Build an ASP.NET Core MVC App with EF Core One-Day Hands-On Lab

Lab 10

This lab adds support for client-side libraries, updates the _Layout.cshtml file by creating partials for the major sections, and adds GDPR support. Prior to starting this lab, you must have completed Lab 9.

Part 1: Manage Client-Side Libraries

Step 1: Install Library Manager

- □ [Visual Studio] Library Manager is part of Visual Studio 2022.
- □ [Command Line] Install the Library Manager CLI Tooling as a global tool:

```
dotnet tool install --global Microsoft.Web.LibraryManager.Cli --version 2.1.113
```

Step 2: Delete the lib directory from the default template

- □ Delete the `wwwroot\lib` folder. It will be replaced with files using library manager.

Step 3: Add the libman.json file

- □ [Visual Studio] Right click on the `AutoLot.Mvc` project and select Manage Client-Side Libraries. This adds the `libman.json` file to the root of the project. Right click on the `libman.json` file and select "Enable restore on build".
- □ [Command Line] Create a new `libman.json` file with the following command:

```
libman init
```

Step 4: Update the libman.json file

- □ Update the `libman.json` file to the following JSON:

```
{
  //https://api.cdnjs.com/libraries?output=human
  //https://api.cdnjs.com/libraries/jquery?output=human
  "version": "1.0",
  "defaultProvider": "cdnjs",
  "defaultDestination": "wwwroot/lib",
  "libraries": [
    {
      "library": "jquery@3.6.0",
      "destination": "wwwroot/lib/jquery",
      "files": [ "jquery.js", "jquery.min.js" ]
    },
    {
      "library": "jquery-validate@1.19.3",
      "destination": "wwwroot/lib/jquery-validation",
      "files": [ "jquery.validate.js", "jquery.validate.min.js", "additional-methods.js",
"additional-methods.min.js" ]
    },
    {
      "library": "jquery-validation-unobtrusive@3.2.12",
      "destination": "wwwroot/lib/jquery-validation-unobtrusive",
      "files": [ "jquery.validate.unobtrusive.js", "jquery.validate.unobtrusive.min.js" ]
    },
    {
      "library": "twitter-bootstrap@5.1.3",
      "destination": "wwwroot/lib/bootstrap",
      "files": [ "css/bootstrap.css", "css/bootstrap.css.map",
        "css/bootstrap.min.css", "css/bootstrap.min.css.map",
        "js/bootstrap.bundle.js", "js/bootstrap.bundle.js.map",
        "js/bootstrap.bundle.min.js","js/bootstrap.bundle.min.js.map",
        "js/bootstrap.js", "js/bootstrap.js.map",
        "js/bootstrap.min.js","js/bootstrap.min.js.map"
      ]
    },
    {
      "library": "font-awesome@5.15.4",
      "destination": "wwwroot/lib/font-awesome/",
      "files": ["js/all.js","css/all.css",
        "sprites/brands.svg","sprites/regular.svg","sprites/solid.svg",
        "webfonts/fa-brands-400.eot","webfonts/fa-brands-400.svg",
        "webfonts/fa-brands-400.ttf","webfonts/fa-brands-400.woff",
        "webfonts/fa-brands-400.woff2",
        "webfonts/fa-regular-400.eot","webfonts/fa-regular-400.svg",
        "webfonts/fa-regular-400.ttf","webfonts/fa-regular-400.woff",
        "webfonts/fa-regular-400.woff2",
        "webfonts/fa-solid-900.eot","webfonts/fa-solid-900.svg",
        "webfonts/fa-solid-900.ttf","webfonts/fa-solid-900.woff",
        "webfonts/fa-solid-900.woff2"
      ]
    }
  ]
}
```

Step 5: Update the client-side libraries

- [Visual Studio] Right click on the `libman.json` file and select Restore Client-Side Libraries.
- [Command Line]

```
libman restore
```

Part 2: Update the Layout and ValidationScriptsPartial views

Step 1: Add the Layout Partials files

- Create a new folder named `Partials` under the Views\Shared folder.
- Create a new view file named `_Head.cshtml` in the `Views\Shared\Partials` folder. Update the content to the following:

```
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>@ViewData["Title"] - AutoLot.Mvc</title>
<environment include="Development,Local">
  <link rel="stylesheet" href="~/lib/bootstrap/css/bootstrap.css" asp-append-version="true" />
</environment>
<environment exclude="Development,Local">
  <link rel="stylesheet"
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
    asp-fallback-href="~/lib/bootstrap/css/bootstrap.min.css"
    asp-fallback-test-class="sr-only" asp-fallback-test-property="position"
    asp-fallback-test-value="absolute" crossorigin="anonymous"
    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"/>
</environment>
<link rel="stylesheet" href="~/lib/font-awesome/css/all.css" asp-append-version="true"/>
<link rel="stylesheet" href="~/css/site.css" asp-append-version="true"/>
<link rel="stylesheet" href="~/AutoLot.Mvc.styles.css" asp-append-version="true" />
```

