ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

SYSTEM EVALUATION AND VALIDATION

Nhóm 4

Môn học: ĐỒ ÁN CÔNG NGHỆ PHẦN MỀM

Giảng viên:

Ngô Huy Biên

Thành phố Hồ Chí Minh – 2024

Thành viên

MSSV	Họ Tên	Vai Trò
20120196	Mai Cường Thịnh	Nhóm trưởng
18120419	Phạm Trường Khoa	Thành viên
20120205	Lê Đông Thức	Nhóm phó
20120293	Võ Phi Hùng	Thành viên
20120340	Trần Nhật Nguyên	Thành viên
20120383	Nguyễn Đức Tiến	Thành viên
20120447	Trịnh Quốc Cường	Thành viên
20120467	Nguyễn Phước Hải	Thành viên
20120633	Viên Hải Yến	Nhóm phó
20120634	Lê Minh Trí	Thành viên

Mục lục

LINK DEMO:

https://youtu.be/JNJ7JZaNs_8

I. UNIT TESTING

a) Đăng ký và cài đặt

Ở thư mục gốc của dự án: **cd server**

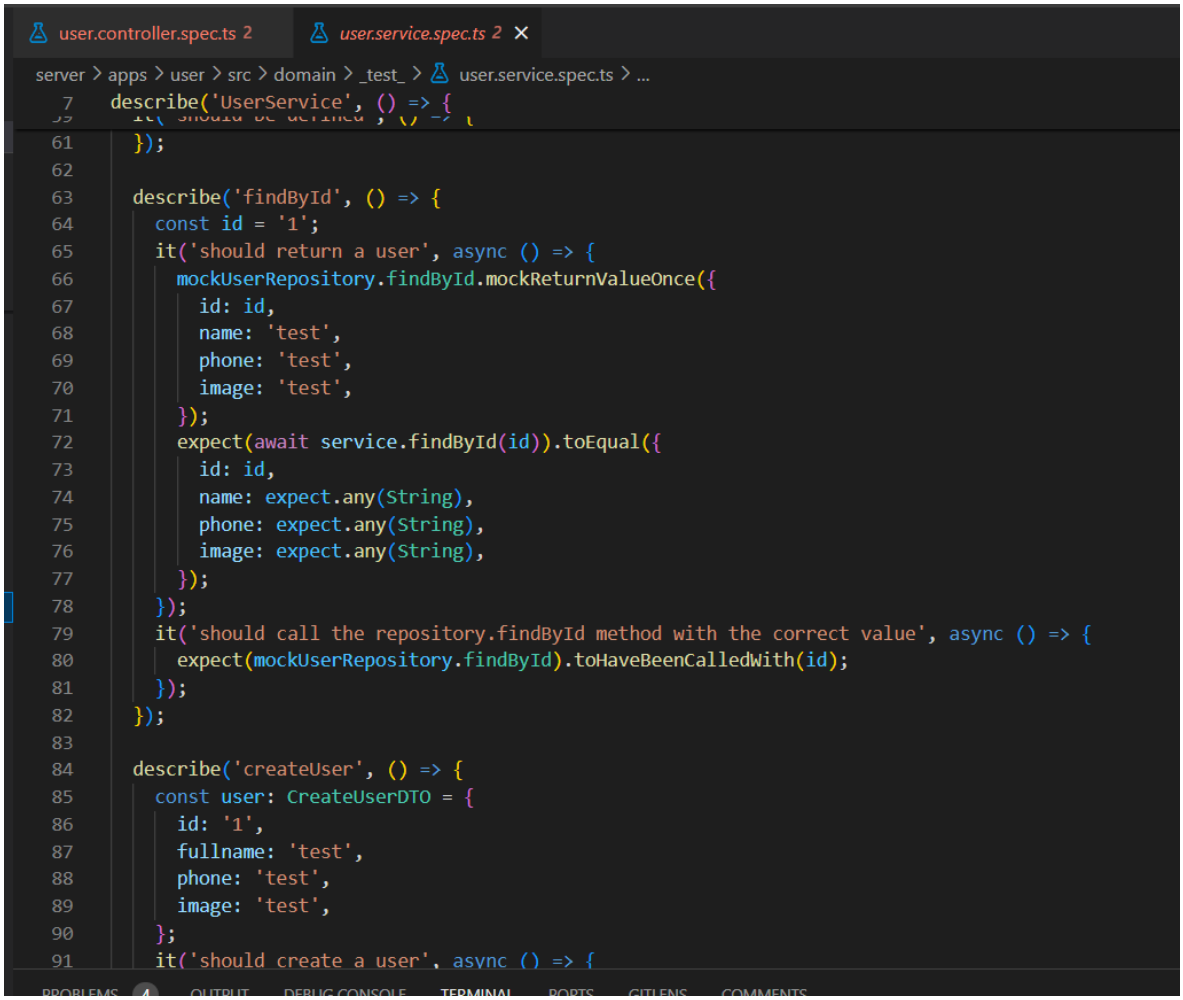
Sử dụng lệnh: **pnpm install jest @types/jest ts-jest** để cài đặt jest qua pnpm

