DỰ ÁN KỸ THUẬT CÁ NHÂN

Tên ứng viên : Phạm Văn Khởi.

Vị trí ứng tuyển : Lập trình viên Fullstack.

1) Tổng quan dự án.

- Tên dự án: Website HADO Ecommerce.
- Mục tiêu: Dự án nhằm xây dựng một hệ thống backend cho phép người dùng đăng ký, đăng nhập, phân quyền và truy cập tài nguyên thông qua API một cách an toàn.
  - Công nghệ sử dụng:
    - Backend: NodeJS Express.
    - Cơ sở dữ liệu: MongoDB.
    - Xác thực: JWT (Json Web Token, Redis).
  - Quy mô:
    - Số lượng API: 15-20 endpoint.
    - Số bảng trong CSDL: ~15 bảng.

2) Khó khăn về mặt kỹ thuật.

- Vấn đề kỹ thuật chính: Thiết kế cơ chế xác thực và phân quyền người dùng trong hệ thống backend, đảm bảo an toàn, chống các lỗ hổng bảo mật phổ biến và dễ mở rộng.
- Khó khăn gặp phải:
  - Việc lưu token trong Local Storage tiềm ẩn nguy cơ bị tấn công XSS do JavaScript có thể truy cập trực tiếp.
  - Đảm bảo tính bảo mật trong quá trình phát hành và xác thực token, đặc biệt là phòng chống XSS và CSRF.
  - Xử lý các tình huống token hết hạn và cơ chế refresh token một cách an toàn.
  - Tránh lộ thông tin nhạy cảm của người dùng trong payload của JWT.
  - Ngăn chặn truy cập trái phép vào các API quan trọng, đồng thời vẫn đảm bảo quyền truy cập đầy đủ cho tài khoản quản trị (admin).
- Cách tôi giải quyết:

    - Áp dụng JWT kết hợp với Role-Based Access Control (RBAC)**.** Token được lưu trong HttpOnly Cookie thay vì Local Storage để ngăn chặn XSS (JavaScript không thể đọc cookie), đồng thời trình duyệt tự động đính kèm cookie trong mỗi request.
    - Sử dụng Redis để quản lý refresh token, cho phép thu hồi token khi cần và tăng mức độ bảo mật.
    - Thiết kế middleware (ví dụ: Authenticated, AuthorizeRole) để kiểm tra JWT cho mọi request, ngoại trừ các endpoint công khai.
    - Phân quyền rõ ràng theo từng role (ADMIN, USER, …) ở tầng controller, tách biệt rõ giữa authentication (xác thực) và authorization (phân quyền).
    - Cấu hình bổ sung để giảm thiểu rủi ro CORS, CSRF khi sử dụng cookie-based authentication.
- Kết quả:
    - Hệ thống luôn xác thực token ở mỗi request và chỉ cho phép người dùng có quyền phù hợp truy cập vào các API tương ứng.
    - Giảm thiểu rủi ro bảo mật và giúp kiến trúc dễ mở rộng khi bổ sung role mới hoặc thêm chức năng API trong tương lai.

3) Vì sao dự án này chuyên sau về mặt kỹ thuật.

- Tôi cho rằng dự án này mang tính chuyên sâu vì:
    - Không chỉ dừng ở việc “viết API”, mà phải:
        - Thiết kế kiến trúc xác thực và phân quyền(User, Role, Permission).
        - Hiểu rõ luồng hoạt động của request/response, middleware, security layer.
        - Tiền đề cho các API hoạt động đúng mục đích.
    - Phải xử lý:
        - Các case về token hết hạn, truy cập trái phép.
    - Áp dụng các khái niệm:
        - Clean Architecture.
        - Separation of Concerns.
        - Security best practices.

- Dự án giúp tôi hiểu rõ hơn về cách xây dựng một hệ thống backend an toàn, có thể mở rộng và vận hành trong môi trường thực tế.

4) Tài liệu liên quan.

- Link Github dự án cá nhân : https://github.com/phamvankhoi35/haho-eccomerce-be
- Link Github Optibot-clone : https://github.com/phamvankhoi35/Optibot-clone
- Deployment Optibot : https://cloud.digitalocean.com/apps/0644c564-a4ad-4c7e-ace5-aead3656f816/logs/optibot-clone?i=c22e53