2274802010846 – Lê Ngô Gia Thịnh

2274802010388 – Trịnh Đăng Khang

Danh sách sản phẩm

A screenshot of a cellphone

AI-generated content may be incorrect.

Xem chi tiết sản phẩm:

A glass of ice coffee with ice cubes

AI-generated content may be incorrect.

Giỏ hàng:

A screenshot of a computer

AI-generated content may be incorrect.

Product.js:

const mongoose = require('mongoose');

const productSchema = new mongoose.Schema({

    name: String,

    description: String,

    price: Number,

    image: String,

});

const Product = mongoose.model('Product', productSchema);

module.exports = Product;

Cart.ejs:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Giỏ hàng</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            text-align: center;

        }

        .container {

            width: 80%;

            margin: auto;

            position: relative;

        }

        h1 {

            color: #c44;

        }

        .cart-item {

            display: flex;

            align-items: center;

            justify-content: space-between;

            border-bottom: 1px solid #ddd;

            padding: 10px 0;

        }

        .cart-item img {

            width: 80px;

            height: 80px;

            border-radius: 10px;

        }

        .cart-item div {

            flex: 1;

            text-align: left;

            margin-left: 15px;

        }

        .cart-item button {

            background-color: #ff4d4d;

            color: white;

            border: none;

            padding: 5px 10px;

            cursor: pointer;

            border-radius: 5px;

        }

        .quantity {

            display: flex;

            align-items: center;

            gap: 10px;

        }

        .quantity button {

            background-color: #28a745;

        }

        .total {

            font-size: 20px;

            font-weight: bold;

            margin-top: 20px;

        }

        /\* Icon quay lại \*/

        .back-icon {

            position: absolute;

            top: 10px;

            left: 10px;

            width: 40px;

            height: 40px;

            cursor: pointer;

            transition: fill 0.3s ease-in-out;

        }

        .back-icon:hover {

            fill: orange;

        }

    </style>

</head>

<body>

    <div class="container">

        <!-- Icon quay lại -->

        <a href="/products">

            <svg class="back-icon" xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="black">

                <path d="M15.5 19l-7-7 7-7" stroke="black" stroke-width="2" stroke-linecap="round" stroke-linejoin="round"/>

            </svg>

        </a>

        <h1>🛒 Giỏ hàng của bạn</h1>

        <% if (cart.length === 0) { %>

            <p>Giỏ hàng trống!</p>

        <% } else { %>

            <% cart.forEach((product) => { %>

                <div class="cart-item">

                    <img src="<%= product.image%>" alt="<%= product.name %>">

                    <div>

                        <strong><%= product.name %></strong> <br>

                        <span><%= product.price.toLocaleString() %> VND</span>

                    </div>

                    <div class="quantity">

                        <form action="/cart/update/<%= product.\_id %>?action=decrease" method="POST">

                            <button type="submit">-</button>

                        </form>

                        <span><%= product.quantity %></span>

                        <form action="/cart/update/<%= product.\_id %>?action=increase" method="POST">

                            <button type="submit">+</button>

                        </form>

                    </div>

                    <form action="/cart/remove/<%= product.\_id %>" method="POST">

                        <button type="submit">🗑️</button>

                    </form>

                </div>

            <% }) %>

            <div class="total">Tổng tiền: <%= cart.reduce((sum, p) => sum + p.price \* p.quantity, 0).toLocaleString() %> VND</div>

        <% } %>

    </div>

</body>

</html>

ProductDetails.ejs:

<!DOCTYPE html>

<html lang="vi">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title><%= product.name %></title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

    <style>

        body {

            font-family: Arial, sans-serif;

            color: #5a2c20;

            background-color: #fff8f3;

        }

        .container {

            width: 80%;

            margin: auto;

            display: flex;

            align-items: center;

            justify-content: space-between;

            padding-top: 50px;

        }

        .product-image img {

            width: 400px;

            border-radius: 10px;

        }

        .product-info {

            width: 50%;

        }

        .breadcrumb {

            font-size: 14px;

            color: #c06c50;

            text-transform: uppercase;

        }

        h1 {

            font-size: 32px;

            color: #8b3a23;

            margin-bottom: 10px;

        }

        .product-price {

            font-size: 24px;

            font-weight: bold;

        }

        .size-options button {

            padding: 10px 20px;

            margin-right: 10px;

            border: 2px solid #8b3a23;

            background: none;

            color: #8b3a23;

            font-weight: bold;

            cursor: pointer;

            border-radius: 5px;

            transition: 0.3s;

