\_Layout.cshtml  
     <!DOCTYPE html>  
     <html>  
     <head>  
         <meta charset="utf-8" />  
         <meta name="viewport" content="width=device-width, initial-scale=1.0">  
         <title>@ViewBag.Title - Business School Management</title>  
         @Styles.Render("~/Content/css")  
         @Scripts.Render("~/bundles/modernizr")  
     </head>  
    <body>  
        <div class="navbar navbar-inverse navbar-fixed-top">  
            <div class="container">  
                <div class="navbar-header">  
                    <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">  
                        <span class="icon-bar"></span>  
                        <span class="icon-bar"></span>  
                       <span class="icon-bar"></span>  
                    </button>  
                  @Html.ActionLink("Business School Grading", "Index", "Student", new { area = "" }, new { @class = "navbar-brand" })  
              </div>  
              <div class="navbar-collapse collapse">  
                 <ul class="nav navbar-nav">  
                        <li>@Html.ActionLink("Home", "Index", "Student")</li>  
                        <li>@Html.ActionLink("About", "About", "Student")</li>  
                      <li>@Html.ActionLink("Contact", "Contact", "Home")</li>  
                  </ul>  
                </div>  
            </div>  
        </div>  
        <div class="container body-content">  
            @RenderBody()  
            <hr />  
            <footer>  
                <p>&copy; @DateTime.Now.Year - Business School Management</p>  
            </footer>  
        </div>  
     
        @Scripts.Render("~/bundles/jquery")  
        @Scripts.Render("~/bundles/bootstrap")  
        @RenderSection("scripts", required: false)  
    </body>  
    </html>  
                   
     
AddMarks.cshtml  
     @using BusinessSchool.Models  
     @model SubjectViewModel  
     @{  
         ViewBag.Title = "AddMarks";  
     }  
       
       
     @using (Html.BeginForm())  
     {  
        @Html.AntiForgeryToken()  
     
    <div class="form-horizontal">  
     
        <p>@ViewBag.message</p>  
        <hr />  
     
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })  
       <div class="form-group">  
         @Html.LabelFor(model => model.Student.RollNo, htmlAttributes: new { @class = "control-label col-md-2" })  
           <div class="col-md-10" style="float:left">  
                @Html.EditorFor(model => model.Student.RollNo, new { htmlAttributes = new { @class = "form-control" } })  
                @Html.ValidationMessageFor(model => model.Student.RollNo, "", new { @class = "text-danger" })  
            </div>  
       </div>  
     
        <div class="form-group">  
            @Html.LabelFor(model => model.Student.StudentName, htmlAttributes: new { @class = "control-label col-md-2" })  
            <div class="col-md-10" style="float:left">  
                @Html.EditorFor(model => model.Student.StudentName, new { htmlAttributes = new { @class = "form-control" } })  
                @Html.ValidationMessageFor(model => model.Student.StudentName, "", new { @class = "text-danger" })  
            </div>  
        </div>  
     
        <div class="form-group">  
            @Html.LabelFor(model => model.Student.Subject1, htmlAttributes: new { @class = "control-label col-md-2" })  
            <div class="col-md-10">  
     
                @Html.DropDownListFor(model => model.Student.Subject1, new SelectList(Model.Grade, "Value", "Text"), "Select Subject1 Grade", htmlAttributes: new { @class = "form-control" })  
     
                @Html.ValidationMessageFor(model => model.Student.Subject1, "", new { @class = "text-danger" })  
            </div>  
        </div>  
        <div class="form-group">  
           @Html.LabelFor(model => model.Student.Subject2, htmlAttributes: new { @class = "control-label col-md-2" })  
            <div class="col-md-10">  
     
               @Html.DropDownListFor(model => model.Student.Subject2, new SelectList(Model.Grade, "Value", "Text"), "Select Subject2 Grade", htmlAttributes: new { @class = "form-control" })  
     
                @Html.ValidationMessageFor(model => model.Student.Subject2, "", new { @class = "text-danger" })  
            </div>  
        </div>  
        <div class="form-group">  
            @Html.LabelFor(model => model.Student.Subject3, htmlAttributes: new { @class = "control-label col-md-2" })  
            <div class="col-md-10">  
     
                @Html.DropDownListFor(model => model.Student.Subject3, new SelectList(Model.Grade, "Value", "Text"), "Select Subject3 Grade", htmlAttributes: new { @class = "form-control" })  
     
                @Html.ValidationMessageFor(model => model.Student.Subject3, "", new { @class = "text-danger" })  
            </div>  
        </div>  
        <div class="form-group">  
            @Html.LabelFor(model => model.Student.Subject4, htmlAttributes: new { @class = "control-label col-md-2" })  
            <div class="col-md-10">  
     