b) Phương pháp thực thi

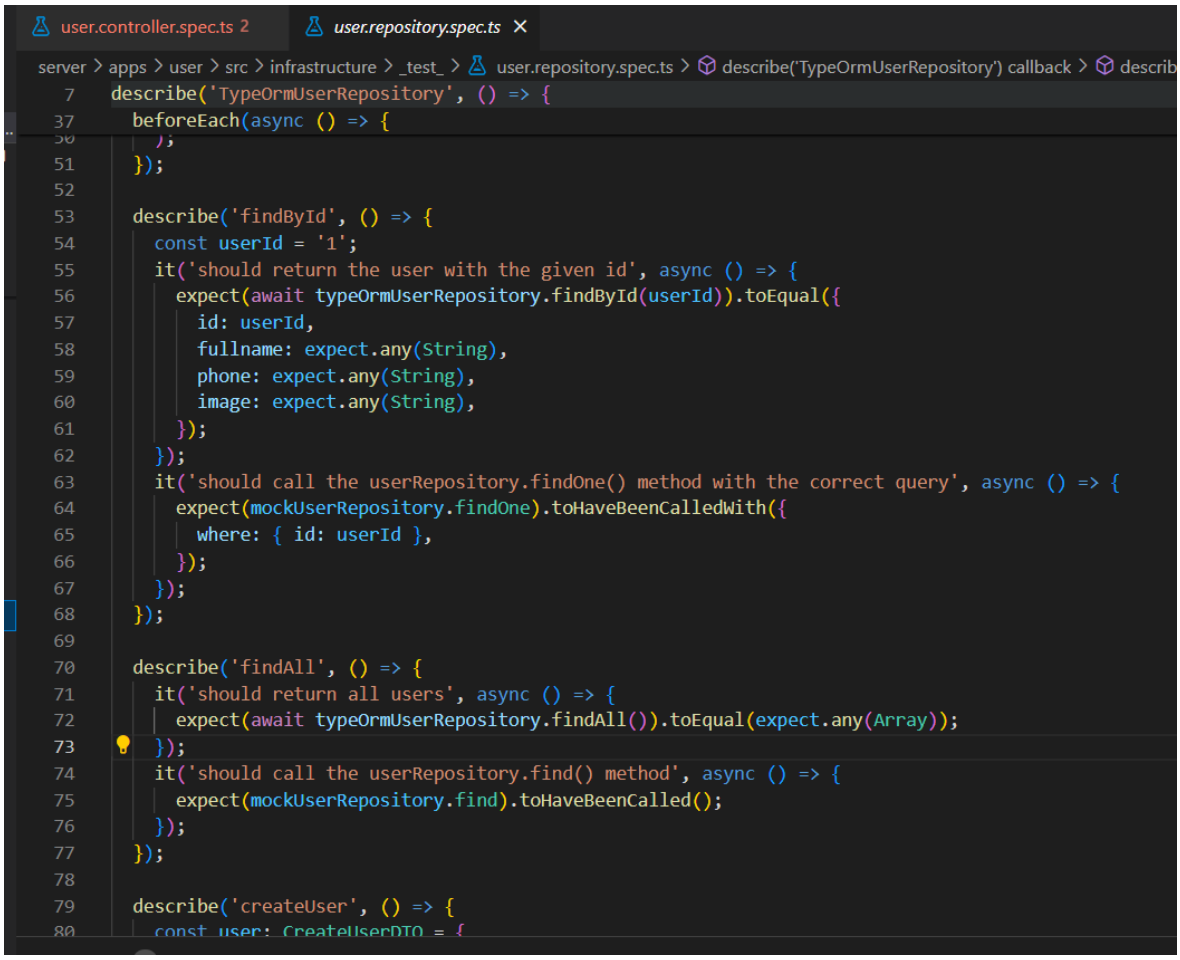
Áp dụng unit test đối với service user gồm controller , service và repository
Viết các file unit test:

