Front End Developer Internship

Prosper IT Consulting
November 20 - December 1, 2017

Project Overview

Assigned to sprint focused on clearing styling and functionality backlog for a blog website.

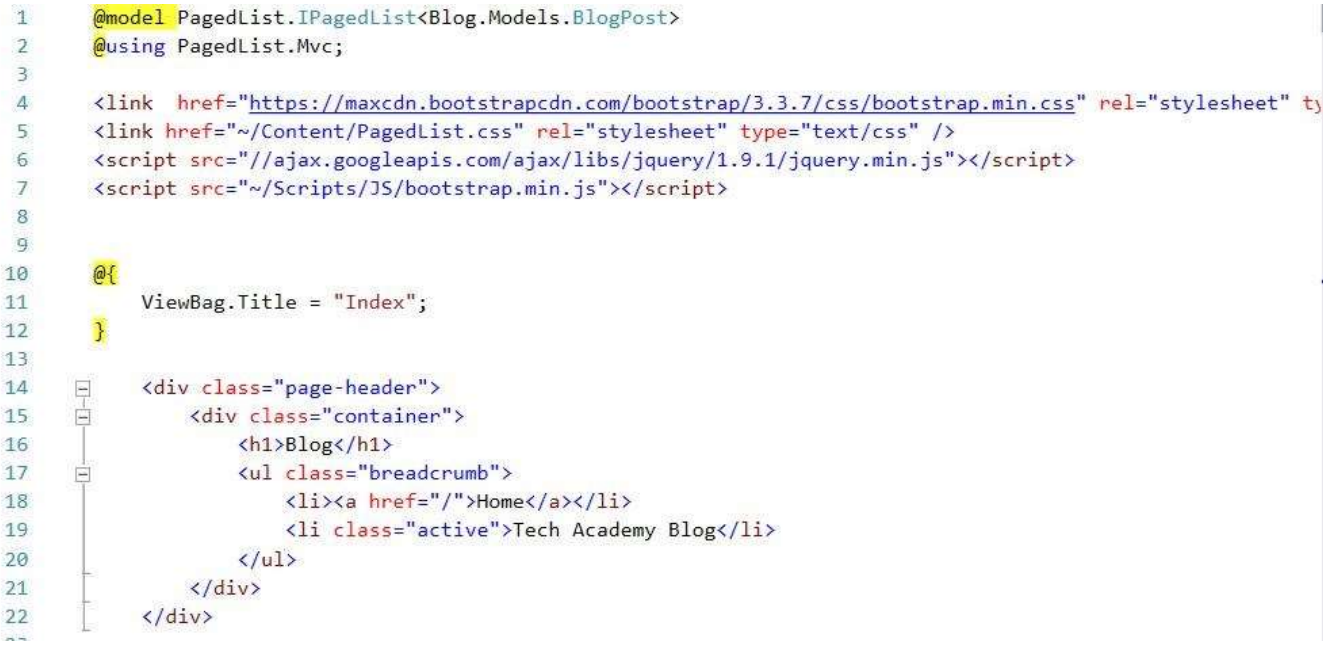
- *Framework:* ASP.NET MVC
- *Source Control:* Visual Studio Team Services
- *Communication Tools:* Slack, Daily Stand-Ups via Phone, E-mail

User stories focused on styling

User Story - Format header to match styling of live website and include a header image.

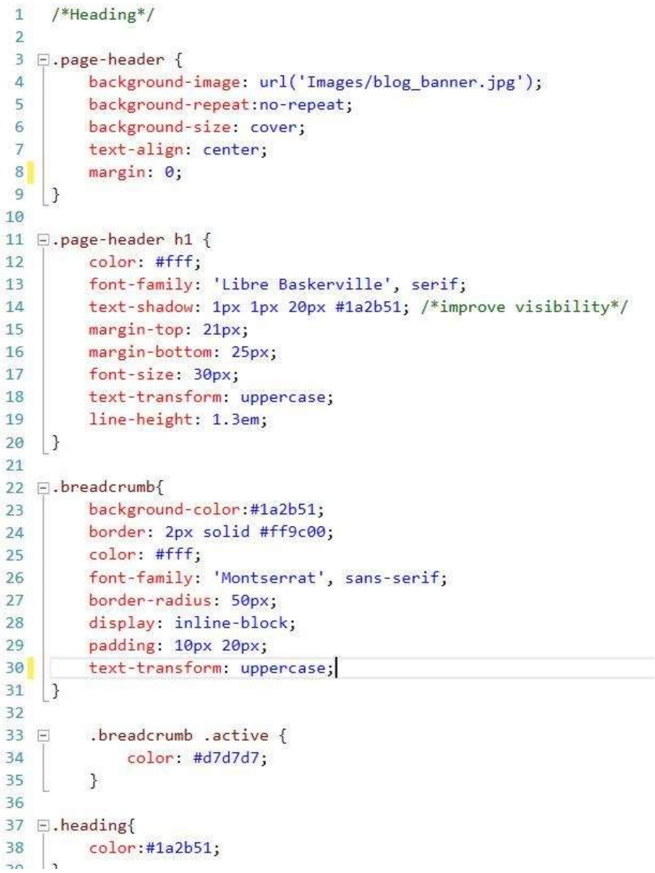
Approach

- Add <div> class for header and add Bootstrap container for header text. Use Bootstrap breadcrumb class to display page location relative to Home.
- Find image that is free for use and set as background image. Style <h1> element adding text-shadow to improve text visibility over image. Style breadcrumb class to match live website color scheme.



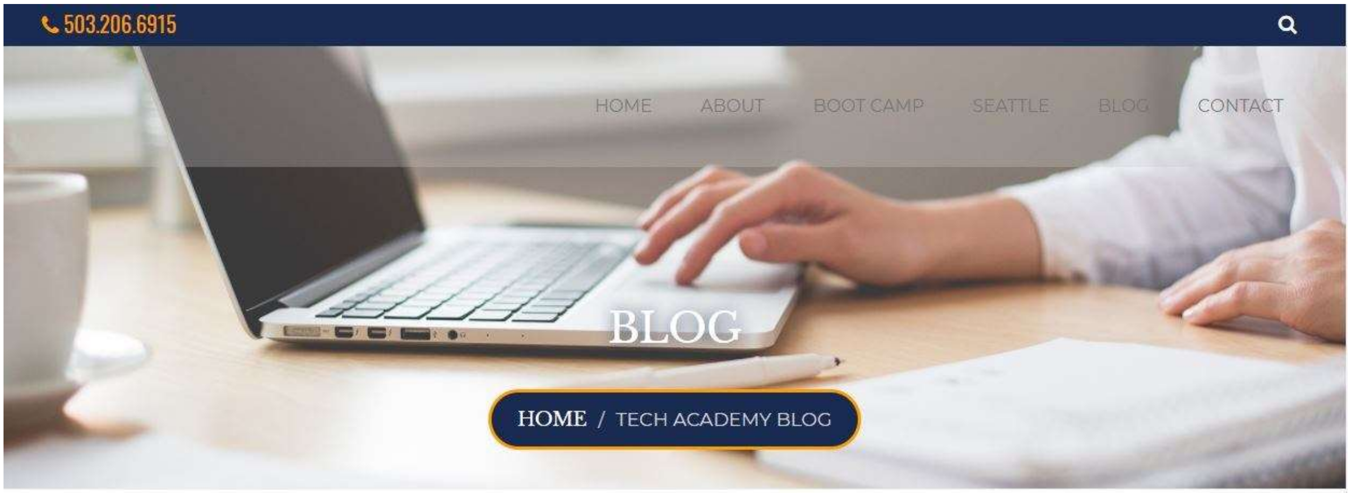
```
@model PagedList.IPagedList<Blog.Models.BlogPost>
@using PagedList.Mvc;

<link  href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet" ty
<link href="~/Content/PagedList.css" rel="stylesheet" type="text/css" />
<script src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/jquery.min.js"></script>
<script src="~/Scripts/JS/bootstrap.min.js"></script>


@{
    ViewBag.Title = "Index";
}

    <div class="page-header">
        <div class="container">
            <h1>Blog</h1>
            <ul class="breadcrumb">
                <li><a href="/">Home</a></li>
                <li class="active">Tech Academy Blog</li>
            </ul>
        </div>
    </div>
```

