```
using Quosten2.Models;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Quosten2.Controllers
{
    public class StudentsController : Controller
    {
        // GET: Students
        public ActionResult ShowStudent()
        {
            SqlConnection cnn = new SqlConnection();
            cnn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
            Catalog=ExamPractice;Integrated Security=True;Connect
            Timeout=30;Encrypt=False;TrustServerCertificate=False;";
            cnn.Open();
            SqlCommand cmdShow = new SqlCommand();
            cmdShow.Connection = cnn;
            cmdShow.CommandType = System.Data.CommandType.Text;
            cmdShow.CommandText = "Select * from Students";
         List<Student> stud = new List<Student>();
            SqlDataReader dr = cmdShow.ExecuteReader();
            while(dr.Read())
            {
                stud.Add(new Student { StudentId = (int)dr["StudentId"], Name =
                dr["Name"].ToString(), JavaMarks = (int)dr["JavaMarks"], DotNetMarks =
                (int)dr["DotNetMarks"]});
            }
            dr.Close();
            cnn.Close();
            return View(stud);
        }

        // GET: Students/Details/5
        public ActionResult StudentDetails(int id)
        {
            SqlConnection cnn = new SqlConnection();
            cnn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
            Catalog=ExamPractice;Integrated Security=True;Connect
            Timeout=30;Encrypt=False;TrustServerCertificate=False;";
            cnn.Open();
            SqlCommand cmdShow = new SqlCommand();
            cmdShow.Connection = cnn;
            cmdShow.CommandType = System.Data.CommandType.Text;
            cmdShow.CommandText = "Select * from Students where StudentId=@StudentId";
            cmdShow.Parameters.AddWithValue("@StudentId",id);
            SqlDataReader dr = cmdShow.ExecuteReader();
            Student stud = null;
            if (dr.Read())
            {
                stud=new Student { StudentId = (int)dr["StudentId"], Name =
                dr["Name"].ToString(), JavaMarks = (int)dr["JavaMarks"], DotNetMarks =
                (int)dr["DotNetMarks"] };
            }
            dr.Close();
            cnn.Close();
            return View(stud);

        }

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

        // POST: Students/Create
```

```
        [HttpPost]
        public ActionResult AddStudent(Student obj)
        {
            SqlConnection cnn = new SqlConnection();
            cnn.ConnectionString= @"Data Source=(localdb)\MSSQLLocalDB;Initial
            Catalog=ExamPractice;Integrated Security=True;Connect
            Timeout=30;Encrypt=False;TrustServerCertificate=False;";
            cnn.Open();
            SqlCommand addStudent = new SqlCommand();
            addStudent.Connection = cnn;
            addStudent.CommandType = System.Data.CommandType.StoredProcedure;
            addStudent.CommandText = "Insert";
            addStudent.Parameters.AddWithValue("@Name", obj.Name);
            addStudent.Parameters.AddWithValue("@JavaMarks", obj.JavaMarks);
            addStudent.Parameters.AddWithValue("@DotNetMarks", obj.DotNetMarks);
            try
            {
                addStudent.ExecuteNonQuery();
                // TODO: Add insert logic here

                return RedirectToAction("ShowStudent");
            }
            catch(Exception ex)
            {
                ViewBag.msg = ex.Message;
                return View();
            }
            finally
            {
                cnn.Close();
            }
        }

        // GET: Students/Edit/5
        public ActionResult UpdateStudent(int id)
        {
            SqlConnection cnn = new SqlConnection();
            cnn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
            Catalog=ExamPractice;Integrated Security=True;Connect
            Timeout=30;Encrypt=False;TrustServerCertificate=False;";
            cnn.Open();
            SqlCommand cmdShow = new SqlCommand();
            cmdShow.Connection = cnn;