- **user.controller.spec.ts:**

- **user.service.spec.ts:**



- **user.repository.spec.ts**



Chạy lệnh thực thi:

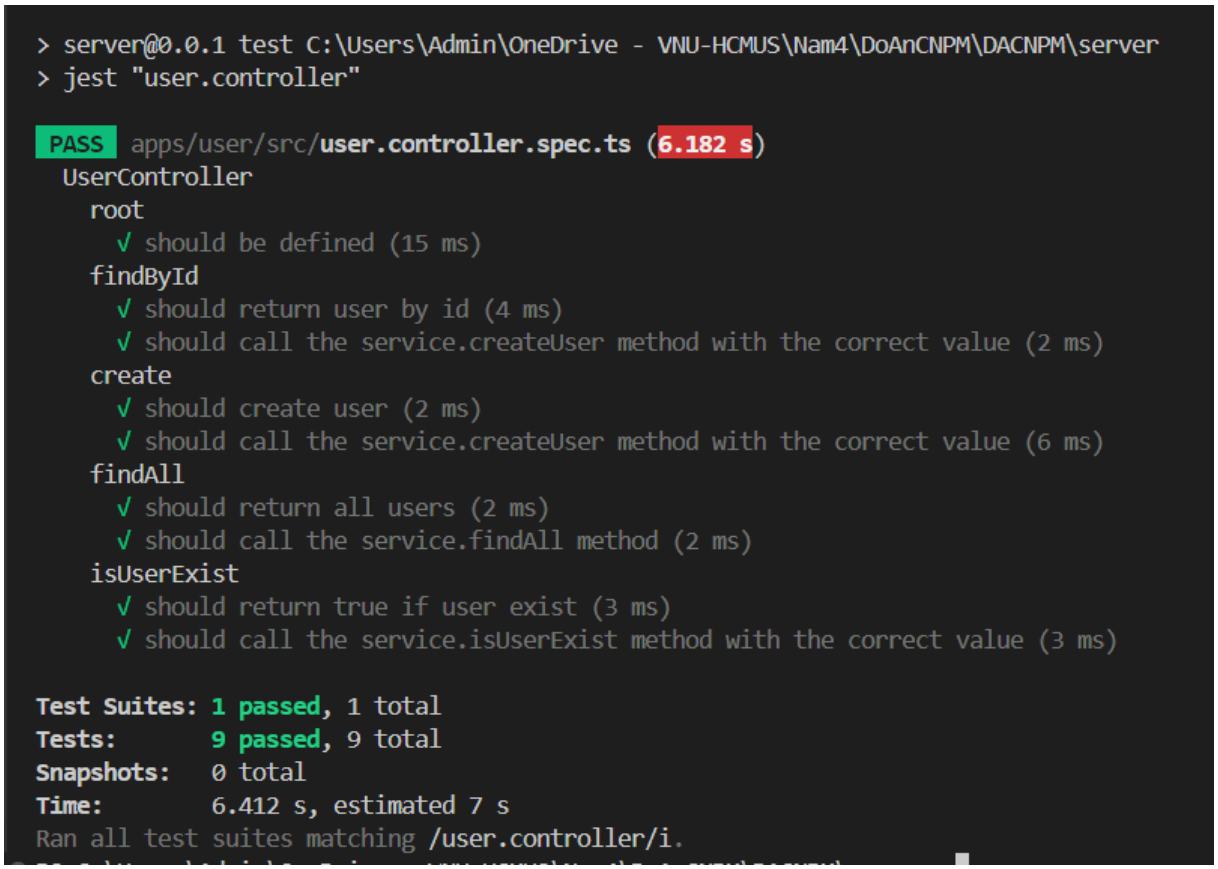
- Đối với user.controller.spec.ts: **pnpm test user.controller.spec.ts**
- Đối với user.service.spec.ts: **pnpm test user. service.spec.ts**
- Đối với user.repository.spec.ts: **pnpm test user. repository.spec.ts**