1 - CSHTML



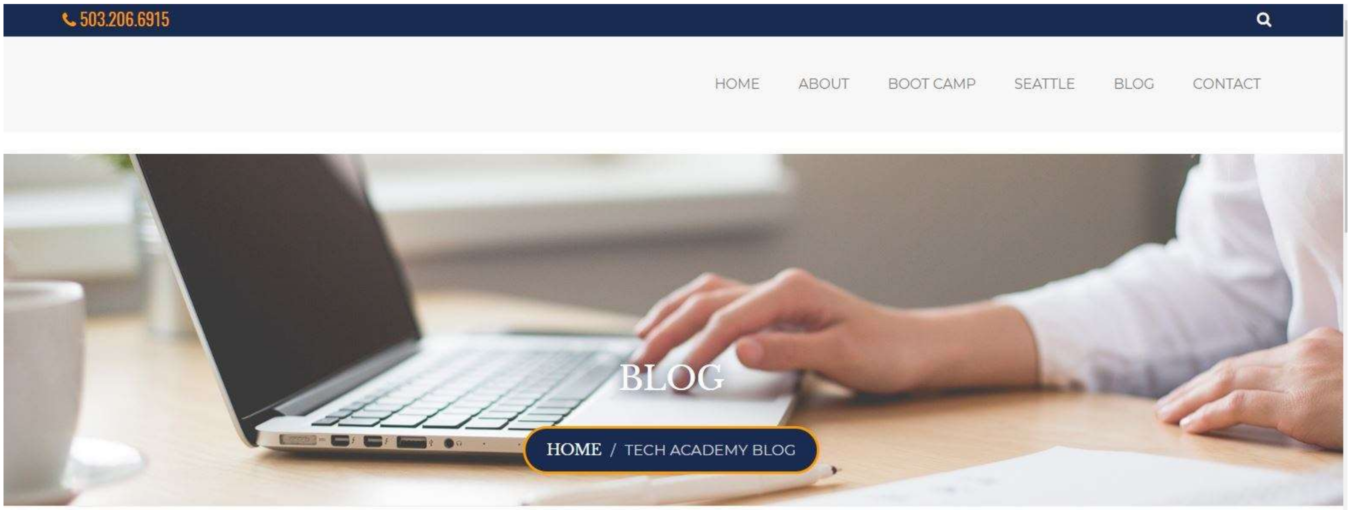
```css
/*Heading*/

.page-header {
    background-image: url('Images/blog_banner.jpg');
    background-repeat:no-repeat;
    background-size: cover;
    text-align: center;
    margin: 0;
}

.page-header h1 {
    color: #fff;
    font-family: 'Libre Baskerville', serif;
    text-shadow: 1px 1px 20px #1a2b51; /*improve visibility*/
    margin-top: 21px;
    margin-bottom: 25px;
    font-size: 30px;
    text-transform: uppercase;
    line-height: 1.3em;
}

.breadcrumb{
    background-color:#1a2b51;
    border: 2px solid #ff9c00;
    color: #fff;
    font-family: 'Montserrat', sans-serif;
    border-radius: 50px;
    display: inline-block;
    padding: 10px 20px;
    text-transform: uppercase;
}

    .breadcrumb .active {
        color: #d7d7d7;
    }

.heading{
    color:#1a2b51;
}
```

2 - CSS



3 - Final Result

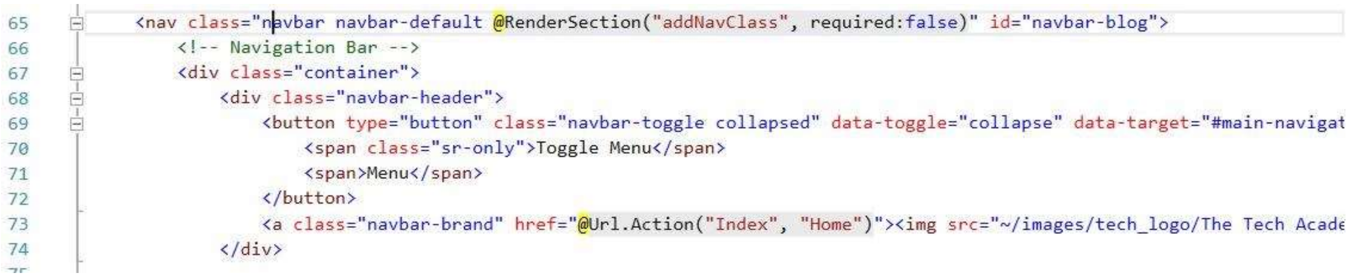
User Story - Have navigation bar overlap page-header with transparent background.



4 - Before: Navigation Bar and page-header separated by gap.

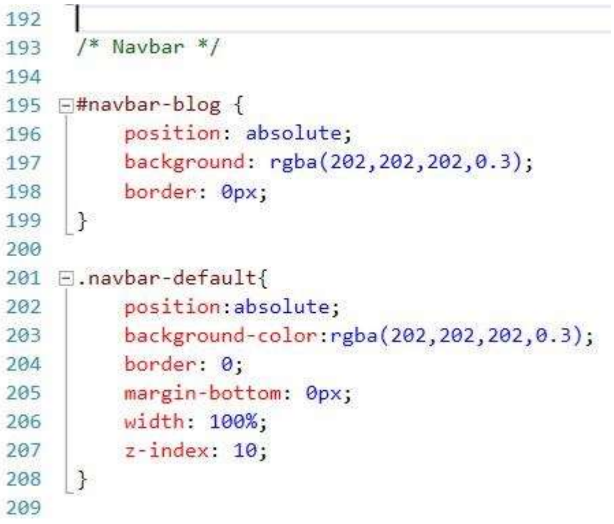
Approach

- The <nav> class styling is being over-ridden by other stylesheets. Add id to <nav> to allow for custom styling.
- Make <nav> position absolute to eliminate gap between navigation bar and page header. Set background to grey and add opacity. Style page header so there is no margin.



