GITHUB LINK- https://github.com/DurgeshSingh01/Simplilearn_Practice_Project/tree/master/Phase_3_Projects/MVC_School_Application

Source Code-

SchoolBAL-

```
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace School_BAl
{
    public class ClassBAL
    {
        public int ClassID { get; set; }
        public string ClassName { get; set; }
    }
    public class StudentsBAL
    {
        public int StudentID { get; set; }
        public string StudentName { get; set; }
        public int ClassID { get; set; }
    }
    public class SubjectsBAL
    {
        public int SubjectID { get; set; }
        public string SubjectName { get; set; }
        public int ClassID { get; set; }
    }
}
```

SchoolDAL:

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using School_BAl;
namespace School_DAL
{
    public class ClassDAL
    {
        public bool InsertClass(ClassBAL Class)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdInsert = new SqlCommand("insert into
class(classid,classname) values(@classid,@classname)", cn);
            cmdInsert.Parameters.AddWithValue("@classid", Class.ClassID);
            cmdInsert.Parameters.AddWithValue("@classname",
Class.ClassName);

            cn.Open();
            int i = cmdInsert.ExecuteNonQuery();
            bool status = false;
            if (i == 1)
            {
                status = true;
            }
            cn.Close();
            cn.Dispose();
            return status;
        }
        public int ClassCount()
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmd = new SqlCommand("select count(*) from Class",
cn);
            cn.Open();

            int cnt = (int)cmd.ExecuteScalar();
            cn.Close();
            cn.Dispose();
            return cnt;
        }
        public bool UpdateClass(ClassBAL Class)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdUpdate = new SqlCommand("[dbo].[sp_UpdateClass]",
cn);

            cmdUpdate.CommandType =
System.Data.CommandType.StoredProcedure;
            cmdUpdate.Parameters.AddWithValue("@p_classid", Class.ClassID);
            cmdUpdate.Parameters.AddWithValue("@p_classname",
Class.ClassName);

            cn.Open();
            int i = cmdUpdate.ExecuteNonQuery();
            bool status = false;
            if (i == 1)
            {
                status = true;
            }
            cn.Close();
            cn.Dispose();
            return status;
        }
        public bool DeleteClass(int classid)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdDelete = new SqlCommand("[dbo].[sp_DeleteClass]",
cn);
```

```
            cmdDelete.CommandType =
System.Data.CommandType.StoredProcedure;
            cmdDelete.Parameters.AddWithValue("@p_classid", classid);
            cn.Open();
            int i = cmdDelete.ExecuteNonQuery();
            bool status = false;
            if (i == 1)
            {
                status = true;
            }
            cn.Close();
            cn.Dispose();
            return status;
        }
        public ClassBAL FindClass(int classid)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);
            SqlCommand cmdSelect = new SqlCommand("[dbo].[sp_FindClass]",
cn);
            cmdSelect.CommandType =
System.Data.CommandType.StoredProcedure;
            cmdSelect.Parameters.AddWithValue("@p_classid", classid);
            SqlParameter p1 = new SqlParameter();
            p1.ParameterName = "@p_classname";
            p1.SqlDbType = System.Data.SqlDbType.NVarChar;
            p1.Size = 5;
            p1.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p1);

            cn.Open();
            cmdSelect.ExecuteNonQuery();

            ClassBAL found = new ClassBAL();

            found.ClassName = p1.Value.ToString();
            cn.Close();
            cn.Dispose();
            return found;
        }
        public void FindClass(int classid, out ClassBAL data)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);
            SqlCommand cmdSelect = new SqlCommand("[dbo].[sp_FindClass]",
cn);
            cmdSelect.CommandType =
System.Data.CommandType.StoredProcedure;
            cmdSelect.Parameters.AddWithValue("@p_classid", classid);
            SqlParameter p1 = new SqlParameter();
            p1.ParameterName = "@p_classname";
            p1.SqlDbType = System.Data.SqlDbType.NVarChar;
            p1.Size = 5;
            p1.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p1);

            cn.Open();
            cmdSelect.ExecuteNonQuery();

            data = new ClassBAL();
            data.ClassName = p1.Value.ToString();

