

MVC Lists Lab

Lab 1: Create an Ordered List

Open the starting solution given to you by the instructor.

Open the **appsettings.Development.json** file and adjust the connection string as appropriate for your SQL Server.

Right mouse-click on the **Views** folder and create a new folder named **Lists**.

Right mouse-click on the **\Lists** folder and select **Add | View ... | Razor View - Empty**.

Set the name to **Index.cshtml**.

Add the following code

```
@{
    ViewData["Title"] = "List Samples";
}

<h1>List Samples</h1>

<div class="list-group">
    <a asp-action="Sample01" asp-controller="Lists"
        class="list-group-item">
        Sample 1 - Ordered List
    </a>
</div>
```

Add First Sample

Add **Sample01.cshtml** to **\Lists** folder.

```
@using AdvWorks.EntityLayer
@model AdvWorks.ViewModelLayer.ProductViewModel

@{
    ViewData["Title"] = "Sample 1";
}

<h2>Sample 1 - Ordered List</h2>

<ol>
    @foreach(Product item in Model.Products) {
        <li>@item.Name</li>
    }
</ol>
```

Add Controller

Right mouse-click on the Controllers folder and select **Add | Controller... | MVC Controller - Empty**.

Set the name to **ListsController.cs** and add some using statements:

```
using AdvWorks.Common;
using AdvWorks.EntityLayer;
using AdvWorks.ViewModelLayer;
```

Add a constructor and a field

```
public ListsController(IRepository<Product, ProductSearch>
productRepo)
{
    _productRepository = productRepo;
}

private readonly IRepository<Product, ProductSearch>
_productRepository;
```

NOTE: The `IRepository<Product, ProductSearch>` is passed in via dependency injection because it was added as a scoped service in the Program.cs file.

Create **Sample01()** method

```
[HttpGet]
public IActionResult Sample01()
{
    // Create view model passing in repository
    ProductViewModel vm = new(_productRepository);

    // Call method to load products
    vm.Search();

    return View(vm);
}
```

Open the `\Views\Home\Index.cshtml` file and add the following<div> at the end of the file

```
<div class="list-group">
  <a asp-action="Index" asp-controller="Lists"
    class="list-group-item">
    List Samples
  </a>
</div>
```

Try it Out

Run the application and view the ordered list sample

Demo 2: Select Tag Helper

Add a new view named **Sample02.cshtml** to \Lists folder.

Replace the HTML with the following:

```
@model AdvWorks.ViewModelLayer.ProductViewModel

@{
    ViewData["Title"] = "Sample 2";
}

<h2>Sample 2 - Using Select Tag Helper</h2>

<div class="form-group">
    <label for="products">Products</label>
    <select class="form-select"
        asp-for="SelectedProduct.ProductID"
        asp-items="@ (new SelectList (Model.Products, "ProductID",
"Name")) ">
    </select>
</div>
```

Open the **ListsController.cs** file and create a **Sample02()** method

```
[HttpGet]
public IActionResult Sample02()
{
    // Create view model passing in repository
    ProductViewModel vm = new(_productRepository);

    // Call method to load products
    vm.Search();

    return View(vm);
}
```

Open the **\Views\Lists\Index.cshtml** file and add a link within the list-group

```
<a asp-action="Sample02" asp-controller="Lists"
    class="list-group-item">
    Sample 2 - Select Tag Helper
</a>
```

Try it Out

Run the application and view the select tag helper page

Lab 3: Datalist

Add a new view named **Sample03.cshtml** to \Lists folder.

Replace the HTML with the following:

```

@using AdvWorks.EntityLayer
@model AdvWorks.ViewModelLayer.ProductViewModel

@{
    ViewData["Title"] = "Sample 3";
}

<h2>Sample 3 - Data List</h2>

<label for="productList" class="col-form-label">Datalist
Example</label>
<input class="form-control" list="products" id="productList"
placeholder="Type to search">
<datalist id="products">
    @foreach (Product item in Model.Products) {
        <option value="@item.Name"></option>
    }
</datalist>

```

Open the **ListsController.cs** file and create a **Sample03()** method

```

[HttpGet]
public IActionResult Sample03()
{
    // Create view model passing in repository
    ProductViewModel vm = new(_productRepository);

    // Call method to load products
    vm.Search();

    return View(vm);
}

```

Open the **\Views\Lists\Index.cshtml** file and add a link within the list-group

```

<a asp-action="Sample03" asp-controller="Lists"
    class="list-group-item">
    Sample 3 - Data List
</a>

```

Try it Out

Run the application and view the datalist page