

1.MyLayout

2.ViewStart

3.Database Queries

4.Models

1. Product

2. Category

5. Controller

1. Index

2. Edit

3. Partial display

6. Views

1. Index

2. Edit

3. Partial display

1.MyLayout

```
<!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>
    <div class="navbar navbar navbar-fixed-top">
        <ul class="nav nav-pills">
            <li role="presentation">@Html.ActionLink("Home", "Index", "Home")</li>
            <li role="presentation">@Html.ActionLink("Product", "Index",
"Products")</li>

        </ul>
    </div>

    <div class="container body-content">
        @RenderBody()
        <hr />
        <footer>
            <p>&copy; @DateTime.Now.Year - My ASP.NET Application</p>
        </footer>
        @Scripts.Render("~/bundles/jquery")
        @Scripts.Render("~/bundles/bootstrap")
        @RenderSection("scripts", required: false)
    </div>
</body>
</html>
```

2.ViewStart

```
@{
    Layout = "~/Views/Shared/_MyLayout.cshtml";
}
```

3.Database Queries

Products Table Query

```
CREATE TABLE [dbo].Products
```

```
(  
    [ProductId] INT NOT NULL PRIMARY KEY,  
    [ProductName] VARCHAR(50) NOT NULL,  
    [Rate] DECIMAL(18, 2) NOT NULL,  
    [Description] VARCHAR(200) NOT NULL,  
    [CategoryId] INT NOT NULL,  
    CONSTRAINT [FK_Products_Categories] FOREIGN KEY ([CategoryId])  
REFERENCES [Categories]([CategoryId])  
)
```

Categories Table Query

```
CREATE TABLE [dbo].Categories
```

```
(  
    [CategoryId] INT NOT NULL PRIMARY KEY,  
    [CategoryName] VARCHAR(50) NOT NULL  
)
```

4.Models

1. Product

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

namespace SplAssignment.Models
{
    public class Product
    {
        [Key]
        [Required(ErrorMessage = "Enter Product Id")]
        [Display(Name = "Product Id")]
        public int ProductId { get; set; }

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

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

        [Required(ErrorMessage = "Enter Product Description")]
        [Display(Name = "Product Description")]
        public string Description { get; set; }
        public int CategoryId { get; set; }
        public string CategoryName { get; set; }

        public static List<Product> GetProducts()
        {
            List<Product> listProduct = new List<Product>();

            SqlConnection con = new SqlConnection();
            con.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=JKJuly2022;Integrated Security=True;";

            try
            {
                con.Open();
                SqlCommand cmdSelect = new SqlCommand();
                cmdSelect.Connection = con;
                cmdSelect.CommandType = CommandType.Text;
                cmdSelect.CommandText = "select * from Products,Categories where
Products.CategoryId=Categories.CategoryId;";

                SqlDataReader dr = cmdSelect.ExecuteReader();
                while (dr.Read())
                {

```

```

        listProduct.Add(new Product { ProductId = (int)dr["ProductId"],
        ProductName = (string)dr["ProductName"], Rate = (decimal)dr["Rate"], Description =
        (string)dr["Description"], CategoryName=(string)dr["CategoryName"] });
    }
    dr.Close();
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
finally
{
    con.Close();
}

return listProduct;
}

public static Product GetSingleProduct(int id)
{
    Product obj = new Product();
    SqlConnection con = new SqlConnection();
    con.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=JKJuly2022;Integrated Security=True;";

    try
    {
        con.Open();
        SqlCommand cmdSelect = new SqlCommand();
        cmdSelect.Connection = con;
        cmdSelect.CommandType = CommandType.Text;
        cmdSelect.CommandText = "select * from Products where
        ProductId=@ProductId ";
        cmdSelect.Parameters.AddWithValue("@ProductId", id);
        SqlDataReader dr = cmdSelect.ExecuteReader();
        if (dr.Read())
        {
            obj.ProductId = (int)dr["ProductId"];
            obj.ProductName = (string)dr["ProductName"];
            obj.Rate = (decimal)dr["Rate"];
            obj.Description = (string)dr["Description"];
            obj.CategoryName = (string)dr["CategoryName"];