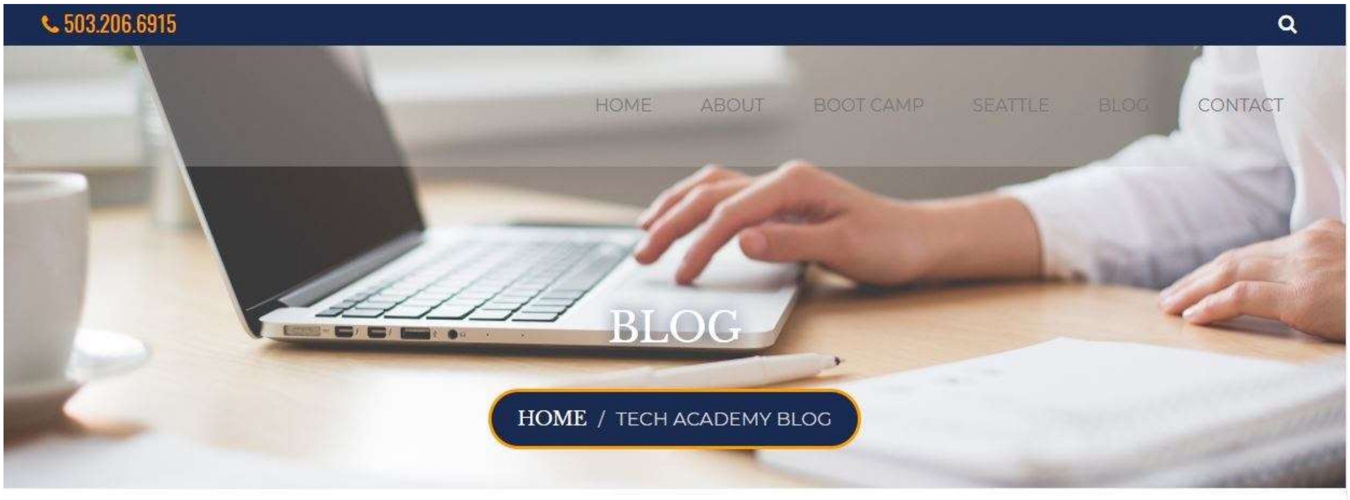
```
<nav class="navbar navbar-default @RenderSection("addNavClass", required:false)" id="navbar-blog">
    <!-- Navigation Bar -->
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#main-navigat
                <span class="sr-only">Toggle Menu</span>
                <span>Menu</span>
            </button>
            <a class="navbar-brand" href="@Url.Action("Index", "Home")"><img src="~/images/tech_logo/The Tech Acade
        </div>
```

5 - CSHTML



```

/* Navbar */

#navbar-blog {
    position: absolute;
    background: rgba(202,202,202,0.3);
    border: 0px;
}

.navbar-default{
    position:absolute;
    background-color:rgba(202,202,202,0.3);
    border: 0;
    margin-bottom: 0px;
    width: 100%;
    z-index: 10;
}

```

6 - CSS

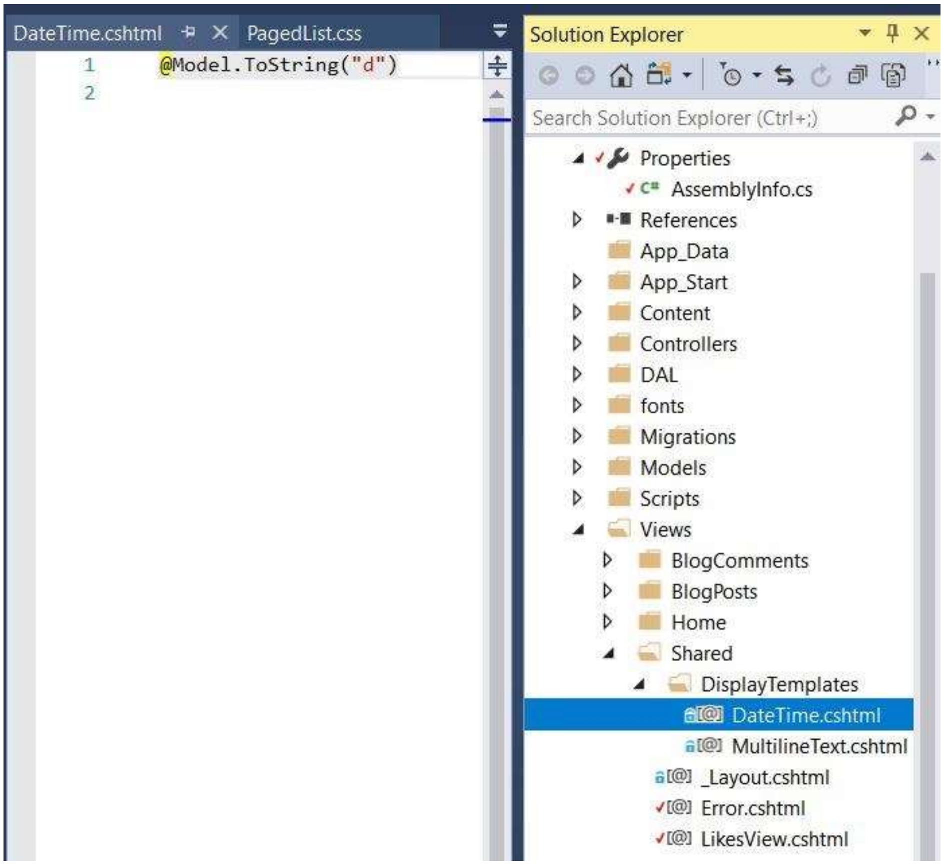


7 - After: Navigation overlaps page header.

User Story - Remove the time of day from the blogpost's timestamp.

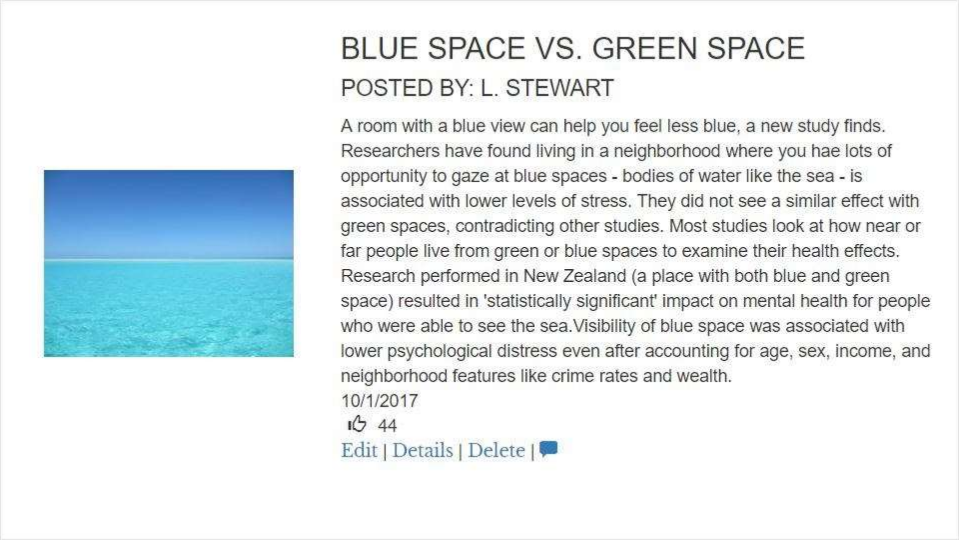
Approach

- Add a new Display Template to the Shared View targeting the DateTime property of the model. Use ToString Method with date format.



```
@Model.ToString("d")
```

