Module VI

ASP.Net MVC

Practical 1: MVC Application With Model view controller display information(students info)

Step 1: Model : Right click on model -> Add-> Class

Student.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;


namespace MVC_demo.Models
{
    public class Student
    {
        public int StudentId { get; set; }
        public string StudentName { get; set; }
        public int Age { get; set; }
    }
}
```

StudInfoController.cs

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.UI.WebControls;
using System.Web.UI;
using MVC_demo.Models;

namespace MVC_demo.Controllers
{
    public class StudInfoController : Controller
    {
        // GET: StudInfo
```

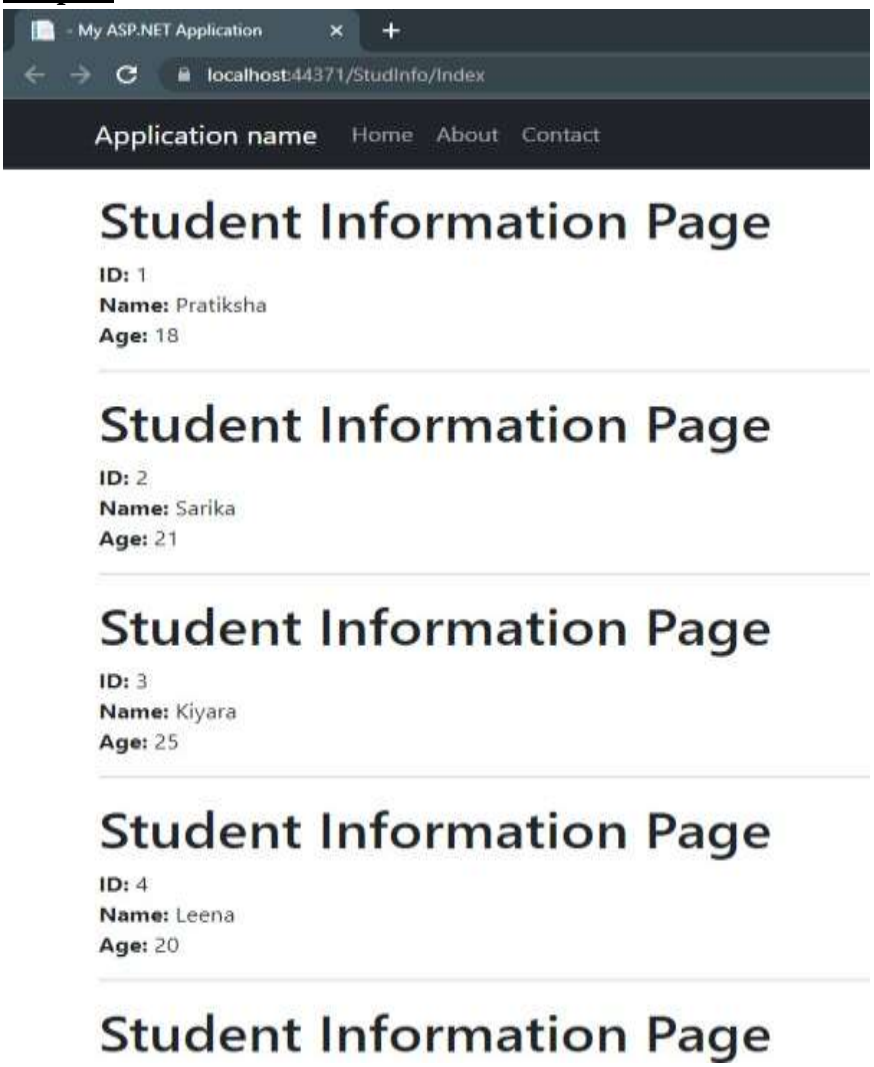
```
        public ActionResult Index()
        {
            ViewBag.ItemList = " Student Information Page ";
            IList < Student > studentList = new List< Student >
            {
                new Student() { StudentId = 1, StudentName =  "Pratiksha ", Age = 18 } ,
                new Student() { StudentId = 2, StudentName = " Sarika ", Age = 21 } ,
                new Student() { StudentId = 3, StudentName = " Kiyara ", Age = 25 } ,
                new Student() { StudentId = 4, StudentName = " Leena ", Age = 20 } ,
                new Student() { StudentId = 5, StudentName = " Richa ", Age = 17 } ,
                new Student() { StudentId = 7, StudentName = " Om ", Age = 19 }
            };
            return View(studentList);
        }
    }
}
```

