Link github lab\_04: [CNM/Tuan04\_Lab04 at main · urglydog/CNM](https://github.com/urglydog/CNM/tree/main/Tuan04_Lab04)

**YÊU CẦU BỔ SUNG Lab3 (ADVANCED EXTENSION – TABLES & FEATURES)**

**Đối tượng áp dụng**:  
– Sinh viên khá, giỏi  
– Nhóm muốn cộng điểm / thay thế một phần đánh giá cuối kỳ  
– Không bắt buộc toàn bộ lớp

**I. MỞ RỘNG THIẾT KẾ CƠ SỞ DỮ LIỆU (DYNAMODB)**

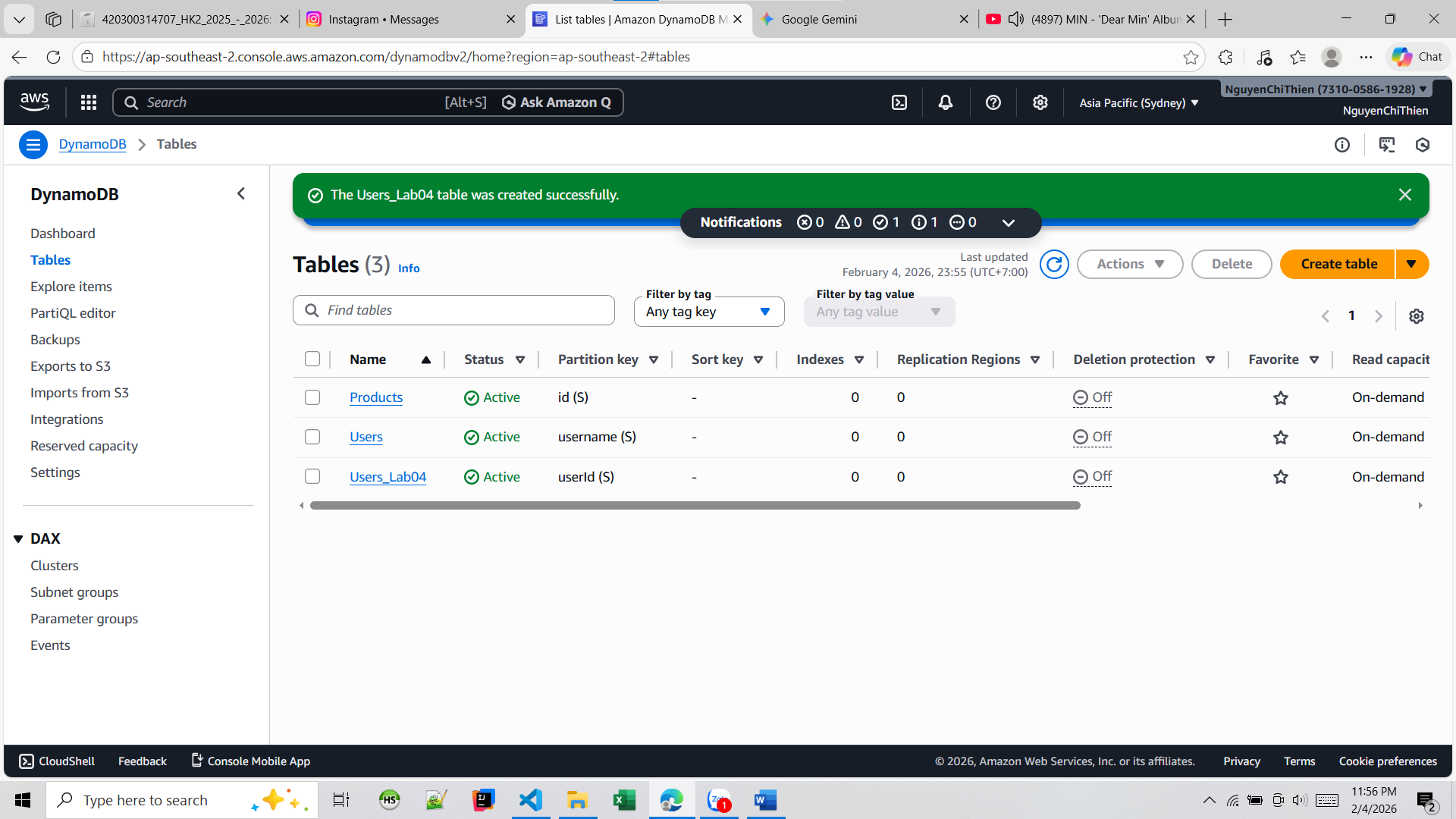
**1️⃣ Bổ sung bảng Users**

**Table: Users**

| **Thuộc tính** | **Kiểu** | **Mô tả** |
| --- | --- | --- |
| userId (PK) | String | UUID |
| username | String | Tên đăng nhập |
| password | String | Mật khẩu (hash) |
| role | String | admin / staff |
| createdAt | String | Ngày tạo |

👉 Yêu cầu:

* Chỉ **admin** được quản lý sản phẩm
* Giải thích vì sao **DynamoDB không join như SQL**

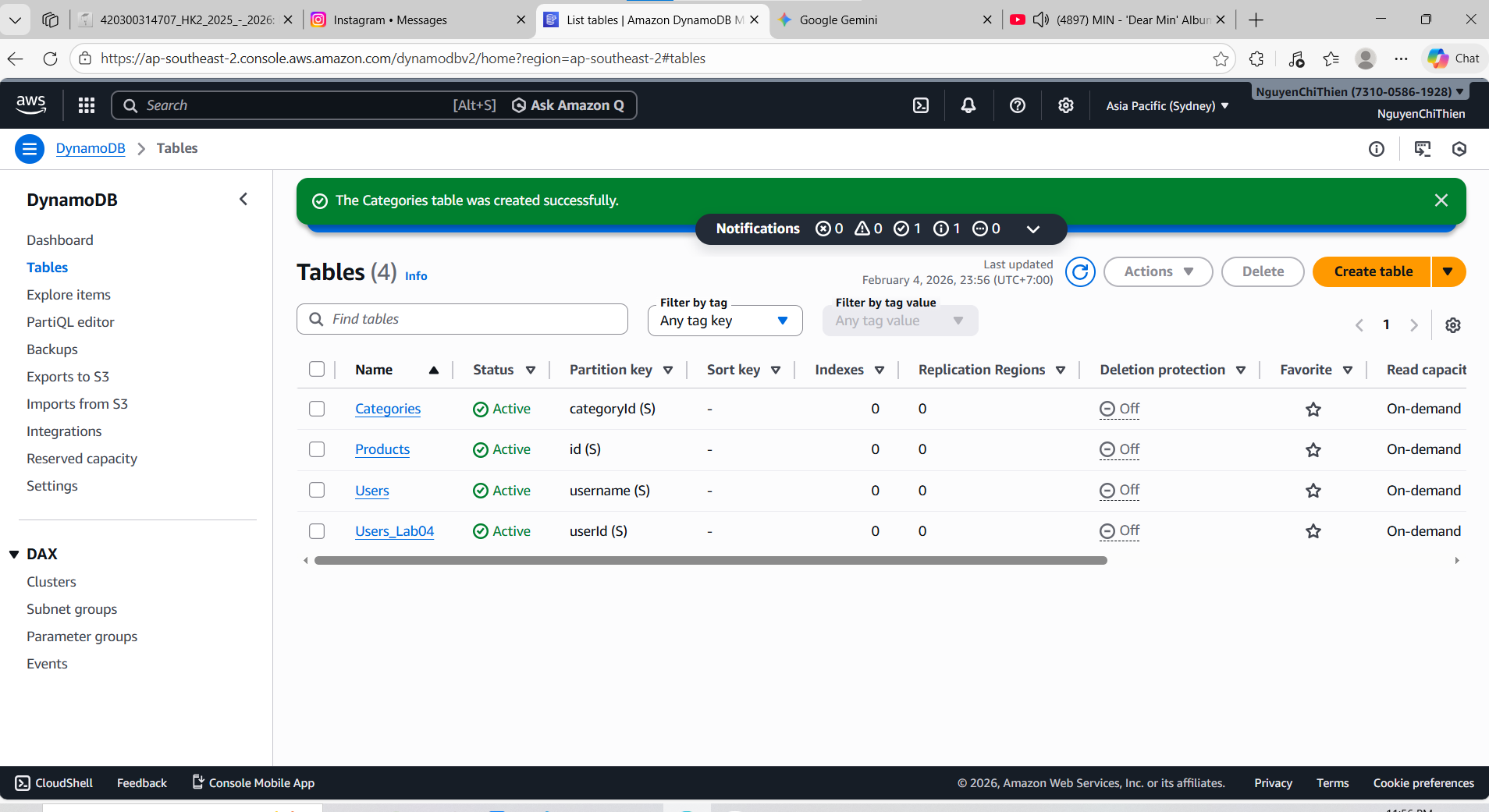


**2️⃣ Bổ sung bảng Categories**

**Table: Categories**

| **Thuộc tính** | **Kiểu** | **Mô tả** |
| --- | --- | --- |
| categoryId (PK) | String | UUID |
| name | String | Tên danh mục |
| description | String | Mô tả |

👉 Sản phẩm **thuộc 1 category**  
👉 Khi xoá category → không xoá sản phẩm (business rule)



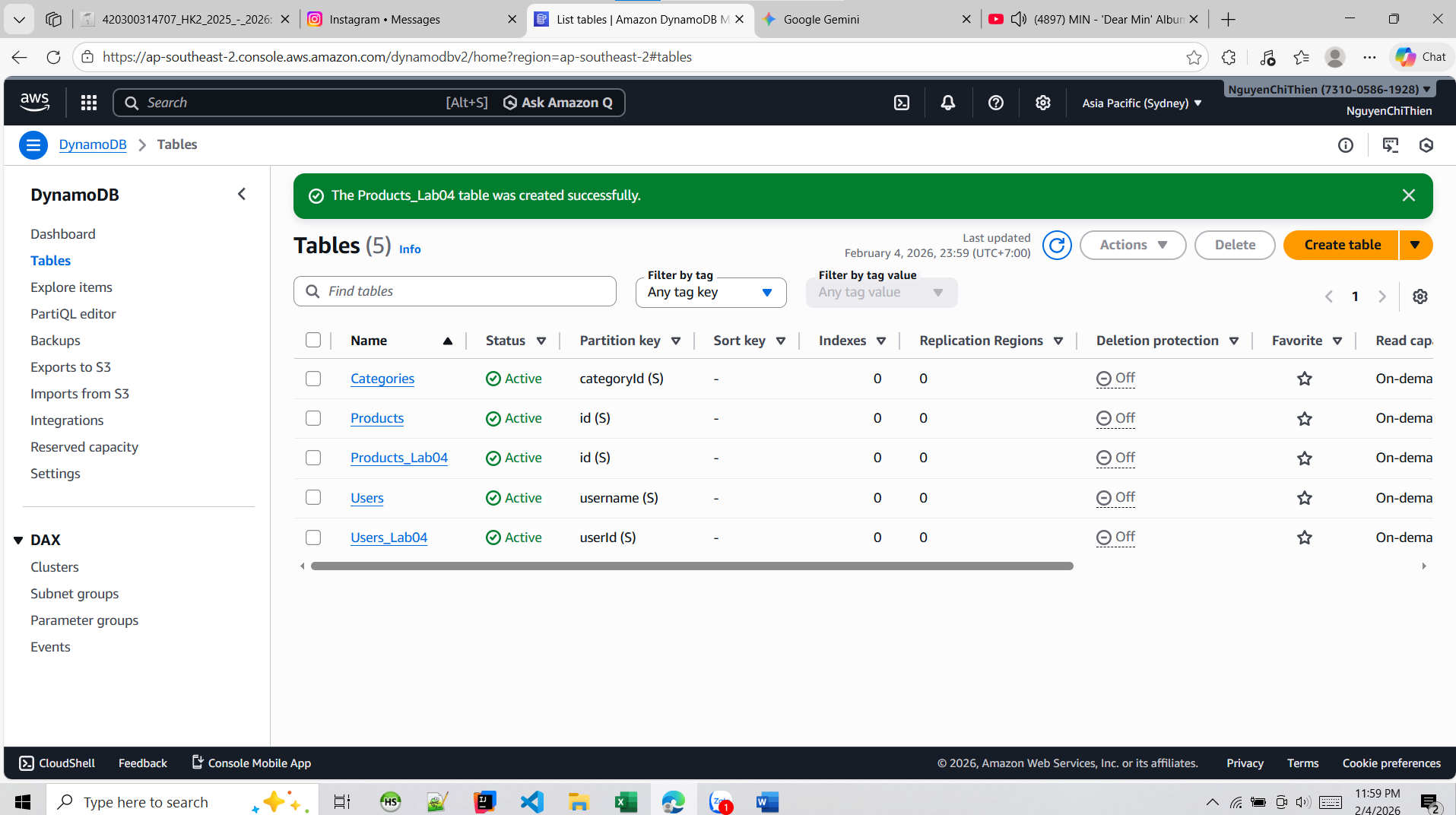
**3️⃣ Cập nhật bảng Products (nâng cao)**