                @Html.DropDownListFor(model => model.Student.Subject4, new SelectList(Model.Grade, "Value", "Text"), "Select Subject4 Grade", htmlAttributes: new { @class = "form-control" })  
     
                @Html.ValidationMessageFor(model => model.Student.Subject4, "", new { @class = "text-danger" })  
            </div>  
        </div>  
        <div class="form-group">  
            @Html.LabelFor(model => model.Student.Subject5, htmlAttributes: new { @class = "control-label col-md-2" })  
            <div class="col-md-10">  
     
                @Html.DropDownListFor(model => model.Student.Subject5, new SelectList(Model.Grade, "Value", "Text"), "Select Subject5 Grade", htmlAttributes: new { @class = "form-control" })  
     
                @Html.ValidationMessageFor(model => model.Student.Subject5, "", new { @class = "text-danger" })  
            </div>  
        </div>  
     
        <div class="form-group">  
            <div class="col-md-offset-2 col-md-10">  
                <input type="submit" id="Submit" value="Add Marks" class="btn btn-default" />  
            </div>  
       </div>  
    </div>  
     
     
     
    }                
    @\*  
     
        @if (ViewBag.ExpenseList != null)  
        {  
            <table class="table table-bordered">  
                <caption><b>Expense Details</b></caption>  
     
                <thead>  
                    <tr><th>Expense Id</th><th>Date of Expense</th><th>Expense Type</th><th>Amount</th><th>Edit</th></tr>  
                </thead>  
             <tbody>  
                  @foreach (var Expense in ViewBag.ExpenseList)  
                   {  
                       <tr>  
                          <td>@Expense.ExpenseId </td>  
                          <td>@Expense.DateOfExpense</td>  
                          <td>@Expense.ExpenseType</td>  
                           <td>@Expense.Amount</td>  
                           <td>@Html.ActionLink("Edit", "EditExpense", new { id = Expense.ExpenseId })</td>  
                      </tr>  
                   }  
               </tbody>  
           </table>  
     
       }  
   \*@  
     
     
                   
     
Student.cs  
     using System;  
     using System.Collections.Generic;  
     using System.Linq;  
     using System.Web;  
     using System.ComponentModel.DataAnnotations;  
       
     namespace BusinessSchool.Models  
     {  
         public class Student  
       {  
           [Key]  
           public int RollNo { get; set; }  
     
           [Required(ErrorMessage = "Please Enter A Valid Name")]  
           public string StudentName { get; set; }  
     
            [Required(ErrorMessage = "Please Select A Grade")]  
          public string Subject1 { get; set; }  
           [Required(ErrorMessage = "Please Select A Grade")]  
          public string Subject2 { get; set; }  
            [Required(ErrorMessage = "Please Select A Grade")]  
            public string Subject3 { get; set; }  
            [Required(ErrorMessage = "Please Select A Grade")]  
            public string Subject4 { get; set; }  
         [Required(ErrorMessage = "Please Select A Grade")]  
           public string Subject5 { get; set; }  
             
        }  
    }                
     
StudentContext.cs  
     using BusinessSchool.Models;  
     using System;  
     using System.Collections.Generic;  
     using System.Data.Entity;  
     using System.Linq;  
     using System.Text;  
     using System.Threading.Tasks;  
       
     namespace BusinessSchool.Models        //DO NOT change the namespace name  
    {  
        class StudentContext : DbContext     //DO NOT change the class name  
        {  
           
            public StudentContext() : base("DataConnection")   //DO NOT change the connection string name  
            {  
           }  
            public DbSet<Student> Students { get; set; }  
    //Declare property 'Students' of type DbSet<Student>.  
        }  
    }  
     
     
               
     
StudentController.cs  
     using System;  
     using System.Collections.Generic;  
     using System.Linq;  
     using System.Web;  
     using System.Web.Mvc;  
     using BusinessSchool.Models;  
       
     namespace BusinessSchool.Controllers  
     {  
        public class StudentController : Controller  
        {  
            // GET: Student  
           StudentDAO sd = new StudentDAO();  
           public ActionResult Index()  
            {  
                List<Student> ct = null;  
               ct = sd.GetStudentList();  
                ViewBag.StudentList = ct;  
                return View(ct);  
            }  
     
            public ActionResult AddMarks()  
           {  
                var viewModel = new SubjectViewModel();  
                return View("AddMarks", viewModel);  
            }  
           [HttpPost]  
           public ActionResult AddMarks(Student Student)  
          {  
               var viewModel = new SubjectViewModel();  
                Student savedExpense = sd.AddMarks(Student);  
                if (savedExpense != null)  
               {  
                    ViewBag.message = "Marks added Sucessfully.";  
                }  
               else  
               {                
                    ViewBag.message = "Marks addition Failed!.";  
     
