```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace MVCSchoolDB.Models
{
    [Table("Class_")]
    public class Class_
    {
        [Key]
        public int ClassId { get; set; }
        [Required]
        [StringLength(50)]
        public string CName { get; set; }
        public virtual ICollection<Subject_> Subject_s { get; set; }
        public virtual ICollection<Student_Class> Student_Classes { get; set; }
    }

}

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace MVCSchoolDB.Models
{
    [Table("Student_Class")]
    public class Student_Class
    {
        [Key]
        public int Id { get; set; }

        [Required]
        [StringLength(50)]
        public string FName { get; set; }

        [Required]
        [StringLength(50)]
        public string LName { get; set; }
        public int CId { get; set; }
        public virtual Class_ Class_ { get; set; }
        public int SubId { get; set; }
        public virtual Subject_ Subject_ { get; set; }
    }

}

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations.Schema;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;

namespace MVCSchoolDB.Models
{
    [Table("Subject")]
    public class Subject_
    {
        [Key]
        public int SubId { get; set; }

        [Required]
        [StringLength(50)]
        public string SubName { get; set; }
        public int CId { get; set; }
        public virtual Class_ Class_ { get; set; }
        public virtual ICollection<Student_Class> Student_Classes { get; set; }
    }

}

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 MVCSchoolDB.Data;
using MVCSchoolDB.Models;

namespace MVCSchoolDB.Controllers
{
    public class Class_sController : Controller
    {
        private RainbowSchoolDBContext db = new RainbowSchoolDBContext();

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

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

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

        // POST: Class_s/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 = "ClassId,CName")] Class_
class_)
        {
            if (ModelState.IsValid)
            {
                db.Class_.Add(class_);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            return View(class_);
        }

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

        // POST: Class_s/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 = "ClassId,CName")] Class_ class_)
        {
            if (ModelState.IsValid)
            {
                db.Entry(class_).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            return View(class_);
        }

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

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

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


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 MVCSchoolDB.Data;
using MVCSchoolDB.Models;

namespace MVCSchoolDB.Controllers
{
    public class Student_ClassController : Controller
    {
        private RainbowSchoolDBContext db = new RainbowSchoolDBContext();

        // GET: Student_Class
        public ActionResult Index()
        {
            var student_Class = db.Student_Class.Include(s => s.Subject_);
            return View(student_Class.ToList());
        }

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

        // GET: Student_Class/Create
        public ActionResult Create()
        {
            ViewBag.SubId = new SelectList(db.Subject_, "SubId", "SubName");
            return View();
        }

        // POST: Student_Class/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 = "Id,FName,LName,CId,SubId")]
Student_Class student_Class)
        {
            if (ModelState.IsValid)
            {
                db.Student_Class.Add(student_Class);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            ViewBag.SubId = new SelectList(db.Subject_, "SubId", "SubName",
student_Class.SubId);
            return View(student_Class);
        }

        // GET: Student_Class/Edit/5
        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Student_Class student_Class = db.Student_Class.Find(id);
            if (student_Class == null)
            {
                return HttpNotFound();
            }
            ViewBag.SubId = new SelectList(db.Subject_, "SubId", "SubName",
student_Class.SubId);
            return View(student_Class);
        }

        // POST: Student_Class/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 = "Id,FName,LName,CId,SubId")]
Student_Class student_Class)
        {
            if (ModelState.IsValid)
            {
                db.Entry(student_Class).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.SubId = new SelectList(db.Subject_, "SubId", "SubName",
student_Class.SubId);
            return View(student_Class);
        }
        // GET: Student_Class/Delete/5
        public ActionResult Delete(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Student_Class student_Class = db.Student_Class.Find(id);
            if (student_Class == null)
            {
                return HttpNotFound();
            }
            return View(student_Class);
        }

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

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

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 MVCSchoolDB.Data;
using MVCSchoolDB.Models;

namespace MVCSchoolDB.Controllers
{
    public class Subject_Controller : Controller
    {
        private RainbowSchoolDBContext db = new RainbowSchoolDBContext();

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

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

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

        // POST: Subject_/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 = "SubId,SubName,CId")] Subject_ 
subject_)
        {
            if (ModelState.IsValid)
            {
                db.Subject_.Add(subject_);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            return View(subject_);
        }

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

        // POST: Subject_/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 = "SubId,SubName,CId")] Subject_ 
subject_)
        {
            if (ModelState.IsValid)
            {
                db.Entry(subject_).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            return View(subject_);
        }

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

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

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

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - Rainbow_SchoolDb</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-
dark">
        <div class="container">
            @Html.ActionLink("Rainbow_SchoolDb", "Index", "Home", new { area = ""
}, new { @class = "navbar-brand" })
            <button type="button" class="navbar-toggler" data-bs-
toggle="collapse" data-bs-target=".navbar-collapse" title="Toggle navigation"
aria-controls="navbarSupportedContent"
                    aria-expanded="false" aria-label="Toggle navigation">
```

```
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse d-sm-inline-flex justify-content-between">
                <ul class="navbar-nav flex-grow-1">
                    <li>@Html.ActionLink("Home", "Index", "Home", new { area = "" }, new { @class = "nav-link" })</li>
                    <li>@Html.ActionLink("Class", "Index", "Class_s", new { area = "" }, new { @class = "nav-link" })</li>
                    <li>@Html.ActionLink("Subject", "Index", "Subject_", new { area = "" }, new { @class = "nav-link" })</li>
                    <li>@Html.ActionLink("Student", "Index", "Student_Class", new { area = "" }, new { @class = "nav-link" })</li>
                    <li>@Html.ActionLink("About", "About", "Home", new { area = "" }, new { @class = "nav-link" })</li>
                    <li>@Html.ActionLink("Contact", "Contact", "Home", new { area = "" }, new { @class = "nav-link" })</li>
                </ul>
            </div>
        </div>
    </nav>
    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - &reg; Rainbow_SchoolDb</p>
        </footer>
    </div>

    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @RenderSection("scripts", required: false)
</body>
</html>
```