**Table: Products (mở rộng)**

| **Thuộc tính** | **Kiểu** |
| --- | --- |
| id (PK) | String |
| name | String |
| price | Number |
| quantity | Number |
| categoryId | String |
| url\_image | String |
| isDeleted | Boolean |
| createdAt | String |

👉 Yêu cầu:

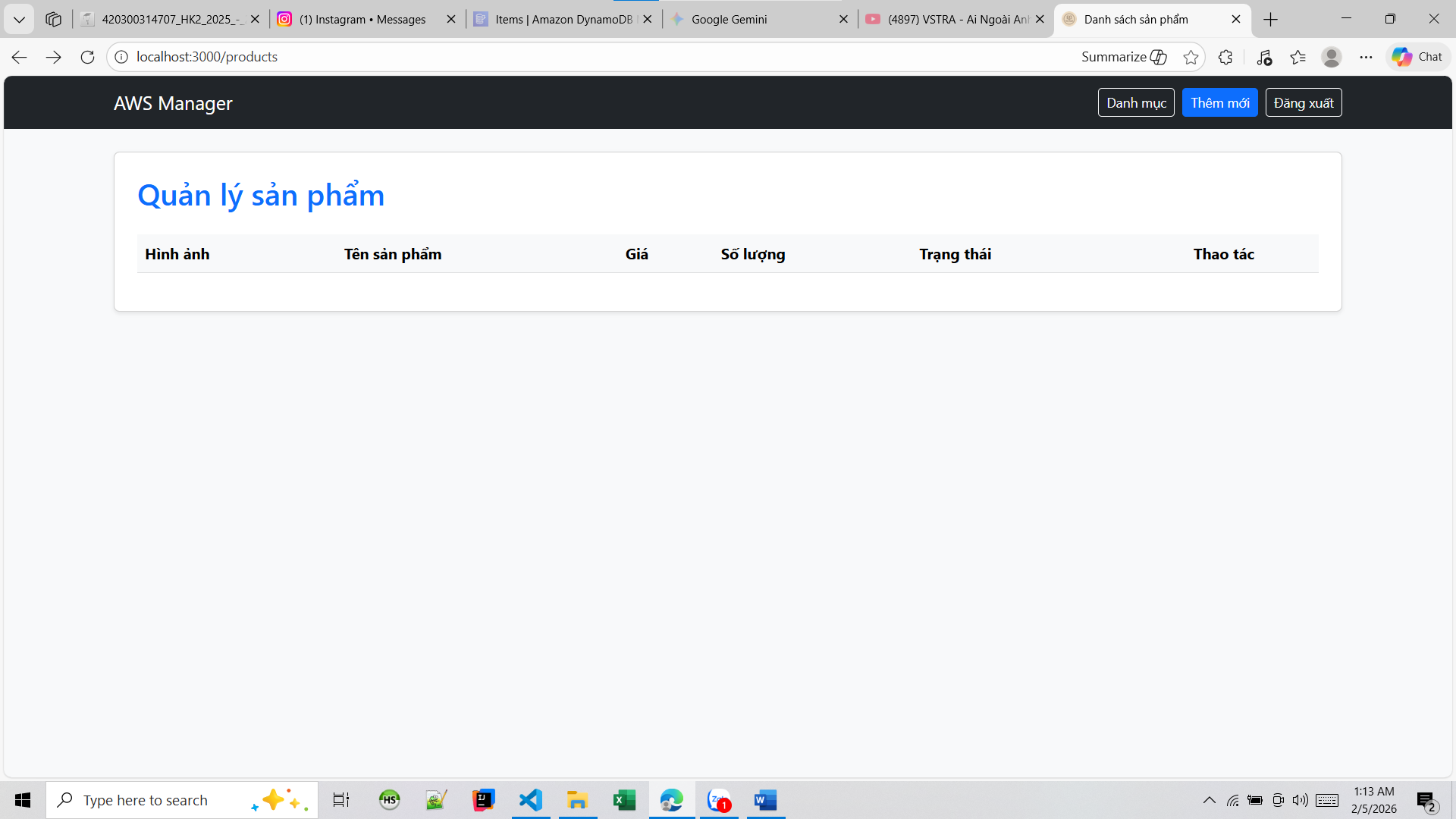
* Áp dụng **Soft Delete**
* Không hiển thị sản phẩm isDeleted = true



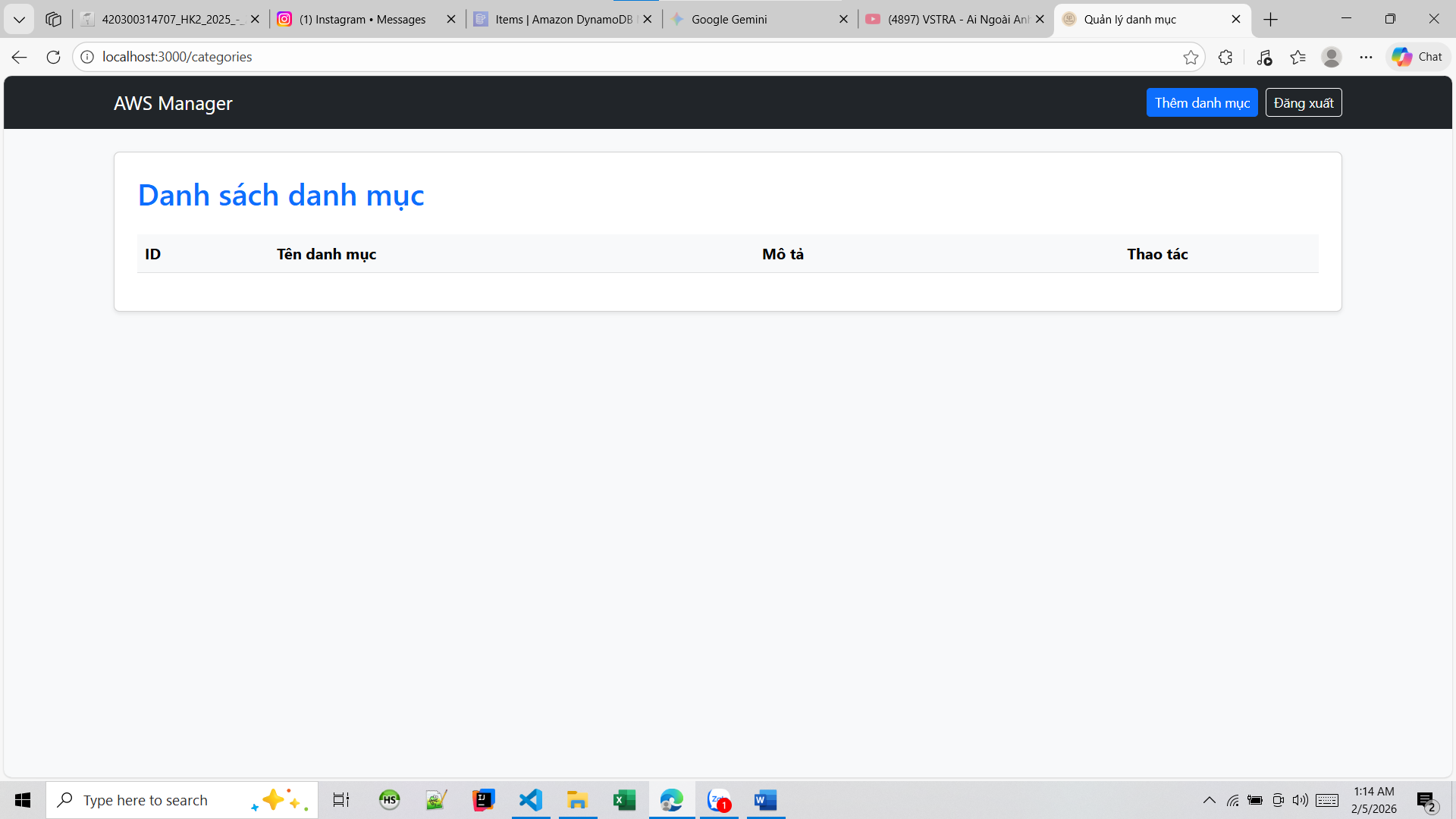
**II. MỞ RỘNG CHỨC NĂNG HỆ THỐNG**

**4️⃣ Chức năng đăng nhập & phân quyền**

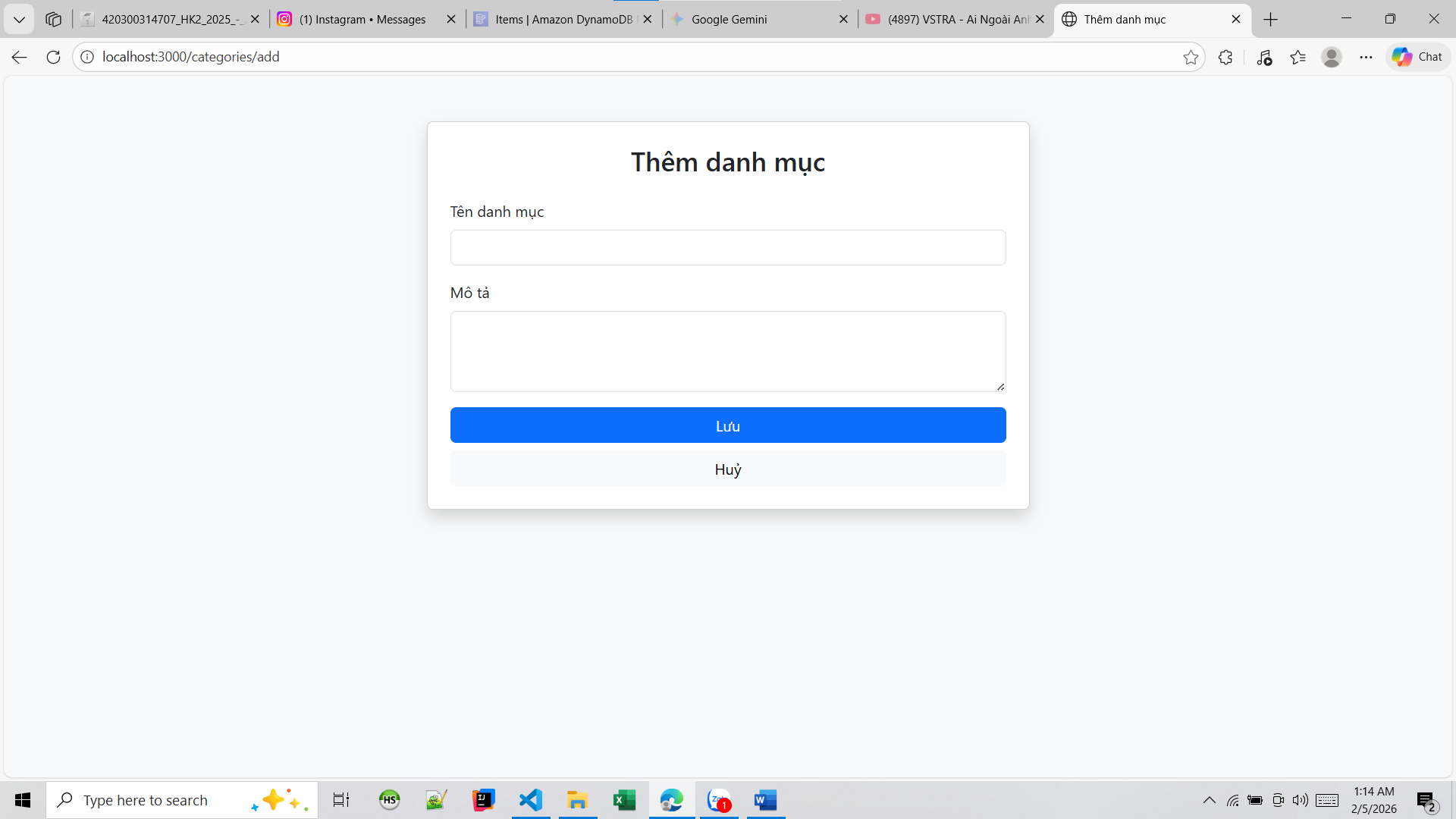
* Trang đăng nhập (/login)
* Session-based authentication
* Phân quyền:
  + admin: CRUD sản phẩm, category