        }

        .size-options button:hover {

            background: #8b3a23;

            color: white;

        }

        .size-options button.active {

            background: #c06c50;

            color: white;

            border-color: #c06c50;

        }

        .quantity {

            display: flex;

            align-items: center;

            margin-top: 20px;

        }

        .quantity button {

            padding: 5px 15px;

            font-size: 18px;

            border: none;

            cursor: pointer;

            background: none;

            color: #8b3a23;

        }

        .quantity input {

            width: 40px;

            text-align: center;

            font-size: 16px;

            border: 1px solid #c06c50;

            border-radius: 5px;

            margin: 0 5px;

        }

        .add-to-cart {

            display: block;

            width: 100%;

            padding: 15px;

            background: none;

            color: #8b3a23;

            font-size: 18px;

            border: 2px solid #8b3a23;

            cursor: pointer;

            margin-top: 20px;

            border-radius: 5px;

            transition: 0.3s;

        }

        .add-to-cart:hover {

            background: #8b3a23;

            color: white;

        }

        .description {

            margin-top: 30px;

            padding: 10px;

            border-top: 2px solid #c06c50;

        }

        .description h3 {

            cursor: pointer;

            display: flex;

            justify-content: space-between;

            align-items: center;

        }

        .description-content {

            display: none;

            padding: 10px 0;

        }

    </style>

</head>

<body>

    <div class="container">

        <div class="product-image">

            <img src="<%= product.image %>" alt="<%= product.name %>">

        </div>

        <div class="product-info">

            <div class="breadcrumb">TRANG CHỦ › <%= product.name.toUpperCase() %></div>

            <h1><%= product.name %></h1>

            <div class="product-price" id="product-price"><%= product.price.toLocaleString() %>₫</div>

            <div class="size-options">

                <button class="size-btn active" data-size="320ML" data-price="<%= product.price %>">320ML</button>

                <button class="size-btn" data-size="450ML" data-price="<%= product.price + 10000 %>">450ML</button>

            </div>

            <div class="quantity">

                <button onclick="changeQuantity(-1)">-</button>

                <input type="text" id="quantity" value="1">

                <button onclick="changeQuantity(1)">+</button>

            </div>

            <form action="/cart/add/<%= product.\_id %>" method="POST">

                <button type="submit" class="add-to-cart" id="add-to-cart">THÊM VÀO GIỎ HÀNG • <%= product.price.toLocaleString() %>₫</button>

            </form>

            <div class="description">

                <h3 onclick="toggleDescription()">MÔ TẢ <i class="fas fa-chevron-down"></i></h3>

                <div class="description-content">

                    <p><%= product.description %></p>

                </div>

            </div>

        </div>

    </div>

    <script>

        function changeQuantity(amount) {

            let quantityInput = document.getElementById("quantity");

            let current = parseInt(quantityInput.value);

            if (current + amount > 0) {

                quantityInput.value = current + amount;

            }

        }

        function toggleDescription() {

            let content = document.querySelector(".description-content");

            content.style.display = content.style.display === "block" ? "none" : "block";

        }

        document.querySelectorAll(".size-options button").forEach(button => {

            button.addEventListener("click", function() {

                document.querySelectorAll(".size-options button").forEach(btn => btn.classList.remove("active"));

                this.classList.add("active");

                let newPrice = parseInt(this.getAttribute("data-price"));

                document.getElementById("product-price").textContent = newPrice.toLocaleString() + "₫";

                document.getElementById("add-to-cart").textContent = "THÊM VÀO GIỎ HÀNG • " + newPrice.toLocaleString() + "₫";

            });

        });

    </script>

</body>

</html>

Product.ejs:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Danh sách sản phẩm</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/all.min.css">

    <style>

        body {

            font-family: Arial, sans-serif;

        }

        .container {

            width: 80%;

            margin: auto;

            text-align: center;

            position: relative;

        }

        h1 {

            color: #333;

        }

        /\* Điều chỉnh vị trí giỏ hàng \*/

        .cart-link {

            position: absolute;

            top: 10px;

            right: 10px;

            font-size: 20px;

            text-decoration: none;

            color: #333;

            display: flex;

            align-items: center;

        }

        .cart-link i {

            margin-right: 5px;

        }

        .cart-link:hover {

            color: #d9534f;

        }

        .product-list {

            display: flex;

            justify-content: center;

            flex-wrap: wrap;

            gap: 20px;

        }

        .product {

            border: 1px solid #ddd;

            padding: 15px;

            border-radius: 10px;

            width: 200px;

            text-align: center;

            position: relative;

            transition: transform 0.2s;

