

Screenshot

The screenshot shows a web application titled "My Shop" with a sidebar menu including Dashboard, Categories, Products, and Orders. The main content area displays a table of products with columns for Sr. No., Category Name, and Action. Two entries are listed: "macbook3233" and "macbook s8". A modal window is open over the table, showing a code editor with C# code for a HomeIndexViewModel class. The code includes a constructor that initializes a _unitOfWork and a dbMyonlineShoppingEntities1 context. It also defines a CreateModel method that returns a new HomeIndexViewModel instance with a ListOfProducts property populated from the repository.

Copyright © Developed By Abdullah

اكتب هنا للبحث

localhost:44328/?page=1

ONLINE SHOPPING STORE

My Shop

Cart



Search Product/Category

Enter Keyword



Our Products trends products



localhost:44328/?page=1

وتحول المراجع، فبع إشراكك المرجعية هنا في تحرير الإشارات المرجعية، جار استناد الإشارات المرجعية إلى ...

قائمة المراجعة

New

Old

Search Product/Category

toshepa

Our Products trends products

toshepa 34234243
3000 Available

toshepa g2
300 Available

© 2021 - Developed By Abdullah

Cart ▾

Checkout

ONLINE SHOPPING STORE

My Shop

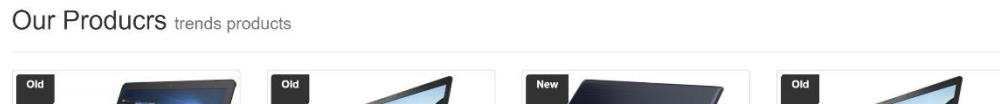
Home Login Location Contact US About

toshepa 34234243 (1) ✕
macbook s8 (1) ✕
macbook L (1) ✕
macbook 92 (1) ✕

Labtop

Search Product/Category

Enter Keyword



ONLINE SHOPPING STORE

My Shop

Cart ▾

Home Login Location Contact US About

Checkout

Product	Quantity	Add
toshepa 34234243	1	+ -
macbook L	1	+ -
macbook s8	2	+ -
macbook 92	3	+ -

[Checkout >>](#)

© 2021 - Developed By Abdullah



ONLINE SHOPPING STORE

My Shop

Cart ▾

Home Login Location Contact US About

Payment info

Product	Price	Quantity	Line Total
toshepa 34234243	3000	1	3000
macbook L	2000	1	2000
macbook s8	2000	2	4000
macbook 92	300	3	900

Total: 9900

Card name:	abdullah saddig alhyani
Card Number:	98798987687888899708
CVV:	...
Expiration Dates:	٢٠٢٣/٠٨/٢٠٢٣

[Payment >>](#)

ONLINE SHOPPING STORE

My Shop



Home Login Location Contact US About

Payment has been successfully done!

[Take Me Home](#)

[Contact Support](#)

© 2021 - Developed By Abdullah



ONLINE SHOPPING STORE

My Shop

Cart ▾

[Home](#) [Login](#) [Location](#) [Contact US](#) [About](#)

Welcome to ONLINE SHOPPING STORE!

[About!..](#)

If you have the desire to buy, a ONLINE SHOPPING STORE satisfies your desires immediately. It is an invitation that is not linked to a specific time, place or occasion. You must fulfill your desire to eat the best of our products in a ONLINE SHOPPING STORE. We strive in a ONLINE SHOPPING STORE to invent the most creative ways to share with you true happiness every day. ONLINE SHOPPING STORE believes in the joy of giving in order to It delivers an unparalleled sense of satisfaction. If you have the desire to buy, a ONLINE SHOPPING STORE satisfies your desires immediately. It is an invitation that is not linked to a specific time, place or occasion. You must fulfill your desire to eat the best of our products in a ONLINE SHOPPING STORE. We strive in a ONLINE SHOPPING STORE to invent the most creative ways to share with you true happiness every day. ONLINE SHOPPING STORE believes in the joy of giving in order to It delivers an unparalleled sense of satisfaction.

© 2021 - Developed By Abdullah



ONLINE SHOPPING STORE

My Shop

Cart ▾

[Home](#) [Login](#) [Location](#) [Contact US](#) [About](#)

Welcome to ONLINE SHOPPING STORE!

[Contact!..](#)

P: 966500227027

Support: abdullah.s.alhyani@gmail.com

© 2021 - Developed By Abdullah



ONLINE SHOPPING STORE

My Shop

Cart ▾

[Home](#) [Login](#) [Location](#) [Contact US](#) [About](#)

Welcome to ONLINE SHOPPING STORE!