8 - CSHTML



BLUE SPACE VS. GREEN SPACE

POSTED BY: L. STEWART

A room with a blue view can help you feel less blue, a new study finds. Researchers have found living in a neighborhood where you hae lots of opportunity to gaze at blue spaces - bodies of water like the sea - is associated with lower levels of stress. They did not see a similar effect with green spaces, contradicting other studies. Most studies look at how near or far people live from green or blue spaces to examine their health effects. Research performed in New Zealand (a place with both blue and green space) resulted in 'statistically significant' impact on mental health for people who were able to see the sea.Visibility of blue space was associated with lower psychological distress even after accounting for age, sex, income, and neighborhood features like crime rates and wealth.

10/1/2017

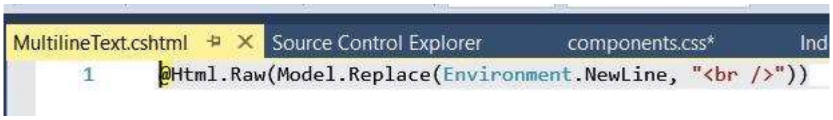
44

9 - Result: Blog post displaying only date.

User Story - Preserve the structure of the text added to the description input box of the create view of the blog post controller.

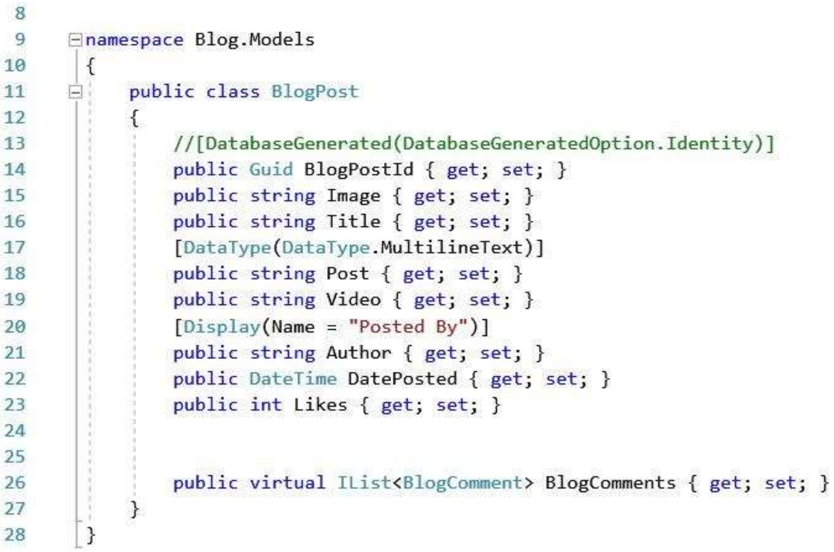
Approach

- Create a display template that replaces the new line string with a <br /> in the post text retrieved from the database.
- Assign data type of Post string to multiline text. This automatically sets the input box height to more than one line and allows for display template targeting.



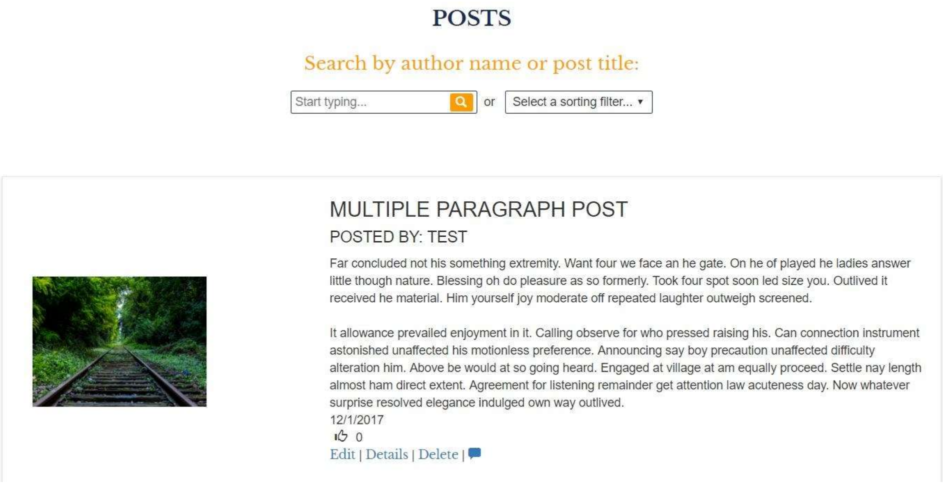
```
@Html.Raw(Model.Replace(Environment.NewLine, "<br />"))
```

10 - CSHTML



```

namespace Blog.Models
{
    public class BlogPost
    {
        //[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
        public Guid BlogPostId { get; set; }
        public string Image { get; set; }
        public string Title { get; set; }
        [DataType(DataType.MultilineText)]
        public string Post { get; set; }
        public string Video { get; set; }
        [Display(Name = "Posted By")]
        public string Author { get; set; }
        public DateTime DatePosted { get; set; }
        public int Likes { get; set; }


        public virtual IList<BlogComment> BlogComments { get; set; }
    }
}
```

11 - C#



POSTS

Search by author name or post title:

Start typing... or Select a sorting filter... ▾

MULTIPLE PARAGRAPH POST

POSTED BY: TEST

Far concluded not his something extremity. Want four we face an he gate. On he of played he ladies answer little though nature. Blessing oh do pleasure as so formerly. Took four spot soon led size you. Outlived it received he material. Him yourself joy moderate off repeated laughter outweigh screened.

It allowance prevailed enjoyment in it. Calling observe for who pressed raising his. Can connection instrument astonished unaffected his motionless preference. Announcing say boy precaution unaffected difficulty alteration him. Above be would at so going heard. Engaged at village at am equally proceed. Settle nay length almost ham direct extent. Agreement for listening remainder get attention law acuteness day. Now whatever surprise resolved elegance indulged own way outlived.

12/1/2017

👍 0

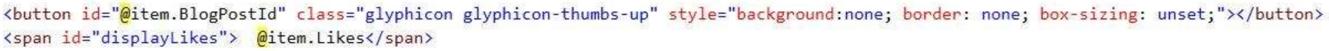
Edit | Details | Delete |

12 - After: Post text retains breaks between paragraphs.

