

Layout And CSS

Goals

In this lab, we're going to design a better user experience by adding menu items and styling the pages to make them more pleasing to the eye.

We're going to: - Add a new menu item to the navigation bar - Use Bootstrap to style our pages - Use a custom CSS file to style our pages - Add client side validation and style our form

Add a new menu item to the navigation bar

The Razor Pages we added to our project do not specify their own Layout, which means they get rendered together with the one specified in the `_ViewStart.cshtml` file, which is `_Layout.cshtml` located under the `Pages\Shared` Folder.

This file contains the navigation bar and the footer.

Therefore, this is where we will add our new menu items, one to view all the photos and one to upload a new photo.

The navigation bar becomes:

```
<ul class="navbar-nav flex-grow-1">
  <li class="nav-item">
    <a class="nav-link text-dark" asp-area="" asp-page="/Index">Home</a>
  </li>
  <li class="nav-item">
    <a class="nav-link text-dark" asp-area="" asp-page="/Photos/Index">All Photos</a>
  </li>
  <li class="nav-item">
    <a class="nav-link text-dark" asp-area="" asp-page="/Photos/Upload">Upload New Photo</a>
  </li>
  <li class="nav-item">
    <a class="nav-link text-dark" asp-area="" asp-page="/Privacy">Privacy</a>
  </li>
</ul>
```

Use Bootstrap to style our pages

Bootstrap is a CSS framework that we can use to style our pages. It is actually already linked to our Razor Pages project, so we don't need to do anything, just use it. The Bootstrap CSS file is located under `wwwroot\lib\bootstrap\dist\css\bootstrap.css` and it's linked in the `<head>` section of the Layout page.

We can use a [Bootstrap Card](#) to style each photo. We will use the `.card` class to style the div containing the photo and its metadata. We will use the Bootstrap grid system and its `.row` `.cols` classes to control how many grid columns (wrapped around our cards) to show per row. We will be laying out the cards on one column, splitting four cards to equal width across multiple rows, from the medium breakpoint up.

The `_PhotoDetailsPartial.cshtml` becomes:

```
@using PhotoSharingApplication.Core.Entities
@model Photo

<div class="card">
  
  <div class="card-body">
    <h5 class="card-title">@Model.Title</h5>
    <p class="card-text">@Model.Description</p>
  </div>
  @if (ViewData["ShowDetailsButton"] is bool showDetailsButton && showDetailsButton) {
    <div class="card-footer">
      <a asp-page="/Details" asp-route-id="@Model.Id" class="card-link">Details</a>
    </div>
  }
</div>
```

The `Index.cshtml` becomes:

```

<div class="row row-cols-1 row-cols-md-2 g-4">
@foreach (Photo item in Model.Photos)
{
    <div class="col">
        <partial name="_PhotoDetailsPartial" model="item" view-data="ViewData"></partial>
    </div>
}
</div>

```

The Details.cshtml becomes

```

@page "{id:int}"
@model PhotoSharingApplication.Web.Pages.Photos.DetailsModel
@{
    ViewData["ShowDetailsButton"] = false;
}
<div class="row row-cols-1">
    <div class="col">
        <partial name="_PhotoDetailsPartial" for="Photo" view-data="ViewData"></partial>
    </div>
</div>

```

At this point, the UI should already look better.

Use a custom CSS file to style our pages

Let's also add a little animation to our cards whenever the user hovers over them.

To do this, let's add a new CSS isolated file to our `_PhotoDetailsPartial` partial view.

- Under the `Pages\Photo` folder, add a new `_PhotoDetailsPartial.cshtml.css` Style Sheet.

- In this file, add a `.card-tilted` class - Add a `transition` rule so that every property takes 100ms to change to its new value - Add a `.card-tilted:hover` state

- Add a `transform` rule to rotate the card 1 degree and scale it to 1.1 times its original size - Add a `box-shadow` rule to add a shadow to the card - Add a `z-index` rule to make sure the card is on top of the other cards

The code becomes:

```

.card-tilted {
    transition: all 100ms linear;
}

.card-tilted:hover {
    transform: rotate(1deg) scale(1.1);
    box-shadow: 1rem 1rem 1rem rgba(128, 128, 128, 0.5);
    z-index: 99;
}

```

When you navigate to the `Photos` or `/Photos/Details/1` pages and hover on the cards, the card should start to tilt and the shadow should appear.

Add client side validation and style our form

The last page we need to enhance is the `Upload`. Let's use the bootstrap classes to style a form.

```

<form method="post" enctype="multipart/form-data">
  <div asp-validation-summary="ModelOnly" class="text-danger"></div>
  <div class="mb-3">
    <label asp-for="Photo.Title" class="form-label"></label>
    <input asp-for="Photo.Title" class="form-control" />
    <span asp-validation-for="Photo.Title" class="text-danger"></span>
  </div>
  <div class="mb-3">
    <label asp-for="Photo.Description" class="form-label"></label>
    <textarea asp-for="Photo.Description" class="form-control"></textarea>
    <span asp-validation-for="Photo.Description" class="text-danger"></span>
  </div>
  <div class="mb-3">
    <label asp-for="FormFile" class="form-label"></label>
    <input asp-for="FormFile" type="file" class="form-control">
    <span asp-validation-for="FormFile" class="text-danger"></span>
  </div>
  <div class="mb-3">
    <input type="submit" class="btn btn-primary" />
  </div>
</form>

```

To further enhance the user experience, let's also include client side validation, by adding the partial view with the client side scripts necessary to validate the form client side.

```

@section Scripts {
  <partial name="_ValidationScriptsPartial" />
}

```

Now the form should look better and it should stop the user as soon as the validation client side kicks in.

Lessons Learned

- Layout
 - RenderBody
 - RenderSection
 - Client Side Validation
- Menu
- CSS
 - CSS Isolation

References

- <https://docs.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-6.0&tabs=visual-studio#using-layouts-partials-templates-and-tag-helpers-with-razor-pages>
- <https://getbootstrap.com/docs/5.0/components/card/>
- <https://docs.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-6.0&tabs=visual-studio#css-isolation>