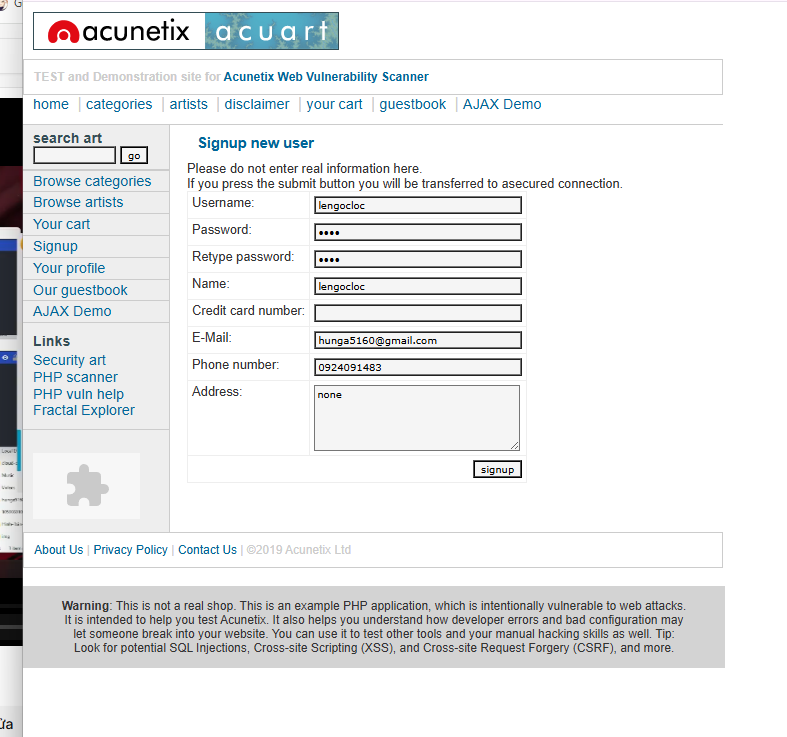
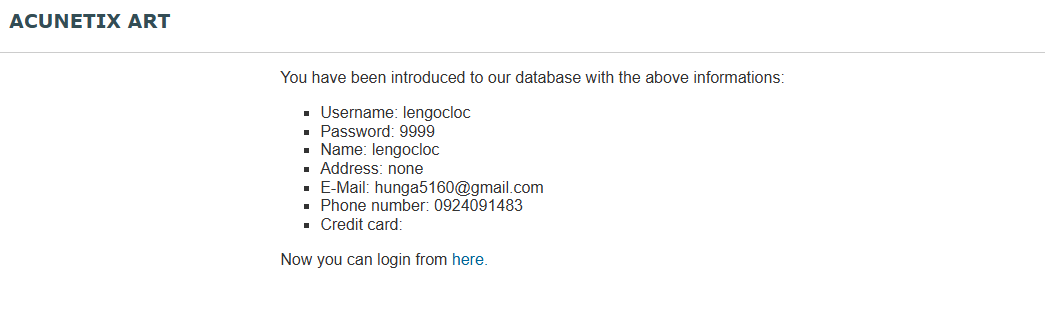
# **Câu 1 (3.5 điểm): Kiểm thử lỗ hổng SQL Injection trên trang register.php**

1. Xác định các tham số đầu vào và cách thức gửi tới server (0.25 điểm)

* Tham số đầu vào: username, password, retype password.
* Phương thức gửi: Dữ liệu được gửi đến server thông qua phương thức POST.





2. Phân tích và phán đoán kiểu truy vấn (0.25 điểm)

Dựa vào chức năng đăng ký, em có thể phán đoán câu truy vấn SQL như sau:

* INSERT INTO users (username, password) VALUES ('$username', '$password');

Bên cạnh đó, nếu không có biện pháp lọc và kiểm tra dữ liệu đầu vào, truy vấn này dễ bị tấn công SQL Injection.

