Module08

Applying Styles to ASP.NET MVC Web Applications

Module Overview

- Using Layouts
- Applying CSS Styles to an MVC Application
- Creating an Adaptive User Interface

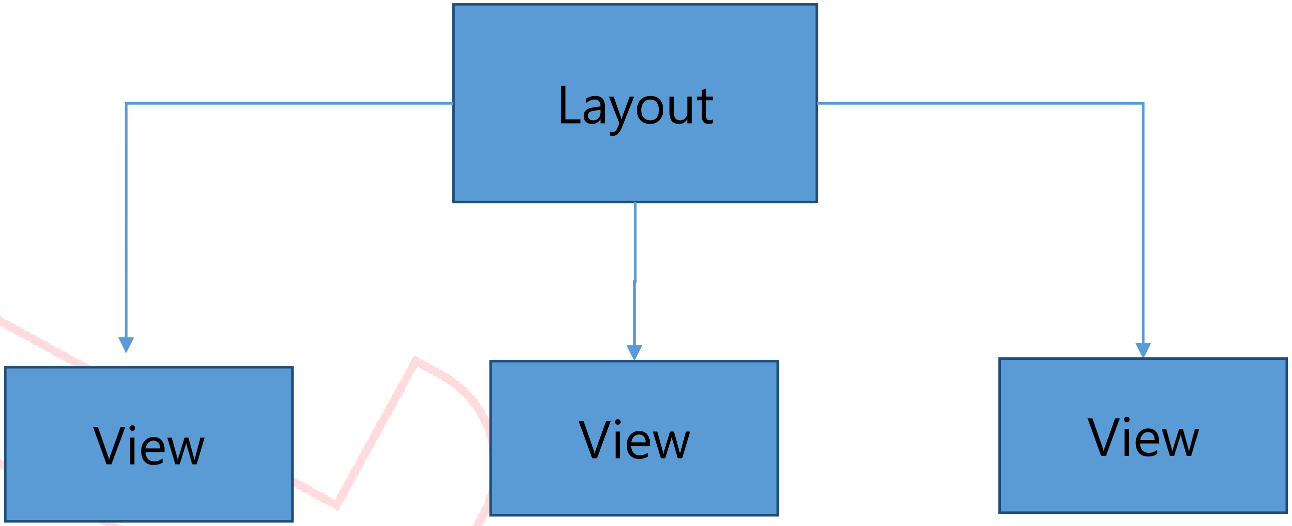
Lesson 1: Using Layouts

- What Are Layouts?
- Creating a Layout
- Linking Views and Layouts

What Are Layouts?

Layouts:

- Allow you to create a style template for a web application
- Allow you to define the content layout, to share across all views



Creating a Layout

While creating layouts:

- You can store the layout in the \Views\Shared folder
- You can use the **@RenderBody()** method to help place content of a view in the layout
- You can use the **ViewBag** object to pass information between a view and a layout

```
<!DOCTYPE html><html><head>
    <meta name="viewport" content="width=device-width" />
    <title>@ViewBag.Title</title>
</head>
<body><div>
        @RenderBody()
</div></body></html>
```

Linking Views and Layouts

To link views and layouts:

- You can add the **Layout** directive at the top of the view file
- You can use the **ViewBag** object:
  - Use properties to pass information between views and templates
  - Use the **@section** directive to define sections in the layout
- You can use the **_ViewStart** file to define the layout
  - Add the _ViewStart.cshtml file in the \Views folder of your project

Lesson 2: Applying CSS Styles to an MVC Application

- Overview of User Interface Design with Expression Blend
- Importing Styles into an MVC Web Application
- Demonstration: How to Apply a Consistent Look and Feel

Overview of User Interface Design with Expression Blend

Expression Blend for HTML:

- Helps create user interfaces for Windows 8 HTML 5 applications and other web applications
- Includes a visual designer
- Includes the interactive mode
- Allows editing CSS on the graphical user interface

Importing Styles into an MVC Web Application

After importing the CSS file:

- You should modify the layout of the web application by using the <link> element
- You can add CSS selectors to define how the styles should be applied:
    - CSS class selectors help specify a style for a group of elements
    - CSS id selectors help specify a style for any unique element in the HTML code

```
.menu
{
    font-weight:bold;
}
```

→

```
<p class="menu"> this is menu</p>
```

Demonstration: How to Apply a Consistent Look and Feel

In this demonstration, you will see how to:

1. Create a new template view
2. Apply the template view to an MVC view file
3. Apply a style sheet to the template view
4. Set the default template view for the MVC web application

Lesson 3: Creating an Adaptive User Interface

- The HTML5 Viewport Attribute
- CSS Media Queries
- MVC 4 Templates and Mobile-Specific Views
- jQuery Mobile

The HTML5 Viewport Attribute

The viewport attribute:

- Helps render webpages in a virtual window, in mobile devices
- Helps eliminate the need to reduce the size of the layout of each webpage
- Supports properties that help to specify the width, height, and scalability of the virtual window

```
<meta name="viewport" content="width=device-width,
  initial-scale=1, maximum-scale=1">
```

CSS Media Queries

Characteristics of media queries:

- Media queries are special selectors that begin with @media
- You can also apply media queries in <link> elements
- Media queries support properties that allow you to specify the size details of the targeted display area

```
@media only screen and (max-width: 500px) {
    header{
      float: none;
    }
}
```

MVC 4 Templates and Mobile-Specific Views

Default Display Mode:

- You can name the view files by using the following syntax:

  [view].mobile.cshtml

Custom Display Mode:

- You can name the view files by using the following syntax:

  [view].[mode name].cshtml

jQuery Mobile

jQuery Mobile library:

- Includes a set of JavaScript and CSS files
- Enables you to create mobile-specific views with minimum changes to HTML elements
- Helps use CDN to bring servers closer to the users

Lab: Applying Styles to MVC 4 Web Applications

- Exercise 1: Creating and Applying Layouts
- Exercise 2: Applying Styles to an MVC Web Application
- Exercise 3: Optional—Adapting Webpages for Mobile Browsers

Estimated Time: 40 minutes

Lab 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 and breadcrumb controls, which should appear on every page of the application.
- *A style sheet and images for all webpages*. The web design team has provided an HTML mock-up application to show how the final product should look. This mock-up includes a style sheet and image files. You need to import these files and apply them to every page of the application.
- *A mobile-specific view*. 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.

Lab Review

- When you first browsed the web application in Exercise 1, why was the menu and the breadcrumb trail visible on the home page, but not on the All Photos page or any other page?
- When you first viewed the site as a mobile browser in Exercise 3, what are the problems you came across with the display of the site heading and menu?

Module Review and Takeaways

- Real-world Issues and Scenarios
- Review Question(s)