**EMPLOYEE**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace ASP\_WebApp\_Inline\_Editing.Pages

{

public class Employee

{

public int EmpID { set; get; }

public string EmpName { set; get; }

public string Designation { set; get; }

public DateTime DOJ { set; get; }

public double Salary { set; get; }

public int DeptId { set; get; }

public string Operation { set; get; }

}

}

**DATABASEOPERATION**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Data;

using System.Data.SqlClient;

namespace ASP\_WebApp\_Inline\_Editing.Pages

{

public class DatabaseOperations

{

SqlConnection Con = new SqlConnection(@"Server=DESKTOP-BS8T9P1\SQL2019;User Id=sa;Password=Hari@123;Database=EmployeeDB1");

SqlCommand Cmd;

SqlDataReader DR;

public int AddEmployee(Employee Emp)

{

string strquery = "Insert into EmpDetails values(@P1,@P2,@P3,@P4,@P5,@P6)";

Cmd = new SqlCommand(strquery, Con);

Cmd.CommandType = CommandType.Text;

Cmd.Parameters.AddWithValue("@P1", Emp.EmpID);

Cmd.Parameters.AddWithValue("@P2", Emp.EmpName);

Cmd.Parameters.AddWithValue("@P3", Emp.Designation);

Cmd.Parameters.AddWithValue("@P4", Emp.DOJ);

Cmd.Parameters.AddWithValue("@P5", Emp.Salary);

Cmd.Parameters.AddWithValue("@P6", Emp.DeptId);

Con.Open();

int i = Cmd.ExecuteNonQuery();

Con.Close();

return i; ;

}

public int DeleteEmployee(int id)

{

string strquery = "Delete EmpDetails Where EmpID=@P1";

Cmd = new SqlCommand(strquery, Con);

Cmd.CommandType = CommandType.Text;

Cmd.Parameters.AddWithValue("@P1", id);

Con.Open();

int i = Cmd.ExecuteNonQuery();

Con.Close();

return i;

}

public int UpdateEmployee(Employee Emp)

{

string strquery = "Update EmpDetails Set EmpName=@P1,Designation=@P2,DOJ=@P3,Salary=@P4,DeptId=@P5 where EmpID=@P6";

Cmd = new SqlCommand(strquery, Con);

Cmd.CommandType = CommandType.Text;

Cmd.Parameters.AddWithValue("@P1", Emp.EmpName);

Cmd.Parameters.AddWithValue("@P2", Emp.Designation);

Cmd.Parameters.AddWithValue("@P3", Emp.DOJ);

Cmd.Parameters.AddWithValue("@P4", Emp.Salary);

Cmd.Parameters.AddWithValue("@P5", Emp.DeptId);

Cmd.Parameters.AddWithValue("@P6", Emp.EmpID);

Con.Open();

int i = Cmd.ExecuteNonQuery();

Con.Close();

return i;

}

//public Employee GetEmployee(int id)

//{

// Employee getEmp = null;

// string strquery = "select \* from EmpDetails where EmpID=@P1";

// Cmd = new SqlCommand(strquery, Con);

// Cmd.CommandType = CommandType.Text;

// Cmd.Parameters.AddWithValue("@P1", id);

// Con.Open();

// DR = Cmd.ExecuteReader();

// if (DR.Read())

// {

// getEmp = new Employee();

// getEmp.EmpID = Convert.ToInt32(DR[0]);

// getEmp.EmpName = DR[1].ToString();

// getEmp.Designation = DR[2].ToString();

// getEmp.DOJ = Convert.ToDateTime(DR[3]);

// getEmp.Salary = Convert.ToDouble(DR[4]);

// getEmp.DeptId=Convert.ToInt32(DR[5]);

// }

// Con.Close();

// return getEmp;

//}

public IEnumerable<Employee> GetAllEmployee()

{

List<Employee> emplist = new List<Employee>();

string strquery = "Select \* from EmpDetails";

Cmd = new SqlCommand(strquery, Con);

Cmd.CommandType = CommandType.Text;

Con.Open();

DR = Cmd.ExecuteReader();

while (DR.Read())

{

Employee emp = new Employee();

emp.EmpID = Convert.ToInt32(DR[0]);

emp.EmpName = DR[1].ToString();

emp.Designation = DR[2].ToString();

emp.DOJ = Convert.ToDateTime(DR[3]);

emp.Salary = Convert.ToDouble(DR[4]);

emp.DeptId = Convert.ToInt32(DR[5]);

emplist.Add(emp);

}

Con.Close();

return emplist;

}

}

}

**INLINEEDITING**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

namespace ASP\_WebApp\_Inline\_Editing.Pages

{

public class InlineEditingModel : PageModel

{

[BindProperty]

public Employee Emp { set; get; }

[BindProperty]

public List<Employee> EmpList { set; get; }

[BindProperty]

public string Operation { set; get; }

public int Editid;