            ClassBAL edata = new ClassBAL();
            edata = data;

            cn.Close();
            cn.Dispose();
        }
        public List<ClassBAL> ClassList()
        {
            SqlConnection cn = new 
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdlist = new SqlCommand("select * from 
[dbo].[fn_Classlist]()", cn);
            cn.Open();
            SqlDataReader dr = cmdlist.ExecuteReader();
            List<ClassBAL> list = new List<ClassBAL>();
            if (dr.HasRows)
            {
                while (dr.Read())
                {
                    ClassBAL bal = new ClassBAL();
                    bal.ClassID = Convert.ToInt32(dr["ClassID"]);
                    bal.ClassName = dr["ClassName"].ToString();

                    list.Add(bal);
                }
            }
            cn.Close();
            cn.Dispose();
            return list;
        }
    }


    //==============================================================
    public class StudentsDAL
    {
        public bool InsertStudent(StudentsBAL s)
        {
            SqlConnection cn = new 
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdInsert = new SqlCommand("insert into 
students(studentid,studentname,classid) 
values(@studentid,@studentname,@classid)", cn);
            cmdInsert.Parameters.AddWithValue("@studentid", s.StudentID);
            cmdInsert.Parameters.AddWithValue("@studentname", 
s.StudentName);
            cmdInsert.Parameters.AddWithValue("@classid", s.ClassID);

            cn.Open();
            int i = cmdInsert.ExecuteNonQuery();
            bool status = false;
            if (i == 1)
            {
                status = true;
            }
            cn.Close();
            cn.Dispose();
            return status;
        }
        public int StudentCount()
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmd = new SqlCommand("select count(*) from
Students", cn);
            cn.Open();

            int cnt = (int)cmd.ExecuteScalar();
            cn.Close();
            cn.Dispose();
            return cnt;
        }
        public bool UpdateStudent(StudentsBAL s)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdUpdate = new
SqlCommand("[dbo].[sp_UpdateStudents]", cn);

            cmdUpdate.CommandType =
System.Data.CommandType.StoredProcedure;

            cmdUpdate.Parameters.AddWithValue("@p_studentid", s.StudentID);
            cmdUpdate.Parameters.AddWithValue("@p_studentname",
s.StudentName);
            cmdUpdate.Parameters.AddWithValue("@p_classid", s.ClassID);

            cn.Open();
            int i = cmdUpdate.ExecuteNonQuery();
            bool status = false;
            if (i == 1)
            {
                status = true;
            }
            cn.Close();
            cn.Dispose();
            return status;
        }
        public bool DeleteStudent(int id)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdDelete = new
SqlCommand("[dbo].[sp_DeleteStudents]", cn);
            cmdDelete.CommandType =
System.Data.CommandType.StoredProcedure;
            cmdDelete.Parameters.AddWithValue("@p_studentid", id);
            cn.Open();
            int i = cmdDelete.ExecuteNonQuery();
            bool status = false;
            if (i == 1)
            {
                status = true;
            }
            cn.Close();
            cn.Dispose();
            return status;
        }
        public StudentsBAL FindStudent(int id)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);
            SqlCommand cmdSelect = new
SqlCommand("[dbo].[sp_FindStudents]", cn);
            cmdSelect.CommandType =
System.Data.CommandType.StoredProcedure;
            cmdSelect.Parameters.AddWithValue("@p_studentid", id);

            SqlParameter p1 = new SqlParameter();
            p1.ParameterName = "@p_studentname";
            p1.SqlDbType = System.Data.SqlDbType.NVarChar;
            p1.Size = 20;
            p1.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p1);

            SqlParameter p2 = new SqlParameter();
            p2.ParameterName = "@p_classid";
            p2.SqlDbType = System.Data.SqlDbType.NVarChar;
            p2.Size = 4;
            p2.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p2);

            cn.Open();
            cmdSelect.ExecuteNonQuery();

            StudentsBAL f = new StudentsBAL();

            f.StudentName = p1.Value.ToString();
            f.ClassID = Convert.ToInt32(p2.Value);
            cn.Close();
            cn.Dispose();
            return f;
        }
        public void FindStudent(int id, out StudentsBAL data)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);
            SqlCommand cmdSelect = new
SqlCommand("[dbo].[sp_FindStudents]", cn);
            cmdSelect.CommandType =
System.Data.CommandType.StoredProcedure;
            cmdSelect.Parameters.AddWithValue("@p_studentid", id);