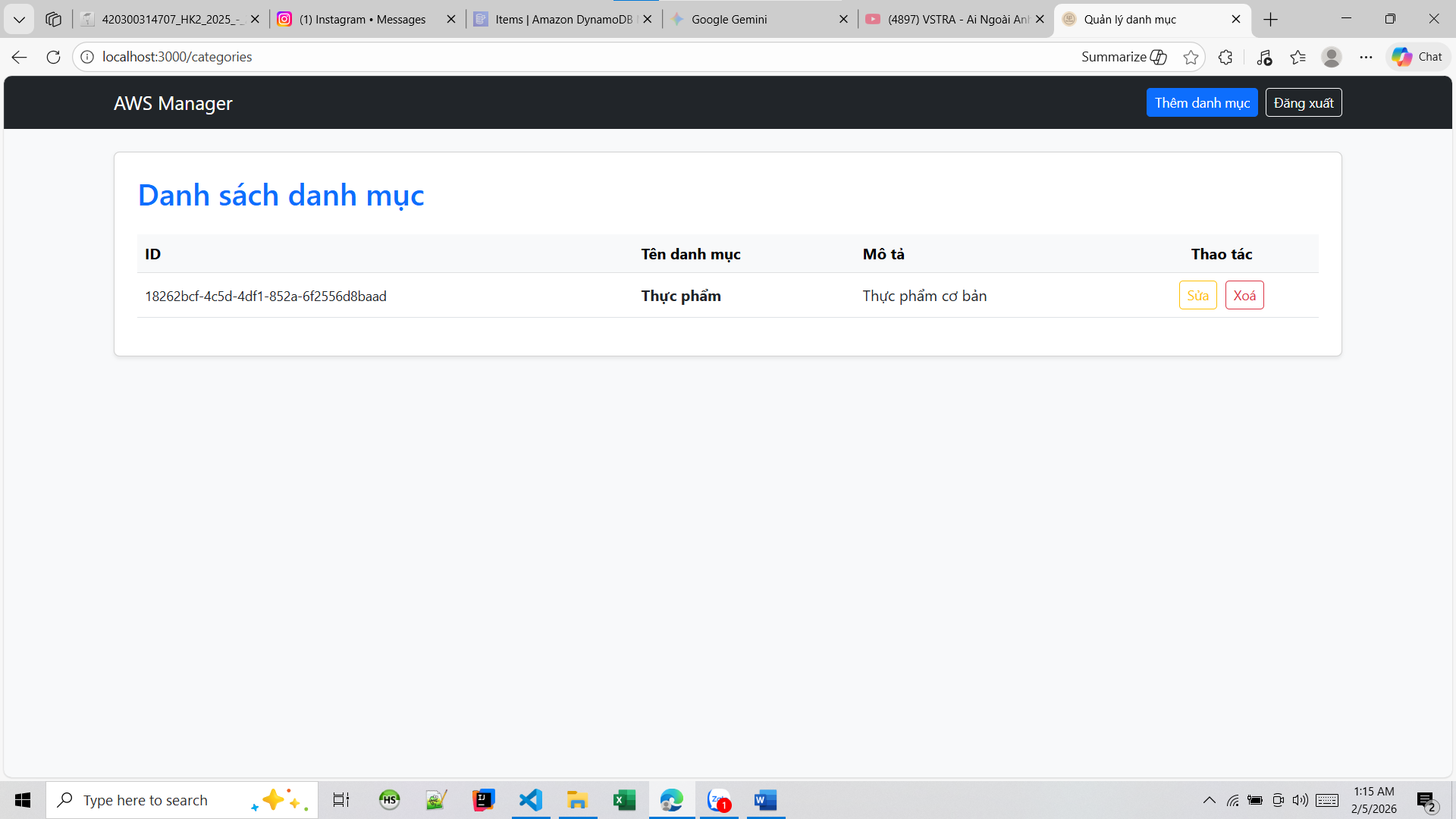
* + - admin có phần thêm mới và danh mục