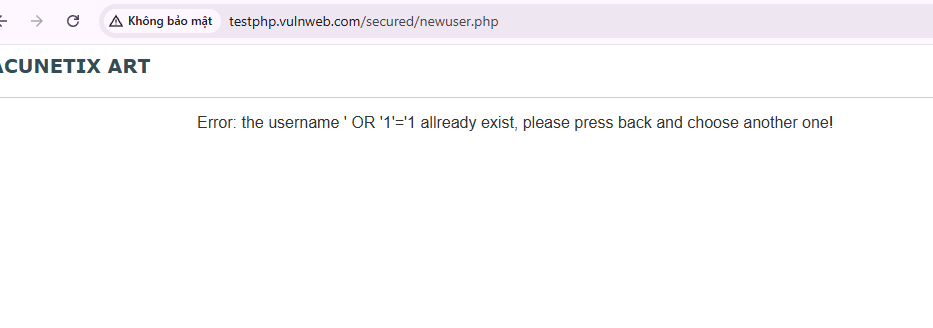
3. Phân tích vị trí tham số đầu vào trong truy vấn (0.5 điểm)

* Tham số username và password được chèn trực tiếp vào phần VALUES của câu truy vấn.
* Tuy nhiên, nếu không xử lý đúng cách, kẻ tấn công có thể chèn mã SQL độc hại vào các trường này.

4. Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào (2.5 điểm)

Kiểm thử 1: Trường username

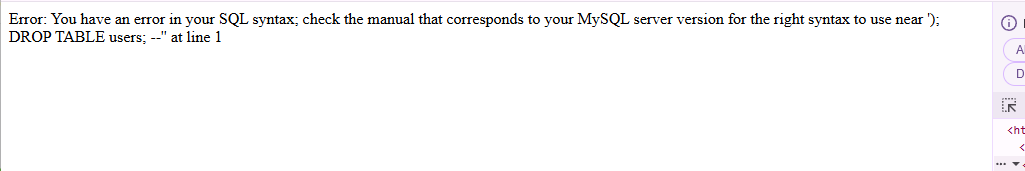
* Dữ liệu thử nghiệm: ' OR '1'='1
* Mục đích: Bypass xác thực bằng cách tạo điều kiện luôn đúng.



* Kết luận : Vì nhận được phản hồi bất thường, chứng tỏ có lỗ hổng SQL Injection.

Kiểm thử 2: Trường password

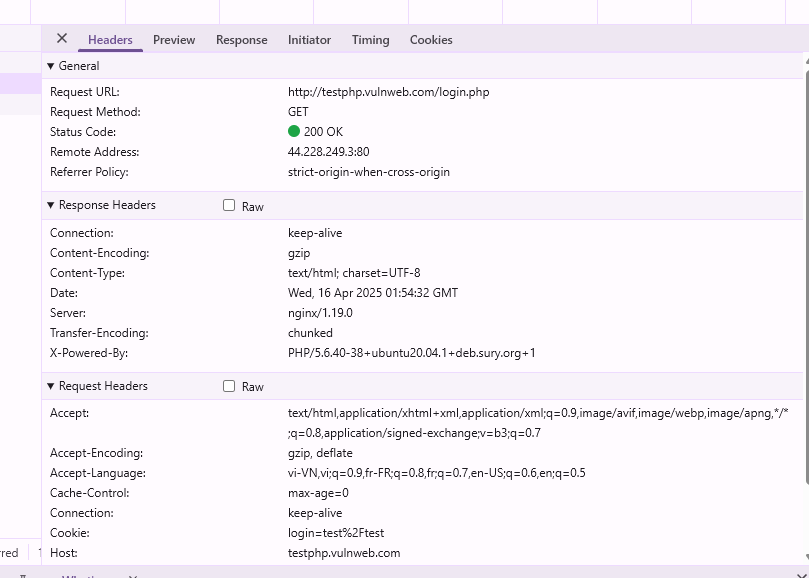
* Dữ liệu thử nghiệm: abc123'); DROP TABLE users; --
* Mục đích: Thử tấn công phá hoại cơ sở dữ liệu.



