

ISC 559 – IS Application Design and Implementation

Assignment 5A – Insert, Edit and Delete with MVC and Entity Framework

In this project, we will add views and controller methods that will allow us to add new venues, and edit and delete existing venues within our database.

In the Venues Folder contained within the Views Folder, edit the Index.cshtml file:

- Use HTML tag helpers to display the Name and MaxCapacity for each Venue
`@Html.DisplayFor(vi=>v.Name)
`
`@Html.DisplayFor(vi=>v.MaxCapacity)
`
- Add the code to create an Edit and a Delete hyperlink for each venue listed, sending the VenueID as a parameter to the “Edit” and “Delete” methods that will be created in the Controller.
`<a asp-action="Edit" asp-route-id="@v.VenueID">Edit`
`<a asp-action="Delete" asp-route-id="@v.VenueID">Delete`
- Add the code to create an Add Venue hyperlink at the bottom of the view (outside of the loop)
`<a asp-action="Create">Create New Venue`

In the Venues Controller

- Code a “Create” method that returns a view with no parameters

```
public IActionResult Create()
{
    return View();
}
```

In the Venues Folder within the Views Folder, add a “Create” view (remove existing code)

- Specify the model for the View
`@model Venue`
- Add input controls to the form to allow users to enter the Venue Name and Max Capacity (with asp-tag helpers) into text boxes with corresponding labels. Be sure to include a Submit button.

```
<form asp-action="Create">
    <label asp-for="Name"></label>
    <input asp-for="Name"/>

    <label asp-for="MaxCapacity"></label>
    <input asp-for="MaxCapacity" />

    <input type="submit" value="Create Venue" />
</form>
```
- At the bottom for the view page, add a link to return users to the Venues page (default Index method of the Venues controller)
`<a asp-action="Index">Back to Venues`

In the Venues Controller

- Code a polymorphic “Create” method that will use the Post method to submit data and have a parameter of a Venue.

```
[HttpPost]
public IActionResult Create (Venue venue)
{
}
```

- Within this method, add the new venue to the database through the data context object and then finalize/save the changes
`_context.Add(venue);`
`_context.SaveChanges();`
- Follow the Post-Redirect-Get Pattern to avoid issues with form resubmission or users hitting the back button
`return RedirectToAction("Index");`

Follow the same format and step (and details from the videos) to implement the Edit venues and Delete venues functionality to this project.

Zip your project folder and submit the zipped project for this assignment.