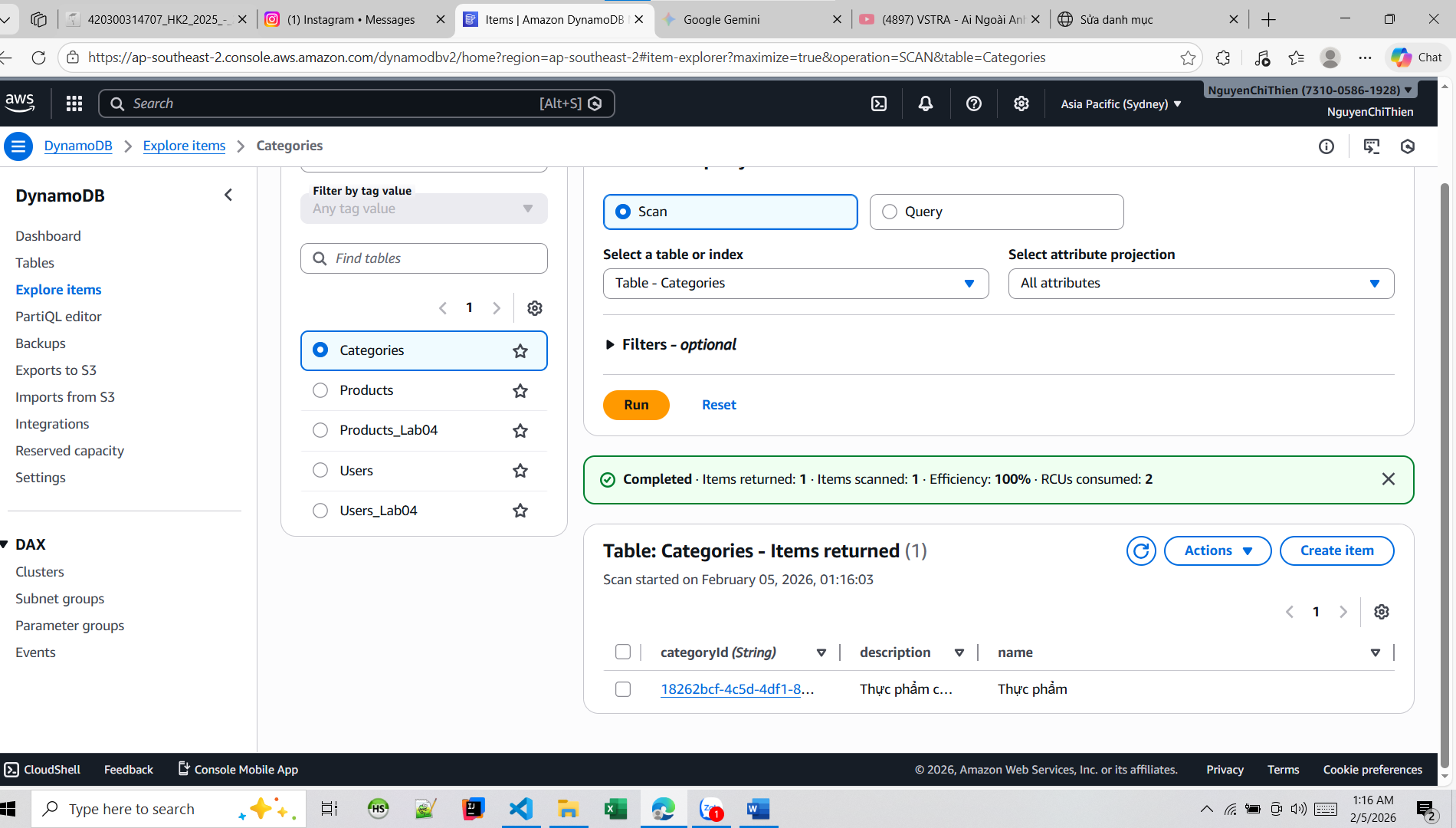


* + - trong danh mục có thêm danh mục

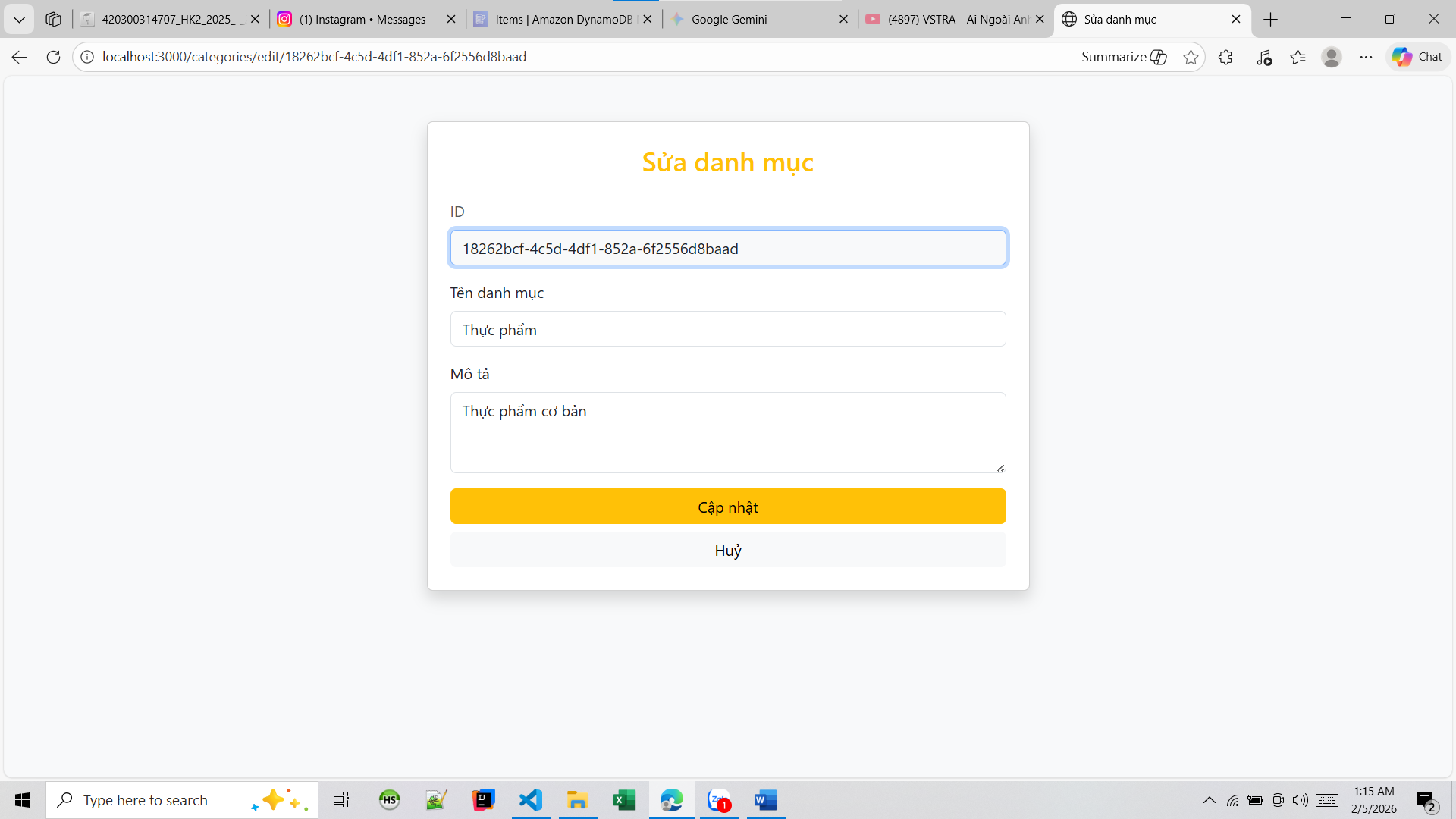


* + - trang thêm danh mục

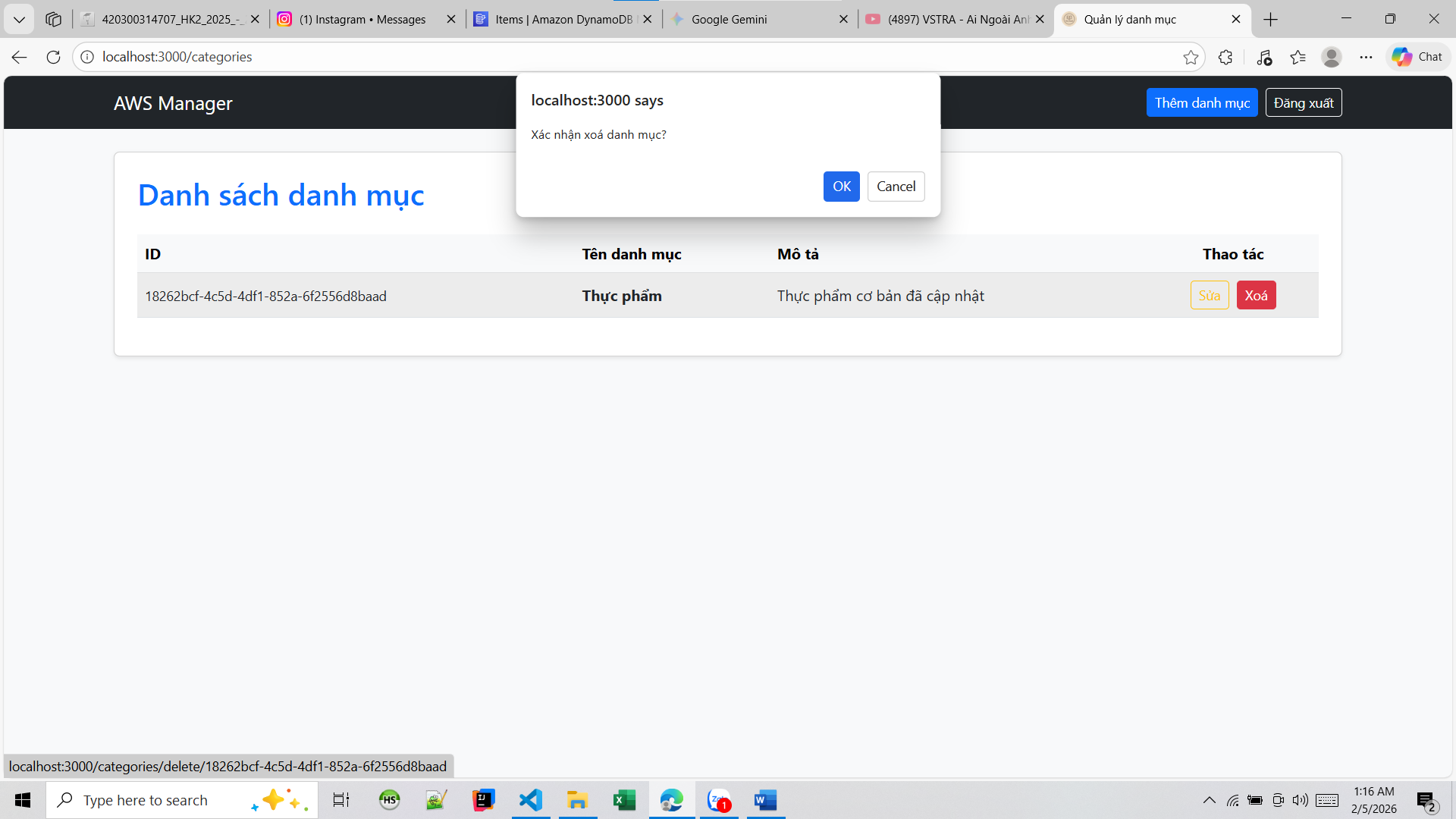




* + - thêm thành công

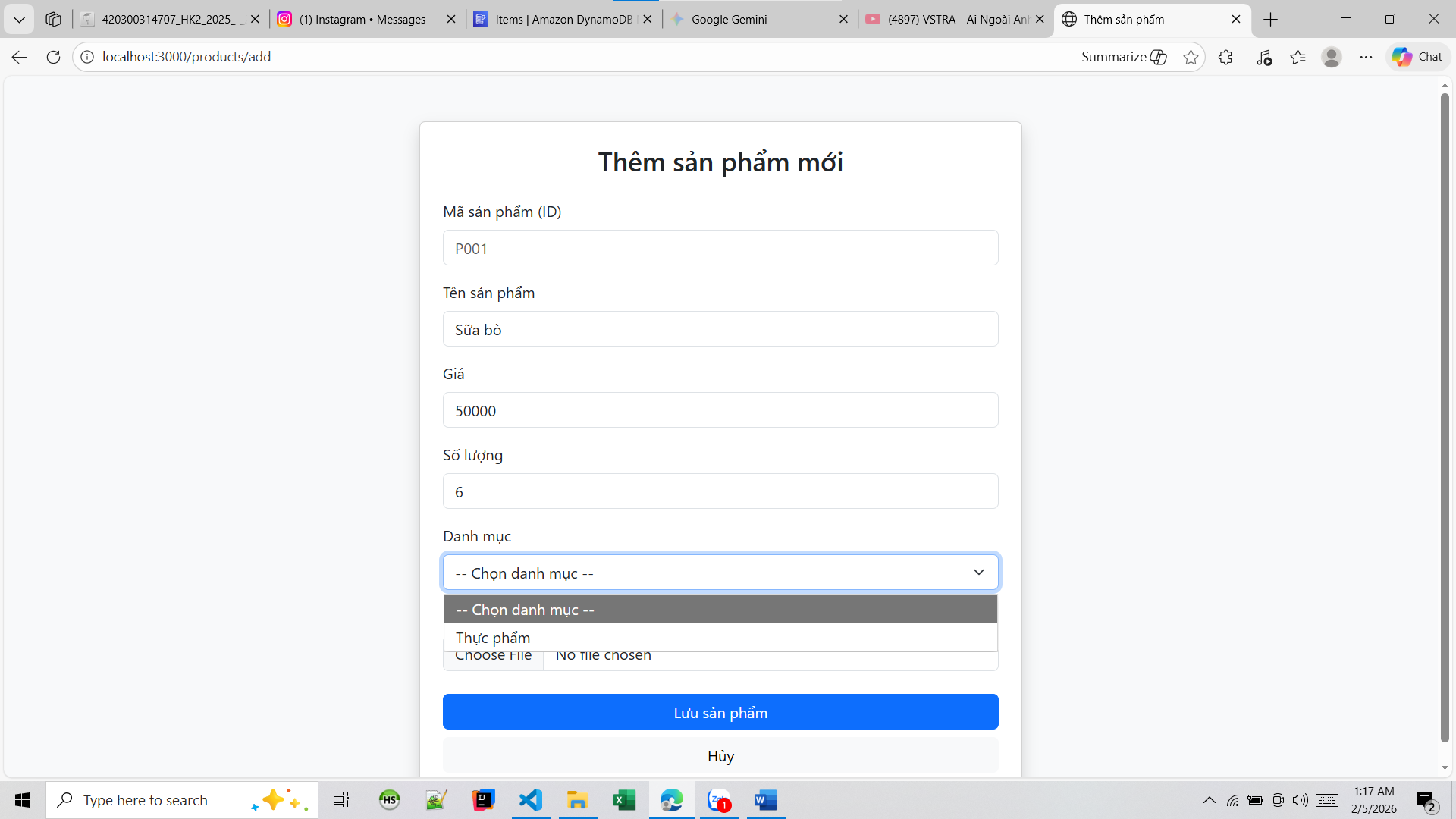


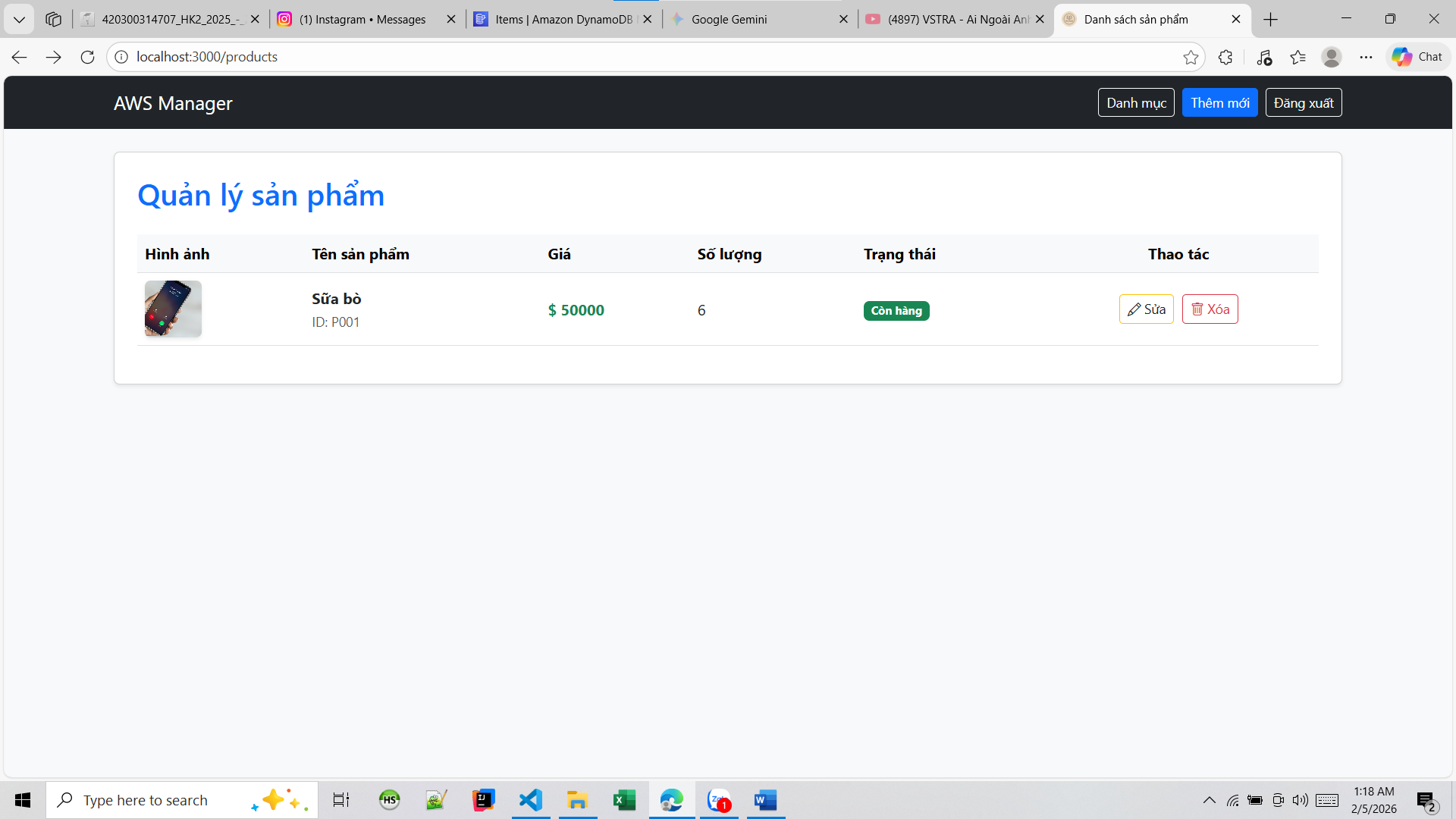
* + - sửa danh mục



Xóa danh mục

Thao tác với sản phẩm (có thêm chọn danh mục)