        }

        .product:hover {

            transform: scale(1.05);

        }

        .product img {

            width: 100%;

            height: auto;

            border-radius: 10px;

        }

        .product-title {

            font-size: 18px;

            font-weight: bold;

            margin-top: 10px;

        }

        .product-price {

            color: #d9534f;

            margin: 10px 0;

        }

        /\* Ẩn nút mặc định \*/

        button {

            background-color: #28a745;

            margin-left: 16%;

            color: white;

            padding: 10px;

            border: none;

            cursor: pointer;

            border-radius: 5px;

            display: none; /\* Ẩn mặc định \*/

        }

        /\* Khi hover vào sản phẩm, hiển thị nút \*/

        .product:hover button {

            display: block;

        }

        button:hover {

            background-color: #218838;

        }

    </style>

</head>

<body>

    <div class="container">

        <!-- Nút giỏ hàng với icon -->

        <a href="/cart" class="cart-link">

            <i class="fas fa-shopping-cart"></i>

        </a>

        <h1>Danh sách sản phẩm</h1>

        <div class="product-list">

            <% products.forEach((product) => { %>

                <div class="product">

                    <a href="/product/<%= product.\_id %>">

                        <img src="<%= product.image%>" alt="<%= product.name %>">

                    </a>

                    <div class="product-title"><%= product.name %></div>

                    <div class="product-price"><%= product.price.toLocaleString() %> VND</div>

                    <form action="/cart/add/<%= product.\_id %>" method="POST">

                        <button type="submit">Thêm vào giỏ hàng</button>

                    </form>

                </div>

            <% }) %>

        </div>

    </div>

</body>

</html>

Db.js:

const mongoose = require('mongoose');

const connectDB = async () => {

    try {

        await mongoose.connect('mongodb://127.0.0.1:27017/shop', {

            useNewUrlParser: true,

            useUnifiedTopology: true,

        });

        console.log('✅ Kết nối MongoDB thành công!');

    } catch (error) {

        console.error('❌ Kết nối MongoDB thất bại:', error);

        process.exit(1);

    }

};

module.exports = connectDB;

Routes.js:

const express = require('express');

const Product = require('./models/Product');

const multer = require('multer');

const path = require('path');

const router = express.Router();

// Trang danh sách sản phẩm

router.get('/products', async (req, res) => {

    const products = await Product.find();

    res.render('products', { products });

});

// API tạo sản phẩm

router.post('/api/products', async (req, res) => {

    const { name, description, price, stock } = req.body;

    const newProduct = new Product({ name, description, price, stock });

    await newProduct.save();

    res.send('Thêm sản phẩm thành công!');

});

// API lấy danh sách sản phẩm

router.get('/api/products', async (req, res) => {

    const products = await Product.find();

    res.json(products);

});

// Thêm sản phẩm vào giỏ hàng

router.post('/cart/add/:id', async (req, res) => {

    try {

        const product = await Product.findById(req.params.id);

        if (!product) {

            return res.status(404).send('Sản phẩm không tồn tại!');

        }

        if (!req.session.cart) req.session.cart = [];

        let cart = req.session.cart;

        let existingProduct = cart.find(p => p.\_id.toString() === product.\_id.toString());

        if (existingProduct) {

            existingProduct.quantity += 1;

        } else {

            cart.push({

                \_id: product.\_id.toString(),

                name: product.name,

                price: product.price,

                quantity: 1,

                image: product.image || "/image/default.jpg"

            });

        }

        req.session.cart = cart;

        res.redirect('/cart');

    } catch (error) {

        console.error('Lỗi thêm vào giỏ hàng:', error);

        res.status(500).send('Lỗi server!');

    }

});

// Xem giỏ hàng

router.get('/cart', (req, res) => {

    res.render('cart', { cart: req.session.cart || [] });

});

// Cấu hình lưu ảnh với Multer

const storage = multer.diskStorage({

    destination: './image/', // Lưu ảnh vào thư mục uploads

    filename: (req, file, cb) => {

        cb(null, file.fieldname + '-' + Date.now() + path.extname(file.originalname));

    }

});

const upload = multer({ storage });

// Cập nhật sản phẩm bằng PUT (nếu dùng API từ Postman)

// 🟡 Route cập nhật sản phẩm

router.put('/api/products/:id', upload.single('image'), async (req, res) => {

    try {

        const { name, price, description } = req.body;

        const imagePath = req.file ? `/image/${req.file.filename}` : req.body.currentImage;

        const updatedProduct = await Product.findByIdAndUpdate(

            req.params.id,

            { name, price, description, image: imagePath },

            { new: true }

        );

        if (!updatedProduct) {

            return res.status(404).json({ message: 'Không tìm thấy sản phẩm!' });

