```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using Assignment7.Models;

namespace Assignment7.Controllers.API
{
    //[RoutePrefix("api/student")]
    public class StudentController : ApiController
    {
        public StudentController()
        {
        }

        [HttpGet]
        public IHttpActionResult GetAllStudents()
        {
            IList<StudentViewModel> students = null;
            using (var ctx = new SchoolDBEntities())
            {
                students = ctx.Students.Include("StudentAddress")
                            .Select(s => new StudentViewModel()
                            {
                                Id = s.StudentID,
                                Name = s.StudentName
                            }).ToList<StudentViewModel>();
            }

            if (students.Count == 0)
            {
                return NotFound();
            }

            return Ok(students);
        }

        [HttpGet]
        public IHttpActionResult GetAllStudentsWithAddress(bool includeAddress)
        {
            IList<StudentViewModel> students = null;

            using (var ctx = new SchoolDBEntities())
            {
                students = ctx.Students.Include("StudentAddress").Select(s => new StudentViewModel()
                {
                    Id = s.StudentID,
                    Name = s.StudentName,
                    Address = s.StudentAddress == null ? null : new AddressViewModel()
                    {
                        StudentId = s.StudentAddress.StudentID,
                        Address1 = s.StudentAddress.Address1,
                        Address2 = s.StudentAddress.Address2,
                        City = s.StudentAddress.City,
                        State = s.StudentAddress.State
                    }
                }).ToList<StudentViewModel>();
            }


            if (students.Count == 0)
            {
                return NotFound();
            }

            return Ok(students);
        }

        [HttpGet]
        public IHttpActionResult GetStudentById(int id)
        {
            StudentViewModel student = null;

            using (var ctx = new SchoolDBEntities())
            {
                student = ctx.Students.Include("StudentAddress")
                    .Where(s => s.StudentID == id)
                    .Select(s => new StudentViewModel()
                    {
                        Id = s.StudentID,
                        Name = s.StudentName
                    }).FirstOrDefault<StudentViewModel>();
            }

            if (student == null)
            {
                return NotFound();
            }

            return Ok(student);
        }

        [HttpGet]
        public IHttpActionResult GetAllStudents(string name)
        {
            IList<StudentViewModel> students = null;

            using (var ctx = new SchoolDBEntities())
            {
                students = ctx.Students.Include("StudentAddress")
                    .Where(s => s.StudentName.ToLower() == name.ToLower())
                    .Select(s => new StudentViewModel()
                    {
                        Id = s.StudentID,
                        Name = s.StudentName,
                        Address = s.StudentAddress == null ? null : new AddressViewModel()
                        {
                            StudentId = s.StudentAddress.StudentID,
                            Address1 = s.StudentAddress.Address1,
                            Address2 = s.StudentAddress.Address2,
                            City = s.StudentAddress.City,
                            State = s.StudentAddress.State
                        }
                    }).ToList<StudentViewModel>();
            }


            if (students.Count == 0)
            {
                return NotFound();
            }
            return Ok(students);
        }

        [HttpGet]
        public IHttpActionResult GetAllStudentsInSameStandard(int standardId)
        {
            IList<StudentViewModel> students = null;

            using (var ctx = new SchoolDBEntities())
            {
                students = ctx.Students.Include("StudentAddress").Include("Standard").Where(s => s.StandardId == standardId)
                            .Select(s => new StudentViewModel()
                            {
                                Id = s.StudentID,
                                Name = s.StudentName,
                                Address = s.StudentAddress == null ? null : new AddressViewModel()
                                {
                                    StudentId = s.StudentAddress.StudentID,
                                    Address1 = s.StudentAddress.Address1,
                                    Address2 = s.StudentAddress.Address2,
                                    City = s.StudentAddress.City,
                                    State = s.StudentAddress.State
                                },
                                Standard = new StandardViewModel()
                                {
                                    StandardId = s.Standard.StandardId,
                                    Name = s.Standard.StandardName
                                }
                            }).ToList<StudentViewModel>();
            }


            if (students.Count == 0)
            {
                return NotFound();
            }

            return Ok(students);
        }

        [HttpPost]
        public IHttpActionResult PostNewStudent(StudentViewModel student)
        {
            if (!ModelState.IsValid)
                return BadRequest("Invalid data.");

            using (var ctx = new SchoolDBEntities())
            {
                ctx.Students.Add(new Student()
                {
                    StudentID = student.Id,
                    StudentName = student.Name
                });

                ctx.SaveChanges();
            }

            return Ok();
        }

        [HttpPut]
        public IHttpActionResult Put(StudentViewModel student)
        {
            if (!ModelState.IsValid)
                return BadRequest("Not a valid model");

            using (var ctx = new SchoolDBEntities())
            {
                var existingStudent = ctx.Students.Where(s => s.StudentID == student.Id)
                                                        .FirstOrDefault<Student>();

                if (existingStudent != null)
                {
                    existingStudent.StudentName = student.Name;

                    ctx.SaveChanges();
                }
                else
                {
                    return NotFound();
                }
            }

            return Ok();
        }

        [HttpDelete]
        public IHttpActionResult Delete(int id)
        {
            if (id <= 0)
                return BadRequest("Not a valid student id");

            using (var ctx = new SchoolDBEntities())
            {
                var student = ctx.Students
                    .Where(s => s.StudentID == id)
                    .FirstOrDefault();

                ctx.Entry(student).State = System.Data.Entity.EntityState.Deleted;
                ctx.SaveChanges();
            }

            return Ok();
        }


    }
}
```