OEL 01

Name: Muhammad Shoaib Akhter Qadri

Enrollment No: 02-131212-009

Reg No: 79290

AttendeeController.cs

using Microsoft.AspNetCore.Mvc;

using Microsoft.Graph.Models;

using System.Linq;

using System.Web.Mvc;

public class AttendeeController : Controller

{

private readonly AppDbContext \_context;

public AttendeeController(AppDbContext context)

{

\_context = context;

}

// GET: Attendees

public IActionResult Index(int eventId)

{

var attendees = \_context.Attendees.Where(a => a.EventID == eventId).ToList();

return View(attendees);

}

// GET: Attendees/Add

public IActionResult Add(int eventId)

{

var model = new AttendeeViewModel { EventID = eventId };

return View(model);

}

// POST: Attendees/Add

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult Add(AttendeeViewModel model)

{

if (ModelState.IsValid)

{

// Add a new attendee to the event and save to the database

var newAttendee = new Attendee { EventID = model.EventID, FullName = model.FullName, Email = model.Email };

\_context.Attendees.Add(newAttendee);

\_context.SaveChanges();

// Redirect to the attendees list page for the event

return RedirectToAction("Index", new { eventId = model.EventID });

}

// Model state is not valid, return the view with validation errors

return View(model);

}

// Other methods for removing attendees, etc.

}

ErrorHandlingController.cs  
using Microsoft.AspNetCore.Mvc;

using System.Web.Mvc;

using System;

public class ErrorHandlingController : Controller

{

private readonly AppDbContext \_context;

public ErrorHandlingController(AppDbContext context)

{

\_context = context;

}

// Log error

public IActionResult LogError(string errorMessage)

{

// Log error to the database

var errorLog = new ErrorLog { ErrorMessage = errorMessage, ErrorDateTime = DateTime.Now };

\_context.ErrorLogs.Add(errorLog);

\_context.SaveChanges();

// Redirect to error page

return RedirectToAction("ErrorPage");

}

// Display error page

public IActionResult ErrorPage()

{

// Retrieve and display error message from the database

var errorMessage = \_context.ErrorLogs.OrderByDescending(e => e.ErrorDateTime).FirstOrDefault()?.ErrorMessage;

return View("Error", errorMessage);

}

}

Database SQL Code:  
-- Create EventManagementSystem database

CREATE DATABASE EventManagementSystem;

-- Use EventManagementSystem database

USE EventManagementSystem;

-- Create Users table for authentication

CREATE TABLE Users (

UserID INT PRIMARY KEY IDENTITY(1,1),

Username VARCHAR(50) NOT NULL,

Password VARCHAR(100) NOT NULL, -- You should hash passwords for security

Email VARCHAR(100) NOT NULL,

FullName VARCHAR(100) NOT NULL,

Role VARCHAR(50) NOT NULL -- Define roles such as 'Admin', 'Organizer', 'Attendee', etc.

);

-- Create Events table for storing event details

CREATE TABLE Events (

EventID INT PRIMARY KEY IDENTITY(1,1),

EventName VARCHAR(100) NOT NULL,

Description TEXT,

EventDate DATE NOT NULL,

OrganizerID INT FOREIGN KEY REFERENCES Users(UserID) -- Reference to the organizer of the event

);

-- Create Attendees table for managing attendees

CREATE TABLE Attendees (

AttendeeID INT PRIMARY KEY IDENTITY(1,1),

EventID INT FOREIGN KEY REFERENCES Events(EventID), -- Reference to the event

UserID INT FOREIGN KEY REFERENCES Users(UserID) -- Reference to the attendee

);

-- Create ErrorLog table for error handling

CREATE TABLE ErrorLog (

ErrorID INT PRIMARY KEY IDENTITY(1,1),

ErrorMessage VARCHAR(500) NOT NULL,

ErrorDateTime DATETIME NOT NULL

);

EventController.cs  
using Microsoft.AspNetCore.Mvc;

using System.Linq;

public class EventController : Controller

{

private readonly AppDbContext \_context;

public EventController(AppDbContext context)

{

\_context = context;

}

// GET: Events

public IActionResult Index()

{

var events = \_context.Events.ToList();

return View(events);

}

// GET: Events/Create

public IActionResult Create()

{

return View();

}

// POST: Events/Create

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult Create(EventViewModel model)

{

if (ModelState.IsValid)

{

// Create a new event and save to the database

var newEvent = new Event { EventName = model.EventName, Description = model.Description, EventDate = model.EventDate };

\_context.Events.Add(newEvent);

\_context.SaveChanges();

// Redirect to the events list page

return RedirectToAction("Index");

}

// Model state is not valid, return the view with validation errors

return View(model);

}

// Other methods for editing, deleting events, etc.

}  
  
AttendeeController.cs  
using Microsoft.AspNetCore.Mvc;

using System.Linq;

public class AttendeeController : Controller

{

private readonly AppDbContext \_context;

public AttendeeController(AppDbContext context)

{

\_context = context;

}

// GET: Attendees

public IActionResult Index(int eventId)

{

var attendees = \_context.Attendees.Where(a => a.EventID == eventId).ToList();

return View(attendees);

}

// GET: Attendees/Add

public IActionResult Add(int eventId)

{

var model = new AttendeeViewModel { EventID = eventId };

return View(model);

}

// POST: Attendees/Add

[HttpPost]

[ValidateAntiForgeryToken]

public IActionResult Add(AttendeeViewModel model)

{

if (ModelState.IsValid)

{

// Add a new attendee to the event and save to the database

var newAttendee = new Attendee { EventID = model.EventID, FullName = model.FullName, Email = model.Email };

\_context.Attendees.Add(newAttendee);

\_context.SaveChanges();

// Redirect to the attendees list page for the event

return RedirectToAction("Index", new { eventId = model.EventID });

}

// Model state is not valid, return the view with validation errors

return View(model);

}

// Other methods for removing attendees, etc.

}  
  
**ErrorHandlingController.cs**using Microsoft.AspNetCore.Mvc;

public class ErrorHandlingController : Controller

{

private readonly AppDbContext \_context;

public ErrorHandlingController(AppDbContext context)

{

\_context = context;

}

// Log error

public IActionResult LogError(string errorMessage)

{

// Log error to the database

var errorLog = new ErrorLog { ErrorMessage = errorMessage, ErrorDateTime = DateTime.Now };

\_context.ErrorLogs.Add(errorLog);

\_context.SaveChanges();

// Redirect to error page

return RedirectToAction("ErrorPage");

}

// Display error page

public IActionResult ErrorPage()

{

// Retrieve and display error message from the database

var errorMessage = \_context.ErrorLogs.OrderByDescending(e => e.ErrorDateTime).FirstOrDefault()?.ErrorMessage;

return View("Error", errorMessage);

}

}

AppDbContext.cs  
using Microsoft.EntityFrameworkCore;

using System.Collections.Generic;

public class AppDbContext : DbContext

{

public DbSet<User> Users { get; set; }

public DbSet<Event> Events { get; set; }

public DbSet<Attendee> Attendees { get; set; }

// Add DbSet properties for other model classes as needed

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

// Configure database connection string

optionsBuilder.UseSqlServer("YourConnectionString");

}

}