

BÁO CÁO BÀI KIỂM TRA ĐẦU VÀO WEBDEV STUDIOS GEN 8

GROUP 3 - WDS SHOP STORE

1. THÔNG TIN NHÓM

Tên Project: WDS Shop Store Fullstack Repository: <https://github.com/xirothedev/wds-shop-store-fullstack>

STT	Thành viên	Role	Ghi chú
1	Thành Trung	Leader / Fullstack	Setup base, Authentication, Admin, Core Backend
2	Phan Kiệt	Member / Fullstack	Search, Product Showcase
3	Minh Phương	Member / Fullstack	Cart Module
4	Trần Tài	Member / Fullstack	Order & Purchase History

2. TÀI KHOẢN KIỂM THỬ (TEST CREDENTIALS)

Hệ thống đã có sẵn tài khoản Admin (được tạo khi seeding). Đối với tài khoản User, vui lòng sử dụng chức năng Đăng ký (Register) để tạo mới.

Tài khoản Admin

- Email: admin@wds.org
- Password: admin123@

Tài khoản User

- Vui lòng truy cập trang Đăng ký để tạo tài khoản mới.
- Chức năng đăng ký/đăng nhập đã hoạt động hoàn thiện với JWT Authentication và mã hóa Argon2.

3. BÁO CÁO PHÂN CHIA CÔNG VIỆC & MỨC ĐỘ THAM GIA

Mức độ tham gia được đánh giá dựa trên khối lượng commit, độ phức tạp của tính năng và trách nhiệm trong dự án (Git History Analysis).

Thành viên	Nhiệm vụ chi tiết	Mức độ tham gia	Tự đánh giá
Thành Trung	- Khởi tạo dự án (Turborepo, Next.js, NestJS). - Thiết kế Database Schema & Seeding (FakerJS).	100% (High)	Leader, chịu trách nhiệm kiến trúc và các module lõi khó nhất.

	<ul style="list-style-type: none"> - Module Authentication (JWT, Guards, Login/Register UI/API). - Admin Panel (Layout, Product Management Base). - Tích hợp Cloudflare R2 để lưu trữ ảnh. - Landing Page UI & Design System foundations. 		
Phan Kiệt	<ul style="list-style-type: none"> - Cart Module (Trọn gói FE + BE): <ul style="list-style-type: none"> - Xây dựng Cart API (CRUD Cart, Validate stock). - Cart UI (Responsive Grid, logic tăng giảm số lượng). - Authentication cho Cart (chỉ user đã login mới có cart). - Xử lý các logic phức tạp về đồng bộ state giỏ hàng. 	95% (High)	Hoàn thành xuất sắc module Giỏ hàng với UI/UX tốt.
Minh Phương	<ul style="list-style-type: none"> - Search & Product Display: <ul style="list-style-type: none"> - API Search & Filter (Backend). - Search Bar UI & Suggestions (Frontend). - Fetch & Render dữ liệu Product Detail Page. - Integration API cho danh sách sản phẩm trang chủ. 	90% (High)	Đóng góp quan trọng vào luồng tìm kiếm và hiển thị sản phẩm chính.
Trần Tài	<ul style="list-style-type: none"> - Order Module: <ul style="list-style-type: none"> - Xây dựng API Order (Tạo đơn hàng từ giỏ hàng). - UI Trang "ĐÃ MUA" (Purchase History). - Hỗ trợ hoàn thiện luồng Admin Product CRUD. - Fix bugs và clean code một số phần. 	85% (Good)	Hoàn thành tốt chức năng cuối cùng của luồng mua hàng (Order).

4. MÔ TẢ CHỨC NĂNG ĐÃ THỰC HIỆN

Dựa trên yêu cầu của đề bài, nhóm đã hoàn thành các hạng mục sau:

☐ 1. Trang chủ (Home)

- Hiển thị danh sách sản phẩm dạng thumbnail.
- Search bar tìm kiếm theo tên (có suggestion).
- Pagination / Infinite Scroll (Lazy loading).
- Điều hướng sang trang chi tiết.

