using Products.Models;

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace Products.Controllers

{

public class ProductController : Controller

{

// GET: Product

public ActionResult Index()

{

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

SqlConnection cn = new SqlConnection();

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

cn.Open();

SqlCommand selectCmd = new SqlCommand();

selectCmd.Connection = cn;

selectCmd.CommandType = System.Data.CommandType.Text;

selectCmd.CommandText = "select \* from Products";

SqlDataReader dr = selectCmd.ExecuteReader();

while (dr.Read())

{

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

}

dr.Close();

cn.Close();

return View(list);

}

// GET: Product/Details/5

public ActionResult Details(int id)

{

return View();

}

// GET: Product/Create

public ActionResult Create()

{

return View();

}

// POST: Product/Create

[HttpPost]

public ActionResult Create(Product objProduct)

{

SqlConnection cn = new SqlConnection();

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

cn.Open();

SqlCommand insertCmd = new SqlCommand();

insertCmd.Connection = cn;

insertCmd.CommandType = System.Data.CommandType.StoredProcedure;

insertCmd.CommandText = "InsertProduct";

insertCmd.Parameters.AddWithValue("@ProductName", objProduct.ProductName);

insertCmd.Parameters.AddWithValue("@Rate", objProduct.Rate);

insertCmd.Parameters.AddWithValue("@Description", objProduct.Description);

insertCmd.Parameters.AddWithValue("@CategoryName", objProduct.CategoryName);

try

{

// TODO: Add insert logic here

insertCmd.ExecuteNonQuery();

return RedirectToAction("Index");

}

catch(Exception e)

{

return View();

}

finally

{

cn.Close();

}

}

// GET: Product/Edit/5

public ActionResult Edit(int id)

{

SqlConnection cn = new SqlConnection();

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

cn.Open();

Product obj = null;

SqlCommand selectCmd = new SqlCommand();

selectCmd.Connection = cn;

selectCmd.CommandType = System.Data.CommandType.Text;

selectCmd.CommandText = "select \* from Products where ProductId=" +id;

SqlDataReader dr = selectCmd.ExecuteReader();

if (dr.Read())

{

obj = new Product

{

ProductId = (int)dr["ProductId"],

ProductName = dr["ProductName"].ToString(),

Rate = (decimal)dr["Rate"],

Description = dr["Description"].ToString(),

CategoryName = dr["CategoryName"].ToString()

};

}

return View();

dr.Close();

cn.Close();

}

// POST: Product/Edit/5

[HttpPost]

public ActionResult Edit(int id, Product obj)

{

SqlConnection cn = new SqlConnection();

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

cn.Open();

SqlCommand updateCmd = new SqlCommand();

updateCmd.Connection = cn;

updateCmd.CommandType = System.Data.CommandType.StoredProcedure;

updateCmd.CommandText = "UpdateProduct";

updateCmd.Parameters.AddWithValue("@ProductId", obj.ProductId);

updateCmd.Parameters.AddWithValue("@ProductName", obj.ProductName);

updateCmd.Parameters.AddWithValue("@Rate", obj.Rate);

updateCmd.Parameters.AddWithValue("@Description", obj.Description);

updateCmd.Parameters.AddWithValue("@CategoryName", obj.CategoryName);

try

{

// TODO: Add update logic here

updateCmd.ExecuteNonQuery();

return RedirectToAction("Index");

}

catch

{

return View("Index");

}

cn.Close();

}

// GET: Product/Delete/5

public ActionResult Delete(int id)

{

return View();

}

// POST: Product/Delete/5

[HttpPost]

public ActionResult Delete(int id, FormCollection collection)

{

try

{

// TODO: Add delete logic here

return RedirectToAction("Index");

}

catch

{

return View();

}

}

}

}