**Dot Net Full Stack Developer**

Capstone Project Problem Statement

**Blog Tracker Application**

**Capstone Project**

GITHUB: <https://github.com/11812142/CapstoneProject.git>

**SOURCE CODE :**

**Blog Application**

Adminmodel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace blog\_application.Models

{

public class Adminmodel

{

[Key]

[Required()]

[EmailAddress]

public string EmailId { get; set; }

[Required()]

public string Password { get; set; }

}

}

Blogmodel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace blog\_application.Models

{

public class blogmodel

{

[Key]

[Required()]

public int BlogId { get; set; }

[Required()]

[MaxLength(20, ErrorMessage = "Maximum 50 Characters only")]

public string Title { get; set; }

[Required()]

[MaxLength(50, ErrorMessage = "Maximum 50 Characters only")]

public string Subject { get; set; }

[Required()]

public DateTime DateOfCreation { get; set; }

[Required()]

public string BlogUrl { get; set; }

[Required()]

public string EmpEmailId { get; set; }

}

}

Empmodel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace blog\_application.Models

{

public class empmodel

{

[Key]

[Required()]

[EmailAddress]

public string EmailId { get; set; }

[Required()]

[MaxLength(50, ErrorMessage = "Maximum 50 Characters only")]

public string Name { get; set; }

[Required()]

public DateTime DateOfJoining { get; set; }

[Required()]

public int PassCode { get; set; }

}

}

BlogController.cs

using blog\_application.Models;

using Dal\_lib;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace blog\_application.Controllers

{

public class blogController : ApiController

{

operation log;

public blogController()

{

log = new operation();

}

public List<blogmodel> Getblog()

{

var ans = log.GetAllblog();

List<blogmodel> deptlist = new List<blogmodel>();

foreach (var item in ans)

{

deptlist.Add(new blogmodel()

{

BlogId = item.BlogId,

Title = item.Title,

Subject = item.Subject,

DateOfCreation = item.DateOfCreation,

BlogUrl = item.BlogUrl,

EmpEmailId = item.EmpEmailId

});

}

return deptlist;

}

// GET api/<controller>/5

// POST api/<controller>

public void Postblog([FromBody] blogmodel value)

{

BlogInfo dept = new BlogInfo();

dept.BlogId = value.BlogId; ;

dept.Title = value.Title;

dept.Subject = value.Subject;

dept.DateOfCreation = value.DateOfCreation;

dept.BlogUrl = value.BlogUrl;

dept.EmpEmailId = value.EmpEmailId;

log.Addblog(dept);

}

// PUT api/<controller>/5

public void Putblog(int id, [FromBody] blogmodel value)

{

BlogInfo dept = new BlogInfo();

dept.BlogId = value.BlogId; ;

dept.Title = value.Title;

dept.Subject = value.Subject;

dept.DateOfCreation = value.DateOfCreation;

dept.BlogUrl = value.BlogUrl;

dept.EmpEmailId = value.EmpEmailId;

log.UpdateBlogDetails(id, dept);

}

// DELETE api/<controller>/5

public void Deleteblog(int id)

{

log.DeleteBlogDetails(id);

}

}

}

employeeController.cs

using blog\_application.Models;

using Dal\_lib;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

namespace blog\_application.Controllers

{

public class employeeController : ApiController

{

operation log;

public employeeController()

{

log = new operation();

}

// [Route("getemp")]

public List<empmodel> Getemp()

{

var ans = log.GetAllEmp();

List<empmodel> emplist = new List<empmodel>();

foreach (var item in ans)

{

emplist.Add(new empmodel()

{

EmailId = item.EmailId,

Name = item.Name,

DateOfJoining = item.DateOfJoining,

PassCode = item.PassCode

});

}

return emplist;

}

// GET api/<controller>/5

// POST api/<controller>

public void Postemp([FromBody] empmodel value)

{

EmpInfo emp = new EmpInfo();

emp.EmailId = value.EmailId;

emp.Name = value.Name;

emp.DateOfJoining = value.DateOfJoining;

emp.PassCode = value.PassCode;

log.AddEmp(emp);

}

// PUT api/<controller>/5

public void Putemp(int id, [FromBody] empmodel value)