* Kết quả: Hệ thống vẫn hoạt động bình thường cho nên không bị hỏng nặng về SQL Injection

# **Câu 2 (3.5 điểm): Kiểm thử lỗ hổng SQL Injection trên trang login.php**

1. Xác định các tham số đầu vào và cách thức gửi tới server (0.25 điểm)

* Tham số đầu vào: admin, admin.
* 
* Phương thức gửi: Dữ liệu được gửi đến server thông qua phương thức POST.

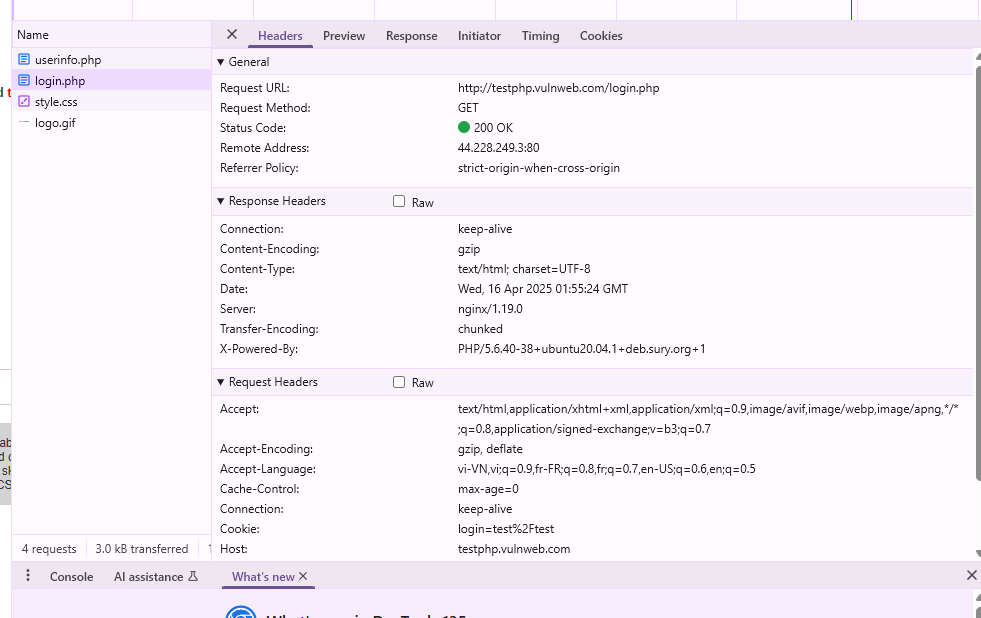
2. Phân tích và phán đoán kiểu truy vấn, vị trí tham số đầu vào (0.25 điểm)

* SELECT \* FROM users WHERE username = '$admin AND password = '$admin;