            SqlParameter p1 = new SqlParameter();
            p1.ParameterName = "@p_studentname";
            p1.SqlDbType = System.Data.SqlDbType.NVarChar;
            p1.Size = 20;
            p1.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p1);

            SqlParameter p2 = new SqlParameter();
            p2.ParameterName = "@p_classid";
            p2.SqlDbType = System.Data.SqlDbType.NVarChar;
            p2.Size = 4;
            p2.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p2);

            cn.Open();
            cmdSelect.ExecuteNonQuery();

            data = new StudentsBAL();

            data.StudentName = p1.Value.ToString();
            data.ClassID = Convert.ToInt32(p2.Value);

            StudentsBAL edata = new StudentsBAL();
            edata = data;

            cn.Close();
            cn.Dispose();
        }
        public List<StudentsBAL> StudentList()
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdlist = new SqlCommand("select * from
[dbo].[fn_Studentslist]()", cn);
            cn.Open();
            SqlDataReader dr = cmdlist.ExecuteReader();
            List<StudentsBAL> list = new List<StudentsBAL>();
            if (dr.HasRows)
            {
                while (dr.Read())
                {
                    StudentsBAL bal = new StudentsBAL();
                    bal.StudentID = Convert.ToInt32(dr["StudentID"]);
                    bal.StudentName = dr["StudentName"].ToString();
                    bal.ClassID = Convert.ToInt32(dr["ClassID"]);

                    list.Add(bal);
                }
            }
            cn.Close();
            cn.Dispose();
            return list;
        }
    }
    //===============================================================

    public class SubjectsDAL
    {
        public bool InsertSubjects(SubjectsBAL s)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdInsert = new SqlCommand("insert into
subjects(subjectid,subjectname,classid)
values(@subjectid,@subjectname,@classid)", cn);
            cmdInsert.Parameters.AddWithValue("@subjectid", s.SubjectID);
            cmdInsert.Parameters.AddWithValue("@subjectname",
s.SubjectName);
            cmdInsert.Parameters.AddWithValue("@classid", s.ClassID);

            cn.Open();
            int i = cmdInsert.ExecuteNonQuery();
            bool status = false;
            if (i == 1)
            {
                status = true;
            }
            cn.Close();
            cn.Dispose();
            return status;
        }
        public int SubjectsCount()
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmd = new SqlCommand("select count(*) from
Subjects", cn);
            cn.Open();

            int cnt = (int)cmd.ExecuteScalar();
            cn.Close();
            cn.Dispose();
            return cnt;
        }
        public bool UpdateSubjects(SubjectsBAL s)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdUpdate = new
SqlCommand("[dbo].[sp_UpdateSubjects]", cn);

            cmdUpdate.CommandType =
System.Data.CommandType.StoredProcedure;

            cmdUpdate.Parameters.AddWithValue("@p_subjectid", s.SubjectID);
            cmdUpdate.Parameters.AddWithValue("@p_subjectname",
s.SubjectName);
            cmdUpdate.Parameters.AddWithValue("@p_classid", s.ClassID);

            cn.Open();
            int i = cmdUpdate.ExecuteNonQuery();
            bool status = false;
            if (i == 1)
            {
                status = true;
            }
            cn.Close();
            cn.Dispose();
            return status;
        }
        public bool DeleteSubjects(int id)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdDelete = new
SqlCommand("[dbo].[sp_DeleteSubjects]", cn);
            cmdDelete.CommandType =
System.Data.CommandType.StoredProcedure;
```

```
            cmdDelete.Parameters.AddWithValue("@p_subjectid", id);
            cn.Open();
            int i = cmdDelete.ExecuteNonQuery();
            bool status = false;
            if (i == 1)
            {
                status = true;
            }
            cn.Close();
            cn.Dispose();
            return status;
        }
        public SubjectsBAL FindSubjects(int id)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);
            SqlCommand cmdSelect = new
SqlCommand("[dbo].[sp_FindSubjects]", cn);
            cmdSelect.CommandType =
System.Data.CommandType.StoredProcedure;
            cmdSelect.Parameters.AddWithValue("@p_subjectid", id);

