



OpenMarket

ASP.NET e-commerce project

Ali Alaghbandrad
Diego Marco Alayo
Leon Digerman
Malik Zafar



What is OpenMarket?

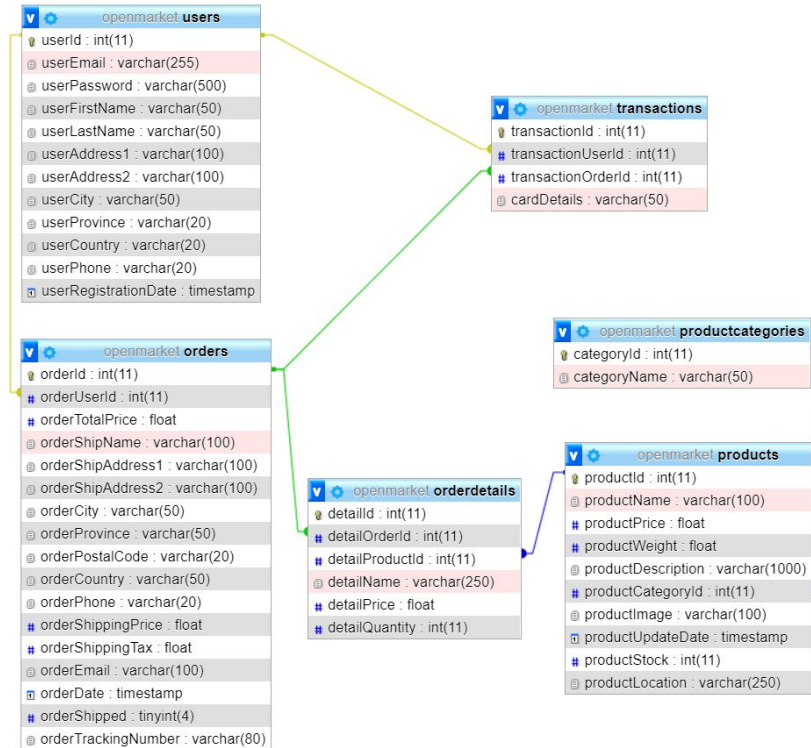
- E-commerce application
- Users (customers): Select products and quantity to purchase
- Users: Browse products by category
- Users: Rate and write reviews for products
- Admin: CRUD products
- Admin: Edit/Delete users