[Location!..](#)

Location

Kingdom Saudi Arabia Makkah Umrah neighborhood Next to the Sayeda Aisha Mosque zipcode:356367

© 2021 - Developed By Abdullah



ONLINE SHOPPING STORE

My Shop

Cart ▾

Home Login Location Contact US About

Welcome to Login page !

in this page you can login to dashboard!..

Enter User name :
Enter Password :
[Login](#)

abdullah
...

© 2021 - Developed By Abdullah



My Shop

Dashboard

Categories Products Orders

Search for... 9+ ✉ 7

Settings Activity Log Logout

Copyright © Developed By Abdullah

My Shop

Dashboard / Categories

Category Details

Show 10 entries Search:

Sr. No.	Category Name	Action
1	Apple	Edit
2	toshsa	Edit
3	HUAWE	Edit
4	HP	Edit
5	MAXUM	Edit

Sr. No. Category Name Action

Showing 1 to 5 of 5 entries Previous 1 Next

Copyright © Developed By Abdullah

My Shop

Dashboard Categories Products Orders

Category Details

Show 10 entries Search:

Sr. No.	Category Name	Action
1	Apple	Edit
2	toshpaa	Edit
3	HUAWE	Edit
4	HP	Edit
5	MAXUM	Edit

Sr. No. Category Name Action

Showing 1 to 5 of 5 entries Previous 1 Next

Copyright © Developed By Abdullah

My Shop

Dashboard Categories Products Orders

Category

CategoryName: Apple

IsActive:

IsDelete:

Save

Back to List

Copyright © Developed By Abdullah

My Shop

- Dashboard
- Categories
- Products
- Orders

mackook (9)

Product

Sr. No.	Category Name	Action
1	macbook3233	Edit
1	macbook s8	Edit
1	macbook 92	Edit
1	toshepa 34234243	Edit
1	macbook L	Edit
1	toshepa g2	Edit

Showing 1 to 6 of 6 entries Previous **1** Next

Copyright © Developed By Abdullah

- Categories
- Products
- Orders

ProductName: macbook3233

CategoryId: Apple

IsActive:

IsDelete:

ProductImage: لم يتم اختيار أي ملف.

IsFeatured:

Quantity: ٣*

Price: 7400.00

[Save](#)

Product

ProductName: macbook 92

CategoryId: Apple

IsActive:

ProductImage: اختيار ملف 2ywd.png

IsFeatured:

Quantity: 2

Price: 3000.0

Create

[Back to List](#)

Index.cshtml

Model1.edmx [Diagram1]

```

classDiagram
    class Tbl_Product {
        ProductId
        ProductName
        CategoryId
        IsActive
        IsDelete
        ProductImage
        IsFeatured
        Quantity
        Price
        CreatedDate
        ModifiedDate
        Description
    }
    class Tbl_Cart {
        CartId
        ProductId
        MemberId
        CartStatusId
    }
    class Tbl_MemberRole {
        MemberRoleId
        MemberId
        RoleId
    }
    class Tbl_CartStatus {
        CartStatusId
        CartStatus
    }
    class Tbl_Members {
        MemberId
        FirstName
        LastName
        EmailId
        Password
        IsActive
        IsDelete
    }
    class Tbl_ShippingDetail {
        ShippingDetailId
        MemberId
        Address
        City
        State
        Country
        ZipCode
        OrderId
    }
    class Tbl_Roles {
        RoleId
        RoleName
    }

    Tbl_Product <|--> Tbl_Cart : 0..1
    Tbl_Product <|--> Tbl_MemberRole : 0..1
    Tbl_Cart <|--> Tbl_CartStatus : 0..1
    Tbl_MemberRole <|--> Tbl_Roles : 0..1
    Tbl_Members <|--> Tbl_ShippingDetail : 0..1
  
```

LAPTOP-A1FCB940\Sql - dbo.Tbl_Cart

Column Name	Data Type	Allow Nulls
CartId	int	<input type="checkbox"/>
ProductId	int	<input checked="" type="checkbox"/>
MemberId	int	<input checked="" type="checkbox"/>
CartStatusId	int	<input checked="" type="checkbox"/>

Results

CartId	ProductId	MemberId	CartStatusId
1	1	1	1

Column Name	Data Type	Allow Nulls
CartStatusId	int	<input type="checkbox"/>
CartStatus	varchar(500)	<input checked="" type="checkbox"/> <input type="checkbox"/>

