Sample Source Code

(Automatically generated code)

Class Model:

```csharp
namespace Student_Management_System.Models
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;

    public partial class Rainbow_SchoolDbEntities : DbContext
    {
        public Rainbow_SchoolDbEntities()
            : base("name=Rainbow_SchoolDbEntities")
        {
        }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            throw new UnintentionalCodeFirstException();
        }

        public virtual DbSet<Class> Classes { get; set; }
        public virtual DbSet<Student> Students { get; set; }
        public virtual DbSet<Subject> Subjects { get; set; }
    }
}
```

Class Controller(ClassesController.cs):

```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 Student_Management_System.Models;

namespace Student_Management_System.Controllers
{
    public class ClassesController : Controller
    {
        private Rainbow_SchoolDbEntities db = new Rainbow_SchoolDbEntities();
```

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

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

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

// POST: Classes/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,ClassName")] Class @class)
{
    if (ModelState.IsValid)
    {
        db.Classes.Add(@class);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(@class);
}

// GET: Classes/Edit/5
```

```csharp
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    Class @class = db.Classes.Find(id);
    if (@class == null)
    {
        return HttpNotFound();
    }
    return View(@class);
}

// POST: Classes/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,ClassName")] Class @class)
{
    if (ModelState.IsValid)
    {
        db.Entry(@class).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(@class);
}

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

// POST: Classes/Delete/5
[HttpPost, ActionName("Delete")]
```

```csharp
        [ValidateAntiForgeryToken]
        public ActionResult DeleteConfirmed(int id)
        {
            Class @class = db.Classes.Find(id);
            db.Classes.Remove(@class);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

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

Class View(Index.cshtml):

```
@model IEnumerable<Student_Management_System.Models.Class>

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.ClassName)
        </th>
        <th></th>
    </tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.ClassName)
        </td>
        <td>
```

```
            @Html.ActionLink("Edit", "Edit", new { id=item.ClassID }) |
            @Html.ActionLink("Details", "Details", new { id=item.ClassID }) |
            @Html.ActionLink("Delete", "Delete", new { id=item.ClassID })
        </td>
    </tr>
}

</table>
```