- □ Create a new view file named `_JavaScriptFiles.cshtml` in the `Views\Shared\Partials` folder. Update the content to the following:

```
<environment include="Development,Local">
    <script src="~/lib/jquery/jquery.js" asp-append-version="true"></script>
    <script src="~/lib/bootstrap/js/bootstrap.bundle.js" asp-append-version="true"></script>
</environment>
<environment exclude="Development,Local">
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"
            asp-fallback-src="~/lib/jquery/jquery.min.js"
            asp-fallback-test="window.jQuery"
            crossorigin="anonymous"
 integrity="sha512-
894YE6QWD5I59HgZOGReFYm4dnWc1Qt5NtvYSaNcOP+u1T9qYdvdihz0PPSiiqn/+/3e7Jo4EaG7TubfWGUrMQ==">
    </script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"
            asp-fallback-src="~/lib/bootstrap/js/bootstrap.bundle.min.js"
            asp-fallback-test="window.jQuery && window.jQuery.fn && window.jQuery.fn.modal"
            crossorigin="anonymous"
            integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p">
    </script>
</environment>
<script src="~/js/site.js" asp-append-version="true"></script>
```

- □ Create a new view file named `_Menu.cshtml` in the `Views\Shared\Partials` folder. Update the content to the following:

```
@inject IOptionsMonitor<DealerInfo> _dealerInfoMonitor
@{
  var dealerInfo = _dealerInfoMonitor.CurrentValue;
}
<nav class="navbar navbar-expand-sm navbar-light bg-white border-bottom box-shadow mb-3">
  <div class="container-fluid">
    <a class="navbar-brand" asp-area="" asp-controller="Home" asp-
action="Index">@dealerInfo.DealerName</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target=".navbar-collapse" aria-controls="navbarSupportedContent"
                aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="navbar-collapse collapse d-sm-inline-flex justify-content-between">
      <ul class="navbar-nav flex-grow-1">
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle text-dark" data-bs-toggle="dropdown">Inventory <i
class="fa fa-car"></i></a>
        </li>
        <li class="nav-item">
          <a class="nav-link text-dark" asp-area="" asp-controller="Home" asp-
action="Privacy">Privacy <i class="fa fa-user-secret"></i></a>
        </li>
      </ul>
    </div>
  </div>
</nav>
```

Step 2: Update the _Layout.cshml file

- □ In the `Views\Shared` directory, update `_Layout.cshtml` file to the following:

```
@inject IWebHostEnvironment _env

<!DOCTYPE html>
<html lang="en">
<head>
    <partial name="Partials/_Head"/>
</head>
<body>
<header>
    <partial name="Partials/_Menu"/>
</header>
<div class="container">
    <main role="main" class="pb-3">
        @RenderBody()
    </main>
</div>

<footer class="border-top footer text-muted">
    <div class="container">
        &copy; 2021 - AutoLot.Mvc - @_env.EnvironmentName - <a asp-area="" asp-controller="Home"
asp-action="Privacy">Privacy</a>
    </div>
</footer>
<partial name="Partials/_JavaScriptFiles"/>
@RenderSection("Scripts", required: false)
</body>
</html>
```

Step 2a: Render the Scripts section asnyc

- □ In the `Views\Shared` directory, update `_Layout.cshtml` file to the following:

```
@await RenderSectionAsync("Scripts", required: false)
```

Step 3: Update the _ValidationScripts.cshml partial file

□ Update the `_ValidationScriptsPartial.cshtml` view to the following:

```
<environment include="Development,Local">
  <script src="~/lib/jquery-validation/jquery.validate.js" asp-append-version="true"></script>
  <script src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.js" asp-append-
version="true"></script>
</environment>
<environment exclude="Development,Local">
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-
validate/1.19.3/jquery.validate.min.js"
   asp-fallback-src="~/lib/jquery-validation/jquery.validate.min.js"
   asp-fallback-test="window.jQuery && window.jQuery.validator"
   crossorigin="anonymous"
   integrity="sha512-
37T7leoNS06R80c8Ulq7cdCDU5MNQBwlYoy1TX/WUsLFC2eYNqtKlV0QjH7r8JpG/S0GUMZwebnVFLPd6SU5yg==">
  </script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery-validation-
unobtrusive/3.2.12/jquery.validate.unobtrusive.min.js"
    asp-fallback-src="~/lib/jquery-validation-unobtrusive/jquery.validate.unobtrusive.min.js"
 asp-fallback-test="window.jQuery && window.jQuery.validator &&
window.jQuery.validator.unobtrusive"
    crossorigin="anonymous"
    integrity="sha512-
o6XqxgrUsKmchwy9G5VRNWSSxTS4Urr4loO6/0hYdpWmFUfHqGzawGxeQGMDqYzxjY9sbktPbNlkIQJWagVZQg==">
  </script>
</environment>
```