☐ 2. Trang chi tiết sản phẩm

- Hiển thị ảnh, tên, giá, mô tả.
- Chọn size/số lượng.
- Nút "Thêm vào giỏ hàng" (tích hợp check authen).

□ 3. Trang giỏ hàng

- Hiển thị danh sách item trong giỏ.
- Tăng/giảm số lượng, xóa item.
- Tự động tính tổng tiền.
- Persist giỏ hàng vào Database (User session).

□ 4. Trang ĐÃ MUA

- Hiển thị danh sách đơn hàng đã đặt.
- Xem lại thông tin sản phẩm đã mua.

□ 5. Trang thêm sản phẩm (Admin)

- Bảo vệ bằng Role Admin (User thường vào bị chặn 401/403).
- Form thêm sản phẩm đầy đủ thông tin.
- Upload ảnh thực tế lên Cloudflare R2 (Tính năng nâng cao).

□ 6. Đăng ký - Đăng nhập & Authentication

- Đăng ký User mới.
- Đăng nhập (Admin & User).
- Bảo mật Password (Argon2).
- Cơ chế JWT Access Token + Refresh Token (HttpOnly Cookie).

5. TỔNG KẾT QUÁ TRÌNH THỰC HIỆN

□ Ưu điểm

1. **Kiến trúc dự án chuẩn:** Sử dụng Monorepo (Turborepo) giúp quản lý code FE và BE chung một nơi dễ dàng. Backend theo mô hình 3-layer (Controller - Service - Repository/Prisma) chuẩn chỉnh, dễ bảo trì.
2. **Công nghệ hiện đại:** Next.js App Router, NestJS, TailwindCSS, Prisma ORM, Tanstack Query. Đây là stack mạnh mẽ và phổ biến hiện nay.
3. **Quy trình làm việc rõ ràng:** Nhóm đã lên kế hoạch chi tiết (trong folder `plans/`) trước khi code, giúp giảm thiểu việc đụng độ logic.
4. **Tính năng nâng cao:** Đã tích hợp được Cloudflare R2 để upload ảnh thật thay vì chỉ lưu link text, và seeding dữ liệu giả lập rất "thật" bằng FakerJS.

□ Vấn đề gặp phải & Giải quyết

1. **Merge Conflicts:** Do làm việc song song nên không tránh khỏi conflict file (đặc biệt là Prisma Schema và các file config chung).
 - *Giải quyết:* Thống nhất format code (ESLint/Prettier) ngay từ đầu. Khi merge code phải có ít nhất 1 người review hoặc tự resolve cẩn thận trên local trước khi push.
2. **Đồng bộ Authentication:** Việc xử lý JWT ở client (Next.js) và server (NestJS) gặp chút khó khăn ban đầu về cookie.
 - *Giải quyết:* Sử dụng thư viện `cookie-parser` ở BE và config `credentials: true` trong Axios instance.

□ **Tự đánh giá chung**

Nhóm đã hoàn thành 100% các yêu cầu cơ bản và một số yêu cầu nâng cao (Upload ảnh, Seeding, Search suggestion). Dự án hoạt động ổn định, clean code và tuân thủ các nguyên tắc thiết kế phần mềm tốt.

6. HƯỚNG DẪN CHẠY DỰ ÁN (TÓM TẮT)

Để chạy dự án local, thầy/cô vui lòng thực hiện:

1. Cài đặt dependencies:

```
pnpm install
```

2. Cấu hình môi trường (`.env`): (Đã setup sẵn file env mẫu hoặc sử dụng file env kèm theo bài nộp).

3. Chạy Database & Seeding:

```
# Push schema  
pnpm turbo db:push --filter=api  
# Seed data (Tạo admin & products)  
pnpm turbo db:seed --filter=api
```

4. Chạy ứng dụng:

```
pnpm dev
```

- Web: <http://localhost:3000>
 - API: <http://localhost:3005>
-