Source Code

SchoolDatabase.sql

```
CREATE DATABASE SCHOOLDB
USE SCHOOLDB
CREATE TABLE Student (
    StudentID INT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    DateOfBirth DATE,
    ClassID INT
);
INSERT INTO Student (StudentID, FirstName, LastName, DateOfBirth, ClassID)
VALUES
    (1, 'Nagashree', 'KS', '2005-03-15', 1),
    (2, 'Manvi', 'S', '2006-07-10', 2),
    (3, 'Appu', 'J', '2005-09-20', 1),
    (4, 'Vijay', 'M', '2007-01-25', 3),
    (5, 'Samantha', 'V', '2006-11-05', 2);
SELECT * FROM Student
CREATE TABLE Subjects (
    SubjectID INT PRIMARY KEY,
    SubjectName VARCHAR(50)
);
INSERT INTO Subjects (SubjectID, SubjectName)
VALUES
    (1, 'Mathematics'),
    (2, 'Science'),
```

```
    (3, 'History'),
    (4, 'English'),
    (5, 'Computer Science');

SELECT * FROM Subjects
CREATE TABLE Classes (
    ClassID INT PRIMARY KEY,
    ClassName VARCHAR(20)
);
INSERT INTO Classes (ClassID, ClassName)
VALUES
    (1, 'Class 01'),
    (2, 'Class 02'),
    (3, 'Class 03'),
    (4, 'Class 04'),
    (5, 'Class 05');
SELECT * FROM Classes
CREATE NONCLUSTERED INDEX IX_Student_ClassID ON Student (ClassID);
CREATE NONCLUSTERED INDEX IX_Student_LastName ON Student (LastName);
CREATE NONCLUSTERED INDEX IX_Subjects_SubjectName ON Subjects (SubjectName);
CREATE NONCLUSTERED INDEX IX_Classes_ClassName ON Classes (ClassName);
```

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 SchoolManagementPracticeProject;
```

```
namespace SchoolManagementPracticeProject.Controllers
{
    public class StudentsController : Controller
    {
        private SCHOOLDBEntities db = new SCHOOLDBEntities();

        // GET: Students
        public ActionResult Index()
        {
            return View(db.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()
        {
            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,DateOfBirth,ClassID")] Student student)
        {
            if (ModelState.IsValid)
            {
                db.Students.Add(student);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            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();
```

```
            }
            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,DateOfBirth,ClassID")] Student student)
        {
            if (ModelState.IsValid)
            {
                db.Entry(student).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            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);
        }
    }
}
```

SubjectsController.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 SchoolManagementPracticeProject;

namespace SchoolManagementPracticeProject.Controllers
{
    public class SubjectsController : Controller
    {
        private SCHOOLDBEntities db = new SCHOOLDBEntities();

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

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

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

        // POST: Subjects/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 = "SubjectID,SubjectName")]
Subject subject)
        {
            if (ModelState.IsValid)
```

```
            {
                db.Subjects.Add(subject);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            return View(subject);
        }

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

        // POST: Subjects/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 = "SubjectID,SubjectName")]
Subject subject)
        {
            if (ModelState.IsValid)
            {
                db.Entry(subject).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            return View(subject);
        }

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

        // POST: Subjects/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public ActionResult DeleteConfirmed(int id)
        {
            Subject subject = db.Subjects.Find(id);
```

```
            db.Subjects.Remove(subject);
            db.SaveChanges();
            return RedirectToAction("Index");
        }

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

ClassesController.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 SchoolManagementPracticeProject;

namespace SchoolManagementPracticeProject.Controllers
{
    public class ClassesController : Controller
    {
        private SCHOOLDBEntities db = new SCHOOLDBEntities();

        // 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
        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")]
        [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);
        }
    }
}
```