Product Crud

1. **Create table Products**

```
CREATE TABLE Products(
Id INT PRIMARY KEY IDENTITY(1,1),
Name NVARCHAR(20),
Price DECIMAL(10,2),
Quantity INT);
```

2. **create Procedure**

```
-- get all products--

CREATE PROCEDURE sp_GetProducts
AS
BEGIN
   SELECT * FROM Products;
END;

--get product by id --

CREATE PROCEDURE sp_GetProductById
   @Id INT
```

```
AS
BEGIN
   SELECT * FROM Products WHERE Id = @Id;
END;

-- add new product --
CREATE PROCEDURE sp_AddProduct
   @Name NVARCHAR(100),
   @Price DECIMAL(10,2),
   @Quantity INT
AS
BEGIN
   INSERT INTO Products (Name, Price, Quantity)
   VALUES (@Name, @Price, @Quantity);
END;

-- update existing product --

CREATE PROCEDURE sp_UpdateProduct
   @Id INT,
   @Name NVARCHAR(100),
   @Price DECIMAL(18,2),
   @Quantity INT
AS
```

```
BEGIN
   UPDATE Products
   SET Name = @Name, Price = @Price, Quantity = @Quantity
   WHERE Id = @Id;
END;

--delete product --

CREATE PROCEDURE sp_DeleteProduct
   @Id INT
AS
BEGIN
   DELETE FROM Products WHERE Id = @Id;
END;
```

3. **Create Model**

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

namespace AD__Product_Crud.Models
{
   public class Product
   {
      public int Id { get; set; }
```

```
        public string Name { get; set; }
        public decimal Price { get; set; }
        public int Quantity { get; set; }
    }
}
```

4. **Create Data Access Layer (DAL)**

```
using AD__Product_Crud.Models;
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;

namespace CRUD_Using_ADO.Models
{
    public class ProductDAL
    {
        String connectionString =
ConfigurationManager.ConnectionStrings["ProductDBConnectio
n"].ToString();

        // Get All Products
        public List<Product> GetAllProducts()
        {
            List<Product> products = new List<Product>();
            using (SqlConnection con = new
SqlConnection(connectionString))
            {
                SqlCommand cmd = new
SqlCommand("sp_GetProducts", con);
```

```
            cmd.CommandType =
CommandType.StoredProcedure;
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
            while (reader.Read())
            {
               products.Add(new Product
               {
                  Id = Convert.ToInt32(reader["Id"]),
                  Name = reader["Name"].ToString(),
                  Price = Convert.ToDecimal(reader["Price"]),
                  Quantity = Convert.ToInt32(reader["Quantity"])
               });
            }
         }
         return products;
      }


      // Get Product By Id
      public Product GetProductById(int id)
      {
         Product product = null;
         using (SqlConnection con = new
SqlConnection(connectionString))
         {
            SqlCommand cmd = new
SqlCommand("sp_GetProductById", con);
            cmd.CommandType =
CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@Id", id);
            con.Open();
            SqlDataReader reader = cmd.ExecuteReader();
```

```
            if (reader.Read())
            {
                product = new Product
                {
                    Id = Convert.ToInt32(reader["Id"]),
                    Name = reader["Name"].ToString(),
                    Price = Convert.ToDecimal(reader["Price"]),
                    Quantity = Convert.ToInt32(reader["Quantity"])
                };
            }
        }
        return product;
    }



    // Add Product
    public void AddProduct(Product product)
    {
        using (SqlConnection con = new
SqlConnection(connectionString))
        {
            SqlCommand cmd = new
SqlCommand("sp_AddProduct", con);
            cmd.CommandType =
CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@Name",
product.Name);
            cmd.Parameters.AddWithValue("@Price",
product.Price);
            cmd.Parameters.AddWithValue("@Quantity",
product.Quantity);
            con.Open();
```

```
            cmd.ExecuteNonQuery();
        }
    }

    // Update Product
    public void UpdateProduct(Product product)
    {
        using (SqlConnection con = new
SqlConnection(connectionString))
        {
            SqlCommand cmd = new
SqlCommand("sp_UpdateProduct", con);
            cmd.CommandType =
CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@Id", product.Id);
            cmd.Parameters.AddWithValue("@Name",
product.Name);
            cmd.Parameters.AddWithValue("@Price",
product.Price);
            cmd.Parameters.AddWithValue("@Quantity",
product.Quantity);
            con.Open();
            cmd.ExecuteNonQuery();
        }
    }

