ISC 559 – IS Application Design and Implementation
Assignment 4: Basic Razor and Database Access (10 points)

For this assignment, you will continue to use the event planning project from the Assignment 3. Make sure that you have completed your Entity Framework migration prior to this assignment so that you have a working database in this assignment.

- Access your EventPlanner database by going to View-> SQL Server Object Explorer. Expand your LocalDB server, then expand Databases, and expand the EventPlanner database. You should see a folder named Tables. Expand that, and you should see the tables that were created from your Entity Framework assignment.
- Right-click on the Venues table, and select View Data. This should open the contents of your table, which will be empty. Add the following records to your table (you will not have to provide a value for VenueID since the database will automatically generate an integer for these. You also don't need to supply an address for these venues):
  Name: *Mercedes-Benz Superdome*, MaxCapacity: *70000*
  Name: *Saenger Theater*, MaxCapacity: *5000*
  Name: *Smoothie King Center*, MaxCapacity: *18000*
  Name: *UNO Lakefront Arena*, MaxCapacity: *8000*
  Name: *Tipitina's*, MaxCapacity: *1000*
- On the root of your project, create a folder named Controllers.
- On the root of your project, create a folder named Views.
- Right-click the Views folder, and go to Add-> New Item... In the dialog box, on the left-hand side, select the Web submenu, and select MVC View Start Page from the list. Don't change its default name (which should be _ViewStart.cshtml)
- Right-click the Views folder, and go to Add-> New Item... In the dialog box, on the left-hand side, select the Web submenu, and select MVC View Imports Page from the list. Don't change its default name (which should be _ViewImports.cshtml)
- In the ViewImports.cshtml file, add the following lines of code (don't put semicolons at the end of these lines):
```
@using YourProjectName.Models
@addTagHelper *, Microsoft.AspNetCore.Mvc.TagHelpers
```
- Right-click the Views folder, and go to Add-> New Folder. Name your new folder Shared
- Right click the Views folder, and go to Add-> New Folder. Name your new folder Venues
- Right-click the Shared subfolder under Views, and go to Add -> New Item... In the dialog box, on the left-hand side, select the Web submenu, and select MVC View Layout Page. Don't change its default name (which should be _Layout.cshtml)
- Right-click the Controllers folder, and go to Add -> Controller... From the list, select Empty Controller, and name your controller VenuesController.
- At the top of your VenuesController class, add this line of code to reference your Models folder and your DAL:
```
using YourProjectName.Models;

using YourProjectName.DAL;
```

- Just above the Index method header (public IActionResult Index()), add the following lines of code, which will provide your controller with database access:

```
private EventPlannerContext _context;
public VenuesController(EventPlannerContext context)
{
    _context = context;
}
```

`EventPlannerContext` – be sure the verify the name of your context item

- Modify the body of the Index method so that it reads:
  return View(_context.Venue.ToList());
  This line of code will access all of the Venues from your Venues table, convert them to a List<Venue> object, and pass it to the view named Index located in your Views/Venues folder.
- Right-click the Venues subfolder under Views, and go to Add -> New Item... In the dialog box, on the left-hand side, select the Web submenu, and select MVC View Page. Don't change its default name (which should be Index.cshtml)
- Delete all of the content from the Index.cshtml page. Add the following lines of code, which will allow the page to access the models and accept a List<Venue> object as its data model:

```
@using ProjectName.Models
@model List<Venue>
```

- Next, add the following code, which will set up the loop that will iterate through your List<Venue> object (which you can refer to using the Model property):

```
@foreach (var v in Model)
{

}
```

- Inside the body of the loop, add the following line of code, which will output the name of the venue, followed by a line break, followed by the maximum capacity of the venue:

```
@v.Name<br />
@v.MaxCapacity<br />
<br />
```

Run the project, and navigate to YourRootURL/Venues. You should see a list of the Venues that you entered into your database at the beginning of the assignment.