c) Kết quả thu được

Sau khi chạy kiểm thử, Jest sẽ cung cấp kết quả chi tiết về các trường hợp kiểm thử đã thực hiện, bao gồm cả những trường hợp thành công và thất bại.

Ví dụ:

- **user.controller.spec.ts**



- **user.service.spec.ts**

```
> server@0.0.1 test C:\Users\Admin\OneDrive - VNU-HCMUS\Nam4\DoAnCNPM\DACNPM\server
> jest "user.service.spec.ts"

 PASS  apps/user/src/domain/_test_/user.service.spec.ts (6.86 s)
  UserService
    √ should be defined (17 ms)
    findById
      √ should return a user (4 ms)
      √ should call the repository.findById method with the correct value (2 ms)
    createUser
      √ should create a user (2 ms)
      √ should call the repository.createUser method if the user is not exits (2 ms)
    findAll
      √ should return all users (2 ms)
      √ should call the repository.searchJob method (2 ms)
    isUserExist
      √ should return true (2 ms)
      √ should call the repository.isUserExist method with the correct value (3 ms)

Test Suites: 1 passed, 1 total
Tests:       9 passed, 9 total
Snapshots:   0 total
Time:        7.06 s, estimated 9 s
Ran all test suites matching /user.service.spec.ts/i.
```

- **user.repository.spec.ts**

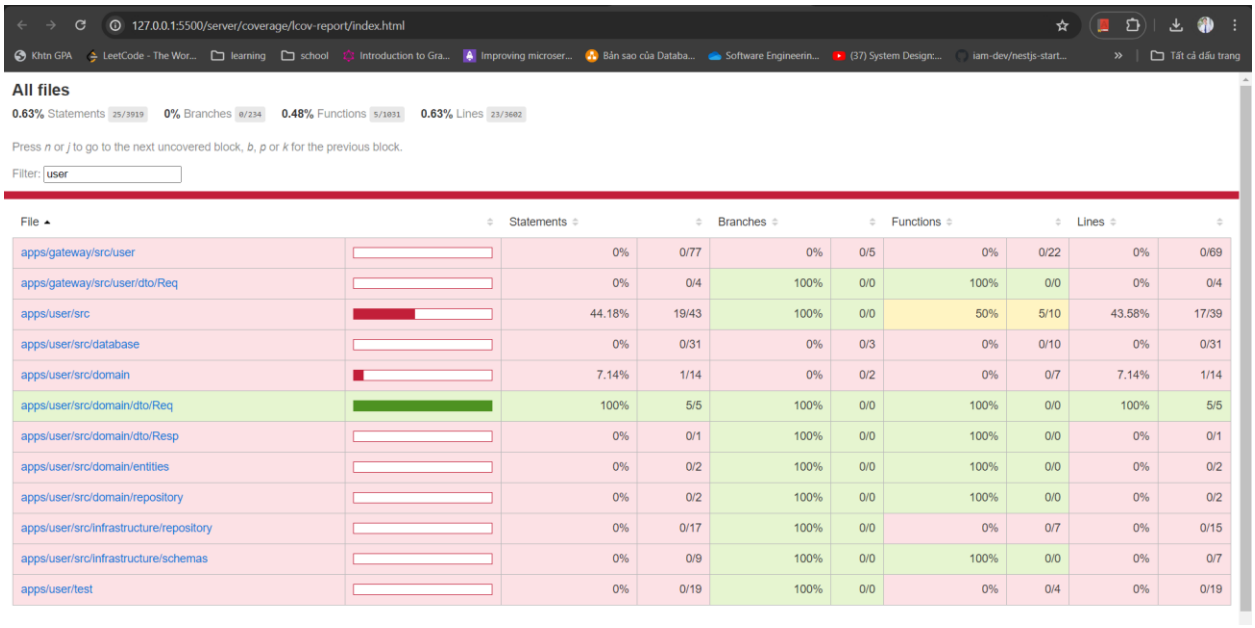
```
> server@0.0.1 test C:\Users\Admin\OneDrive - VNU-HCMUS\Nam4\DoAnCNPM\DACNPM\server
> jest "user.repository.spec.ts"

 PASS  apps/user/src/infrastructure/_test_/user.repository.spec.ts (6.133 s)
  TypeOrmUserRepository
    findById
      √ should return the user with the given id (14 ms)
      √ should call the userRepository.findOne() method with the correct query (3 ms)
    findAll
      √ should return all users (2 ms)
      √ should call the userRepository.find() method (2 ms)
    createUser
      √ should return the created user (3 ms)
      √ should call the userRepository.save() method with the correct user (2 ms)
    isUserExist
      √ should return true if the user exists (2 ms)
      √ should return false if the user does not exist (8 ms)

Test Suites: 1 passed, 1 total
Tests:       8 passed, 8 total
Snapshots:   0 total
Time:        6.319 s, estimated 8 s
Ran all test suites matching /user.repository.spec.ts/i.
```