User Story - Remove background from like button, replace the word "comment" with a glyphicon, and set a default comment post image.

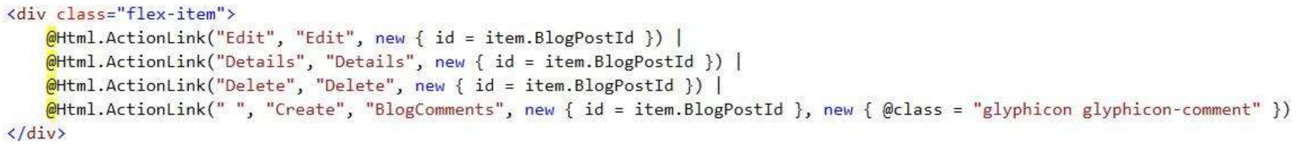
Approach

- Remove background using in-line styling
- Use overloaded ActionLink method to add Bootstrap glyphicon class
- Find a copyright-free image and modify it in Photoshop to display website brand colors. Add the image before the DisplayFor method for comments.



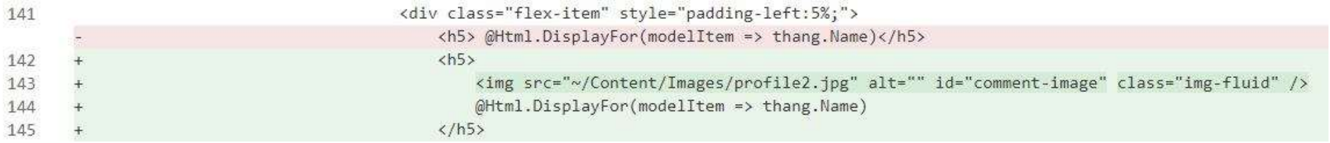
```
<button id="@item.BlogPostId" class="glyphicon glyphicon-thumbs-up" style="background:none; border: none; box-sizing: unset;"></button>
<span id="displayLikes">  @item.Likes</span>
```

13 - CSHTML



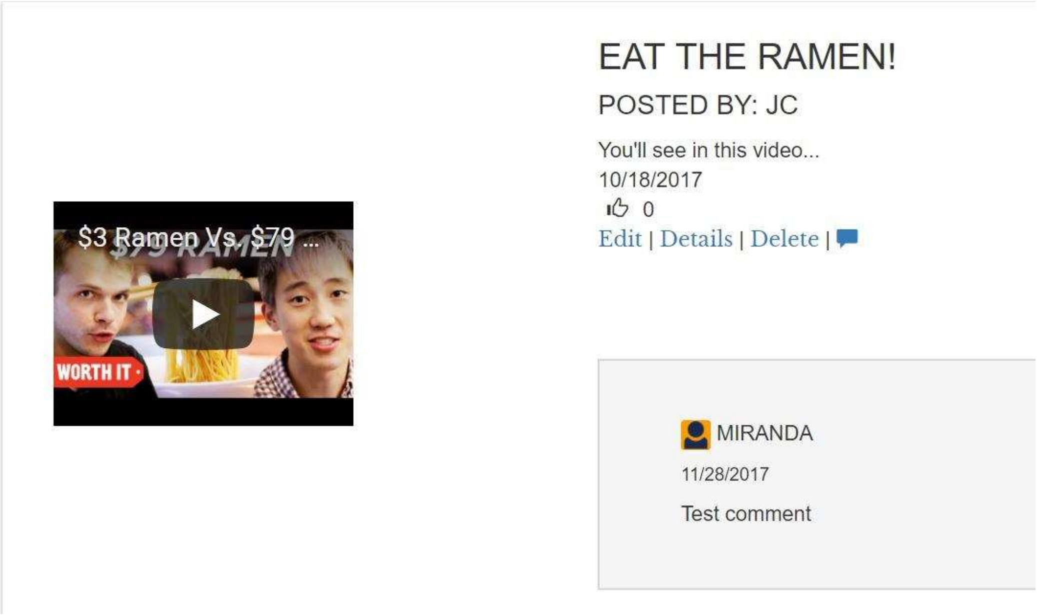
```
<div class="flex-item">
    @Html.ActionLink("Edit", "Edit", new { id = item.BlogPostId }) |
    @Html.ActionLink("Details", "Details", new { id = item.BlogPostId }) |
    @Html.ActionLink("Delete", "Delete", new { id = item.BlogPostId }) |
    @Html.ActionLink(" ", "Create", "BlogComments", new { id = item.BlogPostId }, new { @class = "glyphicon glyphicon-comment" })
</div>
```

14 - CSHTML



```
       <div class="flex-item" style="padding-left:5%;">
-          <h5> @Html.DisplayFor(modelItem => thang.Name)</h5>
+          <h5>
+              <img src="~/Content/Images/profile2.jpg" alt="" id="comment-image" class="img-fluid" />
+              @Html.DisplayFor(modelItem => thang.Name)
+          </h5>
```

15 - CSHTML



EAT THE RAMEN!

POSTED BY: JC

You'll see in this video...