100 %

Results		Messages
CartStatusId	CartStatus	
1	actev	
2	step1	
3	step2	
4	final	

Column Name	Data Type	Allow Nulls
CategoryId	int	<input type="checkbox"/>
CategoryName	varchar(500)	<input checked="" type="checkbox"/>
IsActive	bit	<input checked="" type="checkbox"/>
IsDelete	bit	<input checked="" type="checkbox"/> <input type="checkbox"/>

Results		Messages	
CategoryId	CategoryName	IsActive	IsDelete
1	Apple	1	0
2	toshpa	1	0
3	HUAWE	1	0
4	HP	1	0
5	MAXUM	0	0
6	LG	0	1
7	SAMSONG	1	1

Column Name	Data Type	Allow Nulls
MemberRoleId	int	<input type="checkbox"/>
memberId	int	<input checked="" type="checkbox"/>
RoleId	int	<input checked="" type="checkbox"/> <input type="checkbox"/>

MemberRoleID	memberId	RoleId
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6

Column Name	Data Type	Allow Nulls
MemberId	int	<input type="checkbox"/>
FristName	varchar(200)	<input checked="" type="checkbox"/>
LastNAme	varchar(200)	<input checked="" type="checkbox"/>
EmailId	varchar(200)	<input checked="" type="checkbox"/>
Password	varchar(500)	<input checked="" type="checkbox"/>
IsActive	bit	<input checked="" type="checkbox"/>
IsDelete	bit	<input checked="" type="checkbox"/> <input type="checkbox"/>
CreatedOn	datetime	<input checked="" type="checkbox"/>
ModifiedOn	datetime	<input checked="" type="checkbox"/>

Results Messages

	MemberId	FristName	LastName	EmailId	Password	IsActive	IsDelete	CreatedOn	ModifiedOn
1	1	abdullah	allhyani	a@gmail.com	123	1	0	NULL	NULL
2	2	m	m	m	m	0	0	NULL	NULL
3	3	a	a	a	a	1	0	NULL	NULL

LAPTOP-A1FCB940\S...- dbo.Tbl_Product SQLQuery20.sql - L... A1FCB940\pc (71) LAPTOP-A1FCB940\S...dbo.Tbl_Members

	Column Name	Data Type	Allow Nulls
*	ProductId	int	<input type="checkbox"/>
	ProductName	varchar(500)	<input checked="" type="checkbox"/>
	Categoryld	int	<input checked="" type="checkbox"/>
	IsActive	bit	<input checked="" type="checkbox"/>
	IsDelete	bit	<input checked="" type="checkbox"/>
	ProductImage	varchar(MAX)	<input checked="" type="checkbox"/>
	IsFeatured	bit	<input checked="" type="checkbox"/>
	Quantity	int	<input checked="" type="checkbox"/>
	Price	decimal(18, 0)	<input checked="" type="checkbox"/>
	CreatedDate	varchar(500)	<input checked="" type="checkbox"/>
	ModifiedDate	varchar(500)	<input checked="" type="checkbox"/>
	Description	varchar(500)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Results Messages

	ProductId	ProductName	Categoryld	IsActive	IsDelete	ProductImage	IsFeatured	Quantity	Price	CreatedDate	ModifiedDate	Description
1	1	macbook3233	1	1	0	2ywd.png	0	30	7400	2021-01-01	NULL	NULL
2	2	macbook s8	1	1	0	2.png	0	40	2000	2000-09-09	NULL	NULL
3	5	macbook 92	1	1	0	6.jpg	1	20	300	NULL	NULL	NULL
4	6	toshepa 34234243	2	0	0	2.png	0	4	3000	NULL	NULL	NULL
5	7	macbook L	1	1	0	shsh.png	0	22	2000	NULL	NULL	NULL
6	8	toshepa g2	2	1	0	6.jpg	0	3	300	NULL	NULL	NULL

LAPTOP-A1FCB940\S...g - dbo.Tbl_Roles SQLQuery21.sql - L... A1FCB940\pc (54) LAPTOP-A1FCB940\S...dbo.Tbl_Product

	Column Name	Data Type	Allow Nulls
*	RoleId	int	<input type="checkbox"/>
	RoleName	varchar(200)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Results Messages

	RoleId	RoleName
1	1	ADMIN
2	4	COSTMER
3	3	EMPLOY
4	2	MANGER

LAPTOP-A1FCB940\S...l_ShippingDetails SQLQuery22.sql - L... A1FCB940\pc (56) LAPTOP-A1FCB940\S...g - dbo.Tbl_Roles

