

Online Shopping Management System

Team Members

Name	Role
Balamurali T	Teamlead
Krithika	DB lead
DivyaSri	UI lead
Althaf	Controller lead

Technology Used:

- ❖ Asp.Net Core MVC
- ❖ Database-Sql
- ❖ EntityFramework- DatabasedFirstApproach

Modules:

- ❖ Admin Module
- ❖ Retailer Module
- ❖ Customer Module

Admin Module:

Features:

- ❖ Login & Signup
- ❖ Crud Operation on Retailer-Add/Edit/Delete
- ❖ Product Dashboard-List of products Displayed
- ❖ Customer Dashboard-List of Customer

Retailer Module:

Features:

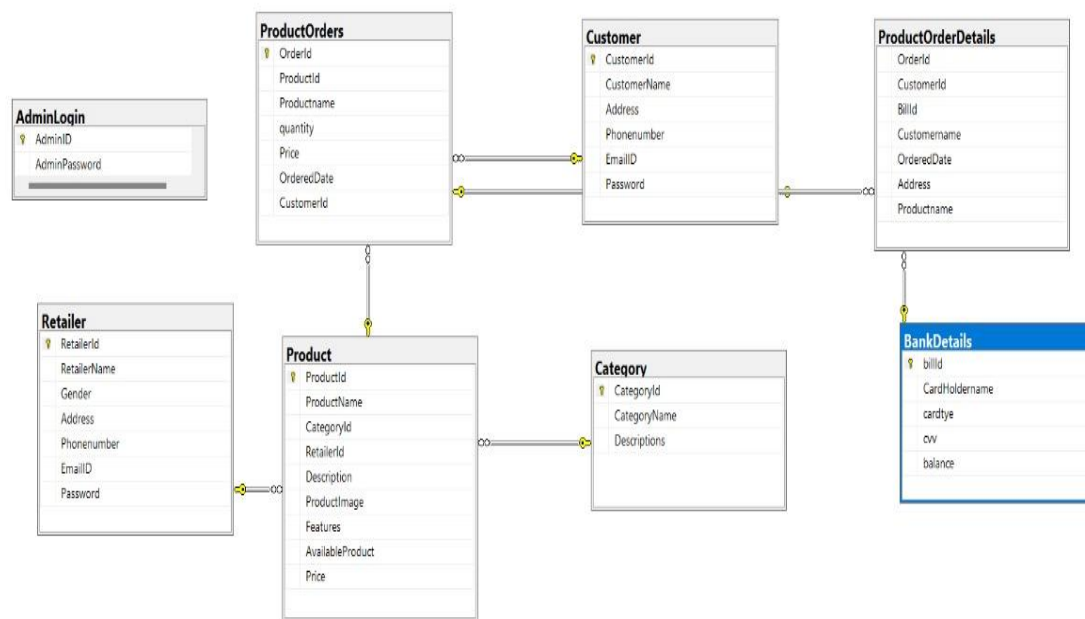
- ❖ Login Module
- ❖ Crud Operation on Category-Add/Edit/Delete
- ❖ Crud Operation on Products-Add/Edit/Delete
- ❖ Search Operation on Products & Category
- ❖ Order DashBoard By Customer
- ❖ My Accounts

Customer Module:

Features:

- ❖ Signup & Login Module
- ❖ Search Operation on Products
- ❖ Add To Cart
- ❖ Product Order Placement Module
- ❖ Payment Module
- ❖ My Order View Module
- ❖ My Accounts
- ❖ Logout

DataBase OverView:



Project Task Done By Teammates

Name	Task
Balamurali	<ul style="list-style-type: none">❖ Implementation Of Admin module❖ Implementation Of Retailer Module❖ Implementation Of HomepageLayout With Product Display & Search Products.❖ Implementation Of ProductImage Upload Module❖ Session Concept Implementation
Krithika	<ul style="list-style-type: none">❖ DB Design❖ Add To Cart Module❖ Product Order Placement Module❖ My Order View Module
DivyaSri	<ul style="list-style-type: none">❖ Crud Operation Product Module❖ Crud Operation On Category Module
Althaf Hussain	<ul style="list-style-type: none">❖ Implementation On Customer Module❖ Login & signup Page❖ Forget Password Module❖ Customer Payment Module

Project Highlights:

❖ Session Concept Implementation

Add Session Service in IConfiguration service in startup.cs and Configured with Pipeline. With Help of Session concept session id will be maintained for 5 minutes.