Tools Used

- ASP.NET Razor
- Paypal API
- Trello, Bitbucket, SourceTree

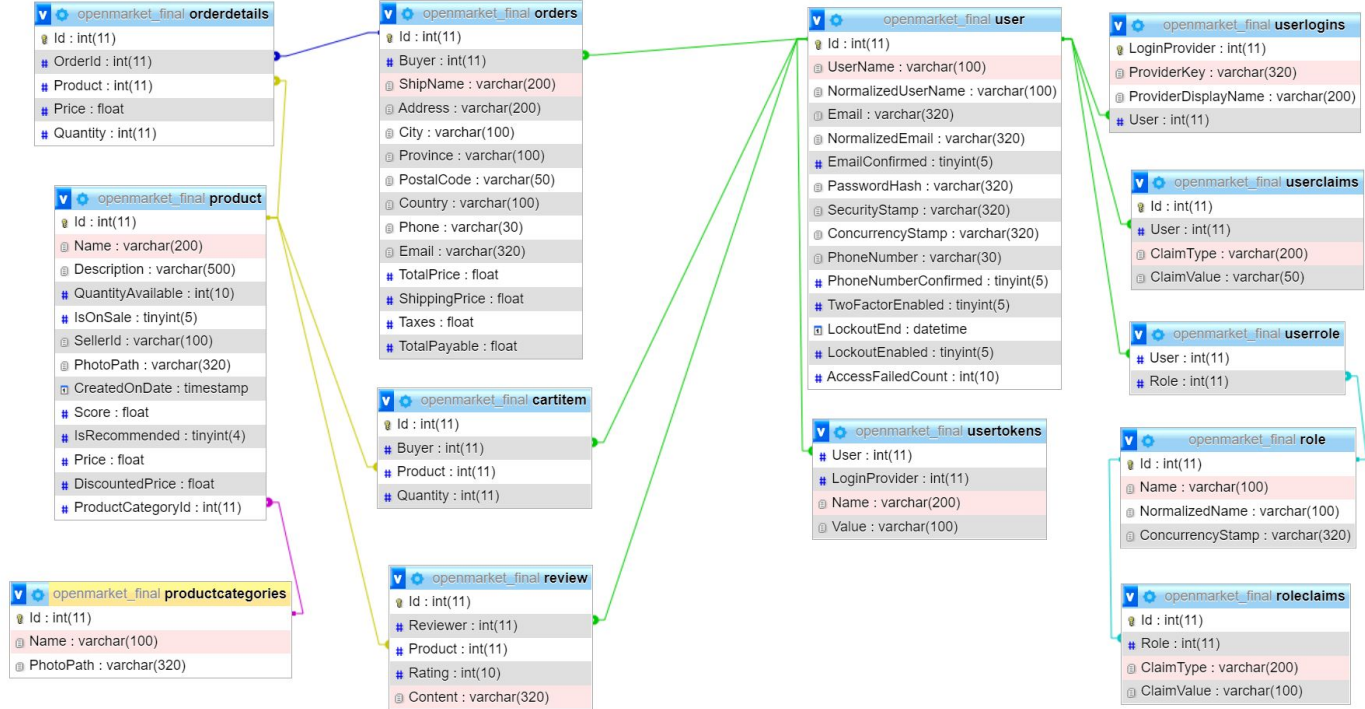
Database

First Draft



Database

Final Draft



Issues Encountered (Products and Payment Section)

- Moving between Javascript and Razorpages C# during integrating payment gateway.
- Easy to pass Razorpages variables to Javascript
- More difficult to pass information from Javascript to razor pages

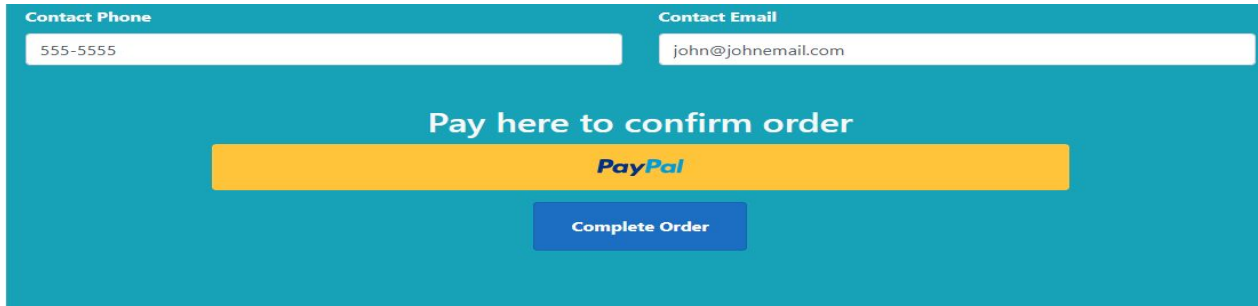
```
<input asp-for="@Model.PaymentDone">
```

```
... <input type="text" data-val="true" data-val-number="The field GrandTotal must be a number."  
data-val-required="The GrandTotal field is required." id="GrandTotal" name="GrandTotal"  
value="42.30"> == $0
```

Issues encountered page-2

```
onApprove: function(data, actions) {  
    return actions.order.capture().then(function(details) {  
        var passForm = document.getElementById("passForm")  
        var inp = document.createElement("input");  
        inp.setAttribute('type', 'hidden');  
        inp.setAttribute('data-val', 'true');  
        inp.setAttribute('data-val-required', 'the PaymentDone field is required');  
        inp.setAttribute('name', 'PaymentDone');  
        inp.setAttribute('id', 'PaymentDone');  
        inp.setAttribute('value', '2');  
        passForm.appendChild(inp);  
    });  
}
```