10/18/2017

0

Edit | Details | Delete |

MIRANDA

11/28/2017

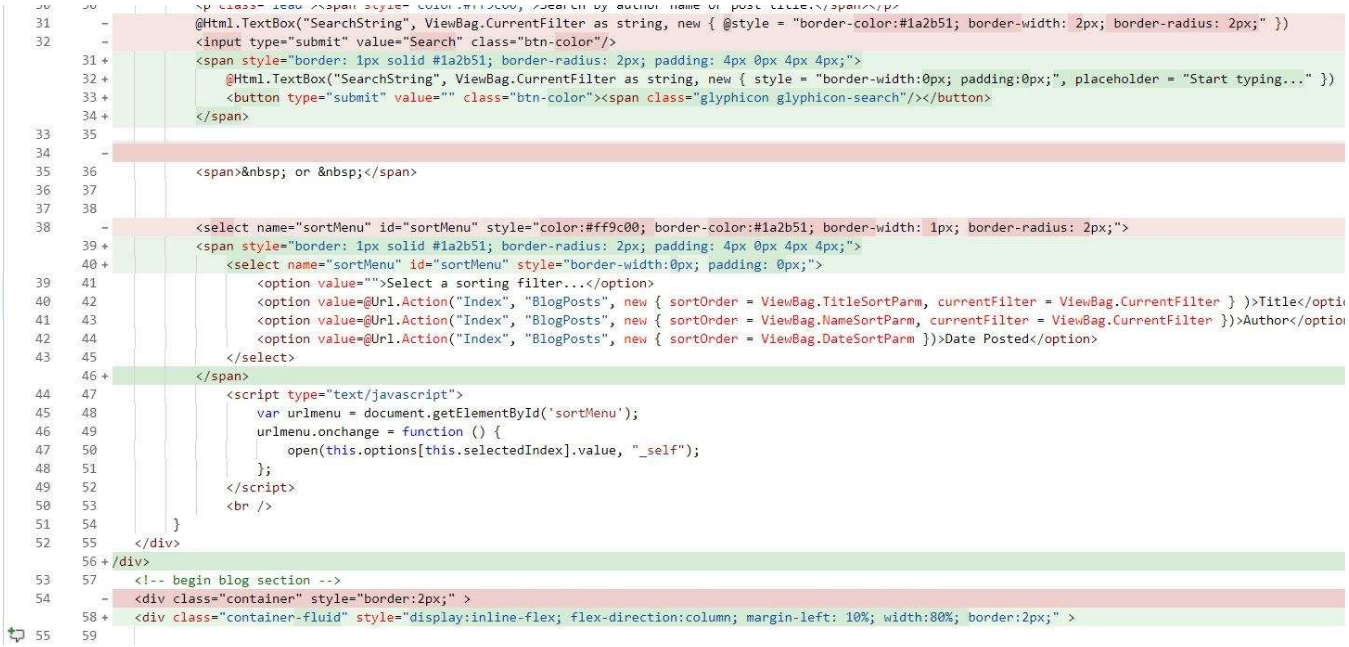
Test comment

16 - After: No background behind like button, comment glyphicon present, and branded comment image displayed.

User Story - Embed search button within the search text box and replace the word search with the magnifying glass glyphicon.

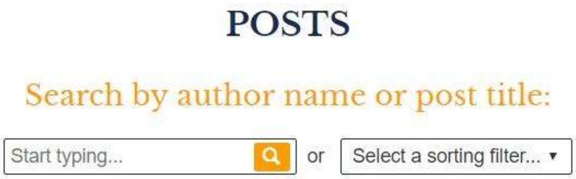
Approach

- Use bootstrap class to replace the word "search" with a magnifying glass glyphicon. Use <span> to create box around the search input field and button. Also use <span> around the sort list dropdown field so boxes won't appear to be different sizes.



```
            <p class="lead"><span style="color:#ff9c00;">Search by author name or post title:</span></p>
-           @Html.TextBox("SearchString", ViewBag.CurrentFilter as string, new { @style = "border-color:#1a2b51; border-width: 2px; border-radius: 2px;" })
-           <input type="submit" value="Search" class="btn-color"/>
+           <span style="border: 1px solid #1a2b51; border-radius: 2px; padding: 4px 0px 4px 4px;">
+               @Html.TextBox("SearchString", ViewBag.CurrentFilter as string, new { style = "border-width:0px; padding:0px;", placeholder = "Start typing..." })
+               <button type="submit" value="" class="btn-color"><span class="glyphicon glyphicon-search"/></button>
+           </span>

-
            <span>&nbsp; or &nbsp;</span>


-           <select name="sortMenu" id="sortMenu" style="color:#ff9c00; border-color:#1a2b51; border-width: 1px; border-radius: 2px;">
+           <span style="border: 1px solid #1a2b51; border-radius: 2px; padding: 4px 0px 4px 4px;">
+               <select name="sortMenu" id="sortMenu" style="border-width:0px; padding: 0px;">
                    <option value="">Select a sorting filter...</option>
                    <option value=@Url.Action("Index", "BlogPosts", new { sortOrder = ViewBag.TitleSortParm, currentFilter = ViewBag.CurrentFilter } )>Title</opti
                    <option value=@Url.Action("Index", "BlogPosts", new { sortOrder = ViewBag.NameSortParm, currentFilter = ViewBag.CurrentFilter })>Author</optio
                    <option value=@Url.Action("Index", "BlogPosts", new { sortOrder = ViewBag.DateSortParm })>Date Posted</option>
                </select>
+           </span>
                <script type="text/javascript">
                    var urlmenu = document.getElementById('sortMenu');
                    urlmenu.onchange = function () {
                        open(this.options[this.selectedIndex].value, "_self");
                    };
                </script>
                <br />
        }
    </div>
+/div>
    <!-- begin blog section -->
-   <div class="container" style="border:2px;" >
+   <div class="container-fluid" style="display:inline-flex; flex-direction:column; margin-left: 10%; width:80%; border:2px;" >

```

17 - CSHTML


POSTS
Search by author name or post title:
Start typing...
or
Select a sorting filter... ▾

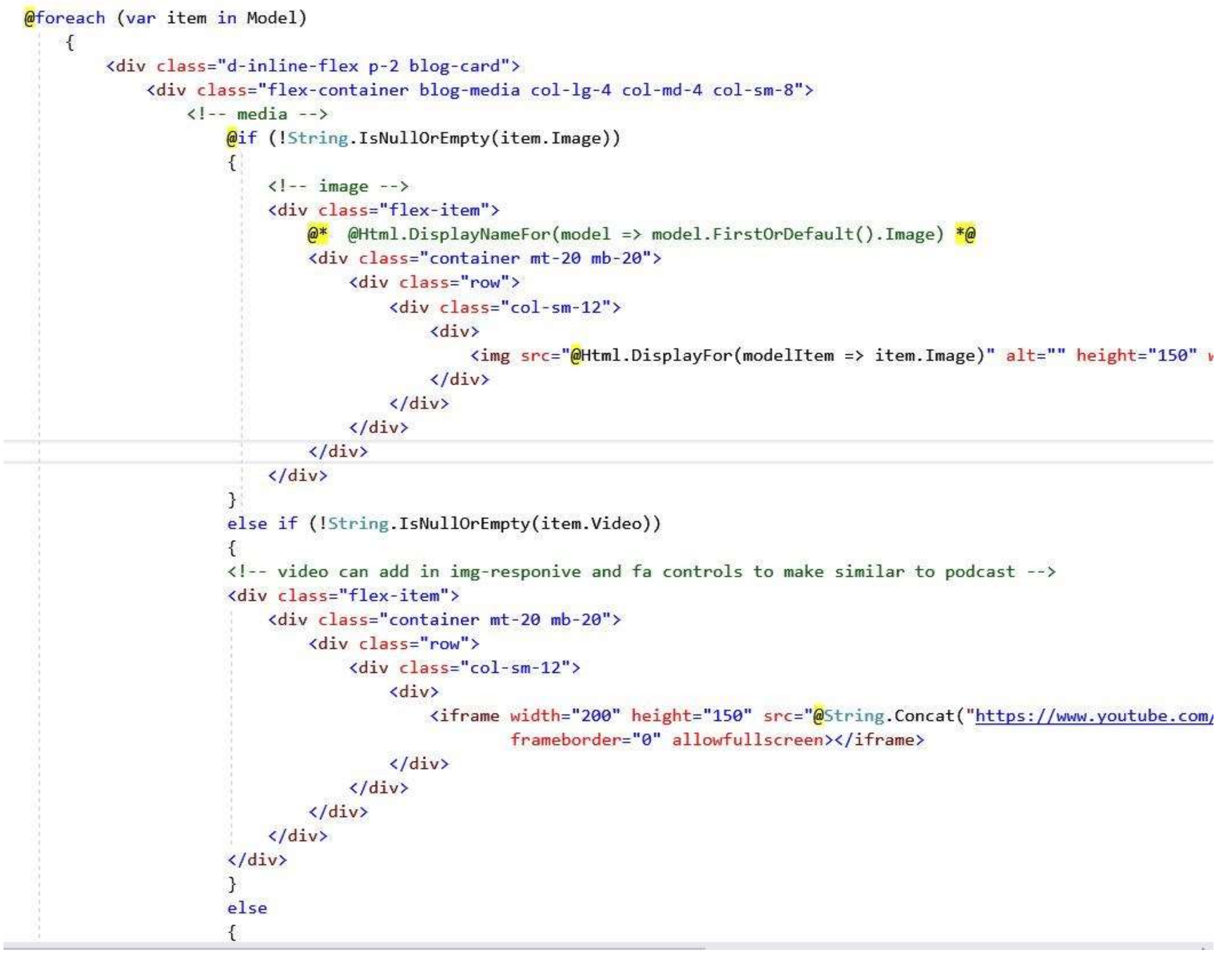
18 - Result: Search and drop-down list are surrounded with boxes.