{

EmpInfo emp = new EmpInfo();

emp.EmailId = value.EmailId;

emp.Name = value.Name;

emp.DateOfJoining = value.DateOfJoining;

emp.PassCode = value.PassCode;

log.UpdateEmployeeDetails(id, emp);

}

// DELETE api/<controller>/5

public void Deleteemp(int id)

{

log.DeleteEmployeeDetails(id);

}

}

}

**Dal Library**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.ComponentModel.DataAnnotations;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Dal\_lib

{

public class AdminInfo

{

[Required()]

[EmailAddress]

[Key]

public string EmailId { get; set; }

[Required()]

public string Password { get; set; }

}

/// <summary>

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

/// </summary>

public class EmpInfo

{

[Required()]

[EmailAddress]

[Key]

public string EmailId { get; set; }

[Required()]

[MaxLength(50, ErrorMessage = "Maximum 50 Characters only")]

public string Name { get; set; }

[Required()]

public DateTime DateOfJoining { get; set; }

[Required()]

public int PassCode { get; set; }

}

/// <summary>

/// ////////

/// </summary>

public class BlogInfo

{

[Required()]

[Key]

public int BlogId { get; set; }

[Required()]

[MaxLength(20, ErrorMessage = "Maximum 50 Characters only")]

public string Title { get; set; }

[Required()]

[MaxLength(50, ErrorMessage = "Maximum 50 Characters only")]

public string Subject { get; set; }

[Required()]

public DateTime DateOfCreation { get; set; }

[Required()]

public string BlogUrl { get; set; }

[Required()]

public string EmpEmailId { get; set; }

}

/// <summary>

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

/// </summary>

public class MyContext : DbContext

{

public MyContext() : base("MyContext")

{

Database.SetInitializer(new Init());

// Database.SetInitializer<MyContext>(new DropCreateDatabaseIfModelChanges<MyContext>());

}

public virtual DbSet<AdminInfo> AdminInfos { get; set; }

public virtual DbSet<EmpInfo> EmpInfos { get; set; }

public virtual DbSet<BlogInfo> BlogInfos { get; set; }

}

/// <summary>

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

/// </summary>

public class Init : DropCreateDatabaseIfModelChanges<MyContext>

{

protected override void Seed(MyContext context)

{

List<AdminInfo> adminlist = new List<AdminInfo>();

adminlist.Add(new AdminInfo() { EmailId = "laawanyasunkara@gmail.com", Password = "lavanya123" });

context.AdminInfos.AddRange(adminlist);

context.SaveChanges();

base.Seed(context);

}

}

/// <summary>

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

/// </summary>

public class operation

{

MyContext context = null;

public operation()

{

context = new MyContext();

}

public List<EmpInfo> GetAllEmp()

{

var ans = context.EmpInfos.ToList();

return ans;

}

public void AddEmp(EmpInfo emp)

{

context.EmpInfos.Add(emp);

context.SaveChanges();

}

public EmpInfo Getempbyid(int id)

{

List<EmpInfo> s = context.EmpInfos.ToList();

EmpInfo r = s.Find(pr => pr.PassCode == id);

return r;

}

public bool DeleteEmployeeDetails(int id)

{

try

{

List<EmpInfo> s = context.EmpInfos.ToList();

EmpInfo r = s.Find(pr => pr.PassCode == id);

context.EmpInfos.Remove(r);

context.SaveChanges();

return true;

}

catch (Exception ex)

{

return false;

}

}

public bool UpdateEmployeeDetails(int id, EmpInfo p)

{

try

{

List<EmpInfo> s = context.EmpInfos.ToList();

EmpInfo k = s.Find(pr => pr.PassCode == id);

k.PassCode = p.PassCode;

k.EmailId = p.EmailId;

k.Name = p.Name;

k.DateOfJoining = p.DateOfJoining;

context.SaveChanges();

return true;

}

catch (Exception ex)

{

return false;

}

}

/// <summary>

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

/// </summary>

/// <returns></returns>

public List<BlogInfo> GetAllblog()

{

return context.BlogInfos.ToList();

}

public void Addblog(BlogInfo m)

{

context.BlogInfos.Add(m);

context.SaveChanges();

}

public BlogInfo Getblogbyid(int id)

