using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace CustomerManagement.Models

{

public class CustomerDetails

{

public int CustomerId { get; set; }

public string Companyname { get; set; }

public string Email { get; set; }

public string Contactperson { get; set; }

public int Phoneno { get; set; }

public int Panno { get; set; }

public int Gstno { get; set; }

public int Cinno { get; set; }

public string District { get; set; }

public string State { get; set; }

public string Address { get; set; }

}

}

/////////CUSTOMER DATABASE CONTEXT

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

namespace CustomerManagement.Models

{

public class CustomerDBContext

{

public string cs = ConfigurationManager.ConnectionStrings["dbcs"].ConnectionString;

public List<CustomerDetails> GetCustomer()

{

List<CustomerDetails> customerDetailslist = new List<CustomerDetails>();

SqlConnection conn = new SqlConnection(cs);

SqlCommand cmd = new SqlCommand("spGetCustomers", conn);

cmd.CommandType = CommandType.StoredProcedure;

conn.Open();

SqlDataReader dr = cmd.ExecuteReader();

while (dr.Read())

{

CustomerDetails customerDetails = new CustomerDetails();

customerDetails.CustomerId = Convert.ToInt32(dr.GetValue(0).ToString());

customerDetails.Companyname = dr.GetValue(1).ToString();

customerDetails.Email = dr.GetValue(2).ToString();

customerDetails.Contactperson = dr.GetValue(2).ToString();

customerDetails.Phoneno= Convert.ToInt32(dr.GetValue(3).ToString());

customerDetails.Panno = Convert.ToInt32(dr.GetValue(4).ToString());

customerDetails.Gstno = Convert.ToInt32(dr.GetValue(4).ToString());

customerDetails.Cinno = Convert.ToInt32(dr.GetValue(4).ToString());

customerDetails.District= dr.GetValue(5).ToString();

customerDetails.State = dr.GetValue(5).ToString();

customerDetails.Address = dr.GetValue(5).ToString();

customerDetailslist.Add(customerDetails);

}

conn.Close();

return customerDetailslist;

}

public bool AddCustomer(CustomerDetails customerDetails)

{

SqlConnection conn = new SqlConnection(cs);

SqlCommand cmd = new SqlCommand("spAddCustomer", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@companyname", customerDetails.Companyname);

cmd.Parameters.AddWithValue("@email", customerDetails.Email);

cmd.Parameters.AddWithValue("@contactperaon", customerDetails.Contactperson);

cmd.Parameters.AddWithValue("@phoneno", customerDetails.Phoneno);

cmd.Parameters.AddWithValue("@panno", customerDetails.Panno);

cmd.Parameters.AddWithValue("@gstno", customerDetails.Gstno);

cmd.Parameters.AddWithValue("@cinno", customerDetails.Cinno);

cmd.Parameters.AddWithValue("@distict", customerDetails.District);

cmd.Parameters.AddWithValue("@state", customerDetails.State);

cmd.Parameters.AddWithValue("@addres", customerDetails.Address);

conn.Open();

int count = cmd.ExecuteNonQuery();

conn.Close();

if (count > 0)

{

return true;

}

else

{

return false;

}

}

///Update method

public bool UpdateCustomer(CustomerDetails customerDetails)

{

SqlConnection conn = new SqlConnection(cs);

SqlCommand cmd = new SqlCommand("spUpdateCustomer", conn);

cmd.CommandType = CommandType.StoredProcedure;

cmd.Parameters.AddWithValue("@id", customerDetails.CustomerId);

cmd.Parameters.AddWithValue("@companyname", customerDetails.Companyname);

cmd.Parameters.AddWithValue("@email", customerDetails.Email);

cmd.Parameters.AddWithValue("@contactperaon", customerDetails.Contactperson);

cmd.Parameters.AddWithValue("@phoneno", customerDetails.Phoneno);

cmd.Parameters.AddWithValue("@panno", customerDetails.Panno);

cmd.Parameters.AddWithValue("@gstno", customerDetails.Gstno);

cmd.Parameters.AddWithValue("@cinno", customerDetails.Cinno);

cmd.Parameters.AddWithValue("@distict", customerDetails.District);

cmd.Parameters.AddWithValue("@state", customerDetails.State);

cmd.Parameters.AddWithValue("@addres", customerDetails.Address);

conn.Open();

int count = cmd.ExecuteNonQuery();

conn.Close();

if (count > 0)

{

return true;

}

else

{

return false;

}

}

//delete

public bool DeleteCustomer(int id)

{

SqlConnection conn = new SqlConnection(cs);

SqlCommand cmd = new SqlCommand("spDeleteCustomer", conn);

cmd.CommandType = CommandType.StoredProcedure;

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

conn.Open();

int count = cmd.ExecuteNonQuery();

conn.Close();

if (count > 0)

{

return true;

}

else

{

return false;

}

}

}

}

/////////////DATBASE EXCEPTION CLASS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

namespace CustomerManagement.Models

{

public class CustomExceptionDataBase

{

}

}

//////////////////

CONTROLLERR

//////////////

using CustomerManagement.Models;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace CustomerManagement.Controllers

{

public class HomeController : Controller

{

// GET: Home

public ActionResult Index()

{

CustomerDBContext db = new CustomerDBContext();

List<CustomerDetails> obj = db.GetCustomer();

return View(obj);

}

public ActionResult Create()

{

return View();

}

[HttpPost]

public ActionResult Create(CustomerDetails customer)

{

try

{

if (ModelState.IsValid == true)

{

CustomerDBContext context = new CustomerDBContext();

bool check = context.AddCustomer(customer);

if (check == true)

{

TempData["InsertMessage"] = "Data has been submitted";

ModelState.Clear();

return RedirectToAction("Index");

}

}

return View();

}

catch

{

return View();

}

}

public ActionResult Edit(int id)

{

CustomerDBContext context = new CustomerDBContext();

var row = context.GetCustomer().Find(model => model.CustomerId == id);

return View(row);

}

[HttpPost]

public ActionResult Edit(int id,CustomerDetails customer)

{

if (ModelState.IsValid == true)

{

CustomerDBContext context = new CustomerDBContext();

bool check = context.UpdateCustomer(customer);

if (check == true)

{

TempData["UpdateMessage"] = "Data has been Updated";

ModelState.Clear();

return RedirectToAction("Index");

}

}

return View();

}

public ActionResult Details(int id)

{

CustomerDBContext context = new CustomerDBContext();

var row = context.GetCustomer().Find(model => model.CustomerId == id);

return View(row);

}

public ActionResult Delete(int id)

{

CustomerDBContext context = new CustomerDBContext();

var row = context.GetCustomer().Find(model => model.CustomerId == id);

return View(row);

}

[HttpPost]

public ActionResult Delete(int id, CustomerDetails customer)

{

CustomerDBContext context = new CustomerDBContext();

bool check = context.DeleteCustomer(id);

if (check == true)

{

TempData["DeleteMessage"] = "Data has been Deleted";

return RedirectToAction("Index");

}

return View();

}

}

}