Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

\_\_\_10\_\_\_

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| **01** | Create Complete calculator program using MVC. |
| **02** | Implement COR on call center scenario. |
|  |  |
|  |  |

Submitted On

02-06-2023

(Date: DD/MM/YY)

**Task No. 1:** Create Complete calculator program using MVC.

**Solution**:

**Model**

namespace Calculator\_MVC.Models {

public class calc {

public int no1 { get; set; }

public int no2 { get; set; }

public int tot { get; set; } } }

**Controller:**

namespace Calculator\_MVC.Controllers {

public class CalcController : Controller {

public ActionResult Index() {

return View(new calc()); }

[HttpPost]

public ActionResult Index(calc c, string calculate) {

if (calculate == "add") {

c.tot = c.no1 + c.no2; }

else if (calculate == "min") {

c.tot = c.no1 - c.no2; }

else if (calculate == "sub") {

c.tot = c.no1 \* c.no2; }

else if (calculate == "divv") {

c.tot = c.no1 / c.no2; }

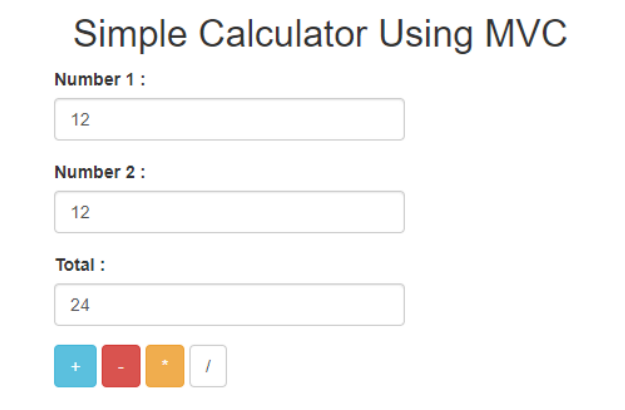
return View(c); } } }

**VIEW:**

@model Calculator\_MVC.Models.calc

@{ ViewBag.Title = "Index";

**Output:**



**Task No. 2:** Create Basic CRUD Application using MVC.

**Solution**:

**Model :**

**A screenshot of a computer

Description automatically generated**

**Controller**

namespace CRUD.Controllers {

public class CustomersController : Controller {

private MvcCrudEntities db = new MvcCrudEntities();

public async Task Index() {

return View(await db.Customers.ToListAsync()); }

public async Task Details(int? id) {

if (id == null) {

return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }

Customer customer = await db.Customers.FindAsync(id);

if (customer == null) {

return HttpNotFound(); }

return View(customer); }

public ActionResult Create() {

return View(); }

ValidateAntiForgeryToken]

public async Task Create([Bind(Include = "Id,Name,Description")] Customer customer) {

if (ModelState.IsValid) {

db.Customers.Add(customer);

await db.SaveChangesAsync();

return RedirectToAction("Index"); }

return View(customer); }

public async Task Edit(int? id) {

if (id == null) {

return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }

Customer customer = await db.Customers.FindAsync(id);

if (customer == null) {

return HttpNotFound(); }

return View(customer); }

ValidateAntiForgeryToken]

public async Task Edit([Bind(Include = "Id,Name,Description")]

Customer customer) {

if (ModelState.IsValid) {

db.Entry(customer).State = EntityState.Modified; await db.SaveChangesAsync();

return RedirectToAction("Index"); }

return View(customer); }

public async Task Delete(int? id) { if (id == null) {

return new HttpStatusCodeResult(HttpStatusCode.BadRequest); }

Customer customer = await db.Customers.FindAsync(id);

if (customer == null) {

return HttpNotFound(); }

return View(customer); }

Customers/Delete/5 [HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public async Task DeleteConfirmed(int id) {

Customer customer = await db.Customers.FindAsync(id);

db.Customers.Remove(customer); await db.SaveChangesAsync();

return RedirectToAction("Index"); }

protected override void Dispose(bool disposing) {

if (disposing) {

db.Dispose(); }

base.Dispose(disposing); } } }

CREATE: @model CRUD.Models.Customer @{ ViewBag.Title = "Create"; }

## **Create**

@using (Html.BeginForm()) { @Html.AntiForgeryToken()

#### **Customer**

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

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

@Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "formcontrol" } }) @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })

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

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

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

@section Scripts { @Scripts.Render("~/bundles/jqueryval") } EDIT: @model CRUD.Models.Customer @{ ViewBag.Title = "Edit"; }

## **Edit**

@using (Html.BeginForm()) { @Html.AntiForgeryToken()