            SqlParameter p1 = new SqlParameter();
            p1.ParameterName = "@p_subjectname";
            p1.SqlDbType = System.Data.SqlDbType.NVarChar;
            p1.Size = 50;
            p1.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p1);

            SqlParameter p2 = new SqlParameter();
            p2.ParameterName = "@p_classid";
            p2.SqlDbType = System.Data.SqlDbType.NVarChar;
            p2.Size = 4;
            p2.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p2);

            cn.Open();
            cmdSelect.ExecuteNonQuery();

            SubjectsBAL f = new SubjectsBAL();

            f.SubjectName = p1.Value.ToString();
            f.ClassID = Convert.ToInt32(p2.Value);
            cn.Close();
            cn.Dispose();
            return f;
        }
        public void FindSubjects(int id, out SubjectsBAL data)
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);
            SqlCommand cmdSelect = new
SqlCommand("[dbo].[sp_FindSubjects]", cn);
            cmdSelect.CommandType =
System.Data.CommandType.StoredProcedure;
            cmdSelect.Parameters.AddWithValue("@p_subjectid", id);

            SqlParameter p1 = new SqlParameter();
            p1.ParameterName = "@p_subjectname";
            p1.SqlDbType = System.Data.SqlDbType.NVarChar;
            p1.Size = 50;
            p1.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p1);

            SqlParameter p2 = new SqlParameter();
            p2.ParameterName = "@p_classid";
            p2.SqlDbType = System.Data.SqlDbType.NVarChar;
            p2.Size = 4;
            p2.Direction = System.Data.ParameterDirection.Output;
            cmdSelect.Parameters.Add(p2);

            cn.Open();
            cmdSelect.ExecuteNonQuery();

            data = new SubjectsBAL();

            data.SubjectName = p1.Value.ToString();
            data.ClassID = Convert.ToInt32(p2.Value);

            SubjectsBAL edata = new SubjectsBAL();
            edata = data;

            cn.Close();
            cn.Dispose();
        }
        public List<SubjectsBAL> SubjectsList()
        {
            SqlConnection cn = new
SqlConnection(ConfigurationManager.ConnectionStrings["RSCnString"].Connect
ionString);

            SqlCommand cmdlist = new SqlCommand("select * from
[dbo].[fn_Subjectslist]()", cn);
            cn.Open();
            SqlDataReader dr = cmdlist.ExecuteReader();
            List<SubjectsBAL> list = new List<SubjectsBAL>();
            if (dr.HasRows)
            {
                while (dr.Read())
                {
                    SubjectsBAL bal = new SubjectsBAL();
                    bal.SubjectID = Convert.ToInt32(dr["SubjectID"]);
                    bal.SubjectName = dr["SubjectName"].ToString();
                    bal.ClassID = Convert.ToInt32(dr["ClassID"]);

                    list.Add(bal);
                }
            }
            cn.Close();
            cn.Dispose();
            return list;
        }
    }
}
```

SchoolHelper:

```
using School_BAl;
using School_DAL;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace School_Helper
{
    public class ClassHelper
    {
        ClassDAL dal = null;
        public ClassHelper()
        {
            dal = new ClassDAL();
        }
        public bool AddClass(ClassBAL Class)
        {
            return dal.InsertClass(Class);
        }
        public bool EditClass(ClassBAL Class)
        {
            return dal.UpdateClass(Class);
        }
        public bool RemoveClass(int classid)
        {
            return dal.DeleteClass(classid);
        }
        public ClassBAL SearchClass(int classid)
        {
            return dal.FindClass(classid);
        }
        public void SearchClass(int classid, out ClassBAL data)
        {

            dal.FindClass(classid, out data);
        }
        public int countClass()
        {
            return dal.ClassCount();

