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

namespace Phase3SchoolDbAzure.Models
{
    public class StatisticsViewModel
    {
        public double AverageScore { get; set; }
        public double HighestScore { get; set; }
        public double LowestScore { 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 Phase3SchoolDbAzure.Models;

namespace Phase3SchoolDbAzure.Controllers
{
    public class MarksController : Controller
    {
        private RainbowSchoolDbEntities db = new RainbowSchoolDbEntities();

        // GET: Marks
        public ActionResult Index()
        {
            var marks = db.Marks.Include(m => m.Student);
            return View(marks.ToList());
        }

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

        // GET: Marks/Create
        public ActionResult Create()
        {
            ViewBag.StudentId = new SelectList(db.Students, "Id", "FirstName");
            return View();
        }

        // POST: Marks/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,StudentId,Score")] Mark 
mark)
        {
            if (ModelState.IsValid)
            {
                db.Marks.Add(mark);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            ViewBag.StudentId = new SelectList(db.Students, "Id", "FirstName", 
mark.StudentId);
            return View(mark);
        }

        // GET: Marks/Edit/5
        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            Mark mark = db.Marks.Find(id);
            if (mark == null)
            {
                return HttpNotFound();
            }
            ViewBag.StudentId = new SelectList(db.Students, "Id", "FirstName", 
mark.StudentId);
            return View(mark);
        }

        // POST: Marks/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,StudentId,Score")] Mark 
mark)
        {
            if (ModelState.IsValid)
            {
                db.Entry(mark).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.StudentId = new SelectList(db.Students, "Id", "FirstName", 
mark.StudentId);
            return View(mark);
        }

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

        // POST: Marks/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public ActionResult DeleteConfirmed(int id)
        {
            Mark mark = db.Marks.Find(id);
            db.Marks.Remove(mark);
            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 Phase3SchoolDbAzure.Models;

namespace Phase3SchoolDbAzure.Controllers
{
    public class StudentsController : Controller
    {
        private RainbowSchoolDbEntities db = new RainbowSchoolDbEntities();

        // 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 =
"Id,FirstName,LastName,DateOfBirth,Gender,Address")] 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 =
"Id,FirstName,LastName,DateOfBirth,Gender,Address")] 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 Statistics()
        {
            var students = db.Students.Include(s => s.Marks).ToList();

            // Calculate the average score for all students
            double averageScore = students.Average(s => s.Marks.Sum(m =>
m.Score));

            // Calculate the highest score among all students
            double highestScore = students.Max(s => s.Marks.Max(m => m.Score));

            // Calculate the lowest score among all students
            double lowestScore = students.Min(s => s.Marks.Min(m => m.Score));

            // Create a StatisticsViewModel to pass the calculated values to the
view
            var model = new StatisticsViewModel
            {
                AverageScore = averageScore,
                HighestScore = highestScore,
                LowestScore = lowestScore
            };

            return View(model);
        }
    }
}
```

```
@model Phase3SchoolDbAzure.Models.StatisticsViewModel

@{
    ViewBag.Title = "Statistics";
}

<h2>Statistics</h2>
<div>
    <p>Average Score: @Model.AverageScore</p>
    <p>Highest Score: @Model.HighestScore</p>
    <p>Lowest Score: @Model.LowestScore</p>
</div>

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - School Database</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")

    <!-- Add custom CSS for styling -->
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f2f2f2;
        }

        .navbar {
            background-color: #343a40 !important;
        }

        .navbar-dark .navbar-nav .nav-link {
            color: #ffffff !important;
        }

        .container.body-content {
            background-color: #ffffff;
            border-radius: 5px;
            padding: 20px;
            margin-top: 20px;
        }

        footer {
            margin-top: 20px;
            text-align: center;
            color: #555;
        }

        /* Add more custom styles as needed */

        /* Styles for tables */
        table {
            width: 100%;
            border-collapse: collapse;
        }

        table, th, td {
            border: 1px solid #ddd;
        }

        th, td {
            padding: 8px;
            text-align: left;
        }
        th {
            background-color: #343a40;
            color: #fff;
        }

        /* Add more table styles as needed */
    </style>
</head>
<body>
    <nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-
dark">
        <div class="container">
            <!-- Update the application name and links as needed -->
            @Html.ActionLink("School Database", "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("Students", "Index", "Students", new {
area = "" }, new { @class = "nav-link" })</li>
                    <li>@Html.ActionLink("Marks", "Index", "Marks", new { area =
"" }, new { @class = "nav-link" })</li>
                    <li>@Html.ActionLink("Statistics", "Statistics", "Students",
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>
    @RenderBody() <!-- This is where child views will inject their content -->
    <hr />

    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @RenderSection("scripts", required: false)

    <footer>
        <p>&copy; @DateTime.Now.Year - School Database</p>
    </footer>
</body>
</html>
```