        }

        res.status(200).json({ message: 'Sản phẩm đã được cập nhật!', product: updatedProduct });

    } catch (error) {

        console.error('Lỗi cập nhật sản phẩm:', error);

        res.status(500).json({ message: 'Có lỗi xảy ra khi cập nhật sản phẩm!' });

    }

});

// Xóa sản phẩm

router.delete('/products/:id', async (req, res) => {

    await Product.findByIdAndDelete(req.params.id);

    res.send('Sản phẩm đã được xóa!');

});

// Route hiển thị chi tiết sản phẩm

router.get('/product/:id', async (req, res) => {

    try {

        const product = await Product.findById(req.params.id);

        if (!product) {

            return res.status(404).send('Sản phẩm không tồn tại');

        }

        res.render('productDetail', { product });

    } catch (error) {

        res.status(500).send('Lỗi server');

    }

});

router.get("/product/:id", async (req, res) => {

    try {

        const productId = req.params.id;

        const product = await Product.findById(productId); // Lấy sản phẩm từ MongoDB

        if (!product) {

            return res.status(404).send("Sản phẩm không tồn tại!");

        }

        res.render("productDetail", { product }); // Render ra trang chi tiết

    } catch (error) {

        console.error(error);

        res.status(500).send("Lỗi server");

    }

});

// API thêm sản phẩm với ảnh

router.post('/api/products', upload.single('image'), async (req, res) => {

    try {

        const { name, price, description } = req.body;

        const imagePath = req.file ? `/image/${req.file.filename}` : '';

        const newProduct = new Product({ name, price, description, image: imagePath });

        await newProduct.save();

        res.status(201).json({ message: 'Sản phẩm đã được tạo!', product: newProduct });

    } catch (error) {

        console.error('Lỗi tạo sản phẩm:', error);

        res.status(500).json({ message: 'Có lỗi xảy ra khi tạo sản phẩm!' });

    }

});

// Route cập nhật sản phẩm

router.post('/cart/update/:id', (req, res) => {

    try {

        const productId = req.params.id;

        const action = req.query.action;

        if (!req.session.cart) req.session.cart = [];

        let cart = req.session.cart;

        let product = cart.find(p => p.\_id.toString() === productId);

        if (product) {

            if (action === "increase") {

                product.quantity += 1;

            } else if (action === "decrease" && product.quantity > 1) {

                product.quantity -= 1;

            }

        }

        req.session.cart = cart;

        res.redirect('/cart');

    } catch (error) {

        console.error('Lỗi cập nhật giỏ hàng:', error);

        res.status(500).send('Lỗi server!');

    }

});

router.post('/cart/remove/:id', (req, res) => {

    try {

        const productId = req.params.id;

        if (!req.session.cart) req.session.cart = [];

        req.session.cart = req.session.cart.filter(p => p.\_id.toString() !== productId);

        res.redirect('/cart');

    } catch (error) {

        console.error('Lỗi xóa sản phẩm khỏi giỏ hàng:', error);

        res.status(500).send('Lỗi server!');

    }

});

module.exports = router;

server.js:

const express = require('express');

const bodyParser = require('body-parser');

const session = require('express-session');

const path = require('path');

const connectDB = require('./db');

const productRoutes = require('./routes');

const app = express();

const PORT = 4000;

// Kết nối DB

connectDB();

// Cấu hình middleware

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

app.use(session({ secret: 'secret', resave: false, saveUninitialized: true }));

app.use('/image', express.static(path.join(\_\_dirname, 'image')));

// Cấu hình thư mục tĩnh

app.use(express.static('public'));

// Cấu hình view engine

app.set('view engine', 'ejs');

app.set('views', './views');

// Sử dụng routes

app.use('/', productRoutes);

// Lắng nghe cổng

app.listen(PORT, () => {

    console.log(`🚀 Server chạy tại: http://localhost:${PORT}`);

});

Upload.js:

const multer = require('multer');

const path = require('path');

const storage = multer.diskStorage({

    destination: './image/', // Thư mục lưu ảnh

    filename: (req, file, cb) => {

        cb(null, file.fieldname + '-' + Date.now() + path.extname(file.originalname));

    }

});

const upload = multer({ storage });

module.exports = upload;

GITHUB: https://github.com/ThinhLe287/WebNC