**Model**

**Class Product**

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace ExamProducts.Models

{

public class Product

{

[Key]

[Display(Name = "Product Id")]

public int ProductId { get; set; }

[DataType(DataType.Text)]

[Required(ErrorMessage ="Please enter Product name..")]

[StringLength(50, ErrorMessage = "The {0} value cant be greater than {1} characters.")]

[Display(Name = "Product Name")]

public string ProductName { get; set; }

[DataType(DataType.Currency)]

[Required(ErrorMessage = "Please enter Rate..")]

[Display(Name = "Rate")]

public decimal Rate { get; set; }

[DataType(DataType.Text)]

[Required(ErrorMessage = "Please enter Product name..")]

[StringLength(200, ErrorMessage = "The {0} value cant be greater than {1} characters.")]

[Display(Name = "Description")]

public string Description { set; get; }

[DataType(DataType.Text)]

[Required(ErrorMessage = "Please enter Product name..")]

[StringLength(50, ErrorMessage = "The {0} value cant be greater than {1} characters.")]

[Display(Name = "Category Name")]

public string CategoryName { set; get; }

}

}

Table

CREATE TABLE [dbo].[Products] (

[ProductId] INT IDENTITY (1, 1) NOT NULL,

[ProductName] VARCHAR (50) NOT NULL,

[Rate] DECIMAL (18, 2) NOT NULL,

[Description] VARCHAR (200) NOT NULL,

[CategoryName] VARCHAR (50) NOT NULL,

PRIMARY KEY CLUSTERED ([ProductId] ASC)

);

Stored Procdedures

CREATE PROCEDURE [dbo].[GetProduct]

@ProductId int

AS

SELECT \* from Products where ProductId = @ProductId;

RETURN 0

CREATE PROCEDURE [dbo].[GetProductList]

AS

SELECT \* from Products ;

RETURN 0

CREATE PROCEDURE [dbo].[UpdateProduct]

@ProductId int,

@ProductName varchar(50),

@Rate decimal (18,2),

@Description varchar(200),

@CategoryName varchar(50)

AS

update Products set ProductName=@ProductName, Rate= @Rate, Description= @Description, CategoryName=@CategoryName where ProductId=@ProductId;

RETURN 0

Controller

using ExamProducts.Models;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace ExamProducts.Controllers

{

public class ProductsController : Controller

{

// GET: Products

public ActionResult Index()

{

List<Product> list = new List<Product>();

SqlConnection cn = new SqlConnection();

cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=Exam;Integrated Security=True";

cn.Open();

SqlCommand cmd = new SqlCommand();

cmd.Connection = cn;

cmd.CommandType = System.Data.CommandType.StoredProcedure;

cmd.CommandText = "GetProductList";

try

{

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

list.Add(new Product { ProductId = (int)dr["ProductId"], ProductName = (string)dr["ProductName"], Rate = (decimal)dr["Rate"], Description = (string)dr["Description"], CategoryName = (string)dr["CategoryName"] });

}

dr.Close();

}

catch(Exception ex)

{

}

finally

{

cn.Close();

}

return View(list);

}

// GET: Products/Details/5

public ActionResult Details(int id)

{

return View();

}

// GET: Products/Edit/5

public ActionResult Edit(int id)

{

Product obj = null;

SqlConnection cn = new SqlConnection();

cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=Exam;Integrated Security=True";

cn.Open();

SqlCommand cmd = new SqlCommand();

cmd.Connection = cn;

cmd.CommandType = System.Data.CommandType.StoredProcedure;

cmd.CommandText = "GetProduct";

cmd.Parameters.AddWithValue("ProductId", id);

try

{

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

obj = new Product { ProductId = (int)dr["ProductId"], ProductName = (string)dr["ProductName"], Rate = (decimal)dr["Rate"], Description = (string)dr["Description"], CategoryName = (string)dr["CategoryName"] };

}

dr.Close();

}

catch (Exception ex)

{

}

finally

{

cn.Close();

}

return View(obj);

}

// POST: Products/Edit/5

[HttpPost]

public ActionResult Edit(int id, Product obj)

{

SqlConnection cn = new SqlConnection();

cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=Exam;Integrated Security=True";

cn.Open();

SqlCommand cmd = new SqlCommand();

cmd.Connection = cn;

cmd.CommandType = System.Data.CommandType.StoredProcedure;

cmd.CommandText = "UpdateProduct";

cmd.Parameters.AddWithValue("ProductId", id);

cmd.Parameters.AddWithValue("ProductName",obj.ProductName);

cmd.Parameters.AddWithValue("Rate", obj.Rate);

cmd.Parameters.AddWithValue("Description", obj.Description);

cmd.Parameters.AddWithValue("CategoryName", obj.CategoryName);

try

{

cmd.ExecuteNonQuery();

return RedirectToAction("Index");

}

catch

{

return View();

}

}

[ChildActionOnly]

public ActionResult MyPView()

{

return View();

}

}

}

Product Views

Index

@model IEnumerable<ExamProducts.Models.Product>

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/MyLayout.cshtml";

}

<h2>Index</h2>

<p>

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

</p>

<table class="table">

<tr>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@\*@Html.DisplayFor(modelItem => item.ProductName)\*@

@Html.ActionLink(item.ProductId.ToString(), "Edit", new { id = item.ProductId })

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

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

@Html.ActionLink("Delete", "Delete", new { id=item.ProductId })\*@

</td>

</tr>

}

</table>

@Html.Partial("MyPView")

Edit

@model IEnumerable<ExamProducts.Models.Product>

@{

ViewBag.Title = "Index";

Layout = "~/Views/Shared/MyLayout.cshtml";

}

<h2>Index</h2>

<p>

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

</p>

<table class="table">

<tr>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th>

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

</th>

<th></th>

</tr>

@foreach (var item in Model) {

<tr>

<td>

@\*@Html.DisplayFor(modelItem => item.ProductName)\*@

@Html.ActionLink(item.ProductId.ToString(), "Edit", new { id = item.ProductId })

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

</td>

<td>

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

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

@Html.ActionLink("Delete", "Delete", new { id=item.ProductId })\*@

</td>

</tr>

}

</table>

@Html.Partial("MyPView")

Partial View

<div>

<div>

<h3>Created by : Shridhar Salunkhe.</h3>

</div>

<div>

<h3>Center : JUHU</h3>

</div>

<div>

<h3>RollNo : 210940520091 </h3>

</div>

</div>

LayOut

@{

Layout = null;

}

<!DOCTYPE html>

<html>

<head>

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

<title>MyLayout</title>

</head>

<body>

<div>

<h1> Product App</h1>

</div>

@RenderBody()

@RenderSection("scripts", required: false)

</body>

</html>