**Deploy an ASP.NET WebForms Application on Azure**

Github Link:

**SourceCode:**

**1.** **StudentController**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using Microsoft.EntityFrameworkCore;

using P4SchoolAzure.Models;

namespace P4SchoolAzure.Controllers

{

public class StudentsController : Controller

{

private readonly AzureSchoolContext \_context;

public StudentsController(AzureSchoolContext context)

{

\_context = context;

}

// GET: Students

public async Task<IActionResult> Index()

{

return \_context.Students != null ?

View(await \_context.Students.ToListAsync()) :

Problem("Entity set 'AzureSchoolContext.Students' is null.");

}

// GET: Students/Details/5

public async Task<IActionResult> Details(int? id)

{

if (id == null || \_context.Students == null)

{

return NotFound();

}

var student = await \_context.Students

.FirstOrDefaultAsync(m => m.StudentId == id);

if (student == null)

{

return NotFound();

}

return View(student);

}

// GET: Students/Create

public IActionResult Create()

{

return View();

}

// POST: Students/Create

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Create([Bind("StudentId,Fname,Lname,Course,Gender")] Student student)

{

if (ModelState.IsValid)

{

\_context.Add(student);

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

return View(student);

}

// GET: Students/Edit/5

public async Task<IActionResult> Edit(int? id)

{

if (id == null || \_context.Students == null)

{

return NotFound();

}

var student = await \_context.Students.FindAsync(id);

if (student == null)

{

return NotFound();

}

return View(student);

}

// POST: Students/Edit/5

// To protect from overposting attacks, enable the specific properties you want to bind to.

// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Edit(int id, [Bind("StudentId,Fname,Lname,Course,Gender")] Student student)

{

if (id != student.StudentId)

{

return NotFound();

}

if (ModelState.IsValid)

{

try

{

\_context.Update(student);

await \_context.SaveChangesAsync();

}

catch (DbUpdateConcurrencyException)

{

if (!StudentExists(student.StudentId))

{

return NotFound();

}

else

{

throw;

}

}

return RedirectToAction(nameof(Index));

}

return View(student);

}

// GET: Students/Delete/5

public async Task<IActionResult> Delete(int? id)

{

if (id == null || \_context.Students == null)

{

return NotFound();

}

var student = await \_context.Students

.FirstOrDefaultAsync(m => m.StudentId == id);

if (student == null)

{

return NotFound();

}

return View(student);

}

// POST: Students/Delete/5

[HttpPost, ActionName("Delete")]

[ValidateAntiForgeryToken]

public async Task<IActionResult> DeleteConfirmed(int id)

{

if (\_context.Students == null)

{

return Problem("Entity set 'AzureSchoolContext.Students' is null.");

}

var student = await \_context.Students.FindAsync(id);

if (student != null)

{

\_context.Students.Remove(student);

}

await \_context.SaveChangesAsync();

return RedirectToAction(nameof(Index));

}

private bool StudentExists(int id)

{

return (\_context.Students?.Any(e => e.StudentId == id)).GetValueOrDefault();

}

}

}

2. Program.cs

using Microsoft.EntityFrameworkCore;

using P4SchoolAzure.Models;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllersWithViews();

builder.Services.AddDbContext<AzureSchoolContext>(options => options.UseSqlServer(builder.Configuration.GetConnectionString("SchoolConstr")));

var app = builder.Build();

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Home/Error");

}

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

3. appsettings.json

{

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"ConnectionStrings": {

"SchoolConstr": "Server=tcp:radhikaasn.database.windows.net,1433;Initial Catalog=AzureSchool;Persist Security Info=False;User ID=radhika;Password=Swamy@8296412406;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;" },

"AllowedHosts": "\*"

}

**4. student Table query in azure**

CREATE TABLE student (

student\_id INT PRIMARY KEY,

Fname VARCHAR(255),

Lname VARCHAR(255),

course VARCHAR(50),

Gender CHAR(1)

);

INSERT INTO student (student\_id, Fname, Lname, course, Gender)

VALUES

(1, 'John', 'Doe', 'Computer Science', 'M'),

(2, 'Jane', 'Smith', 'Mathematics', 'F'),

(3, 'Mike', 'Johnson', 'Physics', 'M'),

(4, 'Emily', 'Williams', 'Biology', 'F');

**5. AzureScoolDbcontext**

using System;

using System.Collections.Generic;

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Metadata;

namespace P4SchoolAzure.Models

{

public partial class AzureSchoolContext : DbContext

{

public AzureSchoolContext()

{

}

public AzureSchoolContext(DbContextOptions<AzureSchoolContext> options)

: base(options)

{

}

public virtual DbSet<Student> Students { get; set; } = null!;

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

#warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the connection string by using the Name= syntax to read it from configuration - see https://go.microsoft.com/fwlink/?linkid=2131148. For more guidance on storing connection strings, see http://go.microsoft.com/fwlink/?LinkId=723263.

optionsBuilder.UseSqlServer("Server=tcp:radhikaasn.database.windows.net,1433;Initial Catalog=AzureSchool;Persist Security Info=False;User ID=radhika;Password=Swamy@8296412406;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;");

}

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Student>(entity =>

{

entity.ToTable("student");

entity.Property(e => e.StudentId)

.ValueGeneratedNever()

.HasColumnName("student\_id");

entity.Property(e => e.Course)

.HasMaxLength(50)

.IsUnicode(false)

.HasColumnName("course");

entity.Property(e => e.Fname)

.HasMaxLength(255)

.IsUnicode(false);

entity.Property(e => e.Gender)

.HasMaxLength(1)

.IsUnicode(false)

.IsFixedLength();

entity.Property(e => e.Lname)

.HasMaxLength(255)

.IsUnicode(false);

});

OnModelCreatingPartial(modelBuilder);

}

partial void OnModelCreatingPartial(ModelBuilder modelBuilder);

}

}