Model

Class Product

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

namespace _210940320048.Models
{
    public class Products
    {


        public int ProductId { get; set; }


        [Required(ErrorMessage ="Enter Product Name") ]
        [Display(Name = "Product Name ")]
        public string ProductName { get; set; }



        [Required(ErrorMessage = "Enter Rate ..!")]
        [Display(Name = "Rate  ")]
        public decimal Rate { get; set; }



        [Required(ErrorMessage = "Enter Description..!")]
        [Display(Name = "Description  ")]
        public string Description { get; set; }


        [Required(ErrorMessage = "Enter Category Name..!")]
        [Display(Name = "Category Name ")]

        public string CategoryName { get; set; }

    }
}
```

Table

```
CREATE TABLE [dbo].[Products] (
    [ProductId]    INT              IDENTITY (101, 200) NOT NULL,
    [ProductName]  VARCHAR (50)     NOT NULL,
    [Rate]         DECIMAL (18, 2) NOT NULL,
    [Description]  VARCHAR (50)     NOT NULL,
    [CategoryName] VARCHAR (50)     NOT NULL,
    PRIMARY KEY CLUSTERED ([ProductId] ASC)
);
```

Stored procedure

1 productEdit

```
CREATE PROCEDURE [dbo].[productEdit]

       @ProductId    INT,
    @ProductName  VARCHAR (50) ,
    @Rate      decimal,
    @Description VARCHAR (50)    ,
    @CategoryName VARCHAR (50)
AS update Products set ProductName = @ProductName,Rate = @Rate,Description =
@Description, CategoryName = @CategoryName Where ProductId= @ProductId

RETURN 0
```

2 productList

```
CREATE PROCEDURE [dbo].[productList]

AS
       SELECT * from Products;
RETURN 0
```

Controller

```
using _210940320048.Models;
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace _210940320048.Controllers
{
    public class ProductsController : Controller
    {
        // GET: Products
        public ActionResult Index()
        {

            List<Products> lst = new List<Products>();
            SqlConnection cn = new SqlConnection();
            cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial 
Catalog=ExamDB;Integrated Security=True;Connect 
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Mult
iSubnetFailover=False";
            cn.Open();

            SqlCommand cmd = new SqlCommand();
            cmd.Connection = cn;
            cmd.CommandType = System.Data.CommandType.StoredProcedure;
            cmd.CommandText = "productList";

            try
            {
                SqlDataReader dr = cmd.ExecuteReader();
                while(dr.Read())
                {
                    lst.Add(new Products { ProductId = (int)dr["ProductId"], 
ProductName = (string) dr["ProductName"], Rate = (decimal) dr["Rate"], Description = 
(string)dr["Description"],
                    CategoryName = (string)dr["CategoryName"]
                    });
                }
                dr.Close();
            }
            catch (Exception e)
            {

                Console.WriteLine(e);
            }
            finally
            {
                cn.Close();
            }


            return View(lst);
        }

        // GET: Products/Details/5
        public ActionResult Details(int id)
        {
            return View();
        }

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

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

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

        // GET: Products/Edit/5
        public ActionResult Edit(int id)
        {
            return View();
        }

        // POST: Products/Edit/5
        [HttpPost]
        public ActionResult Edit(int id, Products obj)
        {
            SqlConnection cn = new SqlConnection();
            cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial 
Catalog=ExamDB;Integrated Security=True;Connect 
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;Mult
iSubnetFailover=False";
            cn.Open();

            SqlCommand cmd = new SqlCommand();
            cmd.Connection = cn;
            cmd.CommandType = System.Data.CommandType.StoredProcedure;
            cmd.CommandText = "productEdit";
            cmd.Parameters.AddWithValue("ProductId", id);
            cmd.Parameters.AddWithValue("@ProductName",obj.ProductName);
            cmd.Parameters.AddWithValue("@Rate", obj.Rate);
            cmd.Parameters.AddWithValue("@Description", obj.Description);
            cmd.Parameters.AddWithValue("@CategoryName", obj.CategoryName);

            try
            {
                cmd.ExecuteNonQuery();
                return RedirectToAction("index");
            }
            catch (Exception e)
            {

                Console.WriteLine(e);
            }
            finally
            {
                cn.Close();
            }


            return RedirectToAction("index");
        }

        // GET: Products/Delete/5
        public ActionResult Delete(int id)
        {
            return View();
        }

        // POST: Products/Delete/5
        [HttpPost]
        public ActionResult Delete(int id, FormCollection collection)
        {
            try
            {
                // TODO: Add delete logic here

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

       public ActionResult ProjectView()
        {
            return View();
        }
    }
}
```

Product view

Index view

```
@model IEnumerable<_210940320048.Models.Products>

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.ProductId)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.ProductName)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Rate)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Description)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.CategoryName)
        </th>
        <th></th>
    </tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.ProductId)

        @*Html.ActionLink(item.ProductId.ToString, "Edit", new { id = item.ProductId
})*@
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.ProductName)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Rate)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Description)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.CategoryName)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new {  id=item.ProductId }) |
            @Html.ActionLink("Details", "Details", new {  id=item.ProductId  }) |
            @Html.ActionLink("Delete", "Delete", new {  id=item.ProductId  })
        </td>
    </tr>
}

</table>

@*@Html.Partial("ProjectView")*@
```

Edit view

```
@model  _210940320048.Models.Products

@{
    ViewBag.Title = "Edit";
}

<h2>Edit</h2>


@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>Products</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.ProductId)
        <div class="form-group">
            @Html.LabelFor(model => model.ProductId, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">

                @Html.EditorFor(model => model.ProductId, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.ProductId, "", new { @class
= "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.ProductName, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.ProductName, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.ProductName, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Rate, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.Rate, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Rate, "", new { @class =
"text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.Description, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
```

```
                @Html.EditorFor(model => model.Description, new { htmlAttributes = new
{ @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.Description, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.CategoryName, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.CategoryName, new { htmlAttributes =
new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.CategoryName, "", new {
@class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save" class="btn btn-default" />
            </div>
        </div>
    </div>
}

<div>
    @Html.ActionLink("Back to List", "Index")
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

@*@Html.Partial("ProjectView")*@
```