CRUD operations : Create Read Update Delete

On a collection

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace WebApplication15.Models

{

public class Student

{

public int Id { get; set; }

public string Name { get; set; }

public int Marks { get; set; }

public string BatchCode { get; set; }

}

}

Normally, Index Action Method give you list of records

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using WebApplication15.Models;

namespace WebApplication15.Controllers

{

public class StudentController : Controller

{

static List<Student> students = null;

public StudentController()

{

if (students == null)

{

students = new List<Student>()

{

new Student(){ Id=1, Name="Aashna", BatchCode="DotNet", Marks=90},

new Student(){ Id=2, Name="Priynaka", BatchCode="DotNet", Marks=87},

new Student(){ Id=3, Name="Tisha", BatchCode="SAP", Marks=98},

new Student(){ Id=4, Name="Naveen", BatchCode="SAP", Marks=90},

new Student(){ Id=5, Name="Siddhant", BatchCode="DotNet", Marks=90},

new Student(){ Id=6, Name="Vaibhav", BatchCode="DotNet", Marks=90},

};

}

}

public ActionResult Index()

{

if (TempData["msg"] != null)

ViewBag.msg = TempData["msg"];

return View(students);

}

public ActionResult Details(int? id)

{

if (id.HasValue)

{

Student student = students.FirstOrDefault(x => x.Id == id);

if (student == null)

{

ViewBag.msg = "Student with this ID not exist";

return View();

}

else

return View(student);

}

else

{

ViewBag.msg = "Please provide ID";

return View();

}

}

// Create

public ActionResult Create()

{

Student student = new Student();

return View(student);

}

[HttpPost]

public ActionResult Create(Student student)

{

if (student != null)

{

students.Add(student);

}

TempData["msg"] = "Record is inserted";

return RedirectToAction("Index");

}

/// Edit

///

public ActionResult Edit(int? id)

{

if (id.HasValue)

{

Student student = students.FirstOrDefault(x => x.Id == id);

if (student == null)

{

ViewBag.msg = "Student with this ID not exist";

return View();

}

else

return View(student);

}

else

{

ViewBag.msg = "Please provide ID";

return View();

}

}

[HttpPost]

public ActionResult Edit(int id, Student student)

{

foreach(Student temp in students)

{

if(temp.Id == id)

{

temp.Name = student.Name;

temp.BatchCode = student.BatchCode;

temp.Marks = student.Marks;

}

}

TempData["msg"] = "Record is updated";

return RedirectToAction("Index");

}

public ActionResult Delete(int? id)

{

if (id.HasValue)

{

Student student = students.FirstOrDefault(x => x.Id == id);

if (student == null)

{

ViewBag.msg = "Student with this ID not exist";

return View();

}

else

return View(student);

}

else

{

ViewBag.msg = "Please provide ID";

return View();

}

}

[HttpPost]

public ActionResult Delete (int id)

{

Student student = students.FirstOrDefault(x => x.Id == id);

students.Remove(student);

TempData["msg"] = "Record is deleted";

return RedirectToAction("Index");

}

}

}

Index View

@model IEnumerable<WebApplication15.Models.Student>

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

@{ if (ViewBag.msg != null)

{

@ViewBag.msg

}

}

<table>

<tr>

<th> ID </th>

<th> Name </th>

<th> Batch Code </th>

<th> Marks </th>

</tr>

@{

foreach(var temp in Model)

{

<tr>

<td> @temp.Id</td>

<td> @temp.Name </td>

<td> @temp.BatchCode</td>

<td> @temp.Marks</td>

</tr>

}}

</table>

Details View

@model WebApplication15.Models.Student

@{

ViewBag.Title = "Details";

}

<h2>Details</h2>

@{

if (@ViewBag.msg != null)

{

<b> @ViewBag.msg </b>

}

else

{

<h3> ID : @Model.Id <br />

Name : @Model.Name <br />

Batch Code : @Model.BatchCode <br />

Marks : @Model.Marks

</h3>

}

}

Create View

@model WebApplication15.Models.Student

@{

ViewBag.Title = "Create";

}

<h2>Create</h2>

@using (Html.BeginForm(FormMethod.Post))

{

@Html.LabelFor(model => model.Id)

@Html.EditorFor(model => model.Id)

<br />

@Html.LabelFor(model => model.Name)

@Html.EditorFor(model => model.Name)

<br />

@Html.LabelFor(model => model.BatchCode)

@Html.EditorFor(model => model.BatchCode)

<br />

@Html.LabelFor(model => model.Marks)

@Html.EditorFor(model => model.Marks)

<br />

<button type="submit"> Create </button>

}

Edit

@model WebApplication15.Models.Student

@{

ViewBag.Title = "Edit";

}

<h2>Edit</h2>

@{

if (ViewBag.msg != null)

{

<b> @ViewBag.msg </b>

}

else

{

using (Html.BeginForm(FormMethod.Post))

{

@Html.LabelFor(model => model.Id)

@Html.EditorFor(model => model.Id)

<br />

@Html.LabelFor(model => model.Name)

@Html.EditorFor(model => model.Name)

<br />

@Html.LabelFor(model => model.BatchCode)

@Html.EditorFor(model => model.BatchCode)

<br />

@Html.LabelFor(model => model.Marks)

@Html.EditorFor(model => model.Marks)

<br />

<button type="submit"> Edit </button>

}

}

}

Delete View

@model WebApplication15.Models.Student

@{

ViewBag.Title = "Delete";

}

<h2>Delete</h2>

@{

if (ViewBag.msg != null)

{

<b> @ViewBag.msg </b>

}

else

{

using (Html.BeginForm(FormMethod.Post))

{

@Html.LabelFor(model => model.Id)

@Html.DisplayFor(model => model.Id)

<br />

@Html.LabelFor(model => model.Name)

@Html.DisplayFor(model => model.Name)

<br />

@Html.LabelFor(model => model.BatchCode)

@Html.DisplayFor(model => model.BatchCode)

<br />

@Html.LabelFor(model => model.Marks)

@Html.DisplayFor(model => model.Marks)

<br />

<button type="submit"> Delete </button>

}

}

}