DatabaseOperations DBOPS = new DatabaseOperations();

public void OnGet()

{

EmpList = DBOPS.GetAllEmployee().ToList();

}

public void OnPost(int Eid)

{

if(Operation == "Edit")

{

Editid = Eid;

}

else if(Operation == "Update")

{

DBOPS.UpdateEmployee(Emp);

}

else if(Operation == "Delete")

{

DBOPS.DeleteEmployee(Eid);

}

else

{

EmpList = DBOPS.GetAllEmployee().ToList();

}

}

}

}

**INLINEEDITING.CSHTML**

@page

@model ASP\_WebApp\_Inline\_Editing.Pages.InlineEditingModel

@{

}

<div>

<h4>Employee Details</h4>

<b><a **asp-page**="./Index">Back To Main</a></b> <br />

<form method="post">

<b><a **asp-page**="./AddEmployee" class="mr-sm-auto"> + New Add </a></b>

<table align="center" cellpadding="5" border="1">

<tr>

<th>EmpID</th>

<th>EmpName</th>

<th>Designation</th>

<th>DOJ</th>

<th>Salary</th>

<th>DeptId</th>

</tr>

@foreach (var Emp in Model.EmpList)

{

<tr>

@if (Model.Operation == "Edit" && Emp.EmpID==Model.Editid)

{

<td>@Html.TextBoxFor(Model => Emp.EmpID)</td>

<td>@Html.TextBoxFor(Model => Emp.EmpName)</td>

<td>@Html.TextBoxFor(Model => Emp.Designation)</td>

<td>@Html.TextBoxFor(Model => Emp.DOJ)</td>

<td>@Html.TextBoxFor(Model => Emp.Salary)</td>

<td>@Html.TextBoxFor(Model => Emp.DeptId)</td>

<td><input type="submit" name="Operation" value="Update" **asp-route-Eid**="@Emp.EmpID" /></td>

<td><input type="submit" name="Operation" value="Cancel" **asp-route-Eid**="@Emp.EmpID" /></td>

}

else

{

<td>@Html.DisplayFor(Model => Emp.EmpID)</td>

<td>@Html.DisplayFor(Model => Emp.EmpName)</td>

<td>@Html.DisplayFor(Model => Emp.Designation)</td>

<td>@Html.DisplayFor(Model => Emp.DOJ)</td>

<td>@Html.DisplayFor(Model => Emp.Salary)</td>

<td>@Html.DisplayFor(Model => Emp.DeptId)</td>

<td><input type="submit" name="Operation" value="Edit" **asp-route-Eid**="@Emp.EmpID" /></td>

<td><input type="submit" name="Operation" value="Delete" **asp-route-Eid**="@Emp.EmpID" /></td>

}

</tr>

}

</table>

</form>

</div>

**AddEmployeeModel**

@page

@model ASP\_WebApp\_Inline\_Editing.Pages.AddEmployeeModel

@{

}

<div class="text-center">

<h4> Add New Employee Fill Below Details</h4>

<p>@Model.Massage</p>

@using (Html.BeginForm())

{

<table align="center" cellspacing="5" border="1">

<tr>

<th>Enter EmpID</th>

<td>@Html.TextBoxFor(Model=> Model.Emp.EmpID)</td>

</tr>

<tr>

<th>Enter EmpName</th>

<td>@Html.TextBoxFor(Model=> Model.Emp.EmpName)</td>

</tr>

<tr>

<th>Enter Designation</th>

<td>@Html.TextBoxFor(Model=> Model.Emp.Designation)</td>

</tr>

<tr>

<th>Enter DOJ</th>

<td>@Html.TextBoxFor(Model=> Model.Emp.DOJ)</td>

</tr>

<tr>

<th>Enter Salary</th>

<td>@Html.TextBoxFor(Model=> Model.Emp.Salary)</td>

</tr>

<tr>

<th>Enter DeptId</th>

<td>@Html.TextBoxFor(Model=> Model.Emp.DeptId)</td>

</tr>

<tr>

<th></th>

<td><input type="Submit" name="Operation" value="Insert" /></td>

</tr>

</table>

}

<b><a **asp-page**="./InlineEditing"><=Back</a></b>

</div>

**AddEmployee**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.RazorPages;

namespace ASP\_WebApp\_Inline\_Editing.Pages

{

public class AddEmployeeModel : PageModel

{

[BindProperty]

public Employee Emp {set; get;}

public string Massage;

DatabaseOperations DBOPS = new DatabaseOperations();

public void OnGet()

{

}

public void OnPost()

{

DBOPS.AddEmployee(Emp);

Massage = "Record Inserted SUCCESSFULLY....!";

}

}

}