**Entity Framework**

it is an open source ORM(Object Relational mapper) Framework developed by MicroSoft .EntityFramework is also used internal ADO.Net .

Object Relational Mapping framework automatically creates classes based on database tables. It can also automatically generate necessary SQL to create database tables based on classes

=>How we can performs CRUD Operation By Using EF .

-----------------------------------------------------

1)Features of EF :-

1) **Configuration**

-----------------------

Any kind of configuration is not required, automatic it establish the connection string with database

2) **Command**

---------------

Command is not required

3) **Object oriented approach**

-------------------------------------

Every table or data base represent in the form of class.

Important method for CRUD operation

------------------------------------

1. for insertion

Add(above 4.0) or AddObject (below 4.0)

2. For save the record

SaveChanges(Object of Table)

**Important Steps of storing record in database using EFWORK**

1. Entity Frame model creation save and rebuild
2. Add Controller
3. Using name space <ProjectName>.models
4. Create the object

ENTITYDATABASEEntities db = new ENTITYDATABASEEntities();

1. Create post index method with parameter(TBL\_emp emp)
2. db.TBL\_EMP.Add(emp);
3. db.SaveChanges(); //commit records permanent storage
4. response.redirect(“/home/index”); without refresh record will be display on the screen.

**Display Record using Dynamic table**

1. if you want to display records on view of html
2. Bind table records with List

List<TBL\_EMP> lst = db.TBL\_EMP.ToList();

1. Create a table header

string table = "<table border='1' cellpadding='10' cellspacing='10' style='color:red'><tr><th>Employee Id</th><th>Employee Name</th><th>Employee City</th><th>Employee Age</th><th>Employee Salary</tr>";

1. Execute a Loop
2. for (int i = 0; i < lst.Count; i++)
3. table+="<tr><td>"+lst[i].EMP\_ID+"</td><td>"+lst[i].EMP\_NAME+"</td><td>"+lst[i].EMP\_CITY+"</td><td>"+lst[i].EMP\_AGE+"</td><td>"+lst[i].EMP\_SALARY+"</td></tr>";
4. Store dynamic table in ViewBag

ViewBag.tbl = table;

1. Now use Razor Block in View

**@{**

**if(ViewBag.tbl!="")**

**{**

**@Html.Raw(ViewBag.tbl);**

**}**

**}**

* **Deletion**

public ActionResult Delete()

{

int id = int.Parse(Request.QueryString["empid"].ToString());

//var del = new TBL\_EMP { EMP\_ID = id };

// db.TBL\_EMP.Attach(del);

TBL\_EMP emp = db.TBL\_EMP.SingleOrDefault(a => a.EMP\_ID == id);

db.TBL\_EMP.Remove(emp);

db.SaveChanges();

Response.Write("<script>alert('Record deleted ')</script>");

Response.Redirect("/Home/Index");

return View();

}

**Index Coading**

public ActionResult Index()

{

List<TBL\_EMP> Lst = db.TBL\_EMP.ToList();

string table = "<table><tr style='background:orange;color:white;font-size:20px;text-align:center;font-weight:bold;'><th>Emplyee Id</th><th>Emplyee Name</th><th>Emplyee City</th><th>Emplyee Age</th><th>Salary</th><th>Delete</th><th>Update</th><th>Details</th></tr>";

for (int i = 0; i < Lst.Count; i++)

{

table += "<tr style='background:red;color:white;font-size:20px;text-align:center;font-weight:bold;'><td>" + Lst[i].EMP\_ID + "</td><td>" + Lst[i].EMP\_NAME + "</td><td>" + Lst[i].EMP\_CITY + "</td><td>" + Lst[i].EMP\_AGE + "</td><td>" + Lst[i].EMP\_SALARY + "</td><td><a href='/Home/Delete?empid=" + Lst[i].EMP\_ID + "'>Delete</a></td><td><a href='#'>Update</a></td><td><a href='/Home/Details?dt="+Lst[i].EMP\_ID+"'>Details</a></td></tr>";

}

table += "</table>";

ViewBag.tab = table;

return View();

}

* **Details**

public ActionResult Details()

{

int id = int.Parse(Request.QueryString["dt"].ToString());

TBL\_EMP emp = db.TBL\_EMP.SingleOrDefault(a => a.EMP\_ID == id);

return View(emp);

}

**Veiw of Details**

@{

Layout = null;

}

<!DOCTYPE html>

@using ENTITYFRAMEWORKDEMO.Models

@model TBL\_EMP

<html>

<head>

<meta name="viewport" content="width=device-width" />

<title>Details</title>

</head>

<body>

<div>

<table>

<tr>

<td>Employee Name</td>

<td>

<input type="text" value="@Model.EMP\_NAME" readonly="true"/>

</td>

<td>Emplyee City</td>

<td>

<input type="text" value="@Model.EMP\_CITY" />

</td>

<td>Employee Age</td>

<td>

<input type="text" value="@Model.EMP\_AGE" />

</td>

<td>Salary</td>

<td>

<input type="text" value="@Model.EMP\_SALARY" />

</td>

</tr>

</table>

</div>

</body>

</html>

**Entity Frame work**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using ENTITYFRAMEWORKDEMO.Models;

namespace ENTITYFRAMEWORKDEMO.Controllers

{

public class HomeController : Controller

{

//

// GET: /Home/

//create DataBase Object as Global

ENTITYDATABASEEntities db = new ENTITYDATABASEEntities();

public ActionResult Index()

{

List<TBL\_EMP> Lst = db.TBL\_EMP.ToList();

string table = "<table><tr style='background:orange;color:white;font-size:20px;text-align:center;font-weight:bold;'><th>Emplyee Id</th><th>Emplyee Name</th><th>Emplyee City</th><th>Emplyee Age</th><th>Salary</th><th>Delete</th><th>Update</th><th>Details</th></tr>";

for (int i = 0; i < Lst.Count; i++)

{

table += "<tr style='background:red;color:white;font-size:20px;text-align:center;font-weight:bold;'><td>" + Lst[i].EMP\_ID + "</td><td>" + Lst[i].EMP\_NAME + "</td><td>" + Lst[i].EMP\_CITY + "</td><td>" + Lst[i].EMP\_AGE + "</td><td>" + Lst[i].EMP\_SALARY + "</td><td><a href='/Home/Delete?empid="+Lst[i].EMP\_ID+"'>Delete</a></td><td><a href='#'>Update</a></td><td><a href='#'>Details</a></td></tr>";

}

table += "</table>";

ViewBag.tab = table;

return View();

}

[HttpPost]

public ActionResult Index(TBL\_EMP emp)

{

db.TBL\_EMP.Add(emp);

db.SaveChanges();

Response.Write("<script>alert('Records saved into database ');</script>");

Response.Redirect("/Home/Index");

return View();

}

public ActionResult Delete()

{

int id = int.Parse(Request.QueryString["empid"].ToString());

var del = new TBL\_EMP { EMP\_ID = id };

db.TBL\_EMP.Attach(del);

db.TBL\_EMP.Remove(del);

db.SaveChanges();

Response.Write("<script>alert('Record deleted ')</script>");

Response.Redirect("/Home/Index");

return View();

}

}

}

**View code**

@{

if(ViewBag.tab!="")

{

@Html.Raw(ViewBag.tab )

}

}

<col width='130'><col width='130'>