                }  
                List<Student> ct = sd.GetStudentList();  
                return View("AddMarks", viewModel);  
           }  
           public ActionResult EditMarks(int id)  
            {  
                var viewModel = new SubjectViewModel();  
              var exp = sd.GetMarksByRollNo(id);  
               viewModel.Student = exp;  
                return View("EditMarks", viewModel);  
            }  
     
            [HttpPost]  
            public ActionResult EditMarks(Student student)  
            {  
                Console.WriteLine("EditMarks " + student.Subject1);  
                sd.UpdateMarks(student);  
                List<Student> ct = sd.GetStudentList();  
                ViewBag.StudentList = ct;  
                //Expense = null;  
                //return RedirectToAction("AddExpense",ct);  
                return View("Index",ct);  
            }  
        }  
    }                
     
SubjectViewModel.cs  
     using System;  
     using System.Collections.Generic;  
     using System.Linq;  
     using System.Web;  
     using System.Web;  
     using System.Web.Mvc;  
       
     namespace BusinessSchool.Models  
     {  
        public class SubjectViewModel  
        {  
            public Student Student { get; set; }  
     
     
           public IEnumerable<SelectListItem> Grade  
            {  
                get  
               {  
                  return new List<SelectListItem>()  
                         {  
                          new SelectListItem { Text = ">=80", Value = "A+" },  
                         new SelectListItem { Text = ">=60 And <80", Value = "A" },  
                            new SelectListItem { Text = ">=40 And <60", Value = "B" },  
                            new SelectListItem { Text = "<40", Value = "F" },  
                       };  
               }  
               set  
               {  
                   }  
     
            }  
        }  
    }                
     
StudentDAO.cs  
     using System;  
     using System.Collections.Generic;  
     using System.Linq;  
     using System.Web;  
       
     namespace BusinessSchool.Models  
     {  
         public class StudentDAO  
         {  
            static int id = 0;  
            public static List<Student> ex = new List<Student>();  
     
            public List<Student> GetStudentList()  
          {  
                StudentContext sc=new StudentContext();  
                List<Student> ex1 = new List<Student>();  
              ex1=sc.Students.ToList();  
              Console.WriteLine("COUNT\*\*\*\*\*\*\*\*\*"+ex1.Count);  
             for (int cnt = 0; cnt < ex1.Count; cnt++)  
               {  
                    Console.WriteLine("GET\*\*\*\*\*\*\*\*\*"+ex1.ElementAt(cnt).RollNo);  
                     
                }  
               return ex1;  
            }  
               public Student AddMarks(Student Student)  
           {  
               try  
                {  
                    //Student.RollNo = ++id;  
                     StudentContext sc=new StudentContext();  
                     Student s1=sc.Students.Add(Student);  
                     sc.SaveChanges();  
                       
                    //ex.Add(Student);  
                }catch(Exception e)  
               {                
                    Console.WriteLine("ADD+++ " +e.Message);  
              }  
     
                return Student;  
            }  
            public Student GetMarksByRollNo(int rollNo)  
            {  
                StudentContext sc=new StudentContext();  
                //Student s1=sc.Students.Find(rollNo);  
                Student s1=(from s in sc.Students where s.RollNo == rollNo select s).FirstOrDefault<Student>();  
               // for (int cnt = 0; cnt < ex.Count; cnt++)  
                // {  
               //     if (ex.ElementAt(cnt).RollNo == rollNo)  
                //     {  
                //         return ex.ElementAt(cnt);  
                //     }  
                // }  
                return s1;  
           }  
            public Student UpdateMarks(Student student)  
            {  
               try{  
                  Console.WriteLine("Student\*\*\*\*\*\*\* " + student.RollNo);  
                    StudentContext sc = new StudentContext();  
                    Student temp=(from s in sc.Students where s.RollNo == student.RollNo select s).FirstOrDefault<Student>();  
                        Console.WriteLine("Student\*\*\*\*\*\*\* " + temp.Subject1+" "+student.Subject1);  
                       temp.Subject1 = student.Subject1;  
                       temp.Subject3 = student.Subject3;  
                       //sc.Entry(temp).CurrentValues.SetValues(student);  
                       sc.SaveChanges();  
              }catch(Exception e)  
                {  
                   Console.WriteLine("UPDATE+++ " +e.Message);  
               }  
               return student;  
           }                
        }  
    }  
EditMarks.cshtml  
       
