**Source Code:**

**LoginContainer.cs**

using MVC.Models;

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data.SqlClient;

using System.Linq;

using System.Web;

using System.Web.Mvc;

using System.Web.Security;

namespace MVC.Controllers

{

public class LoginController : Controller

{

// GET: Login

public ActionResult Admin()

{

return View();

}

[HttpPost]

public ActionResult Admin(LoginInfo loginInfo)

{

string connection = ConfigurationManager.ConnectionStrings["BlogTracker"].ConnectionString;

SqlConnection con = new SqlConnection(connection);

string cmd = "Select EmailId,Password from AdminInfo where EmailId=@Emailid and Password=@Password";

con.Open();

SqlCommand command = new SqlCommand(cmd, con);

command.Parameters.AddWithValue("@EmailId", loginInfo.EmailId);

command.Parameters.AddWithValue("@Password", loginInfo.Password);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

Session["EmailId"] = loginInfo.EmailId.ToString();

return RedirectToAction("Index", "Emp");

}

else

{

ViewData["Message"] = "Admin Login Details Failed";

}

con.Close();

return View();

}

public ActionResult Employee()

{

return View();

}

[HttpPost]

public ActionResult Employee(LoginInfo loginInfo)

{

string connection = ConfigurationManager.ConnectionStrings["BlogTracker"].ConnectionString;

SqlConnection con = new SqlConnection(connection);

string cmd = "Select EmailId, PassCode from EmpInfo where EmailId=@Emailid and PassCode=@Password";

con.Open();

SqlCommand command = new SqlCommand(cmd, con);

command.Parameters.AddWithValue("@EmailId", loginInfo.EmailId);

command.Parameters.AddWithValue("@Password", loginInfo.Password);

SqlDataReader reader = command.ExecuteReader();

if (reader.Read())

{

Session["EmailId"] = loginInfo.EmailId.ToString();

return RedirectToAction("Index", "Blog");

}

else

{

ViewData["Message"] = "Employee Login Details Failed";

}

con.Close();

return View();

}

public ActionResult Logout()

{

FormsAuthentication.SignOut();

Session.Clear();

return RedirectToAction("GuestIndex", "Blog");

}

}

}

**EmpController.cs**

using MVC.Models;

using Newtonsoft.Json;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Text;

using System.Web;

using System.Web.Mvc;

namespace MVC.Controllers

{

public class EmpController : Controller

{

// GET: Emp

Uri baseAddress = new Uri("http://localhost:5230/api");

HttpClient client;

public EmpController()

{

client = new HttpClient();

client.BaseAddress = baseAddress;

}

public ActionResult Index()

{

List<EmpInfo> emps = new List<EmpInfo>();

HttpResponseMessage response = client.GetAsync(client.BaseAddress + "/EmpInfoes").Result;

if (response.IsSuccessStatusCode)

{

string data = response.Content.ReadAsStringAsync().Result;

emps = JsonConvert.DeserializeObject<List<EmpInfo>>(data);

}

return View(emps);

}

public ActionResult Create()

{

return View();

}

[HttpPost]

public ActionResult Create(EmpInfo emps)

{

string data = JsonConvert.SerializeObject(emps);

StringContent content = new StringContent(data, Encoding.UTF8, "application/json");

HttpResponseMessage responce = client.PostAsync(client.BaseAddress + "/EmpInfoes", content).Result;

if (responce.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

return View();

}

[HttpGet]

public ActionResult Edit(int id)

{

EmpInfo emps = new EmpInfo();

HttpResponseMessage response = client.GetAsync(client.BaseAddress + "/EmpInfoes/" + id).Result;

if (response.IsSuccessStatusCode)

{

string data = response.Content.ReadAsStringAsync().Result;

emps = JsonConvert.DeserializeObject<EmpInfo>(data);

}

return View(emps);

}

[HttpPost]

public ActionResult Edit(EmpInfo emp)

{

try

{

string data = JsonConvert.SerializeObject(emp);

StringContent content = new StringContent(data, Encoding.UTF8, "application/json");

HttpResponseMessage response = client.PutAsync(client.BaseAddress + "/EmpInfoes/" + emp.Id, content).Result;

if (response.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

else

{

ModelState.AddModelError(string.Empty, "Error updating emp.");

return View(emp);

}

}

catch (Exception ex)

{

ModelState.AddModelError(string.Empty, "An error occurred: " + ex.Message);

return View(emp);

}

}

[HttpGet]

public ActionResult Delete(int id)

{

try

{

EmpInfo emps = new EmpInfo();

HttpResponseMessage response = client.GetAsync(client.BaseAddress + "/EmpInfoes/" + id).Result;

if (response.IsSuccessStatusCode)

{

string data = response.Content.ReadAsStringAsync().Result;

emps = JsonConvert.DeserializeObject<EmpInfo>(data);

}

return View(emps);

}

catch (Exception)

{

return View();

}

}

[HttpPost, ActionName("Delete")]

public ActionResult DeleteConfirm(int id)

{

try

{

HttpResponseMessage response = client.DeleteAsync(client.BaseAddress + "/EmpInfoes/" + id).Result;

if (response.IsSuccessStatusCode)

{

return RedirectToAction("Index");

}

}

catch (Exception)

{

return View();

throw;

}

return View();

}

}

}