        }
        public List<ClassBAL> ShowClassList()
        {
            return dal.ClassList();
        }
    }


    public class StudentsHelper
    {
        StudentsDAL dal = null;
        public StudentsHelper()
        {
            dal = new StudentsDAL();
        }
        public bool AddStudent(StudentsBAL s)
        {
            return dal.InsertStudent(s);
        }
        public bool EditStudent(StudentsBAL s)
        {
            return dal.UpdateStudent(s);
        }
        public bool RemoveStudent(int id)
        {
            return dal.DeleteStudent(id);
        }
        public StudentsBAL SearchStudent(int id)
        {
            return dal.FindStudent(id);
        }
        public void SearchStudent(int id, out StudentsBAL data)
        {

            dal.FindStudent(id, out data);
        }
        public int countStudent()
        {
            return dal.StudentCount();

        }
        public List<StudentsBAL> ShowStudentList()
        {
            return dal.StudentList();
        }
    }


    public class SubjectsHelper
    {
        SubjectsDAL dal = null;
        public SubjectsHelper()
        {
            dal = new SubjectsDAL();
        }
        public bool AddSubject(SubjectsBAL s)
        {
            return dal.InsertSubjects(s);
        }
        public bool EditSubject(SubjectsBAL s)
        {
            return dal.UpdateSubjects(s);
        }
        public bool RemoveSubject(int id)
        {
            return dal.DeleteSubjects(id);
        }
        public SubjectsBAL SearchSubject(int id)
        {
            return dal.FindSubjects(id);
        }
        public void SearchSubjects(int id, out SubjectsBAL data)
        {

            dal.FindSubjects(id, out data);
        }
        public int countSubject()
        {
            return dal.SubjectsCount();

        }
        public List<SubjectsBAL> ShowSubjectList()
        {
            return dal.SubjectsList();
        }
    }
}
```

App.Config:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
	<connectionStrings>
		<add connectionString="Data Source = DESKTOP-
QQFLCSA\SQLEXPRESS; Initial Catalog = Rainbow_School; Integrated Security
= True" name="RSCnString" providerName="System.Data.SqlClient"></add>
	</connectionStrings>
</configuration>
```

Models:

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

namespace MVCApplication.Models
{
    public class ClassModel
    {
        public int ClassID { get; set; }
        public string ClassName { get; set; }
    }
}
------------------------------------------------------------------
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace MVCApplication.Models
{
    public class StudentsModel
    {
        public int StudentID { get; set; }
        public string StudentName { get; set; }
        public int ClassID { get; set; }
    }
}
---------------------------------------------------------------------
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace MVCApplication.Models
{
    public class SubjectsModel
    {
        public int SubjectID { get; set; }
        public string SubjectName { get; set; }
        public int ClassID { get; set; }

    }
}
```

Controller:

ClassController.cs

```
using MVCApplication.Models;
using School_BAl;
using School_Helper;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace MVCApplication.Controllers
{
    public class ClassController : Controller
    {
        // GET: Class
        ClassHelper helper = null;
        public ClassController()
        {
            helper = new ClassHelper();
        }

        public ActionResult Index()
        {
            var list = helper.ShowClassList();
            List<ClassModel> modelsList = new List<ClassModel>();
            foreach (var item in list)
            {
                modelsList.Add(new ClassModel { ClassID = item.ClassID,
ClassName = item.ClassName });
            }
            return View(modelsList);
        }

        // GET: Class/Details/5
        public ActionResult Details(int id)
        {
            var data = helper.SearchClass(id);
            ClassModel model = new ClassModel();
            model.ClassID = id;
            model.ClassName = data.ClassName;

            return View(model);
        }

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

        // POST: Class/Create
        [HttpPost]
        public ActionResult Create(FormCollection collection)
        {
            try
            {
                // TODO: Add insert logic here

                ClassBAL bal = new ClassBAL();
                bal.ClassID = Convert.ToInt32(Request["ClassID"]);
                bal.ClassName = Request["ClassName"].ToString();

                bool ans = helper.AddClass(bal);
                if (ans)
                {
                    return RedirectToAction("Index");
                }
                else
                {
                    return View();
                }
            }
            catch
            {
                return View();
            }