3. Xác định mức độ ảnh hưởng của lỗ hổng trên mỗi giá trị đầu vào (3 điểm)

Kiểm thử 1: Bypass xác thực

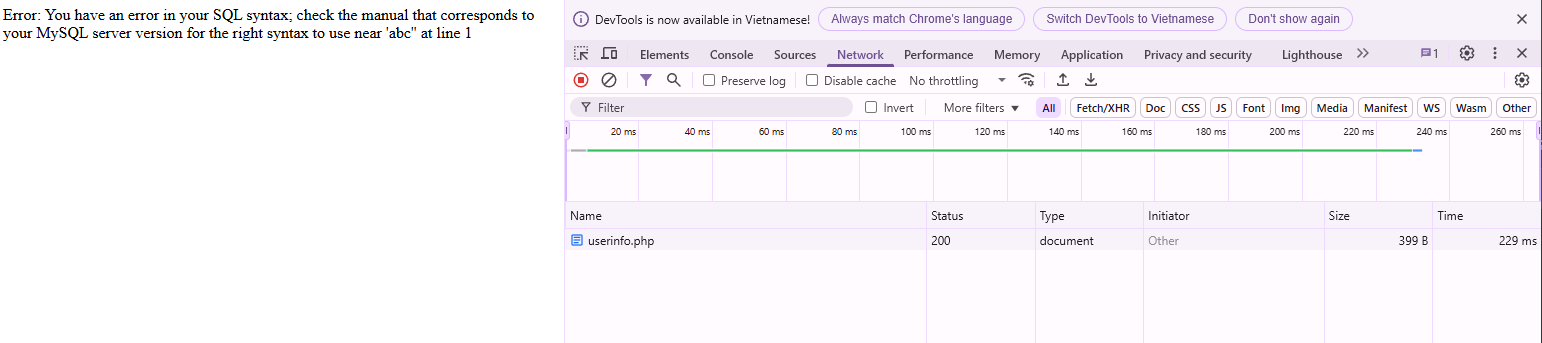
* Dữ liệu thử nghiệm:  
  + username: ' OR '1'='1
  + password: abc123
* Mục đích: Đăng nhập mà không cần mật khẩu hợp lệ.



* Kết quả: Kết quả đăng nhập không thành công, hệ thống bảo mật đang hoạt động ổn định

Kiểm thử 2: Phát hiện lỗi hệ thống

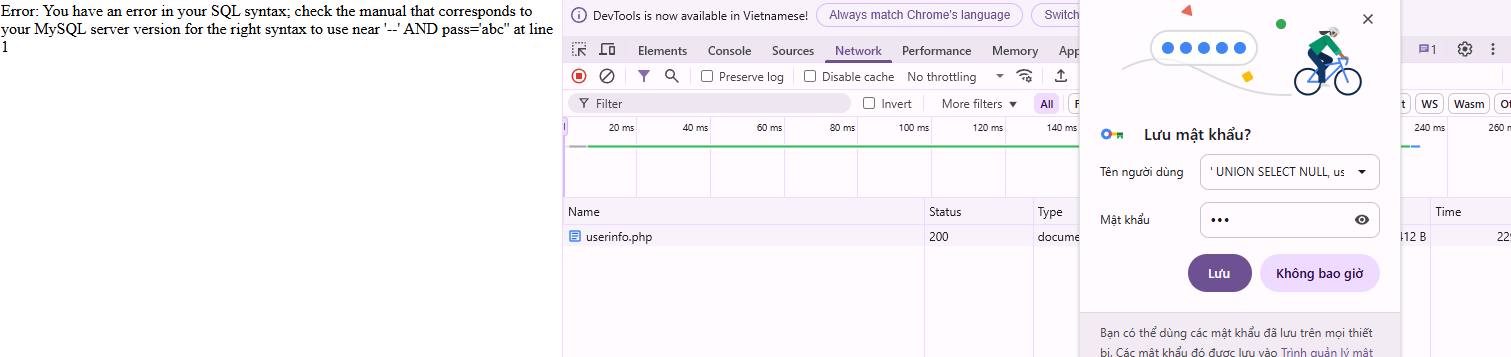
* Dữ liệu thử nghiệm:  
  + username: '
  + password: abc
* Mục đích: Kiểm tra hệ thống có xử lý đúng cú pháp SQL không.



* Kết quả: Kết quả xuất hiện lỗi cú pháp SQL, chứng tỏ không có biện pháp lọc dữ liệu đầu vào.

Kiểm thử 3: Liệt kê người dùng

* Dữ liệu thử nghiệm:  
  + username: ' UNION SELECT NULL, username, password FROM users--
  + password: abc
* Mục đích: Thử lấy danh sách người dùng từ cơ sở dữ liệu.



* Kết quả: Không hiển thị danh sách người dùng cho nên hệ thống không bị lỗ hổng nghiêm trọng.