

# BÀI TẬP THỰC HÀNH 09

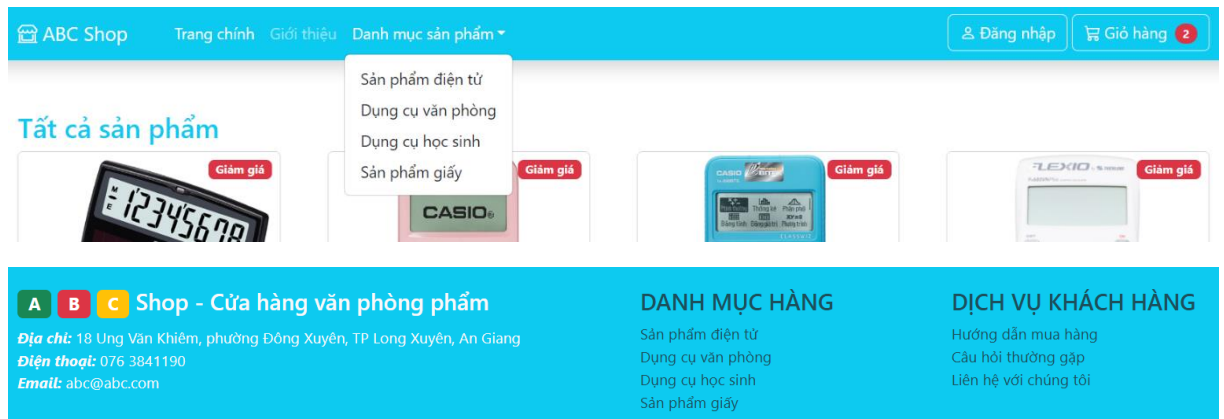
(SV sử dụng kết quả bài 8 để thực hiện tiếp các yêu cầu trong bài tập này.)

## 1. Bổ sung một số dữ liệu trên trang chính

- Đếm số mặt hàng trong giỏ, hiển thị bên cạnh liên kết Giỏ hàng.



- Hiển thị các danh mục lên thanh liên kết, khi chọn một danh mục thì liệt kê các sản phẩm thuộc danh mục đó.



- Tạo View xem sản phẩm thuộc danh mục được chọn.

## 2. Xác thực người dùng

Viết chức năng đăng ký tài khoản khách hàng mới, đăng nhập, đăng xuất,...

Chuẩn bị: Để thực hiện mã hóa mật khẩu, khai báo bổ sung cho các tập tin sau:

- Trong **Controllers/CustomerController.cs**, khai báo thêm biến `_passwordHasher`:

```
using Microsoft.AspNetCore.Identity;

namespace mvc.Controllers
{
    public class HomeController : Controller
    {
        private readonly ApplicationDbContext _context;

        private readonly IPasswordHasher<Khachhang> _passwordHasher;

        public HomeController(ApplicationDbContext context, IPasswordHasher<Khachhang> passwordHasher)
        {
            _context = context;
            _passwordHasher = passwordHasher;
        }
    }
}
```

- Trong **Program.cs**, bổ sung:

```
builder.Services.AddSingleton<IPasswordHasher<Khachhang>, PasswordHasher<Khachhang>>();
```

## 2.1. Đăng ký tài khoản

## ĐĂNG KÝ TÀI KHOẢN

Email

Nhập email

Mật khẩu

Mật khẩu

Họ tên

Nhập họ tên

Số điện thoại

Nhập số điện thoại

Đăng ký

- Trong **Controllers/CustomerController.cs**, viết các phương thức: hiển thị form đăng ký tài khoản khách hàng, lưu thông tin khách hàng

```
public IActionResult Register()
{
    GetData();
    return View();
}
[HttpPost]
public async Task<IActionResult> Register(string hoten, string dienthoai,
string email, string matkhau)
{
    Khachhang kh = new Khachhang();
    kh.Ten = hoten;
    kh.DienThoai = dienthoai;
    kh.Email = email;
    kh.MatKhau = _passwordHasher.HashPassword(kh, matkhau); // mã hóa mk

    if (ModelState.IsValid)
    {
        _context.Add(kh);
        await _context.SaveChangesAsync();
    }
    return RedirectToAction(nameof(Login));
}
```