	Column Name	Data Type	Allow Nulls
*	ShippingDetailId	int	<input type="checkbox"/>
	MemberId	int	<input checked="" type="checkbox"/>
	Adress	varchar(500)	<input checked="" type="checkbox"/>
	City	varchar(500)	<input checked="" type="checkbox"/>
	State	varchar(500)	<input checked="" type="checkbox"/>
	Country	varchar(50)	<input checked="" type="checkbox"/>
	ZipCode	varchar(50)	<input checked="" type="checkbox"/>
	OrderId	int	<input checked="" type="checkbox"/>
	AmountPaid	decimal(18, 0)	<input checked="" type="checkbox"/>
	PaymentType	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

LAPTOP-A1FCB940\S...bo.Tbl_SlideImage SQLQuery23.sql - L... A1FCB940\pc (57) LAPTOP-A1FCB940\S...l_ShippingDetails

	Column Name	Data Type	Allow Nulls
*	SlideId	int	<input type="checkbox"/>
	SlideTitle	varchar(500)	<input checked="" type="checkbox"/>
	SlideImage	varchar(MAX)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Results **Messages**

SlideId	SlideTitle	SlideImage
1	APPLE	6.jpg
2	MAC	1.jpg
3	LONOVO	2.jpg

SQL Server Management Studio - SQLQuery25.sql [LAPTOP-A1FCB940\SQLEXPRESS01.dbMyonlineShopping] Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer Connect Object Explorer Connect

SQLQuery25.sql - L...-A1FCB940\pc(55) SQLQuery24.sql - L...-A1FCB940\pc(60) LAPTOP-A1FCB940\bo.Tbl_SlideImage

```

USE [dbo].[MyonlineShopping]
GO
===== Object: StoredProcedure [dbo].[GetBySearch] Script Date: 21/01/43 06:47:25 =====
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
=====
-- Author:      <Author>,<Name>
-- Create date: <Create Date>,>
-- Description: <Description>,
=====
ALTER PROCEDURE [dbo].[GetBySearch]
    @search nvarchar(max)=null
    AS
BEGIN
    SELECT * from [dbo].[Tbl_Product] P
    left join [dbo].[Tbl_Category] C on P.CategoryId=C.CategoryId
    where
    P.ProductName LIKE CASE WHEN @search is not null then '%' +@search+'%' else P.ProductName END
    OR C.CategoryName LIKE CASE WHEN @search is not null then '%' +@search+'%' else C.CategoryName END
END

```

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q)

Index.cshtml

```

3   using OnlineShoppingStore.Models;
4   using OnlineShoppingStore.Repository;
5   using System;
6   using System.Collections.Generic;
7   using System.Linq;
8   using System.Web;
9   using System.Web.Mvc;
10
11  namespace OnlineShoppingStore.Controllers
12  {
13      public class AdminController : Controller
14      {
15          // GET: Admin
16
17
18          public GenericUnitOfWork _unitOfWork = new GenericUnitOfWork();
19
20          public List<SelectListItem> GetCategory()
21          {
22              List<SelectListItem> list = new List<SelectListItem>();
23              var cat = _unitOfWork.GetRepositoryInstance<Tbl_Category>().GetAllRecords();
24              foreach (var item in cat)
25              {
26                  list.Add(new SelectListItem { Value = item.CategoryId.ToString(), Text = item.CategoryName });
27              }
28              return list;
29          }
30
31          public ActionResult Dashboard()
32          {
33              return View();
34          }
35

