BÁO CÁO BÀI TẬP

Môn học: Lập trình ứng dụng Web

Kỳ báo cáo: Lab 05

Tên chủ đề: Cấu hình web server với Nginx

GV: Nguyễn Bùi Kim Ngân

Ngày báo cáo: 25/05/2025

1. THÔNG TIN CHUNG:

Lớp: NT219.P22.ANTT.1

STT	Họ và tên	MSSV	Email
1	Lương Xuân Anh	23520051	23520051@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

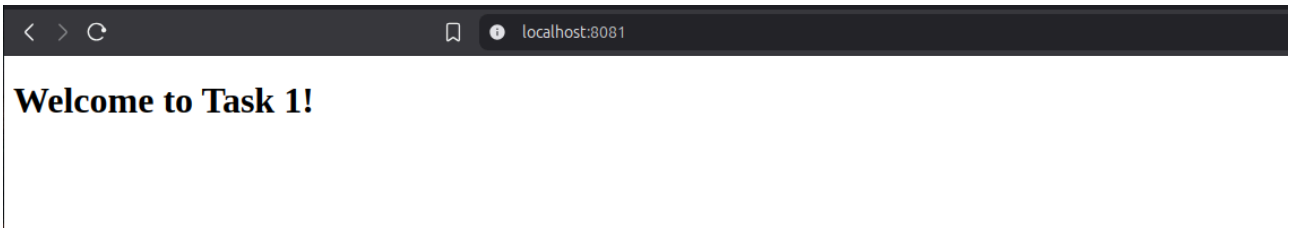
STT	Công việc	Kết quả tự đánh giá
1	Task 1	100%
2	Task 2	100%
3	Task 3	100%
4	Task 4	100%

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết.

[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

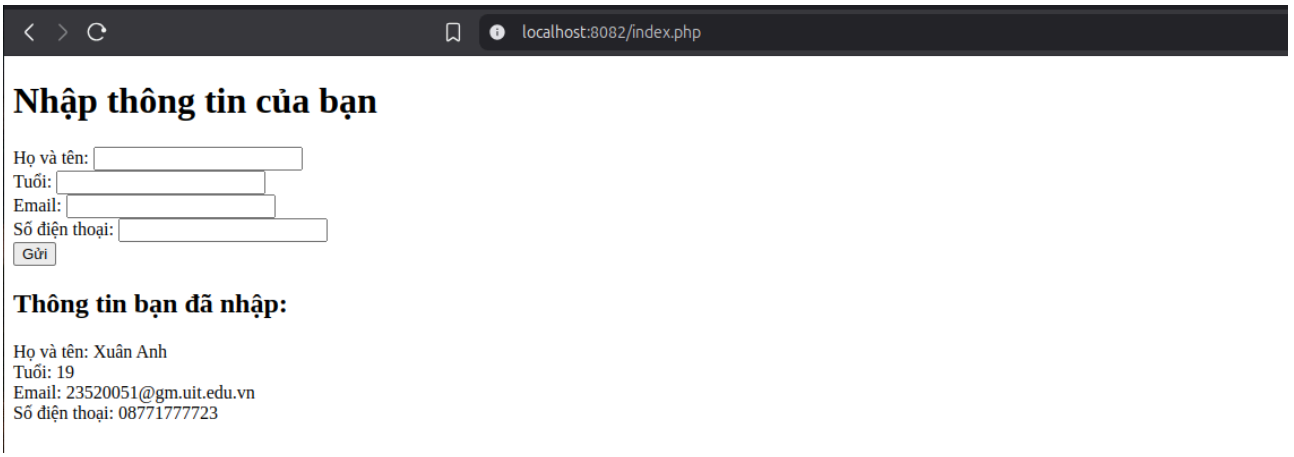
BÁO CÁO CHI TIẾT

1. Task 1



Hình 1.1. Task1

2. Task 2

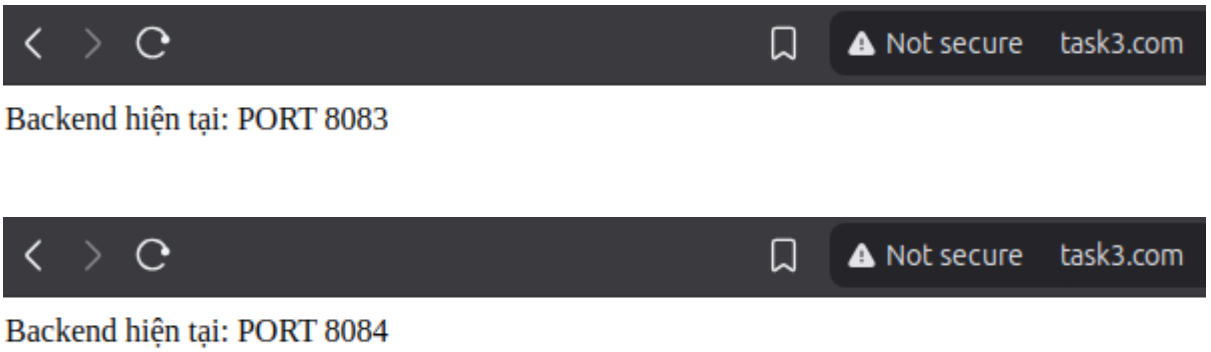


Hình 2.2. Task2

3. Task 3

Hình 3.1. Backend 8083

Hình 3.2. Backend 8084



Hình 3.3. Chuyển trạng thái page (F5)

```
if ($http_x_forwarded_for = "") {
    return 403;
}
```

Hình 3.4. Config không truy cập được port 8083

4. Task 4

Round-robin và least_conn: xen kẽ đều giữa 8083 và 8084.