#### **Customer**

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

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

@Html.EditorFor(model => model.Name, new { htmlAttributes = new { @class = "formcontrol" } }) @Html.ValidationMessageFor(model => model.Name, "", new { @class = "text-danger" })

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

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

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

@section Scripts { @Scripts.Render("~/bundles/jqueryval")} DELETE: @model CRUD.Models.Customer @{ ViewBag.Title = "Delete"; }

## **Delete**

### Are you sure you want to delete this?

#### **Customer**

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

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

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

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

@using (Html.BeginForm()) { @Html.AntiForgeryToken()

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

}

**DETAILS**:

@model CRUD.Models.Customer @{ ViewBag.Title = "Details"; [Lab no.10] [Software Design & Architecture Lab] [ INTERACTIVE SYSTEMS WITH MVC ARCHITECTURAL PATTERN ] }

## **Details**

#### **Customer**

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

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

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

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

@Html.ActionLink("Edit", "Edit", new { id = Model.Id }) | @Html.ActionLink("Back to List", "Index")

INDEX: @model IEnumerable @{ ViewBag.Title = "Index"; }

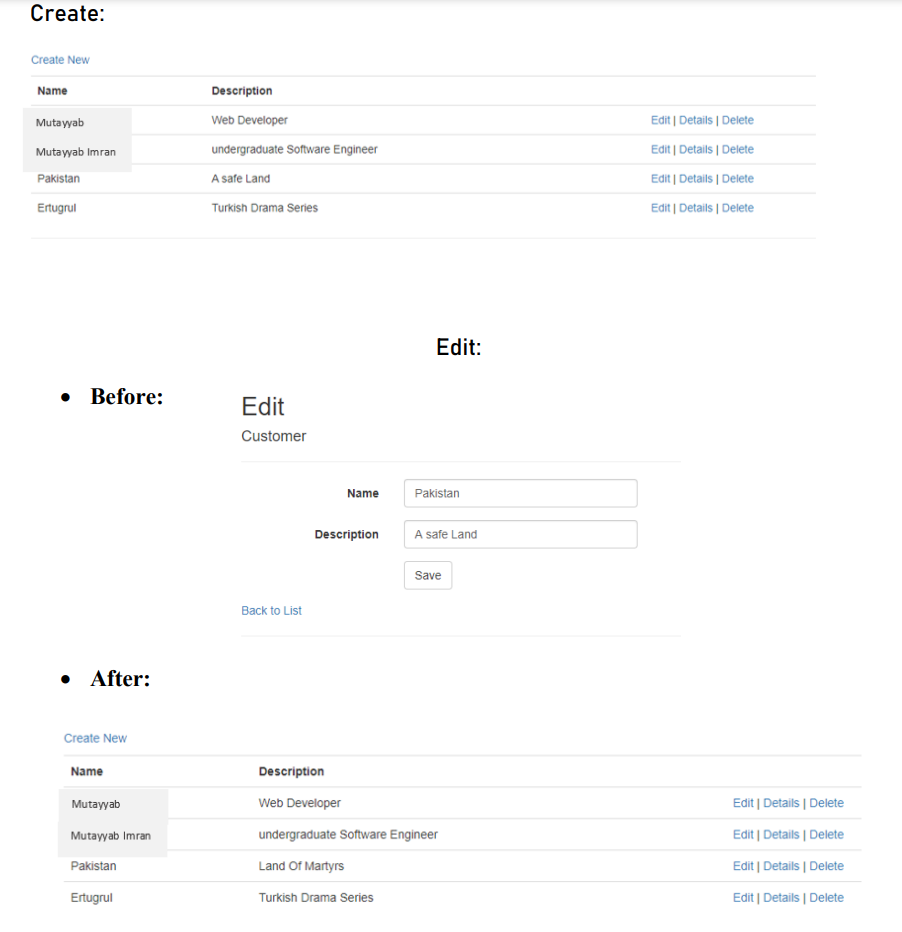
## **Index**

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

|  |  |  |
| --- | --- | --- |
| **@Html.DisplayNameFor(model => model.Name)** | **@Html.DisplayNameFor(model => model.Description)** | @foreach (var item in Model) { |
| @Html.DisplayFor(modelItem => item.Name) | @Html.DisplayFor(modelItem => item.Description) | [Lab no.10] [Software Design & Architecture Lab] [ INTERACTIVE SYSTEMS WITH MVC ARCHITECTURAL PATTERN ]  @Html.ActionLink("Edit", "Edit", new { id=item.Id }) | @Html.ActionLink("Details", "Details", new { id=item.Id }) | @Html.ActionLink("Delete", "Delete", new { id=item.Id })  } |

**Output**:

**A screenshot of a computer

Description automatically generated with medium confidence**