```

```

public ActionResult Categories()
{
    List<Tbl_Category> allcategories = _unitOfWork.GetRepositoryInstance<Tbl_Category>().GetAllRecordsIQueryable().Where(
        return View(allcategories);
}
public ActionResult AddCategory()
{
    return UpdateCategory(0);
}

public ActionResult UpdateCategory(int categoryId)
{
    CategoryDetail cd;
    if (categoryId != null)
    {
        cd = JsonConvert.DeserializeObject<CategoryDetail>(JsonConvert.SerializeObject(_unitOfWork.GetRepositoryInstance<Tbl_Category>().GetFirstOrDefault(categoryId)));
    }
    else
    {
        cd = new CategoryDetail();
    }
    return View("UpdateCategory", cd);
}

public ActionResult CategoryEdit(int catId)
{
    return View(_unitOfWork.GetRepositoryInstance<Tbl_Category>().GetFirstOrDefault(catId));
}
[HttpPost]
public ActionResult CategoryEdit(Tbl_Category tbl)
public ActionResult CategoryEdit(int catId)
{
    return View(_unitOfWork.GetRepositoryInstance<Tbl_Category>().GetFirstOrDefault(catId));
}
[HttpPost]
public ActionResult CategoryEdit(Tbl_Category tbl)
{
    _unitOfWork.GetRepositoryInstance<Tbl_Category>().Update(tbl);
    return RedirectToAction("Categories");
}
public ActionResult Product()
{
    return View(_unitOfWork.GetRepositoryInstance<Tbl_Product>().GetProduct());
}
public ActionResult ProductEdit(int productId)
{
    ViewBag.CategoryList = GetCategory();
    return View(_unitOfWork.GetRepositoryInstance<Tbl_Product>().GetFirstOrDefault(productId));
}
[HttpPost]
public ActionResult ProductEdit(Tbl_Product tbl, HttpPostedFileBase file)
{
    string pic = null;
    if (file != null)
    {
        pic = System.IO.Path.GetFileName(file.FileName);
        string path = System.IO.Path.Combine(Server.MapPath("~/ProductImg/"), pic);
        // file is uploaded
        file.SaveAs(path);
    }
    tbl.ProductImage = file != null ? pic : tbl.ProductImage;
    //tbl.ModifiedDate = DateTime.Now;
}

```

```

        if (file != null)
        {
            pic = System.IO.Path.GetFileName(file.FileName);
            string path = System.IO.Path.Combine(Server.MapPath("~/ProductImg/"), pic);
            // file is uploaded
            file.SaveAs(path);
        }
        tbl.ProductImage = file != null ? pic : tbl.ProductImage;
        //_unitOfWork.GetRepositoryInstance<Tbl_Product>().Update(tbl);
        return RedirectToAction("Product");
    }

    public ActionResult ProductAdd()
    {
        ViewBag.CategoryList = GetCategory();
        return View();
    }

    [HttpPost]
    public ActionResult ProductAdd(Tbl_Product tbl, HttpPostedFileBase file)
    {
        string pic = null;
        if (file != null)
        {
            pic = System.IO.Path.GetFileName(file.FileName);
            string path = System.IO.Path.Combine(Server.MapPath("~/ProductImg/"), pic);
            // file is uploaded
            file.SaveAs(path);
        }
        tbl.ProductImage = pic;
        //_unitOfWork.GetRepositoryInstance<Tbl_Product>().Add(tbl);
        return RedirectToAction("Product");
    }
}

```

```

namespace OnlineShoppingStore.Controllers
{
    public class HomeController : Controller
    {

        dbMyonlineShoppingEntities1 ctx = new dbMyonlineShoppingEntities1();

        public ActionResult Index(string search, int? page)
        {
            HomeIndexViewModel model = new HomeIndexViewModel();
            return View(model.CreateModel(search, 4, page));
        }

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

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

        public ActionResult DecreaseQty(int productId)
        {
            if (Session["cart"] != null)
            {
                List<Item> cart = (List<Item>)Session["cart"];
                var product = ctx.Tbl_Product.Find(productId);
                foreach (var item in cart)
                {
                    if (item.Product.ProductId == productId)
                    {
                        int prevQty = item.Quantity;
                        if (prevQty > 0)
                        {
                            ...
                        }
                    }
                }
            }
        }
    }
}