Bên cạnh đó, có thể chạy lệnh:

pnpm test:cov

để kiểm tra độ bao phủ của các kiểm thử với mã nguồn

Báo cáo coverage sẽ được lưu trong thư mục **coverage/** và có thể mở file **index.html** trong thư mục đó để xem báo cáo chi tiết.



d) Kết quả khảo sát

Có thể thu thập và phân tích dữ liệu từ báo cáo coverage và kết quả kiểm thử hàng ngày. Một số yếu tố cần khảo sát bao gồm:

Tỷ lệ kiểm thử thành công: Số lượng bài kiểm thử thành công so với tổng số bài kiểm thử.
Tỷ lệ độ bao phủ mã nguồn: Tỷ lệ phần trăm mã nguồn được kiểm thử.
Thời gian chạy kiểm thử: Thời gian trung bình để thực hiện các bài kiểm thử.
Từ đó có thể đánh giá chất lượng kiểm thử và cải thiện các phương pháp kiểm thử của mình.

e) So sánh với các hệ thống tương tự

	TopCV của nhóm	**TopCV**	**ITViec**	**MyCV**
Ngành nghề tập trung	Đa dạng	Đa dạng	Công nghệ thông tin	Đa dạng
Tính năng tạo CV	Chỉ có 1 mẫu CV	Cực kì nhiều mẫu CV	Có 4 mẫu CV	Gồm 14 mẫu CV

Tính năng tìm kiếm việc làm	Tốt	Tốt, nhiều ngành nghề	Hệ thống tìm kiếm được chuyên môn hoá cho ngành IT	Không có
Chi phí	Miễn phí	Miễn phí và trả phí	Miễn phí, tính phí với nhà tuyển dụng	Miễn phí và trả phí
Độ phổ biến	Không có	Rất phổ biến	Phổ biến trong ngành IT	Không quá phổ biến
Ưu điểm	Có thể tạo CV, hỗ trợ các tính năng cơ bản	Trang mang đến nguồn ứng viên chất lượng và đa dạng dành cho nhà tuyển dụng	Giao diện đơn giản, chuyên nghiệp, hỗ trợ miễn phí các mẫu CV	Giao diện đơn giản, dễ sử dụng
Nhược điểm	Còn thiếu nhiều tính năng. Mẫu CV còn quá ít	Trang cho phép đăng tin tuyển dụng miễn phí ở mức độ thấp. Hầu hết nhà tuyển dụng đều phải trả phí cho việc đăng tin. Các ứng viên hầu hết đều là sinh viên chưa có kinh nghiệm và chuyên môn.	Chỉ phục vụ cho thị trường IT. Hơi ít mẫu CV	Chỉ có 2 mẫu CV là miễn phí còn lại phải trả phí. Quá ít chức năng

