

Student.cd

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

namespace RainbowSchoolStatistics.Models
{
    public class Student
    {
        [Key]
        public int StudentId { get; set; }
        [Required]
        public string StudentName { get; set; }
        public int Marks { get; set; }
    }
}
```

Statistics.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace RainbowSchoolStatistics.Models
{
    public class Statistics
    {
        public int StudentCount { get; set; }
        public Student MaximumMarks { get; set; }
        public Student MinimumMarks { get; set; }
        public List<Student> LessThan65Marks { get; set; }
        public double PassedPercentage { get; set; }
    }
}
```

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 RainbowSchoolStatistics.Data;
using RainbowSchoolStatistics.Models;

namespace RainbowSchoolStatistics.Controllers
{
    public class StudentsController : Controller
    {
        private RainbowSchoolStatisticsContext db = new
RainbowSchoolStatisticsContext();
    }
}
```

```

// 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(new Student());
}

// 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,StudentName,Marks")]
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,StudentName,Marks")]
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);
    }

    public ActionResult Stats()
    {

```

```

        int total = (db.Students.Count())*100;
        var viewModel = new Statistics
        {
            StudentCount = db.Students.Count(),
            MaximumMarks = db.Students.OrderByDescending(s =>
s.Marks).FirstOrDefault(),
            MinimumMarks = db.Students.OrderBy(s => s.Marks).FirstOrDefault(),
            LessThan65Marks = db.Students.Where(s => s.Marks < 65).ToList(),
            PassedPercentage = (db.Students.Sum(s => s.Marks)/total)*100
        };

        return View(viewModel);
    }
}

```

Stats.cshtml

```
@model RainbowSchoolStatistics.Models.Statistics
```

```

@{
    ViewBag.Title = "Stats";
}
<h2>Statistics</h2>

<div>
    <h4>Student Statistics</h4>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            Total Students
        </dt>
        <dd>
            @Model.StudentCount
        </dd>
        <dt>
            Passed Percentage
        </dt>
        <dd>
            @Model.PassedPercentage
        </dd>
        <dt>
            Student with Maximum Marks
        </dt>
        <dd>
            @Model.MaximumMarks.StudentName
            @Model.MaximumMarks.Marks
        </dd>
        <dt>
            Student with Minimum Marks
        </dt>
        <dd>
            @Model.MinimumMarks.StudentName
            @Model.MinimumMarks.Marks
        </dd>
    </dl>
</div>

```

```
<dt>
    Students with Less than 65 Marks
</dt>
<dd>
    @foreach (var student in Model.LessThan65Marks)
    {
        <span>@student.StudentName </span>
        <span>@student.Marks</span>
        <br />
    }
</dd>
</dl>
</div>
```