Index.cshtml

```
@foreach (var i in Model)
{
   <h1>@ViewBag.ItemList</h1>
   <b> ID: </b> @i.StudentId
   <br/>
   <b> Name: </b> @i.StudentName
   <br/>
   <b> Age: </b> @i.Age
   <br/>
   <hr/>
}
```

Run .cshtml file

Output:



ID: 5
Name: Richa
Age: 17

Student Information Page

ID: 7
Name: Om
Age: 19

Practical 2: MVC Application for customer data entry using HTML helper and validation

Add Class File

Customer.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.ComponentModel.DataAnnotations;
namespace MVCCustomer.Models
{
 public class Customer
 {
 [Required(ErrorMessage = "Please Insert Customer ID!")] public int CustID { get; set; }
 [Required(ErrorMessage = "Please Insert Customer Name!")] public String CustName { get; set; }
 [Required(ErrorMessage = "Please Insert Customer Address!")] public String CustAdd { get; set; }
 }
}
```

Add Controller

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using MVCCustomer.Models;
namespace MVCCustomer.Controllers
{
 public class CustomerController : Controller
 {
 // GET: Customer
public ActionResult Index()
 {
 return View();
 }
 [HttpGet]
 public ViewResult CustomerInput()
 {
 return View();
 }
 [HttpPost]
 public ViewResult CustomerInput(Customer c1)
 {
 if (ModelState.IsValid)
```

```
 {
 return View("CustomerDisplay", c1);
 }
 else
 {
 return View();
 }
 }
 }
```

Index.HTML:

```
<html>
<body>
 <div class="text-center">
 <p>
 <h1/>
 <div class="text-center"/>
 <p><h1>Customer Information System </h1><br /> </p> <div class="btn btn-success">
 @Html.ActionLink("Add Customer Details",
"CustomerInput")
 </div>
 </div>
</body>
</html>
```

CustomerInput.html:

```
@model MVCCustomer.Models.Customer
@{
 ViewBag.Title = "CustomerDisplay";
}
@using (Html.BeginForm())
{
 @Html.ValidationSummary()
<div class="form-group">
 <label>Customer ID :</label>@Html.TextBoxFor(x => x.CustID, new { @class = "form-control" })
 </div>
<div class="form-group">
 <label>Customer Name :</label>@Html.TextBoxFor(x =>
x.CustName, new { @class = "form-control " })
 </div><div class="form-group">
 <label>Customer Address:</label>@Html.TextBoxFor(x =>
x.CustAdd, new { @class = "form-control" })
 </div>
<div class="text-center">
 <input class="btn btn-success" type="submit" value="Submit Customer Data" />
```

```
 </div>
}
```

CustomerDisplay.html:

```
@model MVCCustomer.Models.Customer
@{
 ViewBag.Title = "CustomerDisplay";
}
<!DOCTYPE html>
<html>
<head>
 <meta name="viewport" content="width=device-width" />
 <title>CustomerDisplay</title>
</head>
<body>
 <div class="text-center">
 <h1>Customer Information System</h1>
 <p>Customer ID : @Model.CustID</p>
 <p>Customer Name : @Model.CustName</p>
 <p>Customer Address : @Model.CustAdd</p>
 </div>
</body>
</html>
```

Output:

Application name Home About Contact

Customer Information System

Customer ID : 1

Customer Name : pooja

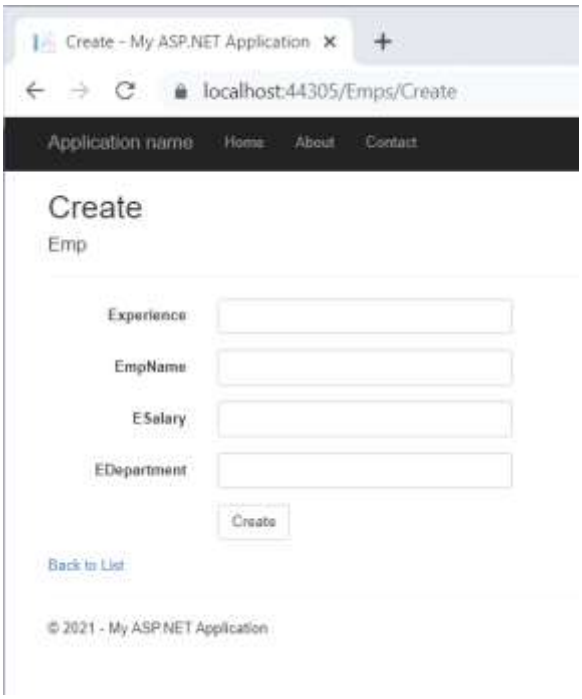
Customer Address : Roadpali

Practical 3: Build MVC Application to from CURD operations using EF.