❖ SetSession ID in Login module

```
HttpContext.Session.SetString("uid",adminLogin.AdminId);
```

❖ GetSessionID

```
ViewBag.uid = HttpContext.Session.GetString("uid");
```

❖ RemoveSession Id in Logout module

```
HttpContext.Session.Remove("uid");
```

❖ Product Image Upload:

In Product Model Class I form File Used For Uploading File.To Access Environment IWebHostEnvironment is used.

Image Upload:

Code:

```
string wwwRootPath = _hostEnvironment.WebRootPath;
    string fileName =
Path.GetFileNameWithoutExtension(product.ImageFile.FileName);
    string extension =
Path.GetExtension(product.ImageFile.FileName);
    product.ProductImage = fileName = fileName +
DateTime.Now.ToString("yymmssff") + extension;
    string path = Path.Combine(wwwRootPath + "/Image", fileName);
    using (var filestream = new FileStream(path, FileMode.Create))
    {
        await product.ImageFile.CopyToAsync(filestream);
    }
```

Delete Image:

```
var imagepath = Path.Combine(_hostEnvironment.WebRootPath,
"Image", product.ProductImage);
    if (System.IO.File.Exists(imagepath))
        System.IO.File.Delete(imagepath);
    _context.Products.Remove(product);
    await _context.SaveChangesAsync();
    return RedirectToAction(nameof(Index));
```

Edit Image:

```
{
    string wwwRootPath = _hostEnvironment.WebRootPath;
    string fileName =
Path.GetFileNameWithoutExtension(product.ImageFile.FileName);
    string extension =
Path.GetExtension(product.ImageFile.FileName);
    product.ProductImage = fileName = fileName +
DateTime.Now.ToString("yymmssff") + extension;
```

```

        string path = Path.Combine(wwwRootPath + "/Image", fileName);
        using (var filestream = new FileStream(path, FileMode.Create))
        {
            await product.ImageFile.CopyToAsync(filestream);
        }

        _context.Update(product);
        await _context.SaveChangesAsync();
    }

```

❖ **Search Products:**

Based On user Input Product will be displayed From Product Table

```

public async Task<IActionResult> SearchProduct(string searchString)
{
    ViewBag.uid = HttpContext.Session.GetString("uid");
    var pro = from m in db.Products
              select m;

    if (!String.IsNullOrEmpty(searchString))
    {
        pro = pro.Where(s => s.ProductName!.Contains(searchString));
    }
    else
    {
        ViewBag.Message = "Product Not Found";
        return View();
    }

    return View(await pro.ToListAsync());
}

```

Product Order Confirmation Module:

Implement Those Module Using Joint Concept In Linq Query based On three Table:

- ❖ Customer Table : Customer Id, Customer Name, Customer Address.
- ❖ Product Table: Product Id, Product Name
- ❖ Product Order: Product Ordered, Quantity, Price, OrderedDate

