=====Person Controller==========

```
using PreviousPaper1.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace PreviousPaper1.Controllers
{
    public class ProductsController : Controller
    {
        // GET: Products
        public ActionResult Index()
        {
            List<Product> prod = Product.GetAllProducts();
            return View(prod);
        }

        // GET: Products/Details/5
        public ActionResult Details(int id)
        {
            Product obj = Product.GetProductById(id);
            return View(obj);
        }

        // 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)
{
    Product product = Product.GetProductById(id);
    List<Category> cata = Category.GetAllCatogories();
    List<SelectListItem> catalist = new List<SelectListItem>();
```

```
            foreach (var item in cata)
            {
                catalist.Add(new SelectListItem { Text = item.CategoryName, Value =
item.CategoryId.ToString() });
            }

            ViewBag.Categories = catalist;
            return View(product);

        }

        // POST: Products/Edit/5
        [HttpPost]
        public ActionResult Edit(int id, Product obj)
        {
            try
            {
                Product.UpdateProduct(obj);
                return RedirectToAction("Index");
            }
            catch
            {

                return View();
            }
        }
```

```
// 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();
   }
}

[ChildActionOnly]
public ActionResult MyPartialView()
{
   return View();
}
```

```
    }
}
```

=====Product.cs==========

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

namespace PreviousPaper1.Models
{
    public class Product
    {
        [Display(Name = "Product Id")]
        [Required(ErrorMessage ="Please Enter Product Id")]
        public int ProductId { get; set; }

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

        [DataType(DataType.Currency)]
        public decimal Rate { get; set; }

        public string Description { get; set; }
```

```
        [Display(Name = "Category Id")]
        [ScaffoldColumn(false)]
        public int CategoryId { get; set; }

        [Display(Name = "Category Name")]
        public string CategoryName { get; set; }

       /* public IEnumerable<SelectListItem> Categories { get; set; }
*/

        public static Product GetProductById(int id)
        {
            Product obj = new Product();
            SqlConnection cn = new SqlConnection();
            cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=JKJuly2022;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFa
ilover=False";
            try
            {
                cn.Open();
                SqlCommand cmdSelect = new SqlCommand();
                cmdSelect.Connection = cn;
                cmdSelect.CommandType = System.Data.CommandType.Text;
                cmdSelect.CommandText = "select * from Products where ProductId=@ProductId";
                cmdSelect.Parameters.AddWithValue("@ProductId", id);

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

```
            {
                obj = new Product { ProductId = (int)dr["ProductId"], ProductName =
(string)dr["ProductName"], Rate = (decimal)dr["Rate"], Description = (string)dr["Description"],
CategoryId = (int)dr["CategoryId"] };


            }
            dr.Close();


        }
        catch
        {


        }
        finally
        {
            cn.Close();
        }
        return obj;
    }


    public static void UpdateProduct(Product obj)
    {
        SqlConnection cn = new SqlConnection();
        cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=JKJuly2022;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFa
ilover=False";
     //   try
      //  {
```

```
            cn.Open();
            SqlCommand cmdSelect = new SqlCommand();
            cmdSelect.Connection = cn;
            cmdSelect.CommandType = System.Data.CommandType.Text;
            cmdSelect.CommandText = "update Products set
ProductName=@ProductName,Rate=@Rate,Description=@Description,CategoryId=CategoryId
where ProductId=@ProductId";



            cmdSelect.Parameters.AddWithValue("@ProductId", obj.ProductId);
            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.ExecuteNonQuery();




        // }
        // catch
        // {

         // }
        // finally
        // {
            cn.Close();
```

```
    // }
}

public static List<Product> GetAllProducts()
{
    List<Product> prod = new List<Product>();
    SqlConnection cn = new SqlConnection();
    cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=JKJuly2022;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFa
ilover=False";
    try
    {
        cn.Open();
        SqlCommand cmdSelect = new SqlCommand();
        cmdSelect.Connection = cn;
        cmdSelect.CommandType = System.Data.CommandType.StoredProcedure;
        cmdSelect.CommandText = "SelectAllProducts";

        SqlDataReader dr = cmdSelect.ExecuteReader();
        while (dr.Read())
        {
            prod.Add(new Product { ProductId = (int)dr["ProductId"], ProductName =
(string)dr["ProductName"], Rate = (decimal)dr["Rate"], Description = (string)dr["Description"],
CategoryId = (int)dr["CategoryId"], CategoryName = (string)dr["CategoryName"] });

        }
        dr.Close();
```

```
        }
        catch
        {

        }
        finally
        {
            cn.Close();
        }
        return prod;

      }

   }
}
```

===Category.cs==========================

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

namespace PreviousPaper1.Models
{
   public class Category
   {
```

```
        public int CategoryId { get; set; }


        [Display(Name ="Category Name")]
        public string CategoryName { get; set; }


        public static List<Category> GetAllCatogories()
        {
            List<Category> cat = new List<Category>();
            SqlConnection cn = new SqlConnection();
            cn.ConnectionString = @"Data Source=(localdb)\MSSQLLocalDB;Initial
Catalog=JKJuly2022;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubnetFa
ilover=False";
            try
            {
                cn.Open();
                SqlCommand cmdSelect = new SqlCommand();
                cmdSelect.Connection = cn;
                cmdSelect.CommandType = System.Data.CommandType.Text;
                cmdSelect.CommandText = "select * from Categories";


                SqlDataReader dr = cmdSelect.ExecuteReader();
                while (dr.Read())
                {
                    cat.Add(new Category { CategoryId = (int)dr["CategoryId"], CategoryName =
(string)dr["CategoryName"] });


                }
                dr.Close();
```

```
        }
        catch
        {


        }
        finally
        {
            cn.Close();
        }
        return cat;
      }
   }
}
```

```
@Html.DropDownListFor(model => model.CategoryName,
(IEnumerable<SelectListItem>)ViewBag.Categories)
```