Emp.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;
namespace WebApplication1.Models
{
 public class Emp
 {
 [Key]
 public virtual int EId { get; set; }
 public virtual int Experience { get; set; }
 [Display(Name = "EmpName")]
 public virtual string EmpName { get; set; }
 public virtual int ESalary { get; set; }
 public virtual string EDepartment { get; set; }
 }
}
```

Output:



Create - My ASP.NET Application

localhost:44305/Emps/Create

Application name Home About Contact

Create

Emp

Experience	1
EmpName	Prashant
ESalary	75000
EDepartment	Full Stack Developer

Create

Back to List

© 2021 - My ASP.NET Application

Index - My ASP.NET Application

localhost:44305/Emps

Application name Home About Contact

Index

Create New

Experience	EmpName	ESalary	EDepartment	
1	Prashant	75000	Full Stack Developer	Edit | Details | Delete

© 2021 - My ASP.NET Application

Practical 4: Build an application Using JQuery.(Table even odd row format,Filter())

Add->project->c#->ASP.net web Application->Web form->then
Right Click->project->ADD->Web form.

.aspx Page

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="oddeven.aspx.cs"
Inherits="jquery.oddeven" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
 <title></title>
</head>
<body>
<form id="form1" runat="server">
<h3>jQuery to cahnge background color of odd even rows</h3>
<script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
<script type="text/javascript">
$(function ()
{
$("table#table1 tr:even").css("background-color", "YELLOW");
$("table#table1 tr:odd").css("background-color", "PINK");
$("table").css("width", "50%");
$("table").attr("border", "true");
});
</script>
<div>
<table id="table1">
<tr><td>Book Number</td>
<td>1001</td></tr>
<tr><td>Isbn</td>
<td>AA</td></tr>
<tr><td>Author</td>
<td>suyash</td></tr>
<tr><td>Publisher</td>
<td>DreamTech</td></tr>
<tr><td>cost</td>
<td>1000</td></tr>
<tr><td>Copies</td>
<td>10</td></tr>
</table>
</div>
</form>
```

```
</body>
</html>
```

Output:

jQuery to cahnge background color of odd even rows

Book Number	1001
Isbn	AA
Author	suyash
Publisher	DreamTech
cost	1000
Copies	10

Practical 5: Build an angular web application.(basic calculator)

.html page

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
 <title></title>
 <script src='https://ajax.googleapis.com/ajax/libs/angularjs/1.5.0-
rc.0/angular.min.js'></script>
 </head>
<body ng-app>
 <h2>Arithmetic Expression In Angularjs</h2> <br />
 <h3>numeric</h3>
 10 + 20 = {{10+20}}
<br /><h3>Subtraction</h3>
 30 - 20 = {{30-20}}
<br /><h3>Multiply</h3>
 10 * 20 = {{10*20}}
 <br /><h3>Division</h3>
 20 / 10 = {{20/10}}
 <br />Enter Numbers for Addition:
 <input type="text" ng-model=Num1 /> + <input type="text" ng-model=Num2 />
 = <span>{{Num1 + Num2}}</span>
 <br />Enter Numbers for Substracion:
 <input type="text" ng-model="Num3" /> - <input type="text" ng-model="Num4" /> =
<span>{{Num3 - Num4}}</span>
 <br />Enter Numbers for Multiplication:
 <input type="text" ng-model="Num5" /> * <input type="text" ng-model="Num6" /> =
<span>{{Num5 * Num6}}</span>
 <br />Enter Numbers for Division:
 <input type="text" ng-model="Num1" /> / <input type="text" ng-model="Num2" /> =
<span>{{Num1 / Num2}}</span>
 </body>
</html>
```

<u>Output:</u>

localhost:61559/HtmlPage2.html

Arithmetic Expression In Angularjs

numeric

10 + 20 = 30

Subtraction

30 - 20 = 10

Multiply

10 * 20 = 200

Division

20 / 10 = 2

Enter Numbers for Addition: 14 + 35 = 1435

Enter Numbers for Substracion: 45 - 7 = 38

Enter Numbers for Multiplication: 2 * 5 = 10

Enter Numbers for Division: 14 / 35 = 0.4