

MVC Application to Manage Data for School Application

Source Code:

Teacher.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace MVCSchool.Models
{
    public class Teacher
    {
        public int TeacherId { get; set; }

        public string FirstName { get; set; }

        public string LastName { get; set; }

        public string Gender { get; set; }

        public string Email { get; set; }

        public string PhoneNumber { get; set; }

        public virtual ICollection<Course> Courses { get; set; } = new
List<Course>();

        public virtual ICollection<Student> Students { get; set; } = new
List<Student>();
    }
}
```

Student.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace MVCSchool.Models
{
    public class Student
    {
        public int StudentId { get; set; }

        public string FirstName { get; set; }

        public string LastName { get; set; }

        public string Gender { get; set; }

        public string Email { get; set; }

        public string PhoneNumber { get; set; }
    }
}
```

```

        public int TeacherId { get; set; }

        public virtual Teacher Teacher { get; set; }
    }
}

```

Course.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace MVCSchool.Models
{
    public class Course
    {
        public int CourseId { get; set; }

        public string CourseName { get; set; }

        public string CourseDescription { get; set; }

        public int TeacherId { get; set; }
        public virtual Teacher Teacher { get; set; }
    }
}

```

TeachersController.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using MVCSchool.Data;
using MVCSchool.Models;

namespace MVCSchool.Controllers
{
    public class TeachersController : Controller
    {
        private SchoolDbContext db = new SchoolDbContext();

        // GET: Teachers
        public ActionResult Index()
        {
            return View(db.Teachers.ToList());
        }

        // GET: Teachers/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
        }
    }
}

```

```

    }
    Teacher teacher = db.Teachers.Find(id);
    if (teacher == null)
    {
        return HttpNotFound();
    }
    return View(teacher);
}

// GET: Teachers/Create
public ActionResult Create()
{
    return View();
}

// POST: Teachers/Create
// To protect from overposting attacks, enable the specific properties you
want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include =
"TeacherId,FirstName,LastName,Gender,Email,PhoneNumber")] Teacher teacher)
{
    if (ModelState.IsValid)
    {
        db.Teachers.Add(teacher);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(teacher);
}

// GET: Teachers/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Teacher teacher = db.Teachers.Find(id);
    if (teacher == null)
    {
        return HttpNotFound();
    }
    return View(teacher);
}

// POST: Teachers/Edit/5
// To protect from overposting attacks, enable the specific properties you
want to bind to, for
// more details see https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"TeacherId,FirstName,LastName,Gender,Email,PhoneNumber")] Teacher teacher)
{

```

```

        if (ModelState.IsValid)
        {
            db.Entry(teacher).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        return View(teacher);
    }

    // GET: Teachers/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Teacher teacher = db.Teachers.Find(id);
        if (teacher == null)
        {
            return HttpNotFound();
        }
        return View(teacher);
    }

    // POST: Teachers/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Teacher teacher = db.Teachers.Find(id);
        db.Teachers.Remove(teacher);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

StudentsController.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using MVCSchool.Data;
using MVCSchool.Models;

```

```

namespace MVCSchool.Controllers
{
    public class StudentsController : Controller
    {
        private SchoolDbContext db = new SchoolDbContext();

        // GET: Students
        public ActionResult Index()
        {
            var students = db.Students.Include(s => s.Teacher);
            return View(students.ToList());
        }

        // GET: Students/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Student student = db.Students.Find(id);
            if (student == null)
            {
                return HttpNotFound();
            }
            return View(student);
        }

        // GET: Students/Create
        public ActionResult Create()
        {
            ViewBag.TeacherId = new SelectList(db.Teachers, "TeacherId",
"FirstName");
            return View();
        }

        // POST: Students/Create
        // To protect from overposting attacks, enable the specific properties you
want to bind to, for
        // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
"StudentId,FirstName,LastName,Gender,Email,PhoneNumber,TeacherId")] Student student)
        {
            if (ModelState.IsValid)
            {
                db.Students.Add(student);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            ViewBag.TeacherId = new SelectList(db.Teachers, "TeacherId",
"FirstName", student.TeacherId);
            return View(student);
        }

        // GET: Students/Edit/5

```

```

public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Student student = db.Students.Find(id);
    if (student == null)
    {
        return HttpNotFound();
    }
    ViewBag.TeacherId = new SelectList(db.Teachers, "TeacherId",
"FirstName", student.TeacherId);
    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 https://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Edit([Bind(Include =
"StudentId,FirstName,LastName,Gender,Email,PhoneNumber,TeacherId")] Student student)
{
    if (ModelState.IsValid)
    {
        db.Entry(student).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    ViewBag.TeacherId = new SelectList(db.Teachers, "TeacherId",
"FirstName", student.TeacherId);
    return View(student);
}

// GET: Students/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Student student = db.Students.Find(id);
    if (student == null)
    {
        return HttpNotFound();
    }
    return View(student);
}

// POST: Students/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    Student student = db.Students.Find(id);
    db.Students.Remove(student);
}

```

```

        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```

CoursesController.cs

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using MVCSchool.Data;
using MVCSchool.Models;

namespace MVCSchool.Controllers
{
    public class CoursesController : Controller
    {
        private SchoolDbContext db = new SchoolDbContext();

        // GET: Courses
        public ActionResult Index()
        {
            var courses = db.Courses.Include(c => c.Teacher);
            return View(courses.ToList());
        }

        // GET: Courses/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Course course = db.Courses.Find(id);
            if (course == null)
            {
                return HttpNotFound();
            }
            return View(course);
        }

        // GET: Courses/Create
        public ActionResult Create()
        {

```

```

        ViewBag.TeacherId = new SelectList(db.Teachers, "TeacherId",
"FirstName");
        return View();
    }

    // POST: Courses/Create
    // To protect from overposting attacks, enable the specific properties you
want to bind to, for
    // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Create([Bind(Include =
"CourseId,CourseName,CourseDescription,TeacherId")] Course course)
    {
        if (ModelState.IsValid)
        {
            db.Courses.Add(course);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

        ViewBag.TeacherId = new SelectList(db.Teachers, "TeacherId",
"FirstName", course.TeacherId);
        return View(course);
    }

    // GET: Courses/Edit/5
    public ActionResult Edit(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Course course = db.Courses.Find(id);
        if (course == null)
        {
            return HttpNotFound();
        }
        ViewBag.TeacherId = new SelectList(db.Teachers, "TeacherId",
"FirstName", course.TeacherId);
        return View(course);
    }

    // POST: Courses/Edit/5
    // To protect from overposting attacks, enable the specific properties you
want to bind to, for
    // more details see https://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
"CourseId,CourseName,CourseDescription,TeacherId")] Course course)
    {
        if (ModelState.IsValid)
        {
            db.Entry(course).State = EntityState.Modified;
            db.SaveChanges();
            return RedirectToAction("Index");
        }
    }

```



```

        ViewBag.TeacherId = new SelectList(db.Teachers, "TeacherId",
"FirstName", course.TeacherId);
        return View(course);
    }

    // GET: Courses/Delete/5
    public ActionResult Delete(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        Course course = db.Courses.Find(id);
        if (course == null)
        {
            return HttpNotFound();
        }
        return View(course);
    }

    // POST: Courses/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public ActionResult DeleteConfirmed(int id)
    {
        Course course = db.Courses.Find(id);
        db.Courses.Remove(course);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing)
        {
            db.Dispose();
        }
        base.Dispose(disposing);
    }
}

```