```

```
public ActionResult DecreaseQty(int productId)
{
    if (Session["cart"] != null)
    {
        List<Item> cart = (List<Item>)Session["cart"];
        var product = ctx.Tbl_Product.Find(productId);
        foreach (var item in cart)
        {
            if (item.Product.ProductId == productId)
            {
                int prevQty = item.Quantity;
                if (prevQty > 0)
                {
                    cart.Remove(item);
                    cart.Add(new Item()
                    {
                        Product = product,
                        Quantity = prevQty - 1
                    });
                }
                break;
            }
        }
        Session["cart"] = cart;
    }
    return RedirectToAction("Checkout");
}
public ActionResult AddToCart(int productId, string url)
{
    if (Session["cart"] == null)
    {
        List<Item> cart = new List<Item>();
        Session["cart"] = cart;
    }
}
```

```
public ActionResult AddToCart(int productId, string url)
{
    if (Session["cart"] == null)
    {
        List<Item> cart = new List<Item>();
        var product = ctx.Tbl_Product.Find(productId);
        cart.Add(new Item()
        {
            Product = product,
            Quantity = 1
        });
        Session["cart"] = cart;
    }
    else
    {
        List<Item> cart = (List<Item>)Session["cart"];
        var count = cart.Count();
        var product = ctx.Tbl_Product.Find(productId);
        for (int i = 0; i < count; i++)
        {
            if (cart[i].Product.ProductId == productId)
            {
                int prevQty = cart[i].Quantity;
                cart.Remove(cart[i]);
                cart.Add(new Item()
                {
                    Product = product,
                    Quantity = prevQty + 1
                });
                break;
            }
            else
            {
```

```
        else
    {
        var prd = cart.Where(x => x.Product.ProductId == productId).SingleOrDefault();
        if (prd == null)
        {
            cart.Add(new Item()
            {
                Product = product,
                Quantity = 1
            });
        }
    }
    Session["cart"] = cart;
}
return RedirectToAction(url);
}
public ActionResult RemoveFromCart(int productId)
{
    List<Item> cart = (List<Item>)Session["cart"];
    foreach (var item in cart)
    {
        if (item.Product.ProductId == productId)
        {
            cart.Remove(item);
            break;
        }
    }
    Session["cart"] = cart;
    return RedirectToAction("Index");
}

public ActionResult About()
{
    if (item.Product.ProductId == productId)
    {
        cart.Remove(item);
        break;
    }
}
Session["cart"] = cart;
return RedirectToAction("Index");
}

public ActionResult About()
{
    return View();
}
public ActionResult Login()
{
    return View();
}
public ActionResult Location()
{
    return View();
}

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

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Mvc;
6
7  namespace OnlineShoppingStore.Controllers
8  {
9      public class PaymentController : Controller
10     {
11         // GET: Payment
12         public ActionResult PaymentView()
13         {
14             return View();
15         }
16         public ActionResult Payment() { return RedirectToAction("Index"); }
17
18         public ActionResult SuccessView()
19         {
20             return View();
21         }
22     }
23 }
```

```
1 ! using OnlineShoppingStore.DAL;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Web;
6
7  namespace OnlineShoppingStore.Models.Home
8  {
9      public class Item
10     {
11         public Tbl_Product Product { get; set; }
12         public int Quantity { get; set; }
13     }
14 }
```

```

using System.Data.SqlClient;
using System.Linq;
using System.Web;
using PagedList;
using PagedList.Mvc;

namespace OnlineShoppingStore.Models.Home
{
    public class HomeIndexViewModel
    {
        public GenericUnitOfWork _unitOfWork = new GenericUnitOfWork();
        dbMyonlineShoppingEntities1 context = new dbMyonlineShoppingEntities1();

        public IPagedList<Tbl_Product> ListOfProducts { get; set; }
        public HomeIndexViewModel CreateModel(string search, int pageSize, int? page)
        {
            SqlParameter[] param = new SqlParameter[]{
                new SqlParameter("@search", search??(object)DBNull.Value)
            };
            IPagedList<Tbl_Product> data = context.Database.SqlQuery<Tbl_Product>("GetBySearch @search", param);
            return new HomeIndexViewModel
            {
                ListOfProducts = data
            };
        }
    }
}

index.cshtml
OnlineShoppingStore          OnlineShoppingStore.Models.productDetail          CategoryDetail.cs
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations;
4  using System.Linq;
5  using System.Web;
6  using System.Web.Mvc;
7
8  namespace OnlineShoppingStore.Models
9  {
10     public class CategoryDetail
11     {
12         public int CategoryId { get; set; }
13         [Required(ErrorMessage ="Category Name Required")]
14         [StringLength(100,ErrorMessage = "Minimum3 and Maximum3 5 and maximum 100 character are allowed",MinimumLength =3)]
15         public string CategoryName { get; set; }
16         public Nullable<bool> IsActive { get; set; }
17         public Nullable<bool> IsDelete { get; set; }
18     }
19
20     public class productDetail ...
21 }
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```

Word - Screenshot-phas-2.docx

```
namespace OnlineShoppingStore.Models
{
    public class Paymentdetail
    {
        public int ProductId { get; set; }
        [Required(ErrorMessage = "Product Name Required")]
        [StringLength(100, ErrorMessage = "Minimum3 and Maximum3 5 and maximum 100 character")]
        public string ProductName { get; set; }
        [Required]
        [Range(1, 50)]
        public Nullable<int> CategoryId { get; set; }
        public Nullable<bool> IsActive { get; set; }
        public Nullable<bool> IsDelete { get; set; }
        public string CreatedDate { get; set; }
        public string ModifiedDate { get; set; }
        [Required(ErrorMessage = "Description Name Required")]
        public string Description { get; set; }
        public string ProductImage { get; set; }
        public Nullable<bool> IsFeatured { get; set; }
        [Required]
        [Range(typeof(int), "1", "500", ErrorMessage = "Invalid Quantity")]
        public Nullable<int> Quantity { get; set; }
        [Required]
        [Range(typeof(decimal), "1", "200000", ErrorMessage = "Invalid Price")]
        public Nullable<decimal> Price { get; set; }
        //public SelectList Categories { get; set; }
        //public SelectList Categories { get; set; }
    }
}