        }
        dr.Close();
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    finally
    {
        con.Close();
    }

    return obj;
}

public static void UpdateProduct(Product obj)
{

```

```

SqlConnection con = new SqlConnection();
con.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=JKJuly2022;Integrated Security=True;";

try
{
    con.Open();
    SqlCommand cmdSelect = new SqlCommand();
    cmdSelect.Connection = con;
    cmdSelect.CommandType = CommandType.StoredProcedure;
    cmdSelect.CommandText = "UpdateProducts";

    cmdSelect.Parameters.AddWithValue("@ProductName", obj.ProductName);
    cmdSelect.Parameters.AddWithValue("@Rate", obj.Rate);
    cmdSelect.Parameters.AddWithValue("@Description", obj.Description);
    cmdSelect.Parameters.AddWithValue("@CategoryId", obj.CategoryId);
    cmdSelect.Parameters.AddWithValue("@ProductId", obj.ProductId);

    cmdSelect.ExecuteNonQuery();

}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
finally
{
    con.Close();
}

}
}
}

```

2.Category

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

namespace SplAssignment.Models
{
    public class Category
    {
        public int CategoryId { get; set; }
        public string CategoryName { get; set; }

        public static List<Category> GetCategoryNames()
        {
            List<Category> listCategory = new List<Category>();
            SqlConnection con = new SqlConnection();
            con.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=JKJuly2022;Integrated Security=True;";

            try
            {
                con.Open();
                SqlCommand cmdSelect = new SqlCommand();
                cmdSelect.Connection = con;
                cmdSelect.CommandType = CommandType.Text;
                cmdSelect.CommandText = "select * from Categories";

                SqlDataReader dr = cmdSelect.ExecuteReader();
                while (dr.Read())
                {
                    listCategory.Add(new Category { CategoryId =
(int)dr["CategoryId"], CategoryName = (string)dr["CategoryName"] });
                }
                dr.Close();
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }
            finally
            {
                con.Close();
            }

            return listCategory;
        }
    }
}
```

7. Controller

1. Index

```
// GET: Products
public ActionResult Index()
{
    List<Product> lstProduct = Product.GetProducts();
    return View(lstProduct);
}
```

2. Edit

```
// GET: Products/Edit/5
public ActionResult Edit(int id=0)
{
    var list = Category.GetCategoryNames();
    ViewBag.list = list;
    Product obj = Product.GetSingleProduct(id);
    return View(obj);
}

// POST: Products/Edit/5
[HttpPost]
public ActionResult Edit(int id=0, Product obj=null)
{
    try
    {
        // TODO: Add update logic here
        Product.UpdateProduct(obj);
        return RedirectToAction("Index");
    }
    catch
    {
        return View();
    }
}
```

3. Partial display

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


8. Views

1. Index

```
@model IEnumerable<SplAssignment.Models.Product>
```

```
@{  
    ViewBag.Title = "Index";  
    Layout = "~/Views/Shared/_MyLayout.cshtml";  
}
```

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

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

```
@foreach (var item in Model) {  
    <tr>  
        <td>  
            @Html.ActionLink($"{@Html.DisplayFor(modelItem => item.ProductId)}", "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>
```

```
</tr>
```

```
}
```

```
</table>
```

```
<div>
```

```
    @Html.Partial("PartialDisplay")
```

```
</div>
```

2. Edit

```
@model SplAssignment.Models.Product
```

```
@{  
    ViewBag.Title = "Edit";  
    Layout = "~/Views/Shared/_MyLayout.cshtml";  
}
```

```
<h2>Edit</h2>
```

```
@using (Html.BeginForm())  
{  
    @Html.AntiForgeryToken()  
  
    <div class="form-horizontal">  
        <h4>Product</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",@disabled="disabled" } })  
                @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.DropDownListFor(model => model.CategoryId, new
SelectList(ViewBag.list, "CategoryId", "CategoryName"), "Select Category Name",
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="Update" class="btn btn-default" />
        </div>
    </div>
</div>
}

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

</div>

<div>
    @Html.Partial("PartialDisplay")
</div>

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

```

3. Partial diplay

```

<h3> Created by Amey Ramesh Patekar</h3>
<h3>Center Juhu</h3>
<h3>Roll Number : 220340520020</h3>

```