

## Practice Project-3

### Section 7 Source Code

#### StudentsMarksController.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 Display_Statistics_of_Marks_Obtained_by_Students.Models;

namespace Display_Statistics_of_Marks_Obtained_by_Students.Content
{
    public class StudentMarksController : Controller
    {
        private StudentDatabaseEntities db = new StudentDatabaseEntities();

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

        // GET: StudentMarks/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
        }
    }
}
```

```

    }
    StudentMark studentMark = db.StudentMarks.Find(id);
    if (studentMark == null)
    {
        return HttpNotFound();
    }
    return View(studentMark);
}

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

// POST: StudentMarks/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,Subject,Marks")] StudentMark
studentMark)
{
    if (ModelState.IsValid)
    {
        db.StudentMarks.Add(studentMark);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    return View(studentMark);
}

```

```
// GET: StudentMarks/Edit/5
```

```
public ActionResult Edit(int? id)
```

```
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    StudentMark studentMark = db.StudentMarks.Find(id);
    if (studentMark == null)
    {
        return HttpNotFound();
    }
    return View(studentMark);
}
```

```
// POST: StudentMarks/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,Subject,Marks")] StudentMark
studentMark)
```

```
{
    if (ModelState.IsValid)
    {
        db.Entry(studentMark).State = EntityState.Modified;
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    return View(studentMark);
}
```

```
// GET: StudentMarks/Delete/5
```

```

public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    StudentMark studentMark = db.StudentMarks.Find(id);
    if (studentMark == null)
    {
        return HttpNotFound();
    }
    return View(studentMark);
}

```

```

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

```

```

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

```

```
}  
}  
}
```

## Layout.cshtml

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>@ViewBag.Title - My ASP.NET Application</title>
```

```
@Styles.Render("~/Content/css")
```

```
@Scripts.Render("~/bundles/modernizr")
```

```
</head>
```

```
<body>
```

```
<nav class="navbar navbar-expand-sm navbar-toggleable-sm navbar-dark bg-dark">
```

```
<div class="container">
```

```
@Html.ActionLink("Application name", "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("About", "About", "Home", new { area = "" }, new { @class = "nav-link" })</li>
```

```
<li>@Html.ActionLink("Contact", "Contact", "Home", new { area = "" }, new { @class = "nav-link" })</li>
```

```
<li>@Html.ActionLink("Student Details", "Index", "Students", new { area = "" }, new { @class = "nav-link"  
})</li>
```

```
</ul>
```

```
</div>
```

```

        </div>
    </nav>
    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
        </footer>
    </div>

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

```

## Index.cshtml

```
@model IEnumerable<Display_Statistics_of_Marks_Obtained_by_Students.Models.StudentMark>
```

```

@{
    ViewBag.Title = "Index";
}

```

```
<h2>Index</h2>
```

```

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

```

```
</th>
<th>
    @Html.DisplayNameFor(model => model.FirstName)
</th>
<th>
    @Html.DisplayNameFor(model => model.LastName)
</th>
<th>
    @Html.DisplayNameFor(model => model.Subject)
</th>
<th>
    @Html.DisplayNameFor(model => model.Marks)
</th>

<th></th>
</tr>
```

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

```

        @Html.DisplayFor(modelItem => item.Marks)
    </td>
    <td>
        @Html.ActionLink("Edit", "Edit", new { id = item.StudentID }) |
        @Html.ActionLink("Details", "Details", new { id = item.StudentID }) |
        @Html.ActionLink("Delete", "Delete", new { id = item.StudentID })
    </td>
</tr>
}

</table>

```

### SQL Query

```
CREATE DATABASE StudentDatabase;
```

```
USE StudentDatabase;
```

```
CREATE TABLE StudentMarks (
```

```
    StudentID INT PRIMARY KEY ,
```

```
    FirstName VARCHAR(50) NOT NULL,
```

```
    LastName VARCHAR(50) NOT NULL,
```

```
    Subject VARCHAR(50) NOT NULL,
```

```
    Marks INT NOT NULL
```

```
)
```

```
INSERT INTO StudentMarks (StudentID,FirstName, LastName, Subject, Marks) VALUES (1,'John', 'Doe', 'Math', 90) ,
```

```
(2,'Jane', 'Smith', 'science', 85) ,
```

```
(3,'Alice', 'Johnson', 'English', 78) ,
```

```
(4,'Bob', 'Brown', 'Advance-java', 92),
```

```
(5,'Eva', 'Lee', 'Advance-python', 88),
```

```
(6,'Michael', 'Clark', 'C#', 75)
```



```
SELECT AVG(Marks)AS AverageMarks,  
MAX(Marks) AS MaximumMarks ,  
MIN(Marks) AS MinimumMarks  
FROM StudentMarks
```

```
SELECT * FROM StudentMarks;
```