Source Code

Student class

```
namespace WebAppMVCDeploy.Models
{
    public class Student
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string Class { get; set; }
        public int Age { get; set; }
        public string Address { get; set; }
    }
}
```

Teacher class

```
namespace WebAppMVCDeploy.Models
{
    public class Teacher
    {
        public int Id { get; set; }
        public string Name { get; set; }
        public string TeachingSubject { get; set; }
    }
}
```

Subject class

```
namespace WebAppMVCDeploy.Models
{
    public class Subject
    {
        public int Id { get; set; }
        public string Name { get; set; }
    }
}
```

StudentController

```
using Microsoft.AspNetCore.Mvc;
using WebAppMVCDeploy.Models;

namespace WebAppMVCDeploy.Controllers
{
    public class StudentController : Controller
    {
        List<Student> student = new List<Student>()
        {
            new Student { Id = 1, Name = "Sam", Class = "8A", Age = 16, Address = "123SomeWhere"},
            new Student { Id = 2, Name = "Arun", Class = "8B", Age = 17, Address = "321 Some Street"},
            new Student { Id = 3, Name = "Arjun", Class = "8C", Age = 16, Address = "432 some where "},
        };
        public IActionResult Index()
        {
            return View(student);
        }

        [HttpGet]
        public IActionResult Create()
        {
            return View(new Student());
        }

        [HttpPost]
        public IActionResult Create(Student model)
        {
            if(ModelState.IsValid)
            {
                student.Add(model);
                return RedirectToAction("Index");
            }
            else
            {
                return View(model);
            }
        }
    }
}
```

TeacherController

```
using Microsoft.AspNetCore.Mvc;
using WebAppMVCDeploy.Models;

namespace WebAppMVCDeploy.Controllers
{
    public class TeacherController : Controller
    {
        List<Teacher> teacher = new List<Teacher>()
        {
            new Teacher { Id = 1, Name = "Mr.Pysics", TeachingSubject = "Physics"},
            new Teacher { Id = 2, Name = "Mr.Biology", TeachingSubject = "Biology"},
            new Teacher { Id = 3, Name = "Mr.Maths", TeachingSubject = "Maths"},
        };
        public IActionResult Index()
        {
            return View(teacher);
        }

        [HttpGet]
        public IActionResult Create()
        {
            return View(new Teacher());
        }

        [HttpPost]
        public IActionResult Create(Teacher model)
        {
            if (ModelState.IsValid)
            {
                teacher.Add(model);
                return RedirectToAction("Index");
            }
            else
            {
                return View(model);
            }
        }
    }
}
```

SubjectController

```
using Microsoft.AspNetCore.Mvc;
using WebAppMVCDeploy.Models;

namespace WebAppMVCDeploy.Controllers
{
    public class SubjectController : Controller
    {
        List<Subject> subject = new List<Subject>()
        {
            new Subject { Id = 1, Name = "Biology"},
            new Subject {Id = 2, Name = "Physics"},
            new Subject {Id = 3, Name = "Maths"},
        };
        public IActionResult Index()
        {
            return View(subject);
        }

        [HttpGet]
        public IActionResult Create()
        {
            return View(new Subject());
        }

        [HttpPost]
        public IActionResult Create(Subject model)
        {
            if (ModelState.IsValid)
            {
                subject.Add(model);
                return RedirectToAction("Index");
            }
            else
            {
                return View(model);
            }
        }
    }
}
```

View/Student/Create

```
@model WebAppMVCDeploy.Models.Student
```

```
@{
    ViewData["Title"] = "Create";
}

<h1>Create</h1>

<h4>Games</h4>
<hr />
<div class="row">
    <div class="col-md-4">
        <form asp-action="Create">
            <div asp-validation-summary="ModelOnly" class="text-danger"></div>
            <div class="form-group">
                <label asp-for="Id" class="control-label"></label>
                <input asp-for="Id" class="form-control" />
                <span asp-validation-for="Id" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Name" class="control-label"></label>
                <input asp-for="Name" class="form-control" />
                <span asp-validation-for="Name" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Class" class="control-label"></label>
                <input asp-for="Class" class="form-control" />
                <span asp-validation-for="Class" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Age" class="control-label"></label>
                <input asp-for="Age" class="form-control" />
                <span asp-validation-for="Age" class="text-danger"></span>
            </div>
            <div class="form-group">
                <label asp-for="Address" class="control-label"></label>
                <input asp-for="Address" class="form-control" />
                <span asp-validation-for="Address" class="text-danger"></span>
            </div>
            <div class="form-group">
                <input type="submit" value="Create" class="btn btn-primary" />
            </div>
        </form>
    </div>
</div>

<div>
    <a asp-action="Index">Back to List</a>
</div>

@section Scripts {
    @{await Html.RenderPartialAsync("_ValidationScriptsPartial");}
}
```

View/Student/Index

```
@model IEnumerable<WebAppMVCDeploy.Models.Student>

@{
    ViewData["Title"] = "Index";
}

<h1>Index</h1>

<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Id)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Name)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Class)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Age)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Address)
            </th>
            <th></th>
        </tr>
    </thead>
    <tbody>
@foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Id)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Name)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Class)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Age)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Address)
            </td>
        </tr>
}
    </tbody>
</table>
```

Other views are similar to this