Input and button appended into form to use OnPost



Contact Phone

555-5555

Contact Email

john@johnemail.com

Pay here to confirm order

PayPal

Complete Order

Layout/Design of Products Section

- Layout tried to accomodate all the variations of product being loaded from database
- Attempt to visually show to buyer the state of the items (out of stock / sale)

Guitar



a guitar

SALE ~~\$100.00~~ \$75.00

[View Detail](#)

Piano




a piano

\$330.00

[View Detail](#)

Piano



a piano

\$330.00

[View Detail](#)

Piano



a piano

SALE ~~\$330.00~~
\$290.00

[View Detail](#)

Code Snippets

```
@page "{id:int}"
@model OpenMarket.Pages.ProductviewModel
@{
    ViewData["Title"] = "View Product";
    Model.CardBg = Model.SelectedProduct.QuantityAvailable >= 1 ? "bg-info" : "bg-danger";
    string PhotoLoc = "images/box1.png";
}

<div class="container text-center">
    <div class="row">
        <div class="col-12 @Model.CardBg m-4 p-5 text-center product-display">
            <h1 class="text-light mb-3">@Model.SelectedProduct.Name</h1>
            @{
                if(@Model.SelectedProduct.PhotoPath != null){
                    PhotoLoc = @Model.SelectedProduct.PhotoPath;
                }else{
                    if(@Model.SelectedProduct.QuantityAvailable <= 0){
                        PhotoLoc = "images/openbox.png";
                    }
                }
            }
        </div>
    </div>
</div>
```

Code used to decide which photo to use for the image of a product

```
}
```

```
Product = await _context.Products.Include(p => p.ProductCategory).FirstOrDefaultAsync(m => m.Id == id);
```

Eager Loading a table using the include keyword

Future Work

- Smoother error handling
- Better layout
- use libraries

Admin: CRUD Product Category

OpenMarket [Home](#) [Privacy](#)

Index

[Add Category](#)

Name	
Furniture	Edit Details Delete
Electronics	Edit Details Delete
Music	Edit Details Delete
Sports	Edit Details Delete
Home Decor	Edit Details Delete
Misc	Edit Details Delete

Admin: CRUD Product Category

OpenMarket Home Privacy

Details

ProductCategory

PhotoPath



Name

Sports

[Edit](#) | [Back to List](#)

OpenMarket Home Privacy

Edit

ProductCategory

Name

[Back to List](#)

Sports

PhotoPath

Uploads\sport.jpg

Image

No file chosen

Admin: CRUD Product Review

OpenMarket [Home](#) [Privacy](#)

[Edit/Manage](#) [Log out \(admin@admin.com\)](#)

Index

Title:

Product	Rating	Content	
Sofa	9	Perfect!	Edit Details Delete
Sofa	10	I just love every thing about this sofa	Edit Details Delete
Sofa	8	Big, Quality and Comfy!	Edit Details Delete

```
Review = await _context.Reviews.Include(p => p.Product).ToListAsync();
```



a sofa
Category: Furniture
\$200.00
14 in stock

Amount

Buy

Product Reviews

(9 / 10)

admin@admin.com posted:

Perfect!

(10 / 10)

admin@admin.com posted:

I just love every thing about this sofa

(8 / 10)

Tom123@mail.com posted:

Big, Quality and Comfy!

User : See all Customers Reviews

```
NewReview.Product = Product;
```

```
var user = User.Identity.Name;
```

```
NewReview.Reviewer = db.Users.FirstOrDefault(u =>  
u.UserName == user);
```

Admin: Manage Users

List of users:

```
public List<Microsoft.AspNetCore.Identity.IdentityUser> UserList { get; set; } = new  
List<Microsoft.AspNetCore.Identity.IdentityUser>();
```

Not secure | https://localhost:5001/Admin/Users

OpenMarket Home Privacy Edit/Manage Log out (admin@admin.com)