        }

        // GET: Class/Edit/5
        public ActionResult Edit(int id)
        {
            var c = helper.SearchClass(id);
            ClassModel model = new ClassModel();
            model.ClassID = id;
            model.ClassName = c.ClassName;

            return View(model);
        }

        // POST: Class/Edit/5
        [HttpPost]
        public ActionResult Edit(int id, FormCollection collection)
        {
            try
            {
                // TODO: Add update logic here
                var c = helper.SearchClass(id);
                c.ClassID = Convert.ToInt32(Request["ClassID"]);
                c.ClassName = Request["ClassName"].ToString();

                bool ans = helper.EditClass(c);

                if (ans)
                {
                    return RedirectToAction("Index");
                }
                else
                {
                    return View();
                }
            }
            catch
            {
                return View();
            }
        }

        // GET: Class/Delete/5
        public ActionResult Delete(int id)
        {
            var c = helper.SearchClass(id);
            ClassModel model = new ClassModel();
            model.ClassID = id;
            model.ClassName = c.ClassName;

            return View(model);
        }

        // POST: Class/Delete/5
        [HttpPost]
        public ActionResult Delete(int id, FormCollection collection)
        {
            try
            {
                // TODO: Add delete logic here
                var dataFound = helper.SearchClass(id);
                if (dataFound != null)
                {
                    bool ans = helper.RemoveClass(id);
                    if (ans)
                    {
                        return RedirectToAction("Index");
                    }
                    else
                    {
                        return View();
                    }
                }

                return RedirectToAction("Index");
            }
            catch
            {
                return View();
            }
        }
    }
}
```

StudentsController.cs

```
using MVCApplication.Models;
using School_BAl;
using School_Helper;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace MVCApplication.Controllers
{
    public class StudentsController : Controller
    {
        // GET: Students
        StudentsHelper helper = null;
        public StudentsController()
        {
            helper = new StudentsHelper();
        }
        public ActionResult Index()
        {
            var list = helper.ShowStudentList();
            List<StudentsModel> modelsList = new List<StudentsModel>();
            foreach (var item in list)
            {
                modelsList.Add(new StudentsModel { StudentID =
item.StudentID, StudentName = item.StudentName, ClassID = item.ClassID });
            }
            return View(modelsList);
        }

        // GET: Students/Details/5
        public ActionResult Details(int id)
        {
            var data = helper.SearchStudent(id);
            StudentsModel model = new StudentsModel();
            model.StudentID = id;
            model.StudentName = data.StudentName;
            model.ClassID = data.ClassID;

            return View(model);
        }

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

        // POST: Students/Create
        [HttpPost]
        public ActionResult Create(FormCollection collection)
        {
            try
            {
                // TODO: Add insert logic here

                StudentsBAL bal = new StudentsBAL();
                bal.StudentID = Convert.ToInt32(Request["StudentID"]);
                bal.StudentName = Request["StudentName"].ToString();
                bal.ClassID = Convert.ToInt32(Request["ClassID"]);

                bool ans = helper.AddStudent(bal);
                if (ans)
                {
                    return RedirectToAction("Index");
                }
                else
                {
                    return View();
                }
            }
            catch
            {
                return View();
            }
        }
```

```
// GET: Students/Edit/5
public ActionResult Edit(int id)
{
    var c = helper.SearchStudent(id);
    StudentsModel model = new StudentsModel();
    model.StudentID = id;
    model.StudentName = c.StudentName;
    model.ClassID = c.ClassID;

    return View(model);
}

// POST: Students/Edit/5
[HttpPost]
public ActionResult Edit(int id, FormCollection collection)
{
    try
    {
        // TODO: Add update logic here
        var c = helper.SearchStudent(id);
        c.StudentID = Convert.ToInt32(Request["StudentID"]);
        c.StudentName = Request["StudentName"].ToString();
        c.ClassID = Convert.ToInt32(Request["ClassID"]);

        bool ans = helper.EditStudent(c);

        if (ans)
        {
            return RedirectToAction("Index");
        }
        else
        {
            return View();
        }
    }
    catch
    {
        return View();
    }
}