            cmdShow.CommandType = System.Data.CommandType.Text;
            cmdShow.CommandText = "Select * from Students where StudentId=@StudentId";
            cmdShow.Parameters.AddWithValue("@StudentId", id);
            SqlDataReader dr = cmdShow.ExecuteReader();
            Student stud = null;
            if (dr.Read())
            {
                stud = new Student { StudentId = (int)dr["StudentId"], Name =
                dr["Name"].ToString(), JavaMarks = (int)dr["JavaMarks"], DotNetMarks =
                (int)dr["DotNetMarks"] };
            }
            dr.Close();
            cnn.Close();
            return View(stud);
        }

        // POST: Students/Edit/5
        [HttpPost]
        public ActionResult UpdateStudent(int id,Student obj)
        {
            SqlConnection cnn = new SqlConnection();
            cnn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
            Catalog=ExamPractice;Integrated Security=True;Connect
            Timeout=30;Encrypt=False;TrustServerCertificate=False;";
            cnn.Open();
            SqlCommand updateStudent = new SqlCommand();
            updateStudent.Connection = cnn;
            updateStudent.CommandType = System.Data.CommandType.Text;
```

```
        updateStudent.CommandText = "Update Students Set
        Name=@Name,JavaMarks=@JavaMarks,DotNetMarks=@DotNetMarks where
        StudentId=@StudentId";
        updateStudent.Parameters.AddWithValue("@StudentId", id);
        updateStudent.Parameters.AddWithValue("@Name", obj.Name);
        updateStudent.Parameters.AddWithValue("@JavaMarks", obj.JavaMarks);
        updateStudent.Parameters.AddWithValue("@DotNetMarks", obj.DotNetMarks);
        try
        {
            updateStudent.ExecuteNonQuery();
            // TODO: Add insert logic here

            return RedirectToAction("ShowStudent");
        }
        catch (Exception ex)
        {
            ViewBag.msg = ex.Message;
            return View();
        }
        finally
        {
            cnn.Close();
        }
    }

    // GET: Students/Delete/5
    public ActionResult DeleteStudent(int id)
    {
        SqlConnection cnn = new SqlConnection();
        cnn.ConnectionString= @"Data Source=(localdb)\MSSQLLocalDB;Initial
        Catalog=ExamPractice;Integrated Security=True;Connect
        Timeout=30;Encrypt=False;TrustServerCertificate=False;";
        cnn.Open();
        SqlCommand cmdDelete = new SqlCommand();
        cmdDelete.Connection = cnn;
        cmdDelete.CommandType = System.Data.CommandType.Text;
        cmdDelete.CommandText = "Select * from Students where StudentId=@StudentId";
        cmdDelete.Parameters.AddWithValue("@StudentId", id);
        SqlDataReader dr = cmdDelete.ExecuteReader();
        Student stud = null;
        if(dr.Read())
        {
            stud = new Student { Name = dr["Name"].ToString(), JavaMarks =
            (int)dr["JavaMarks"], DotNetMarks = (int)dr["DotNetMarks"] };
        }

        dr.Close();
        cnn.Close();
        return View(stud);
    }

    // POST: Students/Delete/5
    [HttpPost]
    public ActionResult DeleteStudent(int id, FormCollection collection)
    {
        SqlConnection cnn = new SqlConnection();
        cnn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
        Catalog=ExamPractice;Integrated Security=True;Connect
        Timeout=30;Encrypt=False;TrustServerCertificate=False;";
        cnn.Open();
        SqlCommand cmdDelete = new SqlCommand();
        cmdDelete.Connection = cnn;
        cmdDelete.CommandType = System.Data.CommandType.Text;
        cmdDelete.CommandText = "Delete from Students where StudentId=@StudentId";
        cmdDelete.Parameters.AddWithValue("@StudentId", id);
        try
        {
            // TODO: Add delete logic here
            cmdDelete.ExecuteNonQuery();
            return RedirectToAction("ShowStudent");
        }
```

```
            catch(Exception e)
            {
                ViewBag.msg = e.Message;
                return View();
            }
            finally
            {
                cnn.Close();
            }
        }
    }
}
```