Drop Down to Table Lab

Lab 1: Modify Repository and View Model

Open the **ProductRepository.cs** file and add a new method

```
public List<Product> GetByColor(string color)
{
  // Get Products by Color
  return _DbContext.Products.Where(row => row.Color ==
color).ToList();
}
```

Modify Product View Model

Open **ProductViewModel.cs** file and add a new using statement

```
using AdvWorks.DataLayer;
```

Add two new properties

```
public IRepository<Color, ColorSearch> ColorRepository { get; set; }
public List<Color> Colors { get; set; }
```

Add a new constructor

```
public ProductViewModel(IRepository<Product, ProductSearch>
repository, IRepository<Color, ColorSearch> colorRepository) :
base() {
  Repository = repository;
  ColorRepository = colorRepository;
}
```

Add a LoadColors() method

```
#region LoadColors Method
public virtual void LoadColors()
{
  if (ColorRepository == null) {
    throw new ApplicationException("Must set the ColorRepository
property.");
  }
  else {
    Colors = ColorRepository. Search(new ColorSearch()).OrderBy(row
=> row.ColorName).ToList();
  }
}
#endregion
```

Add a new LoadProductsByColor() method

```
#region LoadProductsByColor Method
public virtual void LoadProductsByColor(string color)
{
  IsDetailVisible = false;

  if (Repository == null) {
    throw new ApplicationException("Must set the Repository
property.");
  }
  else {
    Products = ((ProductRepository)Repository)
     .GetByColor(color).OrderBy(row => row.Name).ToList();
  }

  if (Products != null) {
    TotalRows = Products.Count;
  }
}
#endregion
```

Lab 2: Add New View

Right mouse-click on the \Views folder and create a new folder named **ProductsByColor**

Right mouse-click on this new folder and create an empty view named **Index.cshtml** and add the following code:

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

@{
 ViewData["Title"] = "Products by Color";
}

<h1>Products by Color</h1>

         <form method="get" asp-action="ProductsByColor">
           <div class="form-group">
             <label asp-for="SelectedProduct.Color"></label>
             <select class="form-select"
                onchange="this.form.submit();"
                asp-for="SelectedProduct.Color"
                asp-items="@(new SelectList(Model.Colors,
"ColorName", "ColorName"))">
             </select>
           </div>
         </form>

<div class="row mt-3">
 <div class="col">
         @if (Model.Products.Count > 0)
         {
           <table class="table table-bordered table-hover table-
striped">
             <thead>
               <tr>
                 <th>Product Name</th>
                 <th>Product Number</th>
                 <th>Color</th>
                 <th class="text-right">Cost</th>
                 <th class="text-right">Price</th>
               </tr>
             </thead>
             <tbody>
               @foreach (Product item in Model.Products)
               {
                 <tr>
                   <td>@Html.DisplayFor(m => item.Name)</td>
                   <td>@Html.DisplayFor(m => item.ProductNumber)</td>
                   <td>@Html.DisplayFor(m => item.Color)</td>
                   <td class="text-right">@Html.DisplayFor(m =>
item.StandardCost)</td>
                   <td class="text-right">@Html.DisplayFor(m =>
item.ListPrice)</td>
                 </tr>
               }
             </tbody>
           </table>
         }
         else
         {
           <span class="text-danger">No products have this
color.</span>
```

```
        }
  </div>
</div>
```

Add Controller

Right mouse-click on the \Controllers folder and add a new empty controller named **ProductsByColorController.cs**.

```
using AdvWorks.Common;
using AdvWorks.EntityLayer;
using AdvWorks.ViewModelLayer;
using Microsoft.AspNetCore.Mvc;

namespace AdvWorks.Controllers
{
  public class ProductsByColorController : Controller
  {
    public ProductsByColorController(
              ILogger<ProductsByColorController> logger,
              IRepository<Product, ProductSearch> repo,
IRepository<Color, ColorSearch> colorRepo)
    {
      _logger = logger;
      _repo = repo;
      _colorRepo = colorRepo;
    }

    private readonly ILogger<ProductsByColorController> _logger;
    private readonly IRepository<Product, ProductSearch> _repo;
    private IRepository<Color, ColorSearch> _colorRepo;

    [HttpGet]
    public IActionResult Index()
    {
      // Create view model passing in repository
      ProductViewModel vm = new(_repo, _colorRepo);

      vm.LoadColors();
      vm.SelectedProduct.Color = "Black";
      vm.LoadProductsByColor(vm.SelectedProduct.Color);

      return View(vm);
    }

    [HttpGet]
    public IActionResult ProductsByColor(ProductViewModel vm)
    {
      // Pass in repositories
      vm.Repository = _repo;
      vm.ColorRepository = _colorRepo;

      vm.LoadProductsByColor(vm.SelectedProduct.Color);
      vm.LoadColors();

      return View("Index", vm);
    }
  }
}
```

Modify Program

Open the **Program.cs** file and add a new scoped interface

```
// Create a scoped version of Color Repository
builder.Services.AddScoped<IRepository<Color, ColorSearch>,
ColorRepository>();
```

Modify Home Page

Open the **\Views\Home\Index.cshtml** file

Add a new anchor tag

```
<a asp-action="Index"
    asp-controller="ProductsByColor" class="list-group-item">
  Display Products by Color
</a>
```

Try it Out

Run the application