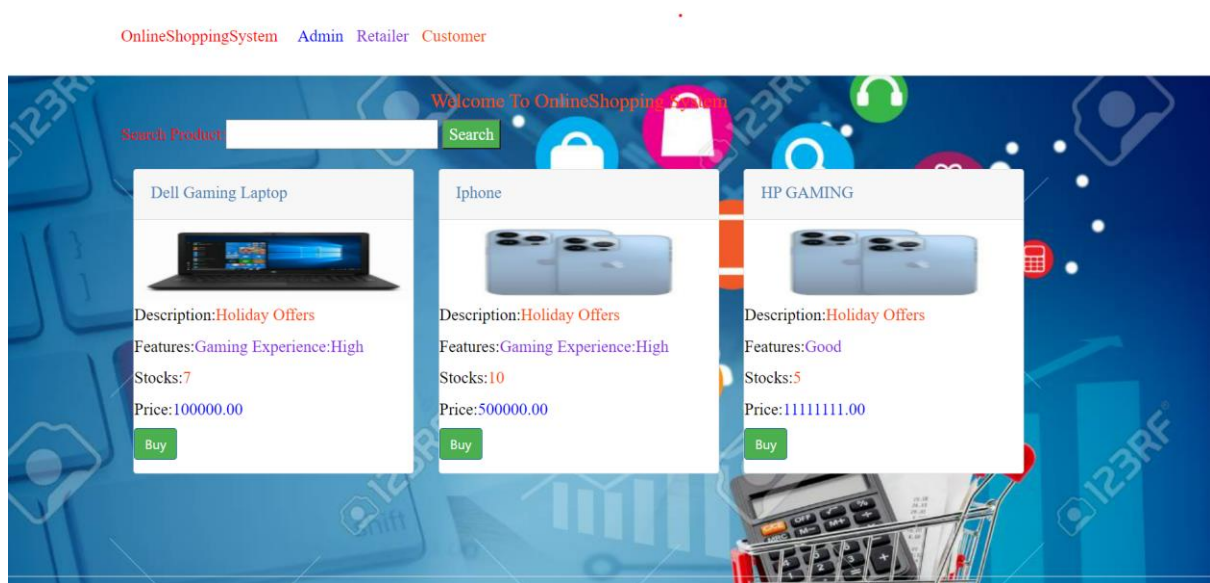
```

public ActionResult MyOrderConfirmation(Customer c)
{
    ViewBag.uid = HttpContext.Session.GetString("uid");
    string email = ViewBag.uid;
    var ad1 = (from p in db.Customers
               where p.EmailId == email
               select p.CustomerId).FirstOrDefault();
    var customerlist = (from cust in db.Customers
                        join
or in db.ProductOrders on cust.CustomerId==ad1 equals or.CustomerId==
ad1
                        join m in db.Products on or.ProductId equals m.ProductId
                        select new { cust.CustomerId, cust.CustomerName,
cust.Address, or.OrderId, or.Productname, or.Quantity,
or.OrderedDate,m.Price }).ToList();
    List<ProductOrderDetail> lvm = new List<ProductOrderDetail>();
    foreach (var item in customerlist)
    {
        ProductOrderDetail objvm = new ProductOrderDetail();
        objvm.OrderId = item.OrderId;
        objvm.Productname = item.Productname;
        objvm.quantity = item.Quantity;
        objvm.Price = (int)item.Price;
        objvm.TotalPrice = (objvm.quantity)*(int)objvm.Price;
        objvm.CustomerId = item.CustomerId;
        objvm.Address = item.Address;
        objvm.OrderedDate = item.OrderedDate;
        objvm.PaymentStatus = "paid";
        objvm.Deliverydate = DateTime.Today.AddDays(10);
        lvm.Add(objvm);
        //db.ProductOrderDetails.Add(objvm);
        //db.SaveChanges();
    }

    return View(lvm);
}

```

HomePage:



Product Cart Design Code:

```
@foreach (var item in Model)
{
    <div class="col-4 float-left h-20 w-20">

        <div class="card h-100 w-100">
            <div class="card-header">
                <h4 class="my-0 font-weight-normal">
                    @*<label style="font-size:23px;color:steelblue">
@Html.DisplayNameFor(model => model.ProductName)</label>*@
                    <label style="font-size: 20px; font-family: 'Times New Roman', Times, serif;
color: steelblue">@Html.DisplayFor(modelItem => item.ProductName) </label>
                </h4>
            </div>
            <div>
                
            </div>
            <div class="d-flex justify-content-between align-items-center">
                <div class="btn-group">
                    <p1 style="font-size: 20px; font-family: 'Times New Roman', Times,
serif;color:black ">Description:</p1><label style="font-size: 20px; font-family: 'Times New
Roman', Times, serif; color: orangered">@Html.DisplayFor(modelItem =>
item.Description)</label>
                </div>
            </div>
            <div class="d-flex justify-content-between align-items-center">
                <div class="btn-group">
                    <p1 style="font-size: 20px; font-family: 'Times New Roman', Times,
serif;color:black ">Features:</p1> <label style="font-size: 20px; font-family: 'Times New
```

```

Roman', Times, serif; color: blueviolet">@Html.DisplayFor(modelItem =>
item.Features)</label>
</div>
</div>
<div class="d-flex justify-content-between align-items-center">
<div class="btn-group">
<p1 style="font-size: 20px; font-family: 'Times New Roman', Times,
serif;color:black ">Stocks:</p1> <label style="font-size: 20px; font-family: 'Times New
Roman', Times, serif; color:orangered">@Html.DisplayFor(modelItem =>
item.AvailableProduct)</label>
</div>
</div>
<div class="d-flex justify-content-between align-items-center">
<div class="btn-group">
<p1 style="font-size: 20px; font-family: 'Times New Roman', Times,
serif;color:black ">Price:</p1> <label style="font-size: 20px; font-family: 'Times New
Roman', Times, serif; color: blue">@Html.DisplayFor(modelItem => item.Price)</label>
</div>
</div>
<form asp-controller="Customer" asp-action="CustomerLogin" method="get">
<div class="form-group">
<input type="submit" value="Buy" class="btn btn-primary" />
</div></form>
</div>

```