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

namespace OnlineShoppingStore.Models
{
    public class Shippingdetail
    {
        public int ShippingDetailId { get; set; }
        [Required]
        public Nullable<int> MemberId { get; set; }
        [Required]
        public string Adress { get; set; }
        [Required]
        public string City { get; set; }
        [Required]
        public string State { get; set; }
        [Required]
        public string Country { get; set; }
        [Required]
        public string ZipCode { get; set; }
        public Nullable<int> OrderId { get; set; }
        public Nullable<decimal> AmountPaid { get; set; }
        [Required]
        public string PaymentType { get; set; }
    }
}
```

```

using OnlineShoppingStore.DAL;
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Linq;
using System.Linq.Expressions;
using System.Web;

namespace OnlineShoppingStore.Repository
{
    public class GenericRepository<Tbl_Entity> : IRepository<Tbl_Entity> where Tbl_Entity : class
    {
        DbSet<Tbl_Entity> _dbSet;
        private dbMyonlineShoppingEntities1 _DBEntity;

        public GenericRepository(dbMyonlineShoppingEntities1 DBEntity)
        {
            _DBEntity = DBEntity;
            _dbSet = _DBEntity.Set<Tbl_Entity>();
        }

        public IEnumerable<Tbl_Entity> GetProduct()
        {
            return _dbSet.ToList();
        }

        public void Add(Tbl_Entity entity)
        {
            _dbSet.Add(entity);
            _DBEntity.SaveChanges();
        }

        public int GetAllrecordCount()
        {
            return _dbSet.Count();
        }

        public int GetAllrecordCount()
        {
            return _dbSet.Count();
        }

        public IEnumerable<Tbl_Entity> GetAllRecords()
        {
            return _dbSet.ToList();
        }

        public IQueryable<Tbl_Entity> GetAllRecordsIQueryable()
        {
            return _dbSet;
        }

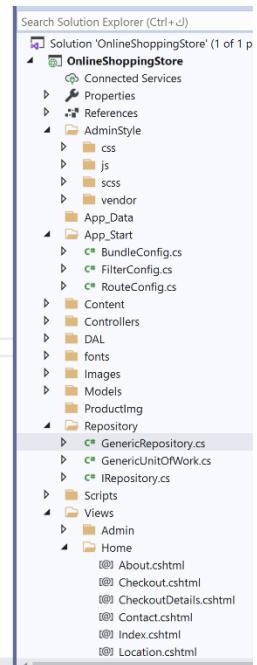
        public Tbl_Entity GetFirstorDefault(int recordId)
        {
            return _dbSet.Find(recordId);
        }

        public Tbl_Entity GetFirstorDefaultByParameter(Expression<Func<Tbl_Entity, bool>> wherePredict)
        {
            return _dbSet.Where(wherePredict).FirstOrDefault();
        }

        public IEnumerable<Tbl_Entity> GetListParameter(Expression<Func<Tbl_Entity, bool>> wherePredict)
        {
            return _dbSet.Where(wherePredict).ToList();
        }

        public IEnumerable<Tbl_Entity> GetResultBySqlprocedure(string query, params object[] parameters)
        {
            if (parameters != null)

```



```

public IEnumerable<Tbl_Entity> GetResultBySqlprocedure(string query, params object[] parameters)
{
    if (parameters != null)
    {
        return _DBEntity.Database.SqlQuery<Tbl_Entity>(query, parameters).ToList();
    }
    else
        return _DBEntity.Database.SqlQuery<Tbl_Entity>(query).ToList();
}

public void InactiveAndDeleteMarkByWhereClause(Expression<Func<Tbl_Entity, bool>> wherePredict, Action<Tbl_Entity> ForEachP
{
    _dbSet.Where(wherePredict).ToList().ForEach(ForEachPredict);
}

public void Remove(Tbl_Entity entity)
{
    if (_DBEntity.Entry(entity).State == EntityState.Detached)
        _dbSet.Attach(entity);
    _dbSet.Remove(entity);
}