Manage Users

Email	Manage
admin@admin.com	Edit Delete
user1234@user.com	Edit Delete
jack@email.com	Edit Delete
user@email.com	Edit Delete
alex@email.com	Edit Delete

```
<table class="table">  
  <thead>  
    <tr>  
      <th scope="col">Email</th>  
      <th scope="col">Manage</th>  
    </tr>  
  </thead>  
  <tbody>  
    @foreach (var u in Model.UserList)  
    {  
      <tr>  
        <td>@u.Email</td>  
        <td>  
          <a asp-page="./Edit" asp-route-Email="@u.Email">Edit</a> |  
          <a asp-page="./Delete" asp-route-Email="@u.Email">Delete</a>  
        </td>  
      </tr>  
    }  
  </tbody>  
</table>
```

Admin: Manage Users

Edit users:

Challenge:

Username/Email change

https://localhost:5001/Admin/Users/Edit?Email=user1234@user.com

OpenMarket Home Privacy Edit/Manage Log out (admin@admin.com)

Edit Profile

Email

Cancel Update Account

```
public async Task<IActionResult> OnPostAsync()
{
    if (!ModelState.IsValid)
    {
        return Page();
    }
    var user = db.Users.FirstOrDefault(x => x.Email == Email);
    if (user != null)
    {
        user.Email = Input.Email;
        user.UserName = Input.Email;
        user.NormalizedEmail = Input.Email.ToUpper();
        user.NormalizedUserName = Input.Email.ToUpper();
        await db.SaveChangesAsync();
    }

    return RedirectToPage("./EditSuccess");
}
```


Admin: Manage Users

Delete users:

Challenge:

Delete User and its role

https://localhost:5001/Admin/Users/Delete?Email=user1234@user.com

OpenMarket Home Privacy Edit/Manage Log out (admin@admin.com)

Delete Profile

Email

user1234@user.com

Cancel Delete

```
public async Task<IActionResult> OnPostAsync()
{
    var user = db.Users.FirstOrDefault(x => x.Email == Email);
    if (user != null)
    {
        try {
            var rolesForUser = await _userManager.GetRolesAsync(user);
            foreach (var item in rolesForUser.ToList())
            {
                var result = await _userManager.RemoveFromRoleAsync(user, item);
            }
            await _userManager.DeleteAsync(user);
            await db.SaveChangesAsync();
            return RedirectToPage("./DeleteSuccess");
        } catch (Microsoft.EntityFrameworkCore.DbUpdateException ex) {
            return NotFound("User cannot be deleted");
        }
    }
    return Page();
}
```

Admin/User: Reset Password & Username

Password &
username change:

Challenge:

ChangePasswordAsync()

OpenMarket Home Privacy Log out Edit (user@email.com)

Account Profile

Email
user@email.com

Current Password:

New Password:

Confirm Password:

```
public async Task<IActionResult> OnPostAsync()
{
    if (!ModelState.IsValid)
    {
        return Page();
    }

    var user = await _userManager.GetUserAsync(User);
    var result = await _userManager.ChangePasswordAsync(user, Input.CurrentPassword, Input.NewPassword);
    user.Email = Input.Email;
    user.UserName = Input.Email;
    user.NormalizedEmail = Input.Email.ToUpper();
    user.NormalizedUserName = Input.Email.ToUpper();
    await db.SaveChangesAsync();
    return RedirectToPage("/Index");
}
```

User: Cart of Products

Visual Studio




Working Cart

ECommerceShoppingCartASPNetCore [Home](#) [Privacy](#)

Product List

Id	Name	Photo	Price	Buy
1	Flower1		2.8	Buy Now
2	Flower2		3.5	Buy Now
3	Flower3		4.4	Buy Now
4	Flower4		3.7	Buy Now
5	Flower5		5.6	Buy Now
6	Flower6		2.8	Buy Now
7	Flower7		4.5	Buy Now
8	Flower8		5.9	Buy Now
9	Flower9		1.7	Buy Now
10	Flower10		3.8	Buy Now

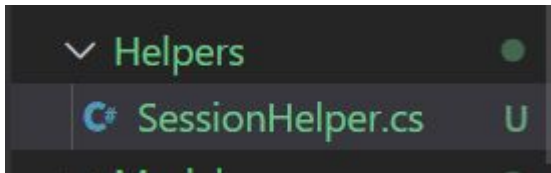
Cart Page

Option	Id	Name	Photo	Price	Quantity	Sub Total
Remove	2	Flower2		3.5	1	3.5
Remove	4	Flower4		3.7	2	7.4
Remove	6	Flower6		2.8	1	2.8
Sum						13.7

[Continue Shopping](#) [Go to Checkout](#)

User: Cart of Products - cont.