     @using BusinessSchool.Models  
     @model SubjectViewModel  
       
     @{  
         ViewBag.Title = "EditMarks";  
     }  
       
     <h3>Update Mark Details</h3>  
     @using (Html.BeginForm())  
    {  
        @Html.AntiForgeryToken()  
     
        <div class="form-horizontal">  
     
            <hr />  
            @Html.HiddenFor(model => model.Student.RollNo)  
     
            <div class="form-group">  
                @Html.LabelFor(model => model.Student.StudentName, htmlAttributes: new { @class = "control-label col-md-2" })  
                <div class="col-md-10">  
                    @Html.EditorFor(model => model.Student.StudentName, new { htmlAttributes = new { @class = "form-control" }, @readonly = "readonly" })  
     
                </div>  
            </div>  
     
           <div class="form-group">  
                @Html.LabelFor(model => model.Student.Subject1, htmlAttributes: new { @class = "control-label col-md-2" })  
               <div class="col-md-10">  
                   @Html.DropDownListFor(m => m.Student.Subject1, new SelectList(Model.Grade, "Value", "Text"), Model.Student.Subject1)  
                </div>  
          </div>  
           <div class="form-group">  
                @Html.LabelFor(model => model.Student.Subject2, htmlAttributes: new { @class = "control-label col-md-2" })  
                <div class="col-md-10">  
                  @Html.DropDownListFor(m => m.Student.Subject2, new SelectList(Model.Grade, "Value", "Text"), Model.Student.Subject2)  
                </div>  
            </div>  
            <div class="form-group">  
                @Html.LabelFor(model => model.Student.Subject3, htmlAttributes: new { @class = "control-label col-md-2" })  
                <div class="col-md-10">  
                    @Html.DropDownListFor(m => m.Student.Subject3, new SelectList(Model.Grade, "Value", "Text"), Model.Student.Subject3)  
                </div>  
            </div>  
           <div class="form-group">  
                @Html.LabelFor(model => model.Student.Subject4, htmlAttributes: new { @class = "control-label col-md-2" })  
                <div class="col-md-10">  
                    @Html.DropDownListFor(m => m.Student.Subject4, new SelectList(Model.Grade, "Value", "Text"), Model.Student.Subject4)  
                </div>  
           </div>  
            <div class="form-group">  
                @Html.LabelFor(model => model.Student.Subject5, htmlAttributes: new { @class = "control-label col-md-2" })  
               <div class="col-md-10">  
                    @Html.DropDownListFor(m => m.Student.Subject5, new SelectList(Model.Grade, "Value", "Text"), Model.Student.Subject5)  
                </div>  
            </div>  
     
     
     
     
     
            <div class="form-group">  
               <div class="col-md-offset-2 col-md-10">  
                    <input type="submit" id="Submit" value="Update Marks" class="btn btn-default" />  
                </div>  
            </div>  
        </div>  
   }                
     
     
     
     
Index.cshtml  
       
     @model IEnumerable<BusinessSchool.Models.Student>  
       
     @{  
         ViewBag.Title = "Index";  
     }  
       
     <p>  
         @Html.ActionLink("Add New Marks", "AddMarks")  
    </p>  
    <table class="table">  
        <tr>  
            <th>  
                @Html.DisplayNameFor(model => model.RollNo)  
           </th>  
            <th>  
                @Html.DisplayNameFor(model => model.StudentName)  
           </th>  
            <th>  
                @Html.DisplayNameFor(model => model.Subject1)  
            </th>  
            <th>  
                @Html.DisplayNameFor(model => model.Subject2)  
            </th>  
     
            <th>  
               @Html.DisplayNameFor(model => model.Subject3)  
            </th>  
            <th>  
                @Html.DisplayNameFor(model => model.Subject4)  
            </th>  
           <th>  
               @Html.DisplayNameFor(model => model.Subject5)  
            </th>  
        </tr>  
     
        @foreach (var mark in Model)  
        {                
    <tr>  
        <td>  
            @Html.DisplayFor(modelItem => mark.RollNo)  
       </td>  
        <td>  
            @Html.DisplayFor(modelItem => mark.StudentName)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => mark.Subject1)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => mark.Subject2)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => mark.Subject3)  
        </td>  
        <td>  
           @Html.DisplayFor(modelItem => mark.Subject4)  
        </td>  
        <td>  
            @Html.DisplayFor(modelItem => mark.Subject5)  
        <td>  
            @Html.ActionLink("Edit", "EditMarks", new { id = mark.RollNo })  
        </td>  
    </tr>  
        }  
     
    </table>

     
                 

|  |  |
| --- | --- |
| https://ssl.gstatic.com/ui/v1/icons/mail/no_photo.png | ReplyForward |