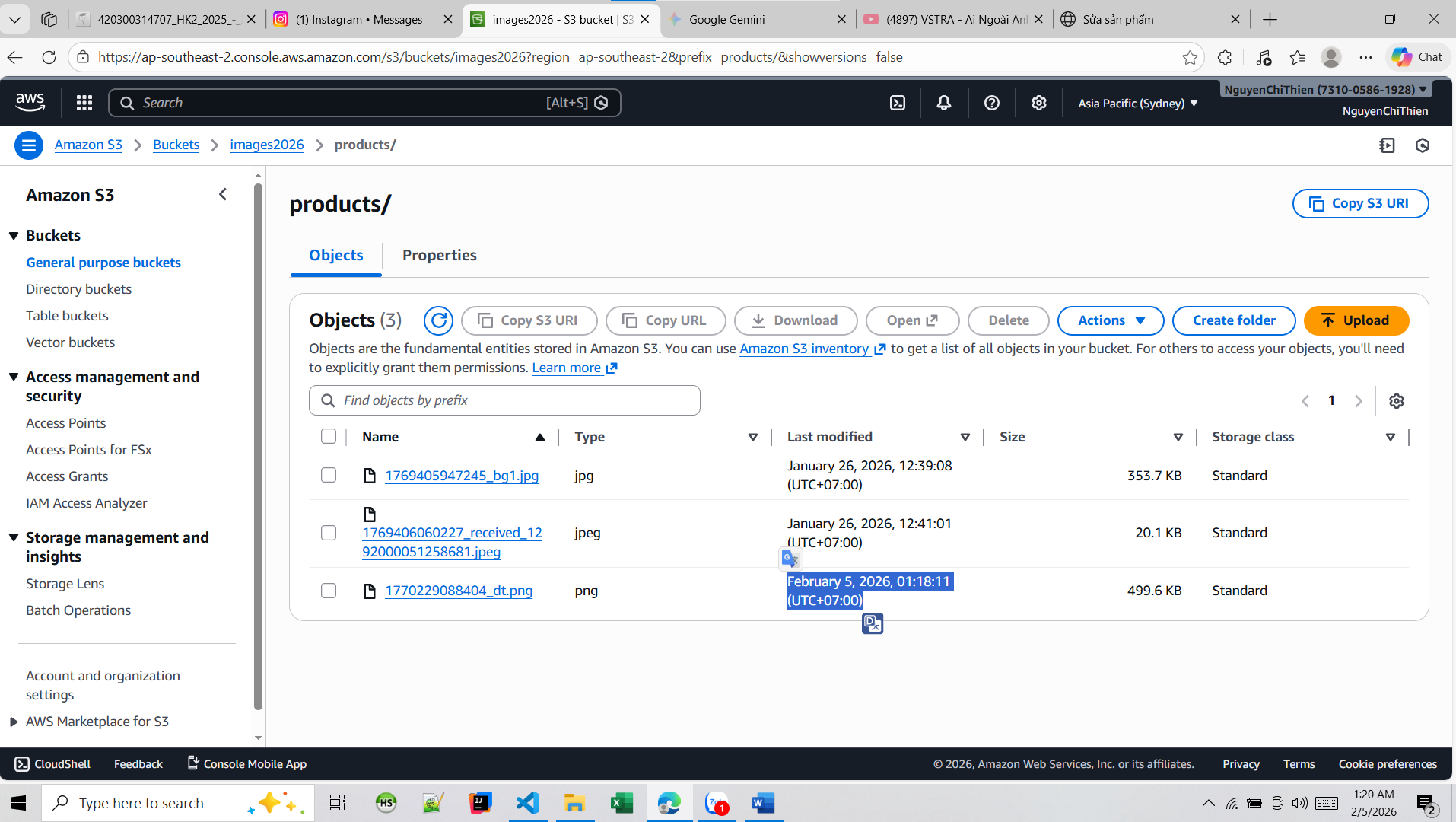




* + - thêm thành công sản phẩm



* + - sửa sản phẩm cũng có danh mục



* + - ảnh vẫn được lưu trong s3
  + staff: chỉ xem danh sách



* + - chỉ xem sản phẩm



* + - chỉ xem danh mục

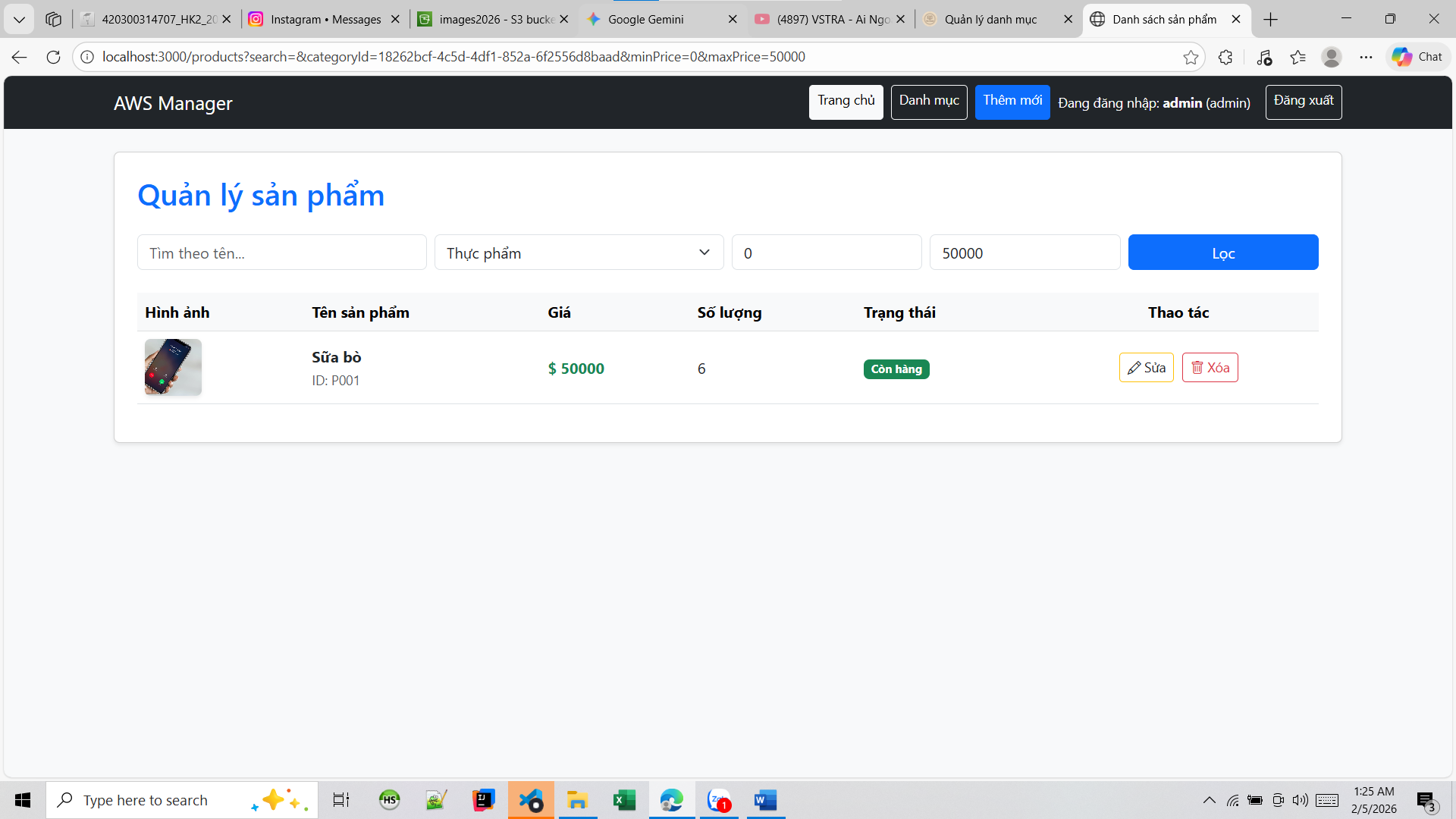
👉 Không yêu cầu JWT (bonus nếu có)

**5️⃣ Quản lý danh mục (Category CRUD)**

* Thêm / sửa / xoá danh mục
* Khi thêm sản phẩm → **chọn category**
* Trang:
  + Danh sách category
  + Form thêm / sửa category

**6️⃣ Tìm kiếm & lọc nâng cao sản phẩm**

* Lọc theo:
  + category
  + khoảng giá
* Tìm theo tên (contains)
* Phân trang danh sách sản phẩm

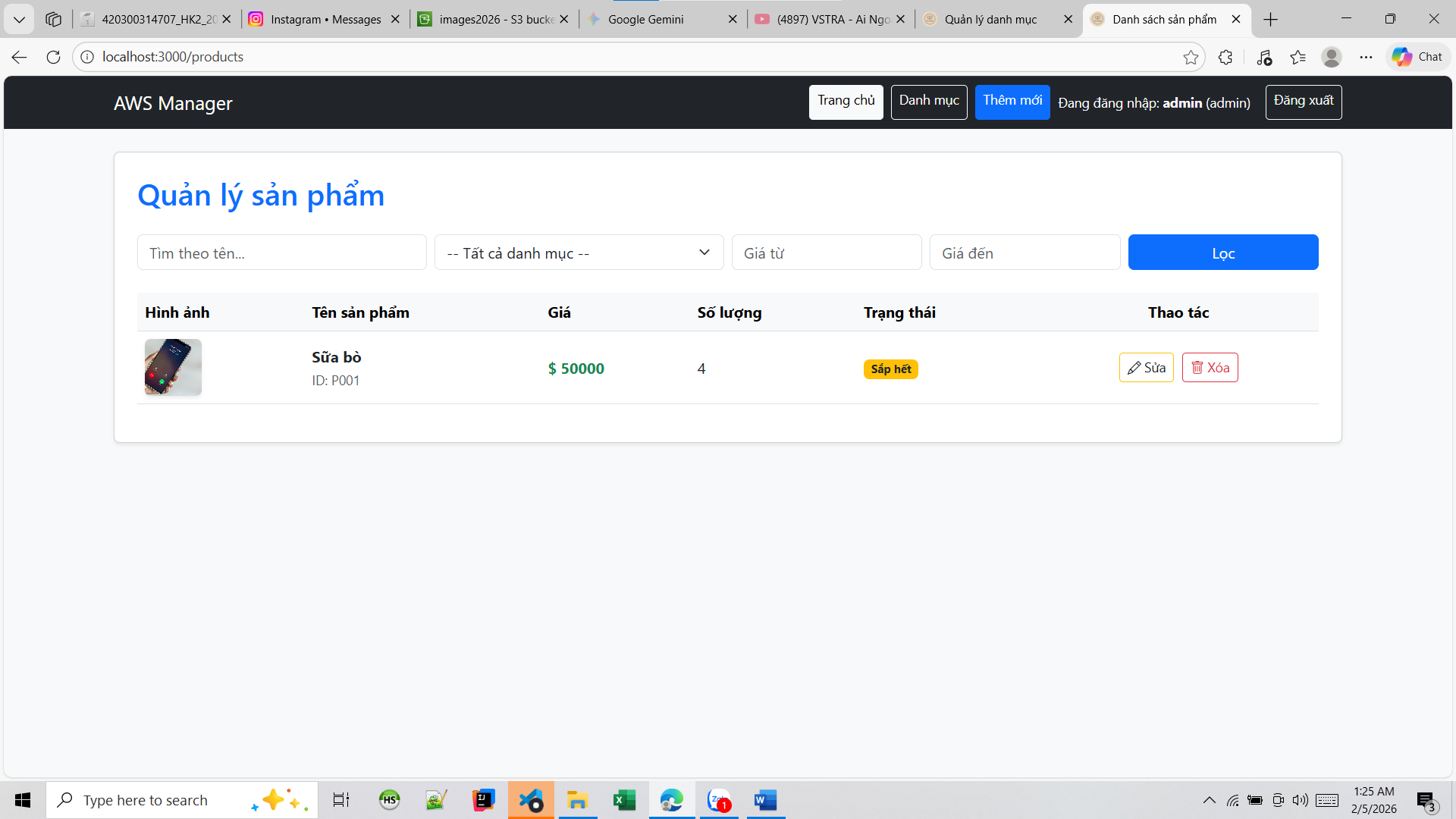
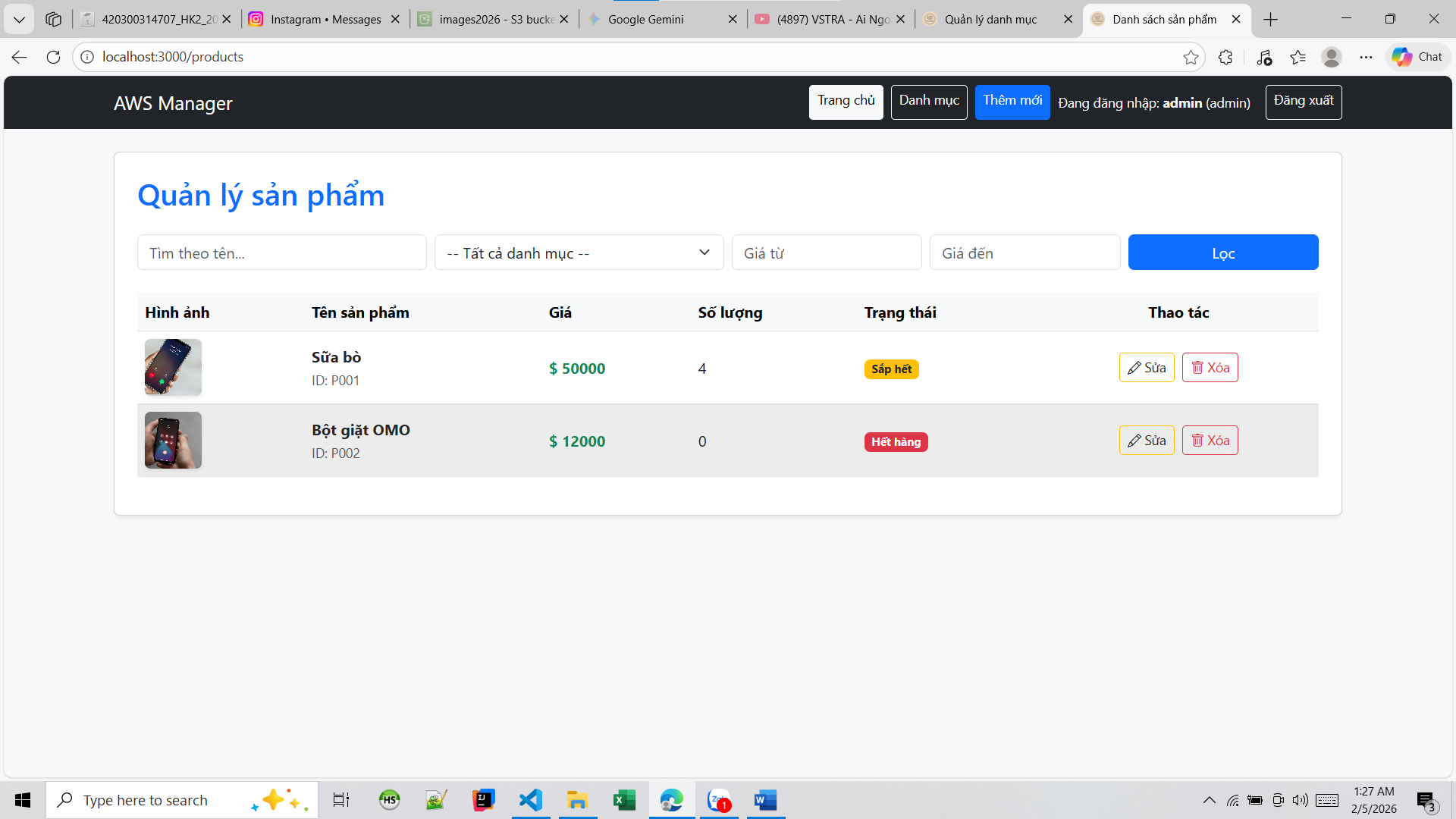


👉 Yêu cầu SV:

