```csharp
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 MVCSchoolContext db = new MVCSchoolContext();

        // 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)
            {
```

```csharp
                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();
```

```csharp
        }
        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);
    }
  }
}
```