    // Delete Product
    public void DeleteProduct(int id)
    {
        using (SqlConnection con = new
SqlConnection(connectionString))
        {
```

```
            SqlCommand cmd = new
SqlCommand("sp_DeleteProduct", con);
            cmd.CommandType =
CommandType.StoredProcedure;
            cmd.Parameters.AddWithValue("@Id", id);
            con.Open();
            cmd.ExecuteNonQuery();
        }
    }
  }
}
```

5. Configure Database Connection

```
<connectionStrings>
        <add name="ProductDBConnection"
connectionString="data source=SHILPA\SQLEXPRESS ;
Database=ProductDB ;Integrated security=True;"/>
</connectionStrings>
```

6. Create Controller

```
using System.Web.Mvc;
using AD__Product_Crud.Models;
using CRUD_Using_ADO.Models;

namespace CRUD_Using_ADO.Controllers
{
  public class ProductController : Controller
  {
    ProductDAL productDal = new ProductDAL();

    public ActionResult Index()
```

```
{
    return View(productDal.GetAllProducts());
}

public ActionResult Create()
{
    return View();
}

[HttpPost]
public ActionResult Create(Product product)
{
    if (ModelState.IsValid)
    {
        productDal.AddProduct(product);
        return RedirectToAction("Index");
    }
    return View(product);
}

public ActionResult Edit(int id)
{
    return View(productDal.GetProductById(id));
}

[HttpPost]
public ActionResult Edit(Product product)
{
    if (ModelState.IsValid)
    {
        productDal.UpdateProduct(product);
        return RedirectToAction("Index");
    }
```

```
            return View(product);
        }

        public ActionResult Delete(int id)
        {
            return View(productDal.GetProductById(id));
        }

        [HttpPost, ActionName("Delete")]
        public ActionResult DeleteConfirmed(int id)
        {
            productDal.DeleteProduct(id);
            return RedirectToAction("Index");
        }
    }
}
```

7. Create View

i. Index.cshtml

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

@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>

<p>
    @Html.ActionLink("Create New", "Create")
```

```
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Name)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Price)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Quantity)
        </th>
        <th></th>
    </tr>

@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Name)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Price)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Quantity)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id=item.Id })
|
            @Html.ActionLink("Details", "Details", new {
id=item.Id }) |
            @Html.ActionLink("Delete", "Delete", new {
id=item.Id })
```

```
            </td>
        </tr>
    }

    </table>
```

ii. Create.cshtml

```
@model AD__Product_Crud.Models.Product

@{
    ViewBag.Title = "Create";
}

<h2>Create</h2>


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

    <div class="form-horizontal">
        <h4>Product</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-
danger" })
```

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

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

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

```
            @Html.EditorFor(model => model.Quantity, new {
htmlAttributes = new { @class = "form-control" } })

            @Html.ValidationMessageFor(model => model.Quantity, "",
new { @class = "text-danger" })

        </div>

    </div>


    <div class="form-group">

        <div class="col-md-offset-2 col-md-10">

            <input type="submit" value="Create" class="btn btn-
default" />

        </div>

    </div>

  </div>

}


<div>

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

</div>


@section Scripts {

  @Scripts.Render("~/bundles/jqueryval")

}
```

iii. Edit.cshtml

```
@model AD__Product_Crud.Models.Product

@{
   ViewBag.Title = "Edit";
}

<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.Id)

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

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

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

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

```
        </div>

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

iv. Delete.cshtml

```
@model AD__Product_Crud.Models.Product

@{
   ViewBag.Title = "Delete";
}

<h2>Delete</h2>

<h3>Are you sure you want to delete this?</h3>
<div>
   <h4>Product</h4>
   <hr />
   <dl class="dl-horizontal">
      <dt>
         @Html.DisplayNameFor(model => model.Name)
      </dt>

      <dd>
         @Html.DisplayFor(model => model.Name)
      </dd>

      <dt>
         @Html.DisplayNameFor(model => model.Price)
      </dt>
```

```
        <dd>
            @Html.DisplayFor(model => model.Price)
        </dd>

        <dt>
            @Html.DisplayNameFor(model => model.Quantity)
        </dt>

        <dd>
            @Html.DisplayFor(model => model.Quantity)
        </dd>

    </dl>

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

        <div class="form-actions no-color">
            <input type="submit" value="Delete" class="btn btn-
danger" /> |
            @Html.ActionLink("Back to List", "Index")
        </div>
    }
</div>
```