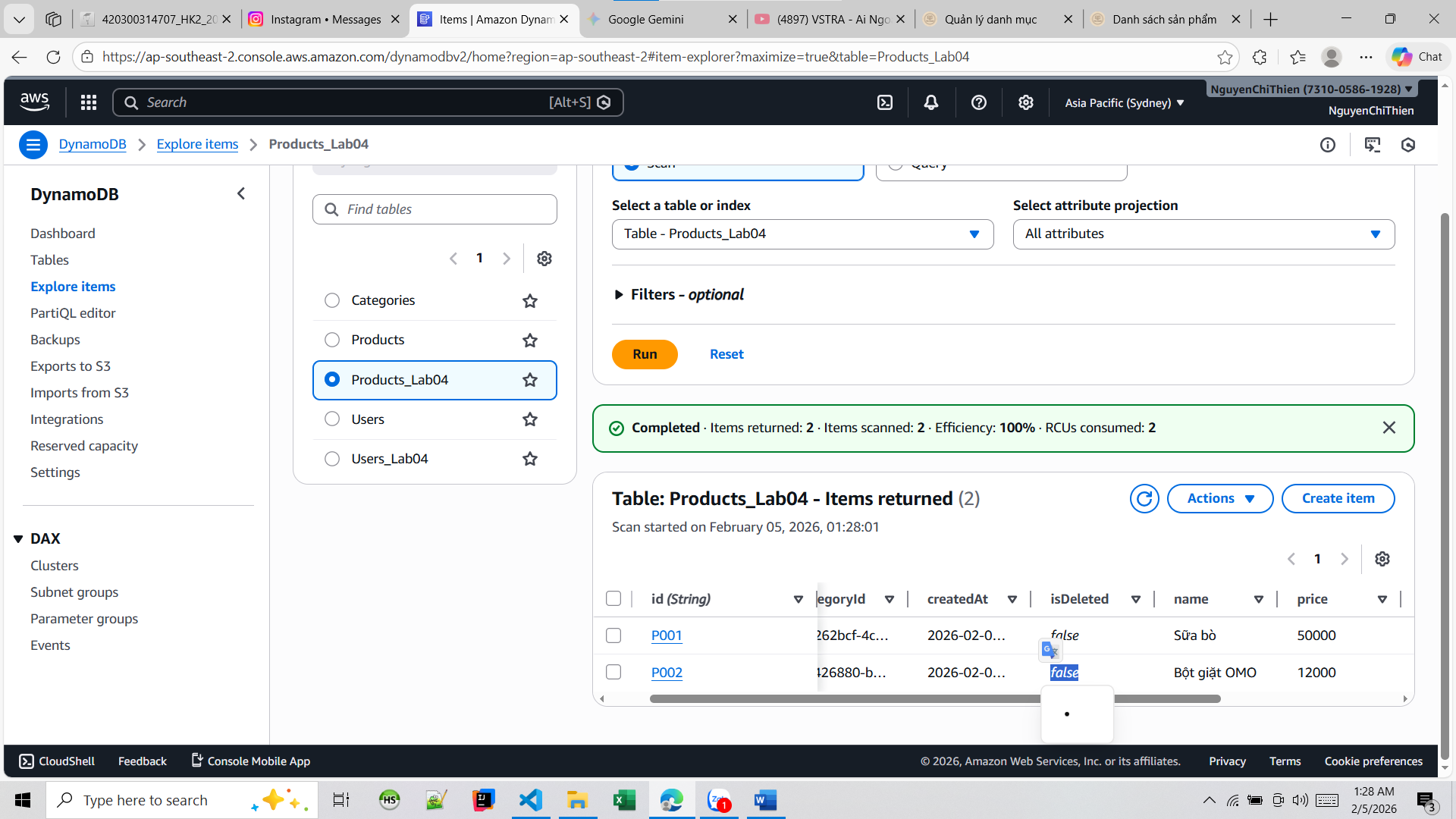
* Phân biệt Query và Scan
* Giải thích **vì sao Scan tốn chi phí**

**7️⃣ Quản lý tồn kho (Inventory Logic)**

* Hiển thị trạng thái:
  + Còn hàng
  + Sắp hết (quantity < 5)
  + Hết hàng
* Cảnh báo trực quan trên giao diện

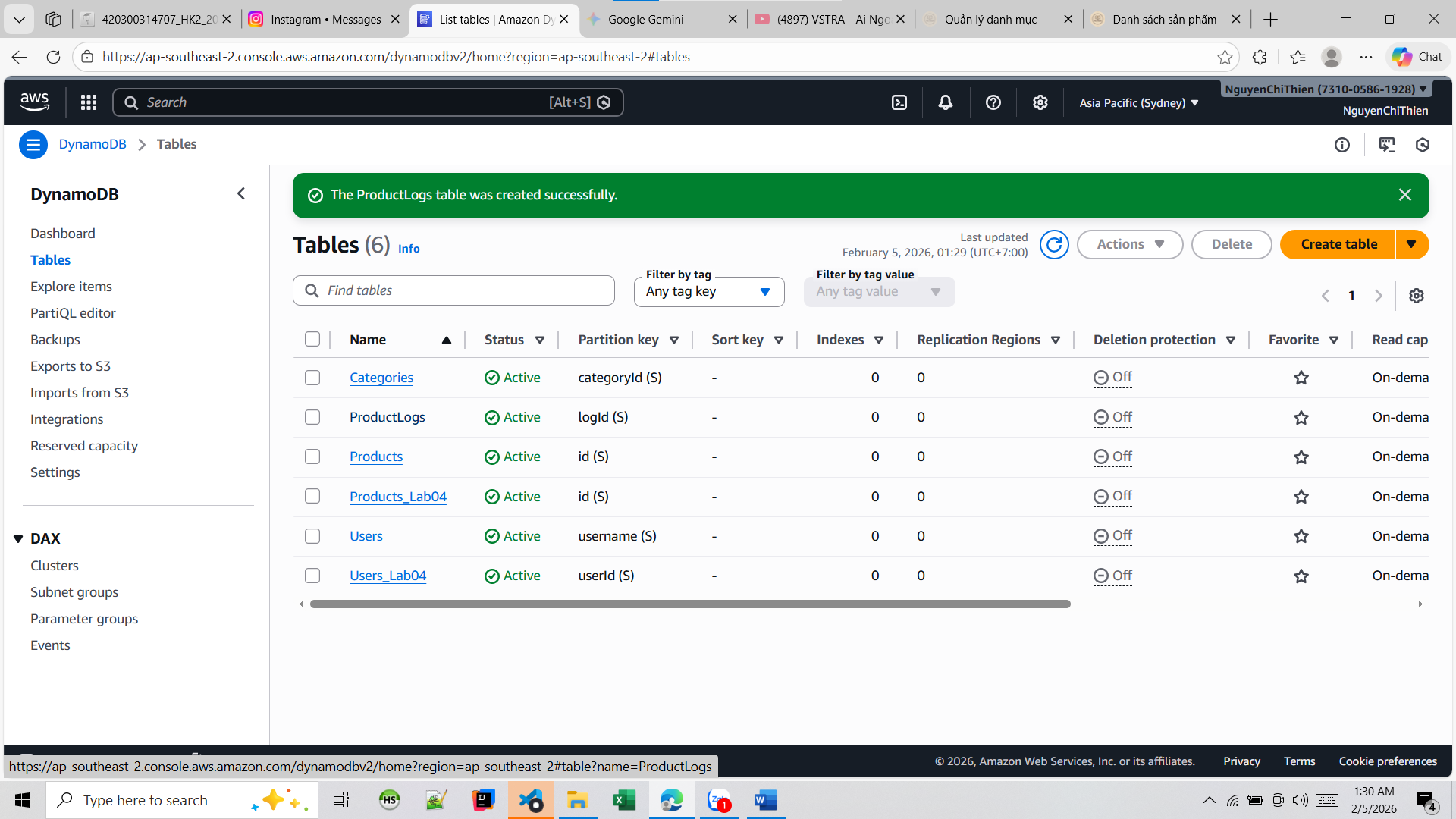
* + - Trạng thái hết hàng: isDeleted = false



**8️⃣ Lịch sử thao tác (Audit – nâng cao)**

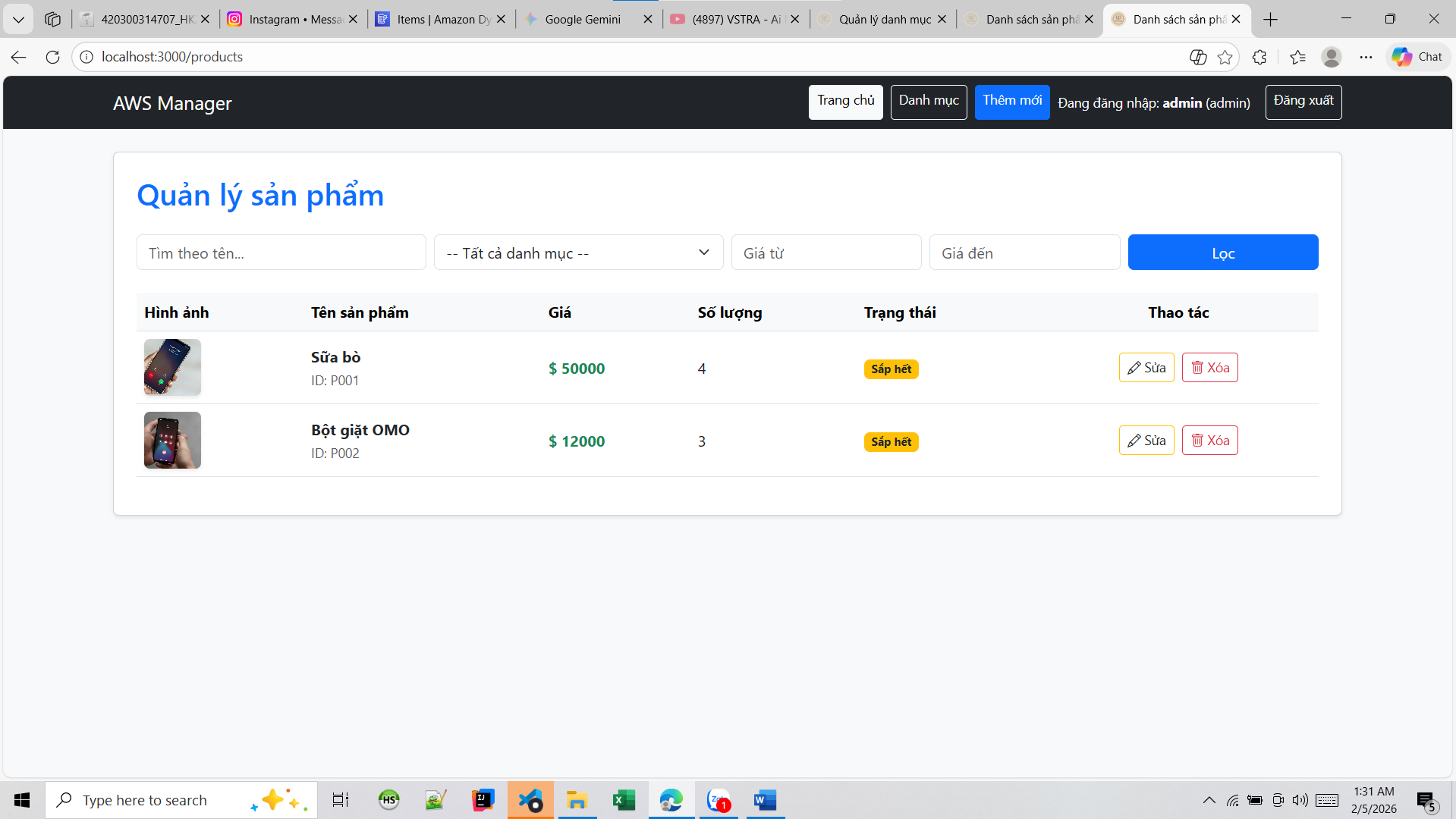
**Table: ProductLogs**

| **Thuộc tính** | **Kiểu** |
| --- | --- |
| logId (PK) | String |
| productId | String |
| action | String (CREATE/UPDATE/DELETE) |
| userId | String |
| time | String |