{

List<BlogInfo> s = context.BlogInfos.ToList();

BlogInfo r = s.Find(pr => pr.BlogId == id);

return r;

}

public bool UpdateBlogDetails(int id, BlogInfo p)

{

try

{

List<BlogInfo> s = context.BlogInfos.ToList();

BlogInfo k = s.Find(pr => pr.BlogId == id);

k.BlogId = p.BlogId;

k.Title = p.Title;

k.Subject = p.Subject;

k.DateOfCreation = p.DateOfCreation;

k.BlogUrl = p.BlogUrl;

k.EmpEmailId = p.EmpEmailId;

context.SaveChanges();

return true;

}

catch (Exception ex)

{

return false;

}

}

public bool DeleteBlogDetails(int id)

{

try

{

List<BlogInfo> s = context.BlogInfos.ToList();

BlogInfo r = s.Find(pr => pr.BlogId == id);

context.BlogInfos.Remove(r);

context.SaveChanges();

return true;

}

catch (Exception ex)

{

return false;

}

}

}

}

**UI**

bloguiCntroller.cs

using Dal\_lib;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Runtime.Remoting.Messaging;

using System.Web;

using System.Web.Mvc;

using UI\_layer.Models;

namespace UI\_layer.Controllers

{

public class bloguiController : Controller

{

// GET: blogui

MyContext db = new MyContext();

public ActionResult Login()

{

return View();

}

[HttpPost]

public ActionResult Login(empmodel log)

{

var user = db.EmpInfos.Where(x => x.EmailId == log.EmailId && x.PassCode == log.PassCode).Count();

if (user > 0)

{

return RedirectToAction("Index");

}

else

{

return View();

}

}

public ActionResult Index()

{

List<blogmodel> emplist = new List<blogmodel>();

using (var client = new HttpClient())

{

client.BaseAddress = new Uri("https://localhost:44378//api/");

var responseTask = client.GetAsync("blog");

responseTask.Wait();

var result = responseTask.Result;

if (result.IsSuccessStatusCode)

{

var readData = result.Content.ReadAsAsync<blogmodel[]>();

readData.Wait();

var empdata = readData.Result;

foreach (var item in empdata)

{

emplist.Add(new blogmodel

{

BlogId = item.BlogId,

Title = item.Title,

Subject = item.Subject,

DateOfCreation = item.DateOfCreation,

BlogUrl = item.BlogUrl,

EmpEmailId = item.EmpEmailId

});

}

}

}

return View(emplist);

}

public ActionResult Create()

{

return View();

}

[HttpPost]

public ActionResult Create(blogmodel empmodel)

{

using (var client = new HttpClient())

{

client.BaseAddress = new Uri("https://localhost:44378//api/blog");

var emp = new blogmodel

{

BlogId = empmodel.BlogId,

Title = empmodel.Title,

Subject = empmodel.Subject,

DateOfCreation = empmodel.DateOfCreation,

BlogUrl= empmodel.BlogUrl,

EmpEmailId= empmodel.EmpEmailId

};

var postTask = client.PostAsJsonAsync<blogmodel>(client.BaseAddress, emp);

postTask.Wait();

var result = postTask.Result;

if (result.IsSuccessStatusCode)

{

var readtaskResult = result.Content.ReadAsAsync<blogmodel>();

readtaskResult.Wait();

var dataInserted = readtaskResult.Result;

}

}

return RedirectToAction("Index");

}

operation op = null;

public bloguiController()

{

op = new operation();

}

public ActionResult EditBlog(int id)

{

var emp = op.Getblogbyid(id);

blogmodel model = new blogmodel();

model.BlogId = id;

model.Title = emp.Title;

model.Subject = emp.Subject;

model.DateOfCreation = emp.DateOfCreation;

model.BlogUrl = emp.BlogUrl;

model.EmpEmailId = emp.EmpEmailId;

return View(model);

}

// POST: Emp/Edit/5

[HttpPost]

public ActionResult EditBlog(int id, FormCollection collection)