□ Create a new folder named `DisplayTemplates` under the `Views\Shared` folder and add a Partial View named `DateTime.cshtml` in the new folder and update the markup to the following:

```
@model DateTime?
@if (Model == null)
{
  @:Unknown
}
else
{
  if (ViewData.ModelMetadata.IsNullableValueType)
  {
    @:@(Model.Value.ToString("d"))
  }
  else
  {
    @:@(((DateTime)Model).ToString("d"))
  }
}
```

Part 4: GDPR Support

Step 1: Add GDPR support to Program.cs

- □ Add the following global using statements to the `GlobalUsings.cs` file:

```
global using Microsoft.AspNetCore.Http.Features;
global using AutoLot.Mvc.Controllers;
global using AutoLot.Services.Utilities;
```

- □ In the top level statements, just after the call to `AddControllersWithView()`, add the following to add GDPR services to the DI container::

```
builder.Services.Configure<CookiePolicyOptions>(options =>
{
  // This lambda determines whether user consent for non-essential cookies is
  // needed for a given request.
  options.CheckConsentNeeded = context => true;
  options.MinimumSameSitePolicy = SameSiteMode.None;
});
// The TempData provider cookie is not essential. Make it essential
// so TempData is functional when tracking is disabled.
builder.Services.Configure<CookieTempDataProviderOptions>(
  options => { options.Cookie.IsEssential = true; });
builder.Services.AddSession(options => { options.Cookie.IsEssential = true; });
```

- □ Add GDPR support to the HTTP pipeline by adding the following line of code before the call to `UseStaticFiles()`:

```
app.UseCookiePolicy();
app.UseStaticFiles();
```

Step 2: Add the _CookieConsentPartial.cshml file

- □ Create a new view file named `_CookieConsentPartial.cshtml` in the `Views\Shared` folder. Update the content to the following:

```
@{
  var consentFeature = Context.Features.Get<ITrackingConsentFeature>();
  var showBanner = !consentFeature?.CanTrack ?? false;
  var cookieString = consentFeature?.CreateConsentCookie();
}

@if (showBanner)
{
  <div id="cookieConsent" class="alert alert-info alert-dismissible fade show" role="alert">
    Use this space to summarize your privacy and cookie use policy. <a asp-area="" asp-
controller="Home" asp-action="Privacy">Learn More</a>.
    <button type="button" class="accept-policy close" data-dismiss="alert" aria-label="Close"
data-cookie-string="@cookieString">
       <span aria-hidden="true">Accept</span>
    </button>
  </div>
  <script>
    (function () {
      var button = document.querySelector("#cookieConsent button[data-cookie-string]");
      button.addEventListener("click", function (event) {
        document.cookie = button.dataset.cookieString;
        window.location =
'@Url.Action(nameof(HomeController.Index),nameof(HomeController).RemoveController())';
      }, false);
    })();
  </script>
}
```

Step 3: Add the Partial to the _Layout

- □ Add the following lines into the `_Layout.cshtml` page:

```
<div class="container">
    <partial name="_CookieConsentPartial"/>
    <main role="main" class="pb-3">
        @RenderBody()
    </main>
</div>
```

Step 4: Update _Menu.cshtml

- □ Add the following two lines into the Razor block at the top of `_Menu.cshtml`:

```
@{
  var dealerInfo = _dealerInfoMonitor.CurrentValue;
  var consentFeature = Context.Features.Get<ITrackingConsentFeature>();
  var showBanner = !consentFeature?.CanTrack ?? false;
}
```

- □ Add the following Razor block after the closing </li> For the Privacy menu item:

```
@if (showBanner)
{
  <li class="nav-item">
    <a class="nav-link text-dark" asp-controller="Home" asp-action="GrantConsent" title="Accept
Cookie Policy">Accept Cookie Policy <i class="fas fa-cookie-bite"></i></a>
  </li>
}
else
{
  <li class="nav-item">
    <a class="nav-link text-dark" asp-controller="Home" asp-action="WithdrawConsent" title="Revoke
Cookie Policy">Revoke Cookie Policy <i class="fas fa-cookie"></i></a>
  </li>
}
```

Step 5: Update the HomeController for GDPR

- □ Add the `GrantConsent` action method:

```
[HttpGet]
public IActionResult GrantConsent()
{
  HttpContext.Features.Get<ITrackingConsentFeature>().GrantConsent();
  return RedirectToAction(nameof(Index), nameof(HomeController).RemoveController(),
    new {area = ""});
}
```

- □ Add the `WithdrawConsent` action method:

```
[HttpGet]
public IActionResult WithdrawConsent()
{
  HttpContext.Features.Get<ITrackingConsentFeature>().WithdrawConsent();
  return RedirectToAction(nameof(Index), nameof(HomeController).RemoveController(),
    new {area = ""});
}
```

Summary

In this lab you finished the added support for client-side libraries, updated the layout, and GDPR support.

Next steps

In the next part of this tutorial series, you will create the Menu View Component and custom Tag Helpers.