- Tạo View/Customer/Register.cshtml

```
@{
    ViewData["Title"] = "Đăng ký tài khoản";
}
<div class="row">
    <div class="col-sm-4"></div>
    <div class="col-sm-4">
        <div class="p-3 shadow">
            <h3 class="text-center text-info">ĐĂNG KÝ TÀI KHOẢN</h3>
            <form method="post" asp-action="Register">

                <div class="my-3">
                    <label>Email</label>
                    <input class="form-control" type="email" name="email"
placeholder="Nhập email" required>
                </div>
                <div class="my-3">
                    <label>Mật khẩu</label>
                    <input class="form-control" type="password" name="matkhau"
placeholder="Mật khẩu" required>
                </div>
                <div class="my-3">
                    <label>Họ tên</label>
                    <input class="form-control" type="text" name="hoten"
placeholder="Nhập họ tên" required>
                </div>
                <div class="my-3">
                    <label>Số điện thoại</label>
                    <input class="form-control" type="text" name="dienthoai"
placeholder="Nhập số điện thoại" required>
                </div>
                <br>
                <div class="d-grid">
                    <input class="btn btn-info btn-block" type="submit"
value="Đăng ký">
                </div>
            </form>
        </div>
    </div>
    <div class="col-sm-4"></div>
</div>
```

## 2.2. Đăng nhập

The image shows a login form with the following elements:

- Title:** ĐĂNG NHẬP
- Email Input:** A text box with the placeholder text "Email".
- Password Input:** A text box with the placeholder text "Mật khẩu".
- Login Button:** A blue button with the text "Đăng nhập".
- Registration Link:** A link that says "Chưa có tài khoản? [Đăng ký](#)".

- Trong **Controllers/CustomerController.cs**, viết các phương thức: hiển thị form đăng nhập, kiểm tra thông tin khách hàng hợp lệ

```

public IActionResult Login()
{
    GetData();
    return View();
}
[HttpPost]
public async Task<IActionResult> Login(string email, string matkhou)
{
    var kh = await _context.Khachhangs
        .FirstOrDefaultAsync(m => m.Email == email);
    if (kh != null && _passwordHasher.VerifyHashedPassword(kh,
kh.MatKhou, matkhou) == PasswordVerificationResult.Success)
    {
        // Đăng nhập thành công, thực hiện các hành động cần thiết
        // Ví dụ: Ghi thông tin người dùng vào Session
        HttpContext.Session.SetString("khachhang", kh.Email);
        return RedirectToAction(nameof(CustomerInfo));
    }
    return RedirectToAction(nameof(Login));
}
public IActionResult CustomerInfo()
{
    GetData();
    return View();
}

```

- Tạo View/View/View/Login.cshtml

```

@{
    ViewData["Title"] = "Đăng nhập";
}
<div class="row">
    <div class="col-sm-4"></div>
    <div class="col-sm-4">
        <div class="p-3 shadow">
            <h3 class="text-center text-info">ĐĂNG NHẬP</h3>
            <form method="post" asp-action="Login">

                <input class="form-control" type="text" name="email"
placeholder="Email" required><br>
                <input class="form-control" type="password" name="matkhou"
placeholder="Mật khẩu" required><br>
                <div class="d-grid">
                    <input class="btn btn-info btn-block" type="submit"
value="Đăng nhập">
                </div>
            </form>
            <div class="my-3 text-center">Chưa có tài khoản? <a asp-
action="Register">Đăng ký</a></div>

        </div>
    </div>
</div>

```

- Tạo View/View/View/CustomerInfo.cshtml

```

@{
    ViewData["Title"] = "Hồ sơ khách hàng";
}
@if(ViewBag.khachhang != null)

```

```

{
    <h3>Chào @ViewBag.khachhang.Ten!</h3>
    <p>SV bổ sung nội dung tùy ý</p>
    <p><a asp-action="Signout">Đăng xuất</a></p>
}
else
{
    <p>Vui lòng <a asp-action="Login">Đăng nhập</a></p>
}

```

### 2.3. Đăng xuất

- Trong **Controllers/CustomerController.cs**, viết phương thức đăng xuất

```

public IActionResult Signout()
{
    HttpContext.Session.SetString("khachhang", "");
    GetData();
    return RedirectToAction("Index");
}

```

### 2.4. Xử lý đơn hàng cho trường hợp khách đã đăng nhập

- Nếu khách hàng đã đăng nhập thì hiển thị sẵn thông tin khách hàng tại bước Thanh toán.

## 3. Bài tập về nhà: Hoàn thiện các chức năng quản trị

- Xử lý hình ảnh, file upload
- Áp dụng giao diện admin (kết quả bài tập 5) cho trang quản trị.