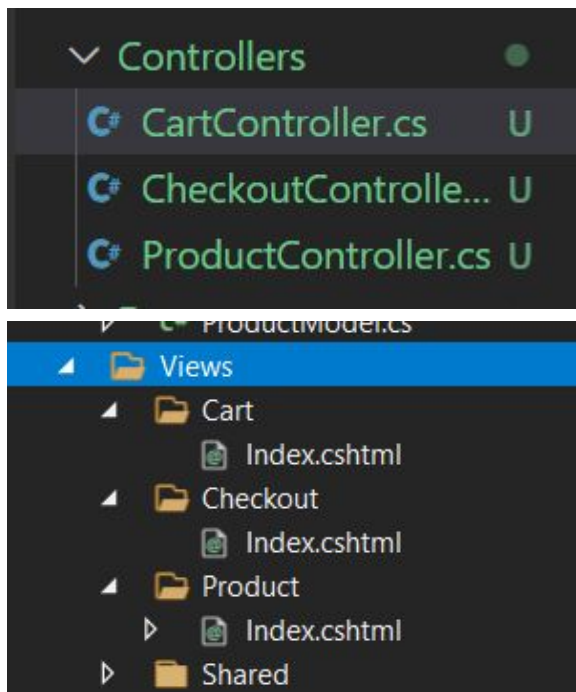
Session Helpers



```
1  using Microsoft.AspNetCore.Http;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Threading.Tasks;
6  using System.Text.Json.Serialization;
7  using Newtonsoft.Json;
8
9  namespace OpenMarket.Helpers
10 {
11     9 references
12     public static class SessionHelper
13     {
14         3 references
15         public static void SetObjectAsJson(this ISession session, string key, object value)
16         {
17             session.SetString(key, JsonConvert.SerializeObject(value));
18         }
19
20         6 references
21         public static T GetObjectFromJson<T> (this ISession session, string key)
22         {
23             var value = session.GetString(key);
24             return value == null? default : JsonConvert.DeserializeObject<T>(value);
25         }
26     }
27 }
```

User: Cart of Products - cont.

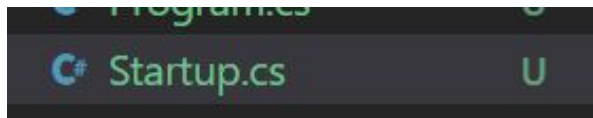
Controller Interfaces



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Threading.Tasks;
5  using OpenMarket.ViewModels;
6  using Microsoft.AspNetCore.Mvc;
7
8  namespace OpenMarket.Controllers
9  {
10     0 references
11     public class ProductController : Controller
12     {
13         0 references
14         public IActionResult Index()
15         {
16             ProductModel productModel = new ProductModel();
17             ViewBag.products = productModel.findAll();
18             return View();
19         }
20     }
```

User: Cart of Products - cont.

EndPoint Map Controller Route



```
app.UseEndpoints(endpoints =>
{
    endpoints.MapRazorPages();
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller}/{action=Index}/{id?}"
    );
});
```

Future Work

Use of Controller Interface

Learn more Json

Implement a system where any user can post product (multiple sellers)

Team-work

Task	Register User & Admin	Login User & Admin	Admin can view, update & delete users	User/Admin can update username & password	Admin can CRUD products	Admin can CRUD product category	User can browse/search for products by category	User can leave a product review	User can pay for product	User manage items in cart
Ali	*	*	*	*	*					
Diego										*
Leon						*		*		
Malik			*	*	*		*		*	

The end

Thank you