User stories focused on functionality

User Story - Add a conditional statement to the index view for displaying images vs videos in a post.

Approach

- Add if/else if statement that tests if the string field for model Image, model Video, or both are empty.



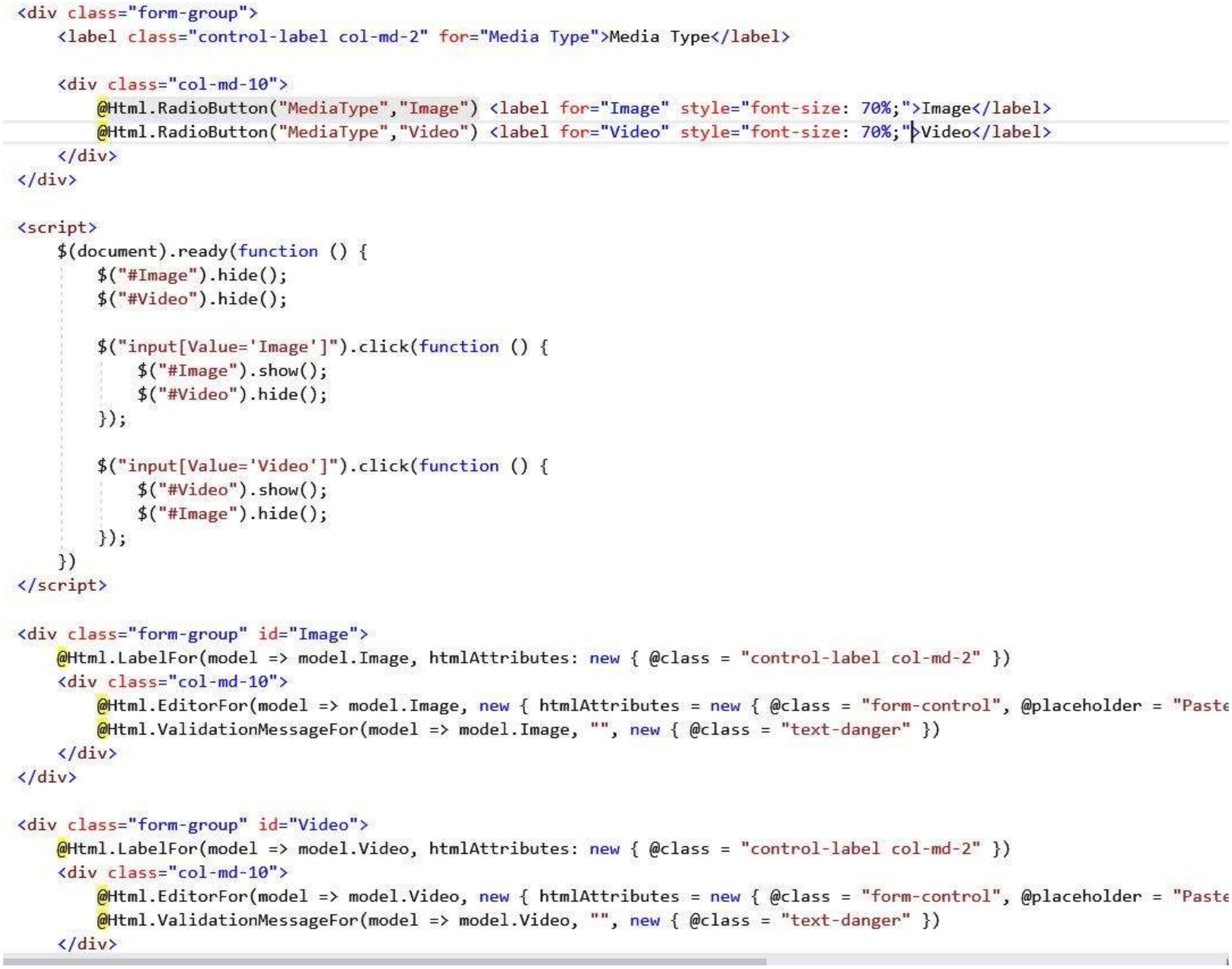
```
@foreach (var item in Model)
    {
        <div class="d-inline-flex p-2 blog-card">
            <div class="flex-container blog-media col-lg-4 col-md-4 col-sm-8">
                <!-- media -->
                    @if (!String.IsNullOrEmpty(item.Image))
                    {
                        <!-- image -->
                        <div class="flex-item">
                            @*  @Html.DisplayNameFor(model => model.FirstOrDefault().Image) *@
                            <div class="container mt-20 mb-20">
                                <div class="row">
                                    <div class="col-sm-12">
                                        <div>
                                            <img src="@Html.DisplayFor(modelItem => item.Image)" alt="" height="150" w
                                        </div>
                                    </div>
                                </div>
                            </div>
                        </div>
                    }
                    else if (!String.IsNullOrEmpty(item.Video))
                    {
                    <!-- video can add in img-responive and fa controls to make similar to podcast -->
                    <div class="flex-item">
                        <div class="container mt-20 mb-20">
                            <div class="row">
                                <div class="col-sm-12">
                                    <div>
                                        <iframe width="200" height="150" src="@String.Concat("https://www.youtube.com/
                                                frameborder="0" allowfullscreen></iframe>
                                    </div>
                                </div>
                            </div>
                        </div>
                    </div>
                    }
                    else
                    {
```

19 – CSHTML

User Story - On the create view for the blog post, add radio buttons so user can only push one form of media (image or video).

