A. Bài test frontend:

Xây dựng giao diện theo figma: https://www.figma.com/design/kCQs5qoqcPinFTqLN2jpnF/Test?node-id=0-1&node-type=canvas&t=FZMsX23IB9AqNrDT-0

-> cuối mỗi ngày gửi file lên sky

B. Bài test backend

Mô tả bài test:

Bạn sẽ xây dựng API cho phần ecommerce đơn giản. API này sẽ cung cấp các chức năng để quản lý sản phẩm, quản lý đơn hàng, giỏ, thanh toán. Bạn sẽ sử dụng PHP thuần và MySQL để xây dựng API. Thời gian hoàn thành bài test tối đa là 2 ngày.

Yêu cầu cụ thể:

1. Chức năng API cơ bản:

Đăng ký & Đăng nhập (Authentication):

Xây dựng API để người dùng có thể đăng ký tài khoản mới (/api/register).

Xây dựng API để người dùng đăng nhập vào hệ thống và nhận về token để xác thực các yêu cầu tiếp theo (/api/login).

Token có thể được quản lý bằng JWT hoặc một cơ chế session dựa trên PHP.

Quản lý sản phẩm:

Tạo sản phẩm mới: API để tạo sản phẩm mới với tên, ảnh và nội dung (/api/product/create).

Xem chi tiết sản phẩm: API để xem chi tiết sản phẩm của người dùng đã đăng nhập (/api/product/url\_sanpham/).

Xem tất cả sản phẩm: API để lấy danh sách tất cả sản phẩm từ mọi người dùng (/api/products/all).

2. Thiết kế cơ sở dữ liệu (MySQL):

Bạn cần thiết kế và cung cấp script tạo các bảng cơ sở dữ liệu cho hệ thống API, bao gồm:

Bảng người dùng (users): Lưu trữ thông tin người dùng (id, tên, email, mật khẩu).

Bảng sản phẩm (product): Lưu trữ sản phẩm của chủ shop

Bảng lưu giỏ hàng:

Bảng đơn hàng

Bảng chủ shop (vender): Lưu thông tin về chủ shop

3. Yêu cầu về API và code:

Xây dựng API theo kiến trúc RESTful.

Sử dụng design pattern phù hợp trong PHP:

Singleton pattern để quản lý kết nối cơ sở dữ liệu.

Repository pattern (nếu có thể) để truy xuất và quản lý dữ liệu.

4. Xử lý bảo mật:

JWT (JSON Web Token) để xác thực người dùng cho các API cần đăng nhập. Khi người dùng đăng nhập thành công, trả về token để người dùng sử dụng cho các request sau này.

Mã hóa mật khẩu người dùng bằng cách sử dụng PHP password\_hash().

Kiểm tra và xác thực dữ liệu đầu vào của người dùng để đảm bảo an toàn (Validation).

5. Cách gửi dữ liệu và phản hồi:

Các API cần hỗ trợ gửi và nhận dữ liệu ở định dạng JSON.

Mỗi API cần trả về mã trạng thái HTTP phù hợp (200, 201 cho thành công, 400 cho lỗi yêu cầu không hợp lệ, 401 cho yêu cầu không được xác thực, 500 cho lỗi server).

Ví dụ về cấu trúc response:

json

Copy code

{

"status": "success",

"message": "Post created successfully",

"data": {

"id": 1,

"title": "My First Post",

"content": "This is the content of my first post",

"created\_at": "2024-10-20 10:30:00"

}

}

6. Yêu cầu phụ:

Đảm bảo cơ chế phân trang cho các danh sách sản phẩm

Ghi log các lỗi xảy ra trên hệ thống.

Kết quả mong đợi:

Mã nguồn hoàn chỉnh của API với cấu trúc thư mục rõ ràng.

Script tạo cơ sở dữ liệu MySQL.

Document API (có thể sử dụng file markdown hoặc Postman collection) để mô tả cách sử dụng từng API.

-> gửi code đã làm vào cuối ngày