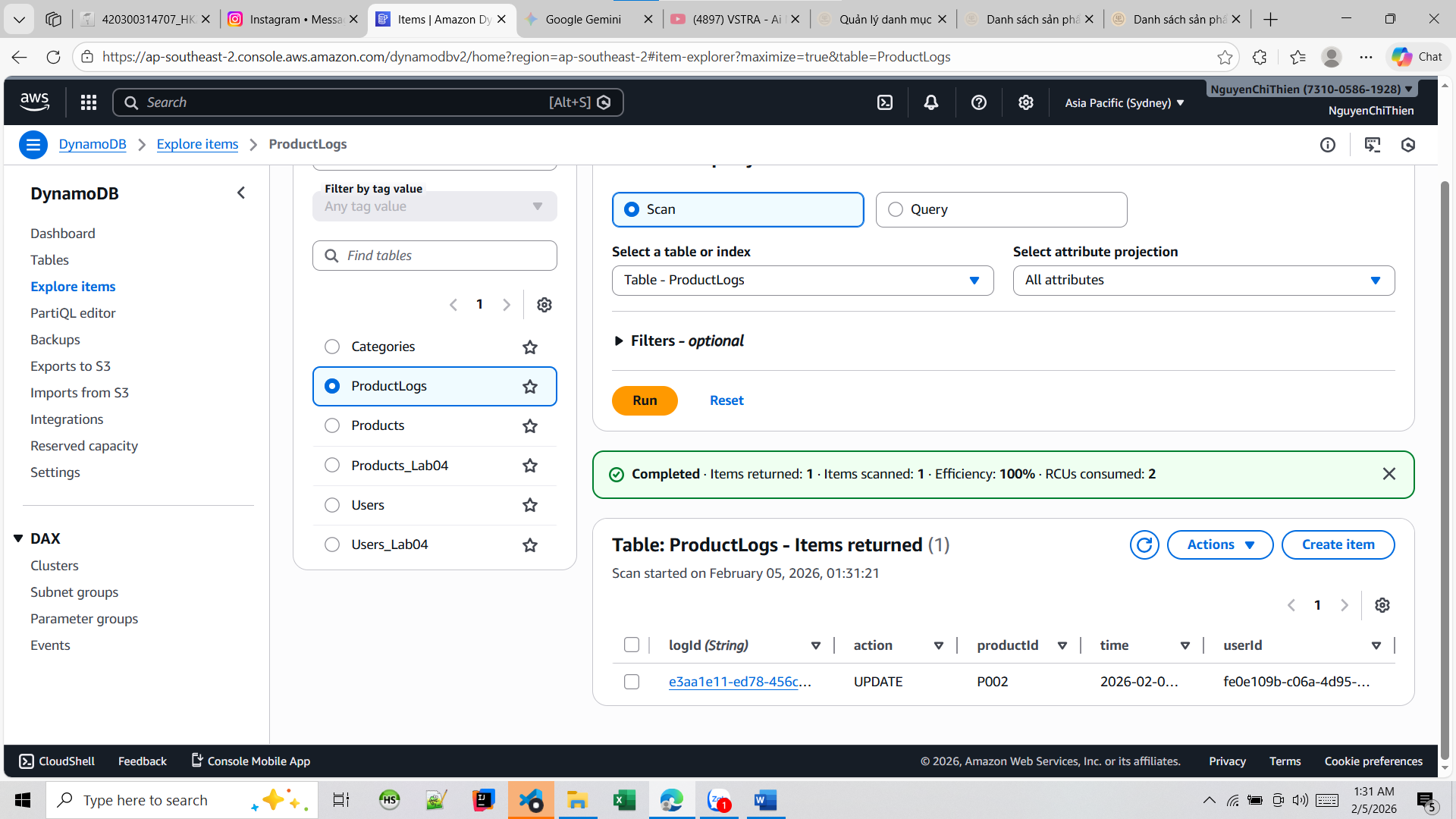


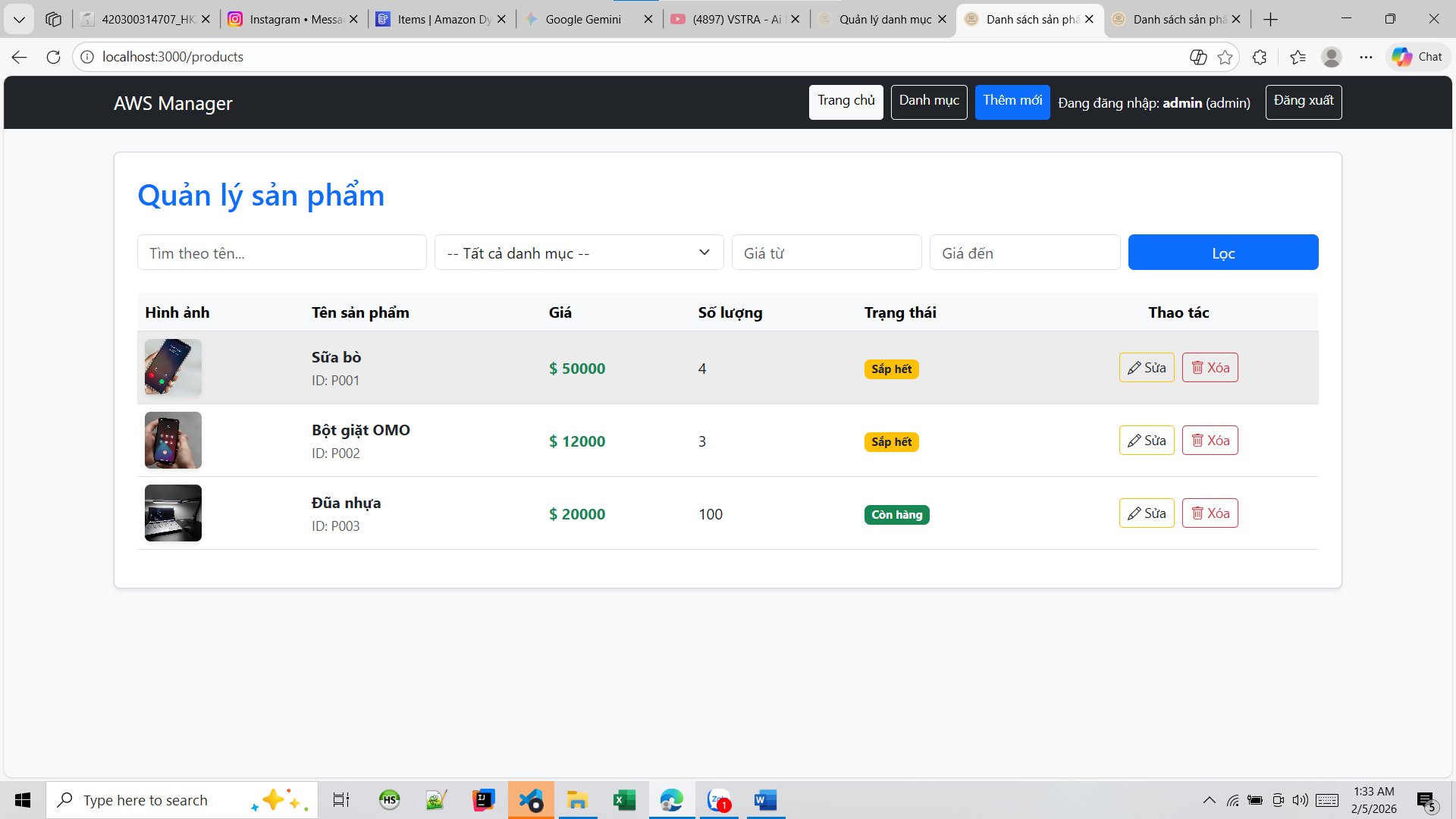
👉 Ghi log khi:

* thêm
* sửa
* xoá sản phẩm



* + - update số lượng của P002 thì 0 - > 3





* + - add sản phẩm



**III. YÊU CẦU KIẾN TRÚC & TRIỂN KHAI**

**9️⃣ Tách lớp rõ ràng**

Bắt buộc:

* controllers
* services
* repositories
* middlewares

👉 Không viết logic DynamoDB trực tiếp trong route

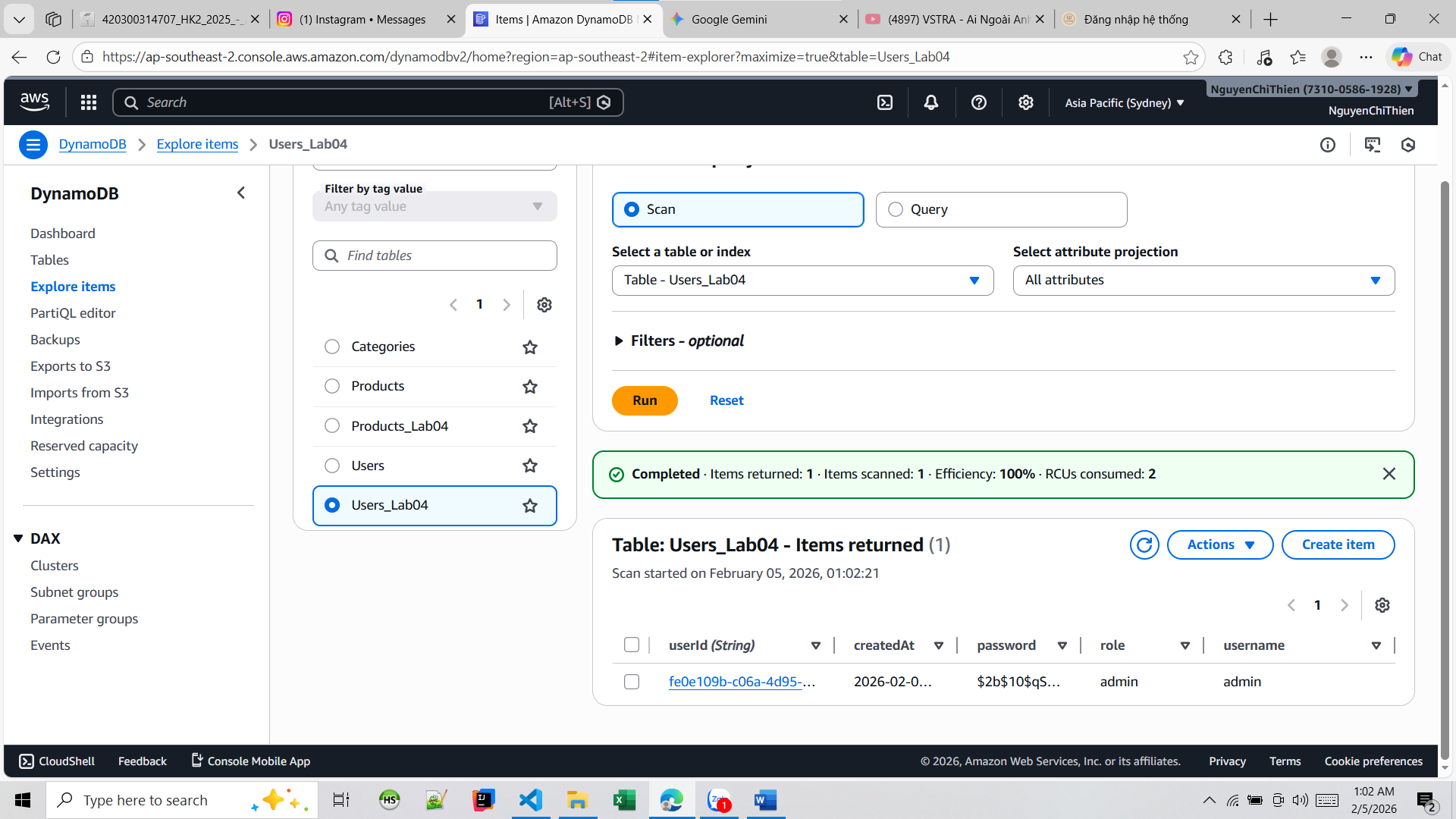
**🔟 AWS & Cloud**

* EC2 dùng **IAM Role**
* S3:
  + upload ảnh
  + xoá ảnh khi xoá sản phẩm
* Không hard-code:
  + access key
  + secret key

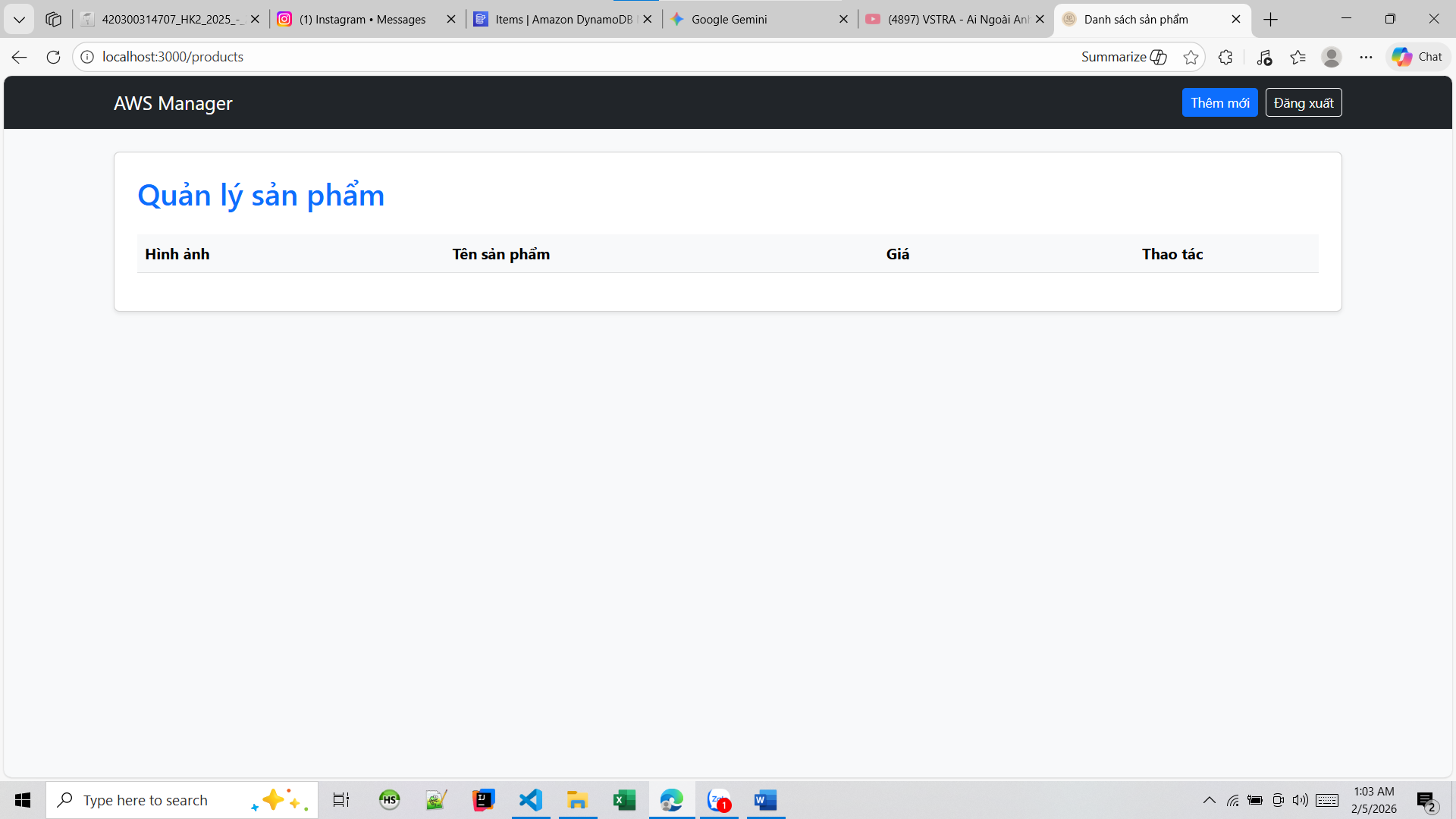
**IV. YÊU CẦU BÁO CÁO BỔ SUNG**

Sinh viên làm phần mở rộng phải trình bày thêm:

