Sample Code (Autogenerated)

Mark Statistics(index.cshtml)

```
@model IEnumerable<MarksStatistics.Models.MarkStatistic>

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

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

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.AverageMarks)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.MaxMarks)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.MinMarks)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.TotalStudents)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Class.ClassName)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id=item.Id }) |
            @Html.ActionLink("Details", "Details", new { id=item.Id }) |
            @Html.ActionLink("Delete", "Delete", new { id=item.Id })
        </td>
    </tr>
}

</table>
```

StatisticsController

```
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web.Mvc;
using MarksStatistics.Models;

namespace MarksStatistics.Controllers
{
    public class MarkStatisticsController : Controller
    {
        private Rainbow_SchoolDbEntities db = new Rainbow_SchoolDbEntities();

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

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

        // GET: MarkStatistics/Create
        public ActionResult Create()
        {
            ViewBag.ClassId = new SelectList(db.Classes, "ClassID", "ClassName");
            ViewBag.StudentId = new SelectList(db.Students, "StudentID",
"FirstName");
            return View();
        }

        // POST: MarkStatistics/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,ClassId,StudentId,AverageMarks,MaxMarks,MinMarks,TotalStudents")]
MarkStatistic markStatistic)
        {
            if (ModelState.IsValid)
            {
                db.MarkStatistics.Add(markStatistic);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.ClassId = new SelectList(db.Classes, "ClassID", "ClassName",
markStatistic.ClassId);
            ViewBag.StudentId = new SelectList(db.Students, "StudentID",
"FirstName", markStatistic.StudentId);
            return View(markStatistic);
        }

        // GET: MarkStatistics/Edit/5
        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            MarkStatistic markStatistic = db.MarkStatistics.Find(id);
            if (markStatistic == null)
            {
                return HttpNotFound();
            }
            ViewBag.ClassId = new SelectList(db.Classes, "ClassID", "ClassName",
markStatistic.ClassId);
            ViewBag.StudentId = new SelectList(db.Students, "StudentID",
"FirstName", markStatistic.StudentId);
            return View(markStatistic);
        }

        // POST: MarkStatistics/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,ClassId,StudentId,AverageMarks,MaxMarks,MinMarks,TotalStudents")]
MarkStatistic markStatistic)
        {
            if (ModelState.IsValid)
            {
                db.Entry(markStatistic).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.ClassId = new SelectList(db.Classes, "ClassID", "ClassName",
markStatistic.ClassId);
            ViewBag.StudentId = new SelectList(db.Students, "StudentID",
"FirstName", markStatistic.StudentId);
            return View(markStatistic);
        }

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

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

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