{

try

{

// TODO: Add update logic here

var emp = op.Getblogbyid(id);

emp.BlogId = Convert.ToInt32(Request["BlogId"]);

emp.Title = Request["Title"].ToString();

emp.Subject = Request["Subject"].ToString();

emp.DateOfCreation = Convert.ToDateTime(Request["DateOfCreation"]);

emp.BlogUrl = Request["BlogUrl"].ToString();

emp.EmpEmailId = Request["EmpEmailId"].ToString();

bool ans = op.UpdateBlogDetails(id, emp);

if (ans)

{

return RedirectToAction("Index");

}

else

{

return View();

}

}

catch

{

return View();

}

}

// GET: Emp/Delete/5

public ActionResult DeleteBlog(int id)

{

var emp = op.Getblogbyid(id);

blogmodel model = new blogmodel();

model.BlogId = id;

model.Title = emp.Title;

model.Subject = emp.Subject;

model.DateOfCreation = emp.DateOfCreation;

model.BlogUrl = emp.BlogUrl;

model.EmpEmailId = emp.EmpEmailId;

return View(model);

}

// POST: Emp/Delete/5

[HttpPost]

public ActionResult DeleteBlog(int id, FormCollection collection)

{

try

{

// TODO: Add delete logic here

var dataFound = op.Getblogbyid(id);

if (dataFound != null)

{

bool ans = op.DeleteBlogDetails(id);

if (ans)

{

return RedirectToAction("Index");

}

else

{

return View();

}

}

return RedirectToAction("Index");

}

catch

{

return View();

}

}

}

}

empuiController.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Web;

using System.Web.Mvc;

using UI\_layer.Models;

using Dal\_lib;

using System.Runtime.Remoting.Messaging;

namespace UI\_layer.Controllers

{

public class empuiController : Controller

{

// GET: empui

MyContext db=new MyContext();

public ActionResult Login()

{

return View();

}

[HttpPost]

public ActionResult Login(Adminmodel log)

{

var user = db.AdminInfos.Where(x => x.EmailId == log.EmailId && x.Password == log.Password).Count();

if (user > 0)

{

return RedirectToAction("Index");

}

else

{

return View();

}

}

public ActionResult Index()

{

List<empmodel> emplist = new List<empmodel>();

using (var client = new HttpClient())

{

client.BaseAddress = new Uri("https://localhost:44378//api/");

var responseTask = client.GetAsync("employee");

responseTask.Wait();

var result = responseTask.Result;

if (result.IsSuccessStatusCode)

{

var readData = result.Content.ReadAsAsync<empmodel[]>();

readData.Wait();

var empdata = readData.Result;

foreach (var item in empdata)

{

emplist.Add(new empmodel

{

EmailId = item.EmailId,

Name = item.Name,

DateOfJoining = item.DateOfJoining,

PassCode = item.PassCode

});

}

}

}

return View(emplist);

}

public ActionResult Create()

{

return View();

}

[HttpPost]

public ActionResult Create(empmodel empmodel)

{

using (var client = new HttpClient())

{

client.BaseAddress = new Uri("https://localhost:44378//api/employee");

var emp = new empmodel

{

EmailId = empmodel.EmailId,

Name = empmodel.Name,

DateOfJoining = empmodel.DateOfJoining,

PassCode = empmodel.PassCode

};

var postTask = client.PostAsJsonAsync<empmodel>(client.BaseAddress, emp);

postTask.Wait();

var result = postTask.Result;

if (result.IsSuccessStatusCode)

{

var readtaskResult = result.Content.ReadAsAsync<empmodel>();

readtaskResult.Wait();

var dataInserted = readtaskResult.Result;

}

}

return RedirectToAction("Index");

}

operation op = null;

public empuiController()

{

op = new operation();

}

public ActionResult Edit(int id)

{

var emp = op.Getempbyid(id);

empmodel model = new empmodel();

model.PassCode = id;

model.EmailId=emp.EmailId;

model.Name = emp.Name;

model.DateOfJoining = emp.DateOfJoining;

return View(model);

}

// POST: Emp/Edit/5

[HttpPost]

public ActionResult Edit(int id, FormCollection collection)

{

try

{

// TODO: Add update logic here

var emp = op.Getempbyid(id);

emp.EmailId = Request["EmailId"].ToString();

emp.Name = Request["Name"].ToString();

emp.DateOfJoining = Convert.ToDateTime(Request["DateOfJoining"]);

emp.PassCode = Convert.ToInt32(Request["PassCode"]);

bool ans =op.UpdateEmployeeDetails(id, emp);

if (ans)