1. Sơ đồ quan hệ giữa các bảng DynamoDB
2. Luồng xử lý:
   * đăng nhập, đăng kí

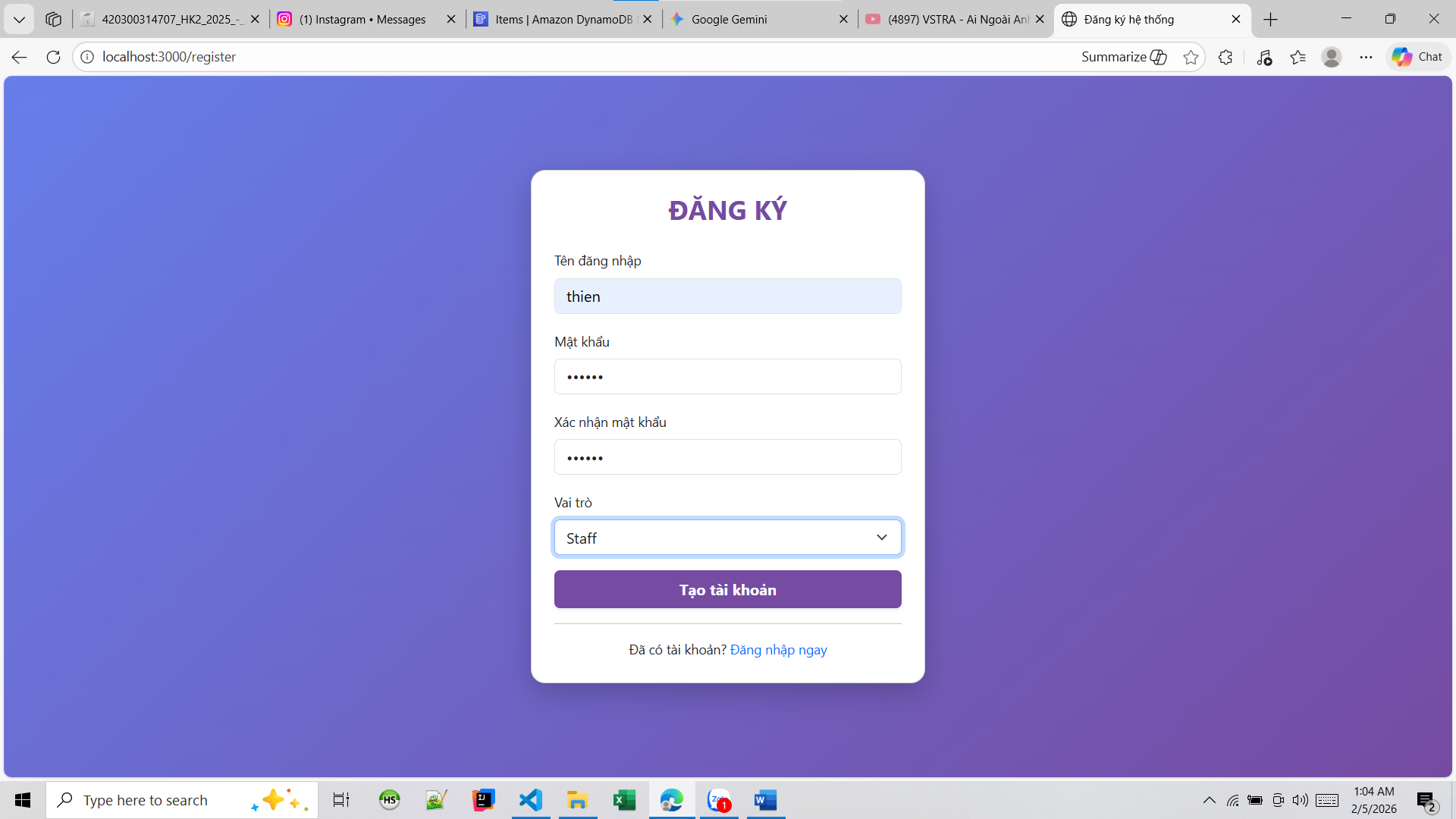


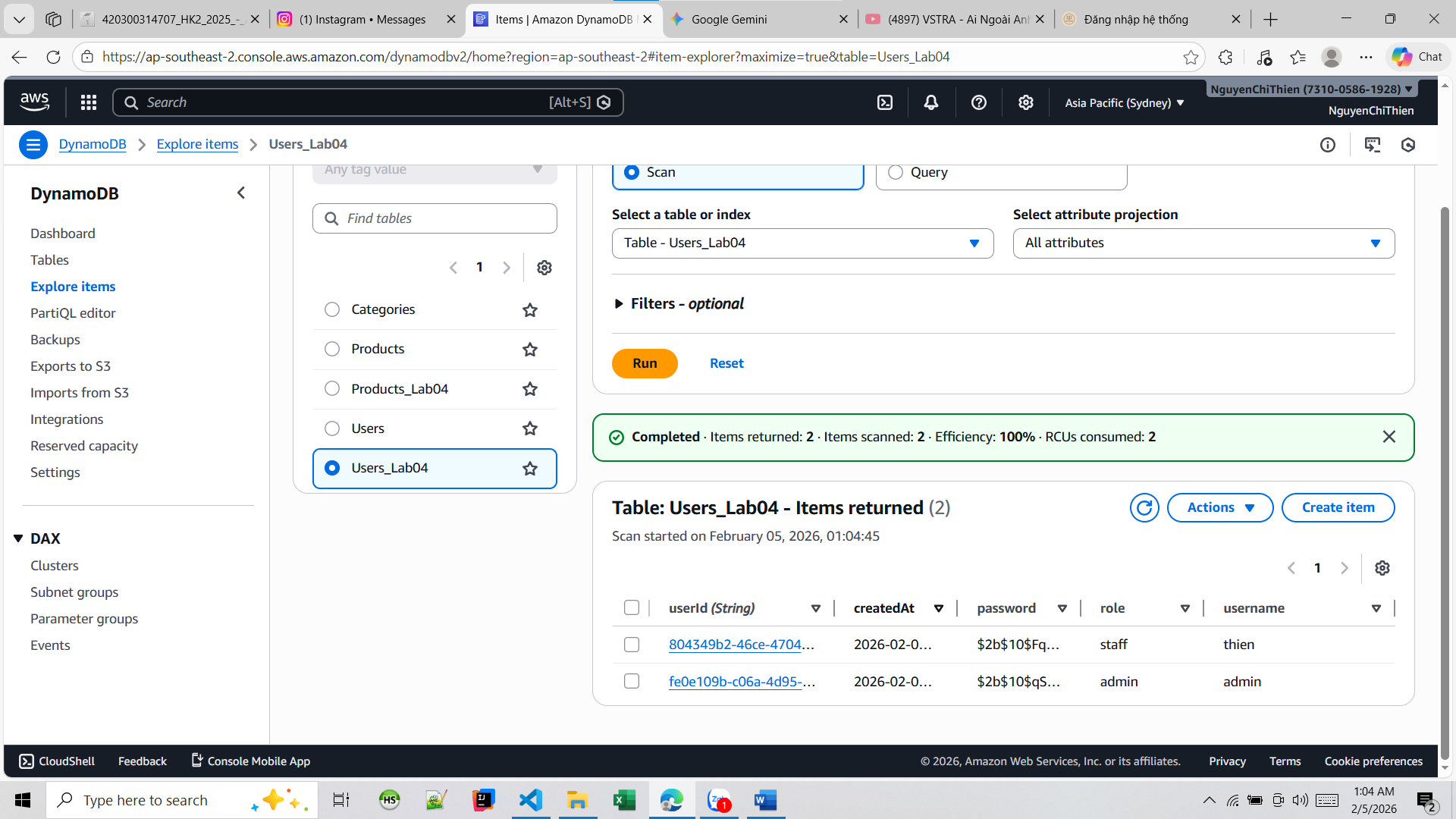
* đăng kí thành công với bảng Users mới



* đăng nhập thành công với vai trò admin

Bonus: tạo thêm một tài khoản là staff để thấy rõ phân quyền





* + thêm sản phẩm

1. So sánh:
   * DynamoDB vs MySQL cho bài toán này

| **Tiêu chí** | **DynamoDB (NoSQL)** | **MySQL (Relational DB)** |
| --- | --- | --- |
| **Cấu trúc (Schema)** | **Flexible:** Không cần định nghĩa trước toàn bộ trường. Dễ thêm thuộc tính cho sản phẩm (size, màu, cấu hình). | **Strict:** Phải định nghĩa bảng rõ ràng. Thay đổi Schema (ALTER TABLE) tốn thời gian khi dữ liệu lớn. |
| **Khả năng Scale** | **Scale ngang (Horizontal):** Tự động xử lý khi lượng user/order tăng đột biến (ví dụ: đợt Flash Sale). | **Scale dọc (Vertical):** Thường phải nâng cấp cấu hình server. Scale ngang (Sharding) rất phức tạp. |
| **Truy vấn (Query)** | **Hạn chế:** Chỉ nhanh khi dùng Key. Khó làm báo cáo doanh thu theo tháng hoặc lọc sản phẩm đa điều kiện. | **Linh hoạt:** SQL cực mạnh. Dễ dàng JOIN bảng Users với Orders để lấy thông tin chi tiết đơn hàng. |
| **Ràng buộc dữ liệu** | **Yếu:** Không có Foreign Key. Việc kiểm tra tồn kho (Inventory) phải xử lý cẩn thận ở tầng Logic Code. | **Mạnh:** Có Foreign Key, Constrain. Đảm bảo tính nhất quán (ACID) tuyệt đối cho các giao dịch thanh toán. |
| **Thanh toán/Giao dịch** | Hỗ trợ DynamoDB Transactions nhưng giới hạn và khó cấu hình hơn SQL truyền thống. | Là thế mạnh. Phù hợp cho các bài toán trừ tiền ví, cập nhật trạng thái đơn hàng phức tạp. |

1. Nhận xét ưu / nhược điểm mô hình NoSQL

| **Mặt** | **Đặc điểm chi tiết** |
| --- | --- |
| **Ưu điểm** | ● **Hiệu năng cực cao:** Đáp ứng thời gian phản hồi mili-giây cho các thao tác đọc/ghi profile, giỏ hàng.    ● **Lược đồ linh hoạt:** Sản phẩm điện tử có cấu hình khác sản phẩm thời trang, NoSQL cho phép lưu tất cả trong 1 bảng.    ● **Serverless:** Không cần quản lý server, tự động backup và sẵn sàng cao (High Availability). |
| **Nhược điểm** | ● **Không có JOIN:** Dữ liệu bị phân mảnh hoặc phải lưu lặp (Denormalization).    ● **Chi phí Scan:** Nếu không thiết kế Index (GSI/LSI) chuẩn, việc tìm kiếm sản phẩm sẽ rất tốn kém và chậm.    ● **Độ phức tạp ứng dụng:** Lập trình viên phải viết code để đảm bảo tính toàn vẹn dữ liệu thay vì dựa vào DB. |

**mở rộng vừa (2–3 bảng)** để có chức năng **như mini e-commerce (**khách hàng mua hàng**)**?

* Đã pull về trên EC2 và chạy thành công