public void RemovebyWhereClause(Expression<Func<Tbl_Entity, bool>> wherePredict)
{
    Tbl_Entity entity = _dbSet.Where(wherePredict).FirstOrDefault();
    Remove(entity);
}

public void RemoveRangeBywhereClause(Expression<Func<Tbl_Entity, bool>> wherePredict)
{
    List<Tbl_Entity> entity = _dbSet.Where(wherePredict).ToList();
    foreach (var ent in entity)
    {
        Remove(ent);
    }
}

public void RemoveRangeBywhereClause(Expression<Func<Tbl_Entity, bool>> wherePredict)
{
    List<Tbl_Entity> entity = _dbSet.Where(wherePredict).ToList();
    foreach (var ent in entity)
    {
        Remove(ent);
    }
}

public void Update(Tbl_Entity entity)
{
    _dbSet.Attach(entity);
    _DBEntity.Entry(entity).State = EntityState.Modified;
    _DBEntity.SaveChanges();
}

public void UpdateByWhereClause(Expression<Func<Tbl_Entity, bool>> wherePredict, Action<Tbl_Entity> ForEachPredict)
{
    _dbSet.Where(wherePredict).ToList().ForEach(ForEachPredict);
}

public IEnumerable<Tbl_Entity> GetRecordsToShow(int pageNo, int pageSize, int currentPage, Expression<Func<Tbl_Entity, bool
{
    if (wherePredict != null)
    {
        return _dbSet.OrderBy(orderByPredict).Where(wherePredict).ToList();
    }

    else
    {
        return _dbSet.OrderBy(orderByPredict).ToList();
    }
}

```

```

public IEnumerable<Tbl_Entity> GetRecordsToShow(int pageNo, int pageSize,
{
    if (wherePredict != null)
    {
        return _dbSet.OrderBy(orderByPredict).Where(wherePredict).ToList();
    }

    else
    {
        return _dbSet.OrderBy(orderByPredict).ToList();
    }
}

namespace OnlineShoppingStore.Repository
{
    public class GenericUnitOfWork : IDisposable
    {
        private dbMyonlineShoppingEntities1 DBEntity = new dbMyonlineShoppingEntities1();
        public IRepository<Tbl_EntityType> GetRepositoryInstance<Tbl_EntityType>() where Tbl_EntityType : class
        {
            return new GenericRepository<Tbl_EntityType>(DBEntity);
        }

        public void SaveChanges()
        {
            DBEntity.SaveChanges();
        }

        protected virtual void Dispose(bool disposing)
        {
            if (!this.disposed)
            {
                if (disposing)
                {
                    DBEntity.Dispose();
                }
                this.disposed = true;
            }
        }

        public void Dispose()
        {
            Dispose(true);
            GC.SuppressFinalize(this);
        }

        public void Dispose()
        {
            Dispose(true);
            GC.SuppressFinalize(this);
        }
    }

    private bool disposed = false;
}

```

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

namespace OnlineShoppingStore.Repository
{
    public interface IRepository<Tbl_Entity> where Tbl_Entity : class
    {
        IEnumerable<Tbl_Entity> GetProduct();
        IEnumerable<Tbl_Entity> GetAllRecords();
        IQueryable<Tbl_Entity> GetAllRecordsIQueryable();
        int GetAllRecordCount();
        void Add(Tbl_Entity entity);
        void Update(Tbl_Entity entity);
        void UpdateByWhereClause(Expression<Func<Tbl_Entity, bool>> wherePredict, Action<Tbl_Entity> ForEachPredict);
        Tbl_Entity GetFirstOrDefault(int recordId);
        void Remove(Tbl_Entity entity);
        void RemovebyWhereClause(Expression<Func<Tbl_Entity, bool>> wherePredict);
        void RemoveRangeByWhereClause(Expression<Func<Tbl_Entity, bool>> wherePredict);
        void InactiveAndDeleteMarkByWhereClause(Expression<Func<Tbl_Entity, bool>> wherePredict, Action<Tbl_Entity> ForEachPredict);
        Tbl_Entity GetFirstOrDefaultByParameter(Expression<Func<Tbl_Entity, bool>> wherePredict);
        IEnumerable<Tbl_Entity> GetListParameter(Expression<Func<Tbl_Entity, bool>> wherePredict);
        IEnumerable<Tbl_Entity> GetResultBySqlprocedure(string query, params object[] parameters);
        IEnumerable<Tbl_Entity> GetRecordsToShow(int pageNo, int pageSize, int currentPage, Expression<Func<Tbl_Entity, bool>> wherePred
    }
}
```