// GET: Students/Delete/5
public ActionResult Delete(int id)
{
    var c = helper.SearchStudent(id);
    StudentsModel model = new StudentsModel();
    model.StudentID = id;
    model.StudentName = c.StudentName;
    model.ClassID = c.ClassID;

    return View(model);
}

// POST: Students/Delete/5
[HttpPost]
public ActionResult Delete(int id, FormCollection collection)
{
    try
    {
        // TODO: Add delete logic here
        var dataFound = helper.SearchStudent(id);
        if (dataFound != null)
        {
```

```
                    bool ans = helper.RemoveStudent(id);
                    if (ans)
                    {
                        return RedirectToAction("Index");
                    }
                    else
                    {
                        return View();
                    }
                }

                return RedirectToAction("Index");
            }
            catch
            {
                return View();
            }
        }
    }
}
```

SubjectsController.cs

```
using MVCApplication.Models;
using School_BAl;
using School_Helper;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace MVCApplication.Controllers
{
    public class SubjectsController : Controller
    {
        // GET: Subjects
        SubjectsHelper helper = null;
        public SubjectsController()
        {
            helper = new SubjectsHelper();
        }
        public ActionResult Index()
        {
            var list = helper.ShowSubjectList();
            List<SubjectsModel> modelsList = new List<SubjectsModel>();
            foreach (var item in list)
            {
                modelsList.Add(new SubjectsModel { SubjectID =
item.SubjectID, SubjectName = item.SubjectName, ClassID = item.ClassID });
            }
            return View(modelsList);
        }

        // GET: Subjects/Details/5
        public ActionResult Details(int id)
        {
            var data = helper.SearchSubject(id);
            SubjectsModel model = new SubjectsModel();
            model.SubjectID = id;
            model.SubjectName = data.SubjectName;
            model.ClassID = data.ClassID;

            return View(model);
        }

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

        // POST: Subjects/Create
        [HttpPost]
        public ActionResult Create(FormCollection collection)
        {
            try
            {
                // TODO: Add insert logic here

                SubjectsBAL bal = new SubjectsBAL();
                bal.SubjectID = Convert.ToInt32(Request["SubjectID"]);
                bal.SubjectName = Request["SubjectName"].ToString();
                bal.ClassID = Convert.ToInt32(Request["ClassID"]);

                bool ans = helper.AddSubject(bal);
                if (ans)
                {
                    return RedirectToAction("Index");
                }
                else
                {
                    return View();
                }
            }
            catch
            {
                return View();
            }
        }

        // GET: Subjects/Edit/5
        public ActionResult Edit(int id)
        {
            var c = helper.SearchSubject(id);
            SubjectsModel model = new SubjectsModel();
            model.SubjectID = id;
            model.SubjectName = c.SubjectName;
            model.ClassID = c.ClassID;

            return View(model);
        }

        // POST: Subjects/Edit/5
        [HttpPost]
        public ActionResult Edit(int id, FormCollection collection)
        {
            try
            {
                // TODO: Add update logic here
                var c = helper.SearchSubject(id);
                c.SubjectID = Convert.ToInt32(Request["SubjectID"]);
```

```
                c.SubjectName = Request["SubjectName"].ToString();
                c.ClassID = Convert.ToInt32(Request["ClassID"]);

                bool ans = helper.EditSubject(c);

                if (ans)
                {
                    return RedirectToAction("Index");
                }
                else
                {
                    return View();
                }
            }
            catch
            {
                return View();
            }
        }

        // GET: Subjects/Delete/5
        public ActionResult Delete(int id)
        {
            var c = helper.SearchSubject(id);
            SubjectsModel model = new SubjectsModel();
            model.SubjectID = id;
            model.SubjectName = c.SubjectName;
            model.ClassID = c.ClassID;

            return View(model);
        }

        // POST: Subjects/Delete/5
        [HttpPost]
        public ActionResult Delete(int id, FormCollection collection)
        {
            try
            {
                // TODO: Add delete logic here
                var dataFound = helper.SearchSubject(id);
                if (dataFound != null)
                {
                    bool ans = helper.RemoveSubject(id);
                    if (ans)
                    {
                        return RedirectToAction("Index");
                    }
                    else
                    {
                        return View();
                    }
                }

                return RedirectToAction("Index");
            }
            catch
            {
                return View();
            }
        }
    }
}
```