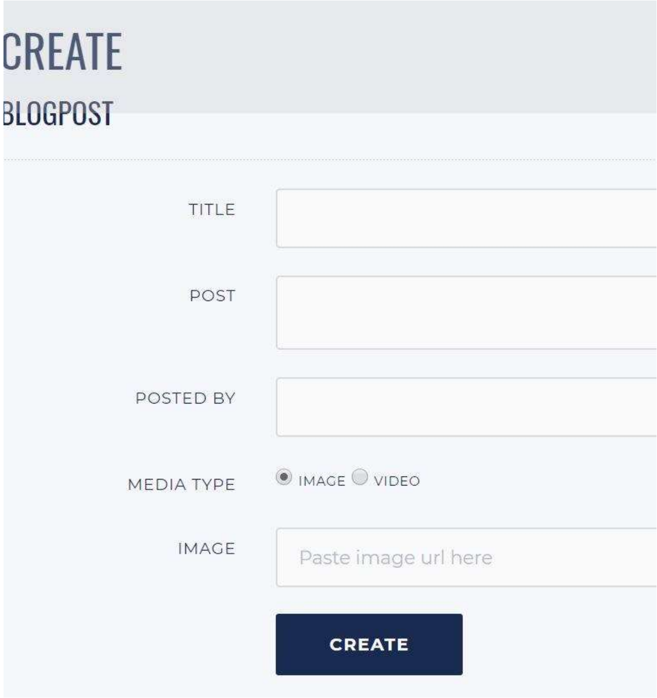
Approach

- Use html helper class to add radio button for Image and Video. Add id to each form field for targeting. Write jQuery script that will initially hide both fields and will then display the form field associated with each radio button on click.



```
<div class="form-group">
    <label class="control-label col-md-2" for="Media Type">Media Type</label>

    <div class="col-md-10">
        @Html.RadioButton("MediaType","Image") <label for="Image" style="font-size: 70%;">Image</label>
        @Html.RadioButton("MediaType","Video") <label for="Video" style="font-size: 70%;">Video</label>
    </div>
</div>

<script>
    $(document).ready(function () {
        $("#Image").hide();
        $("#Video").hide();

        $("input[Value='Image']").click(function () {
            $("#Image").show();
            $("#Video").hide();
        });

        $("input[Value='Video']").click(function () {
            $("#Video").show();
            $("#Image").hide();
        });
    })
</script>

<div class="form-group" id="Image">
    @Html.LabelFor(model => model.Image, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Image, new { htmlAttributes = new { @class = "form-control", @placeholder = "Paste
        @Html.ValidationMessageFor(model => model.Image, "", new { @class = "text-danger" })
    </div>
</div>

<div class="form-group" id="Video">
    @Html.LabelFor(model => model.Video, htmlAttributes: new { @class = "control-label col-md-2" })
    <div class="col-md-10">
        @Html.EditorFor(model => model.Video, new { htmlAttributes = new { @class = "form-control", @placeholder = "Paste
        @Html.ValidationMessageFor(model => model.Video, "", new { @class = "text-danger" })
    </div>
```

20 - CSHTML + jQuery



CREATE

BLOGPOST

TITLE

POST

POSTED BY

MEDIA TYPE ● IMAGE ○ VIDEO

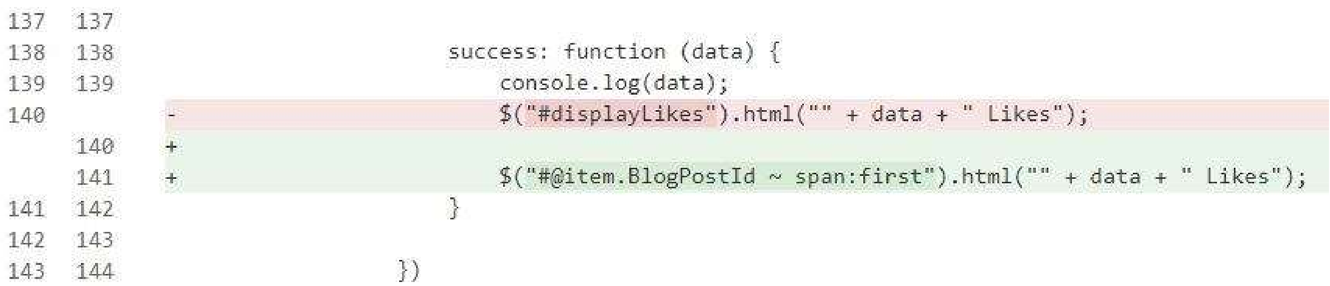
IMAGE Paste image url here

21 – Result

User Story - Debug like functionality for the blog posts (only the first like button on each page seems to be working)

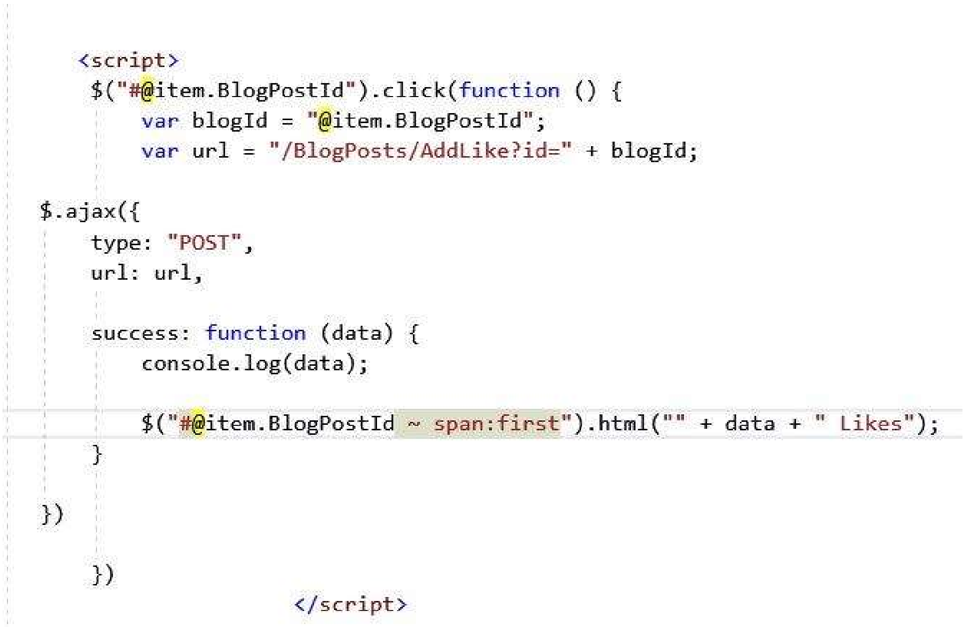
Approach

- Testing of the like buttons on the blog posts revealed that clicking on any like button on the page would only increase the like count next to the first like button. Review of the jQuery script revealed that the code to display likes did not target a specific blog post.
- Modified the script to only display the like click for the specified blog post. Targeting was achieved with the blog post id property and directing to the first span, where the number of likes is displayed.



```

                          success: function (data) {
                              console.log(data);
-                             $("#displayLikes").html("" + data + " Likes");
+
+                             $("#@item.BlogPostId ~ span:first").html("" + data + " Likes");
                          }

                      })
```

22 - jQuery



```
    <script>
     $("#@item.BlogPostId").click(function () {
         var blogId = "@item.BlogPostId";
         var url = "/BlogPosts/AddLike?id=" + blogId;

 $.ajax({
     type: "POST",
     url: url,

     success: function (data) {
         console.log(data);

         $("#@item.BlogPostId ~ span:first").html("" + data + " Likes");
     }

 })

     })
                    </script>
```

23 - jQuery + CSHTML