Giải thích (Round-robin):

- Cách hoạt động:

Phân phối các request tuần tự theo vòng tròn tới các backend trong danh sách.

Ví dụ:

Request 1 → Backend A

Request 2 → Backend B

Request 3 → Backend A

Request 4 → Backend B

… (vòng lặp)

- Ưu điểm:

Cài đặt đơn giản.

Cân bằng số lượng request tương đối đều cho các backend.

- Nhược điểm:

Không tính đến tải thực tế từng backend.

Nếu 1 backend yếu hơn hoặc bị treo, phải cấu hình thêm để loại backend đó ra.

```
❯ for i in {1..20}; do curl -s http://task3.com; echo ""; sleep 1; done

Backend hiện tại: PORT 8083

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8083

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8083

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8083

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8083

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8083
```

Hình 4.1. Round robin

Giải thích (least_conn):

- Cách hoạt động:

Request tiếp theo sẽ được gửi tới backend đang có ít kết nối đang mở nhất.

Phù hợp với các backend xử lý lâu hoặc không đồng đều về hiệu năng.

- Ưu điểm:

Cân bằng tải tốt hơn cho các backend không đồng đều.

Giảm nguy cơ backend yếu bị quá tải.

- Nhược điểm:

Không phải lúc nào cũng tối ưu nếu request nhanh hoặc backend ngang nhau.

Tính toán số kết nối có thể tốn thêm tài nguyên.

```
❯ for i in {1..20}; do curl -s http://task3.com; echo ""; sleep 1; done
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8084
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8084
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8084
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8084
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8084
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8084
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8084
```

Hình 4.2. least conn

ip_hash: cùng IP sẽ luôn là 1 backend.

Giải thích:

- Cách hoạt động:

Dựa vào IP của client, hash ra backend cố định.

Nghĩa là cùng 1 IP sẽ luôn gửi về cùng 1 backend (trừ khi backend đó unavailable).

- Ưu điểm:

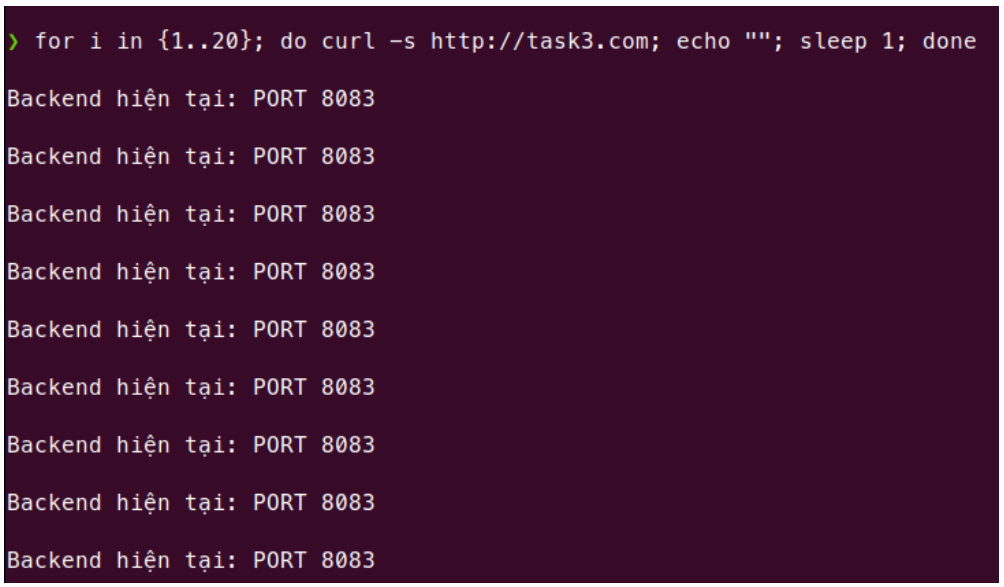
Giữ session ổn định (ví dụ giỏ hàng, đăng nhập…)

Phù hợp cho các hệ thống cần session sticky.

- Nhược điểm:

Nếu 1 backend bị down, session liên quan đến backend đó mất.

Không cân bằng tải tốt bằng các thuật toán khác.



weight: backend có trọng số lớn hơn sẽ xuất hiện nhiều hơn.

Giải thích:

- Cách hoạt động:

Mỗi backend được gán trọng số (weight), backend có weight lớn hơn nhận nhiều request hơn.

Ví dụ:

Backend A weight=3

Backend B weight=1

→ A nhận 3 request thì B nhận 1.

- Ưu điểm:

Phù hợp khi backend có cấu hình phần cứng khác nhau.

Linh hoạt hơn so với round-robin.

- Nhược điểm:

Phải tính toán đặt weight phù hợp.

Nếu weight chênh lệch lớn dễ gây overload nếu không cấu hình cẩn thận.

```
❯ for i in {1..20}; do curl -s http://task3.com; echo ""; sleep 1; done
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8084
Backend hiện tại: PORT 8083
Backend hiện tại: PORT 8083
```

Hình 4.4. Trọng số port 8083 lớn hơn 8084

```
nginx: configuration file /etc/nginx/nginx.conf test is successful
❯ for i in {1..20}; do curl -s http://task3.com; echo ""; sleep 1; done

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8083

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084

Backend hiện tại: PORT 8084
```

Hình 4.5. Trọng số port 8083 lớn hơn 8084

HẾT