Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

\_\_\_12\_\_\_

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| 1 | You have to consider a MVC CRUD operation application for your project and implement it on asp.net webforms. |

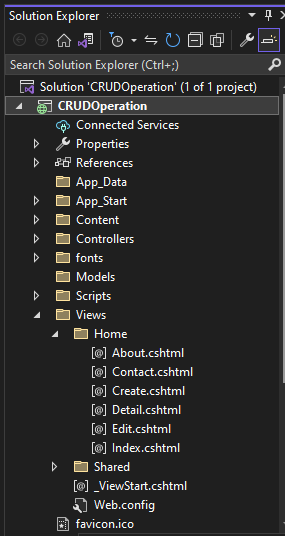
Submitted On:

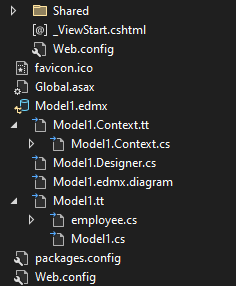
16-01-2022

(Date: DD/MM/YY)

**Task 1:**

**Solution:**

****

****

**Controller Class:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace CRUDOperation.Controllers

{

public class HomeController : Controller

{

employeeEntities \_context = new employeeEntities();

public ActionResult Index()

{

var listofData = \_context.employees.ToList();

return View(listofData);

}

public ActionResult About()

{

ViewBag.Message = "Your application description page.";

return View();

}

public ActionResult Contact()

{

ViewBag.Message = "Your contact page.";

return View();

}

[HttpGet]

public ActionResult Create()

{

return View();

}

[HttpPost]

public ActionResult Create(employee model)

{

\_context.employees.Add(model);

\_context.SaveChanges();

ViewBag.Message = "Data Insert Successfully";

return View();

}

[HttpGet]

public ActionResult Edit(int id)

{

var data = \_context.employees.Where(x => x.EmployeeId == id).FirstOrDefault();

return View(data);

}

[HttpPost]

public ActionResult Edit(employee Model)

{

var data = \_context.employees.Where(x => x.EmployeeId == Model.EmployeeId).FirstOrDefault();

if (data != null)

{

data.EmployeeCity = Model.EmployeeCity;

data.EmployeeName = Model.EmployeeName;

data.EmployeeSalary = Model.EmployeeSalary;

\_context.SaveChanges();

}

return RedirectToAction("index");

}

public ActionResult Detail(int id)

{

var data = \_context.employees.Where(x => x.EmployeeId == id).FirstOrDefault();

return View(data);

}

public ActionResult Delete(int id)

{

var data = \_context.employees.Where(x => x.EmployeeId == id).FirstOrDefault();

\_context.employees.Remove(data);

\_context.SaveChanges();

ViewBag.Messsage = "Record Delete Successfully";

return RedirectToAction("index");

}

}

}

Index.HTML:

@model IEnumerable<CRUDOperation.employee>

@{

ViewBag.Title = "Index";

}

<h2>Index</h2>

<p>

@Html.ActionLink("Create New", "Create")

</p>

<table class="table">

<tr>

<th>

@Html.DisplayNameFor(model => model.EmployeeName)

</th>

<th>

@Html.DisplayNameFor(model => model.EmployeeSalary)

</th>

<th>

@Html.DisplayNameFor(model => model.EmployeeCity)

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@Html.DisplayFor(modelItem => item.EmployeeName)

</td>

<td>

@Html.DisplayFor(modelItem => item.EmployeeSalary)

</td>

<td>

@Html.DisplayFor(modelItem => item.EmployeeCity)

</td>

<td>

@Html.ActionLink("Edit", "Edit", new { id=item.EmployeeId }) |

@Html.ActionLink("Details", "Detail", new { id=item.EmployeeId }) |

@Html.ActionLink("Delete", "Delete", new { id=item.EmployeeId })

</td>

</tr>

}

</table>

**Create.cs Html:**

@model CRUDOperation.employee

@{

ViewBag.Title = "Create";

}

<h2>Create</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>employee</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

<div class="form-group">

@Html.LabelFor(model => model.EmployeeId, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.EmployeeId, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.EmployeeId, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.EmployeeName, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.EmployeeName, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.EmployeeName, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.EmployeeSalary, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.EmployeeSalary, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.EmployeeSalary, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.EmployeeCity, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.EmployeeCity, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.EmployeeCity, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="Create" class="btn btn-default" />

</div>

</div>

</div>

}

<div>

@Html.ActionLink("Back to List", "Index")

</div>

@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

}

**Detail.CSHTML:**

@model CRUDOperation.employee

@{

ViewBag.Title = "Detail";

}

<h2>Detail</h2>

<div>

<h4>employee</h4>

<hr />

<dl class="dl-horizontal">

<dt>

@Html.DisplayNameFor(model => model.EmployeeName)

