* 1. Locate the <environment Exclude=”Development”> tag and replace its content with

<**environment** **exclude**="Development">

<**script** **src**="https://code.jquery.com/jquery-3.2.1.slim.min.js"

integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"

crossorigin="anonymous"

**asp-fallback-src**="~/lib/jquery/dist/jquery.min.js"

**asp-fallback-test**="window.jQuery">

</**script**>

<**script** **src**="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"

**asp-fallback-src**="~/lib/popper.js/dist/popper.min.js"

integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"

crossorigin="anonymous">

</**script**>

<**script** **src**="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"

integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"

crossorigin="anonymous"

**asp-fallback-src**="~/lib/bootstrap/dist/js/bootstrap.min.js"

**asp-fallback-test**="window.jQuery && window.jQuery.fn && window.jQuery.fn.modal">

</**script**>

<**script** **src**="~/js/site.min.js" **asp-append-version**="true"></**script**>

</**environment**>

**NOTE: Your versions may vary. Find src and integrity values on** [**https://getbootstrap.com/docs/4.0/getting-started/download/#bootstrapcdn**](https://getbootstrap.com/docs/4.0/getting-started/download/#bootstrapcdn)

### Task 2: Update the element of the views to use the same style.

1. Update the Layout navigation
   1. In the Views/Shared/\_Layout.cshtml file and replace the <nav> tag with the following code

<nav class="navbar fixed-top navbar-expand-lg navbar-dark bg-dark">

<**a** **asp-area**="" **asp-controller**="Home" **asp-action**="Index" class="navbar-brand">Photo Sharing Application</**a**>

<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNavDropdown" aria-controls="navbarNavDropdown" aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbarNavDropdown">

<ul class="navbar-nav">

<li class="nav-item"><**a** **asp-area**="" **asp-controller**="Home" **asp-action**="Index" class="nav-link">Latest Photos</**a**></li>

<li class="nav-item"><**a** **asp-area**="" **asp-controller**="Photos" **asp-action**="Index" class="nav-link">All Photos</**a**></li>

<li class="nav-item"><**a** **asp-area**="" **asp-controller**="Photos" **asp-action**="Create" class="nav-link">Create Photo</**a**></li>

</ul>

</div>

</nav>

1. Update the PhotoGallery ViewComponent so that each photo becomes a card. Ensure that a maximum of 3 photos per row are displayed. In the end your code could look like this:

@model IEnumerable<Photo>

<div class="row">

@foreach (var item in Model) {

<div class="col-sm-4">

<div class="card">

<img class="card-img-top" src="@Url.Action("GetImage", "Photos", new { id = item.Id })" alt="@item.Title">

<div class="card-body">

<h5 class="card-title">@item.Title</h5>

<p class="card-text">

Created By @Html.DisplayFor(model => item.UserName) on @Html.DisplayFor(model => item.CreatedDate)

</p>

<**a** **asp-action**="Details" **asp-controller**="Photos" **asp-route-id**="@item.Id" class="btn btn-primary">Display</**a**>

<**a** **asp-action**="Delete" **asp-controller**="Photos" **asp-route-id**="@item.Id" class="btn btn-info">Delete</**a**>

</div>

</div>

</div>

}

</div>

1. Modify the Details ad Delete view by using a card as well. If you created a partial view, in the end it might look something like this:

@model Photo

<div class="row">

<div class="col-sm-12">

<div class="card">

<img class="card-img-top" src="@Url.Action("GetImage", "Photos", new { id = Model.Id })" alt="@Model.Title">

<div class="card-body">

<h5 class="card-title">@Model.Title</h5>

<p class="card-text">

@Html.DisplayFor(model => model.Description)

</p>

</div>

<div class="card-footer">

@Html.DisplayFor(model => model.ImageMimeType) format, created By @Html.DisplayFor(model => model.UserName) on @Html.DisplayFor(model => model.CreatedDate)

</div>

</div>

</div>

</div>

1. Modify the Create view by removing the classes for the labels and updating the class for the fileupload to form-control-file. In the end your code might look like this:

<div class="row">

<div class="col-md-6">

<**form** **asp-action**="Create" enctype="multipart/form-data">

<**div** **asp-validation-summary**="ModelOnly" class="text-danger"></**div**>

<div class="form-group">

<**label** **asp-for**="Title"></**label**>

<**input** **asp-for**="Title" class="form-control" />

<**span** **asp-validation-for**="Title" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="PhotoFile"></**label**>

<input type="file" name="Image" class="form-control-file" />

</div>

<div class="form-group">

<**label** **asp-for**="Description"></**label**>

<**textarea** **asp-for**="Description" class="form-control" ></**textarea**>

<**span** **asp-validation-for**="Description" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="CreatedDate"></**label**>

<**input** **asp-for**="CreatedDate" class="form-control" />

<**span** **asp-validation-for**="CreatedDate" class="text-danger"></**span**>

</div>

<div class="form-group">

<**label** **asp-for**="UserName"></**label**>

<**input** **asp-for**="UserName" class="form-control" />

<**span** **asp-validation-for**="UserName" class="text-danger"></**span**>

</div>

<div class="form-group">

<input type="submit" value="Create" class="btn btn-info" />

</div>

</**form**>

</div>

</div>

### Task 3: Browse the styled web application.

1. Start the web application in debugging mode to examine the home page with the new style applied.
2. Browse to All Photos to examine the page with the new style applied.
3. Display a photo of your choice to examine the new style applied.
4. Create a photo to examine the new style applied

### Task 4: Browse the styled web application and verify that the pages resize correctly on a mobile device.

1. Start the web application in debugging mode to examine the home page with the new style applied.
2. Browse to All Photos to examine the page with the new style applied.
3. Resize the browser and examine how the layout changes.
4. Display a photo of your choice to examine the new style applied.
5. Resize the browser and examine how the layout changes.
6. Set the browser profile to a mobile phone and examine how the layout changes
7. Stop debugging.

**Results**: After completing this exercise, you will be able to create a Photo Sharing application with a consistent look and feel.