{

return RedirectToAction("Index");

}

else

{

return View();

}

}

catch

{

return View();

}

}

// GET: Emp/Delete/5

public ActionResult Delete(int id)

{

var emp = op.Getempbyid(id);

empmodel model = new empmodel();

model.PassCode = id;

model.EmailId = emp.EmailId;

model.Name = emp.Name;

model.DateOfJoining = emp.DateOfJoining;

return View(model);

}

// POST: Emp/Delete/5

[HttpPost]

public ActionResult Delete(int id, FormCollection collection)

{

try

{

// TODO: Add delete logic here

var dataFound = op.Getempbyid(id);

if (dataFound != null)

{

bool ans = op.DeleteEmployeeDetails(id);

if (ans)

{

return RedirectToAction("Index");

}

else

{

return View();

}

}

return RedirectToAction("Index");

}

catch

{

return View();

}

}

}

}

Home Controller.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Web;

using System.Web.Mvc;

using UI\_layer.Models;

namespace UI\_layer.Controllers

{

public class HomeController : Controller

{

public ActionResult Index()

{

List<blogmodel> emplist = new List<blogmodel>();

using (var client = new HttpClient())

{

client.BaseAddress = new Uri("https://localhost:44378//api/");

var responseTask = client.GetAsync("blog");

responseTask.Wait();

var result = responseTask.Result;

if (result.IsSuccessStatusCode)

{

var readData = result.Content.ReadAsAsync<blogmodel[]>();

readData.Wait();

var empdata = readData.Result;

foreach (var item in empdata)

{

emplist.Add(new blogmodel

{

BlogId = item.BlogId,

Title = item.Title,

Subject = item.Subject,

DateOfCreation = item.DateOfCreation,

BlogUrl = item.BlogUrl,

EmpEmailId = item.EmpEmailId

});

}

}

}

return View(emplist);

}

}

}

Admin Model

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace UI\_layer.Models

{

public class Adminmodel

{

[Key]

[Required()]

[EmailAddress]

public string EmailId { get; set; }

[Required()]

public string Password { get; set; }

}

}

Blog Model

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace UI\_layer.Models

{

public class blogmodel

{

[Key]

[Required()]

public int BlogId { get; set; }

[Required()]

[MaxLength(20, ErrorMessage = "Maximum 50 Characters only")]

public string Title { get; set; }

[Required()]

[MaxLength(50, ErrorMessage = "Maximum 50 Characters only")]

public string Subject { get; set; }

[Required()]

public DateTime DateOfCreation { get; set; }

[Required()]

public string BlogUrl { get; set; }

[Required()]

public string EmpEmailId { get; set; }

}

}

Employee Model

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Web;

namespace UI\_layer.Models

{

public class empmodel

{

[Key]

[Required()]

[EmailAddress]

public string EmailId { get; set; }

[Required()]

[MaxLength(50, ErrorMessage = "Maximum 50 Characters only")]

public string Name { get; set; }

[Required()]

public DateTime DateOfJoining { get; set; }

[Required()]

public int PassCode { get; set; }

}

}

**Unit Test**

Blog app test

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using Dal\_lib;

using NUnit.Framework;

namespace Unit\_Test

{

[TestFixture]

public class blog\_app\_test

{

[TestCase]

public void Admin\_mail\_Test()

{

MyContext db = new MyContext();

var found = db.AdminInfos.ToList();

Assert.AreEqual("pratibha@gmail.com", found[0].EmailId);

}

[TestCase]

public void Admin\_password\_Test()

{

MyContext db = new MyContext();

var found = db.AdminInfos.ToList();

Assert.AreEqual("pratibha123", found[0].Password);

}

validate v=new validate();

[TestCase]

public void AdminTest()

{

v.check();

}

}

}

Validate.cs

using Dal\_lib;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Unit\_Test

{

public class validate

{

MyContext db=new MyContext();

public bool check()

{

bool ans = false;

var found = db.AdminInfos.ToList();

if (found[0].EmailId == "pratibha@gmail.com" && found[0].Password == "pratibha123")

{

ans = true;

}

return ans;

}

}

}