</dt>

<dd>

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

</dd>

<dt>

@Html.DisplayNameFor(model => model.EmployeeSalary)

</dt>

<dd>

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

</dd>

<dt>

@Html.DisplayNameFor(model => model.EmployeeCity)

</dt>

<dd>

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

</dd>

</dl>

</div>

<p>

@Html.ActionLink("Edit", "Edit", new { id = Model.EmployeeId }) |

@Html.ActionLink("Back to List", "Index")

</p>

**Edit.cshtml:**

@model CRUDOperation.employee

@{

ViewBag.Title = "Edit";

}

<h2>Edit</h2>

@using (Html.BeginForm())

{

@Html.AntiForgeryToken()

<div class="form-horizontal">

<h4>employee</h4>

<hr />

@Html.ValidationSummary(true, "", new { @class = "text-danger" })

@Html.HiddenFor(model => model.EmployeeId)

<div class="form-group">

@Html.LabelFor(model => model.EmployeeName, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.EmployeeName, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.EmployeeName, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.EmployeeSalary, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.EmployeeSalary, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.EmployeeSalary, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

@Html.LabelFor(model => model.EmployeeCity, htmlAttributes: new { @class = "control-label col-md-2" })

<div class="col-md-10">

@Html.EditorFor(model => model.EmployeeCity, new { htmlAttributes = new { @class = "form-control" } })

@Html.ValidationMessageFor(model => model.EmployeeCity, "", new { @class = "text-danger" })

</div>

</div>

<div class="form-group">

<div class="col-md-offset-2 col-md-10">

<input type="submit" value="Save" class="btn btn-default" />

</div>

</div>

</div>

}

<div>

@Html.ActionLink("Back to List", "Index")

</div>

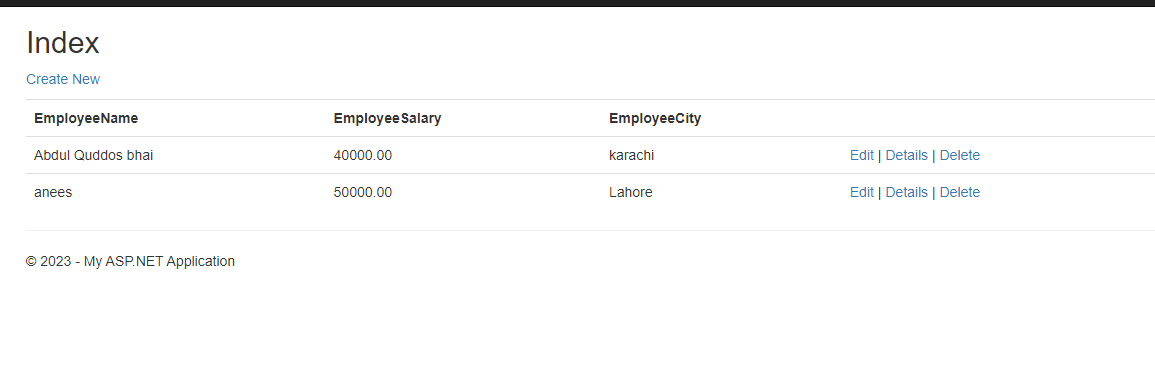
@section Scripts {

@Scripts.Render("~/bundles/jqueryval")

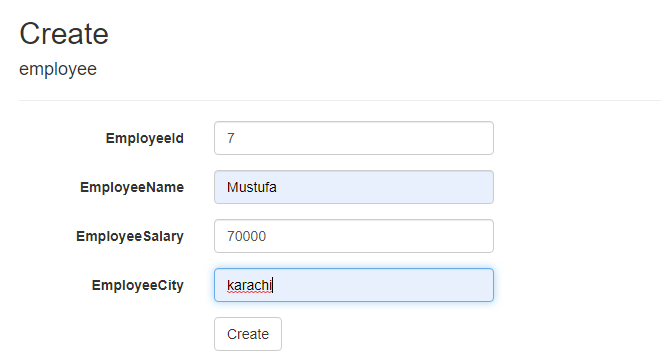
}

**Output:**

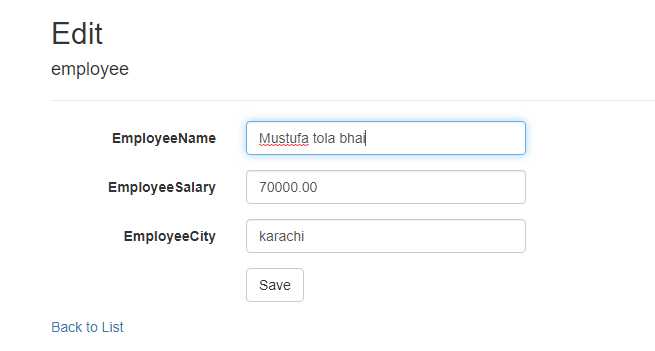
**Index:**



**Create:**

****

**Edit:**

****