f) Video demo unit testing

Link youtube: https://youtu.be/BJRSbgcHHPs

II. Load test

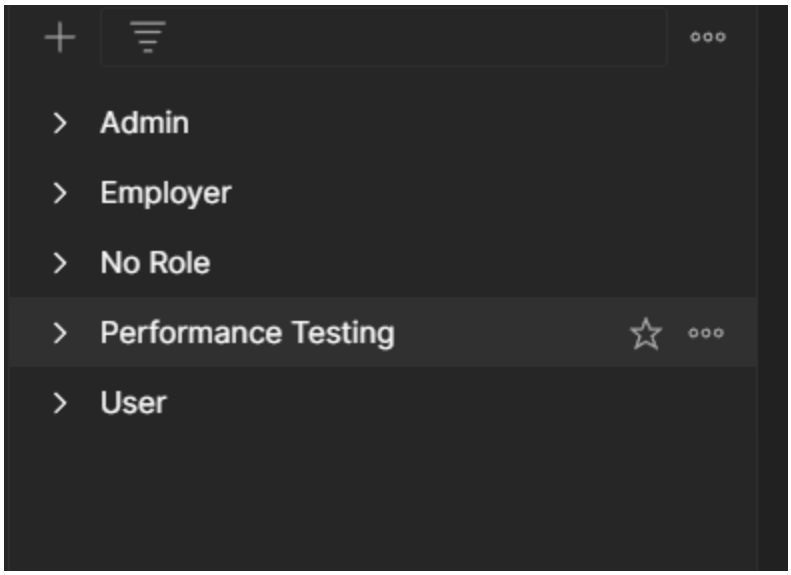
a) Đăng ký và cài đặt

Sử dụng postman để load test

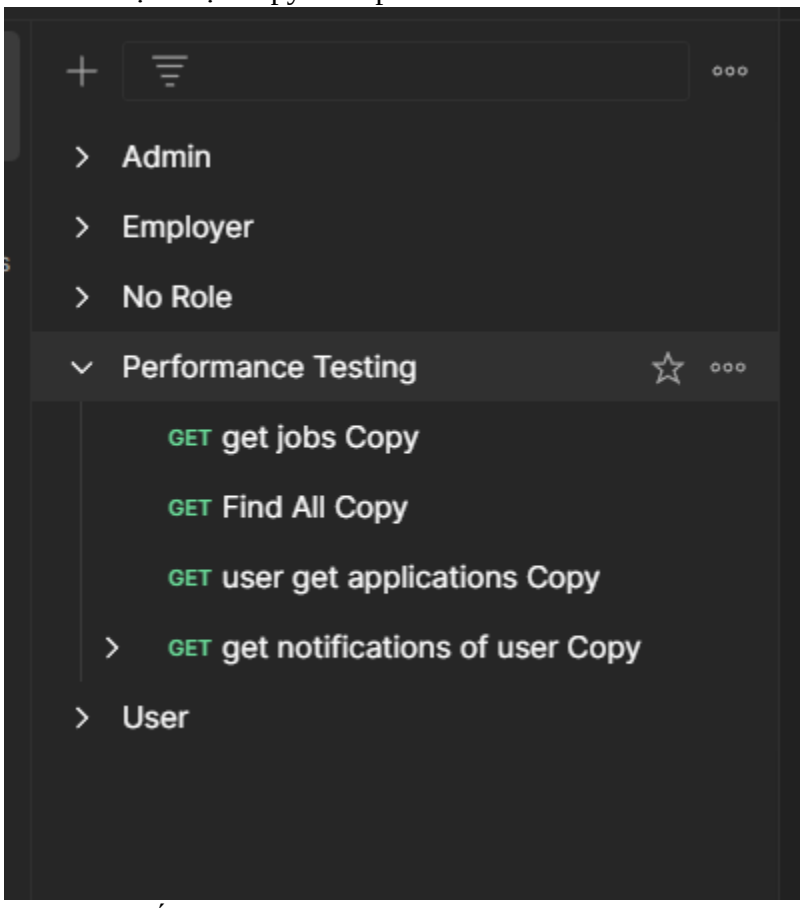
Link collection: https://www.postman.com/martian-star-52215/workspace/authorized-api

b) Phương pháp thực thi

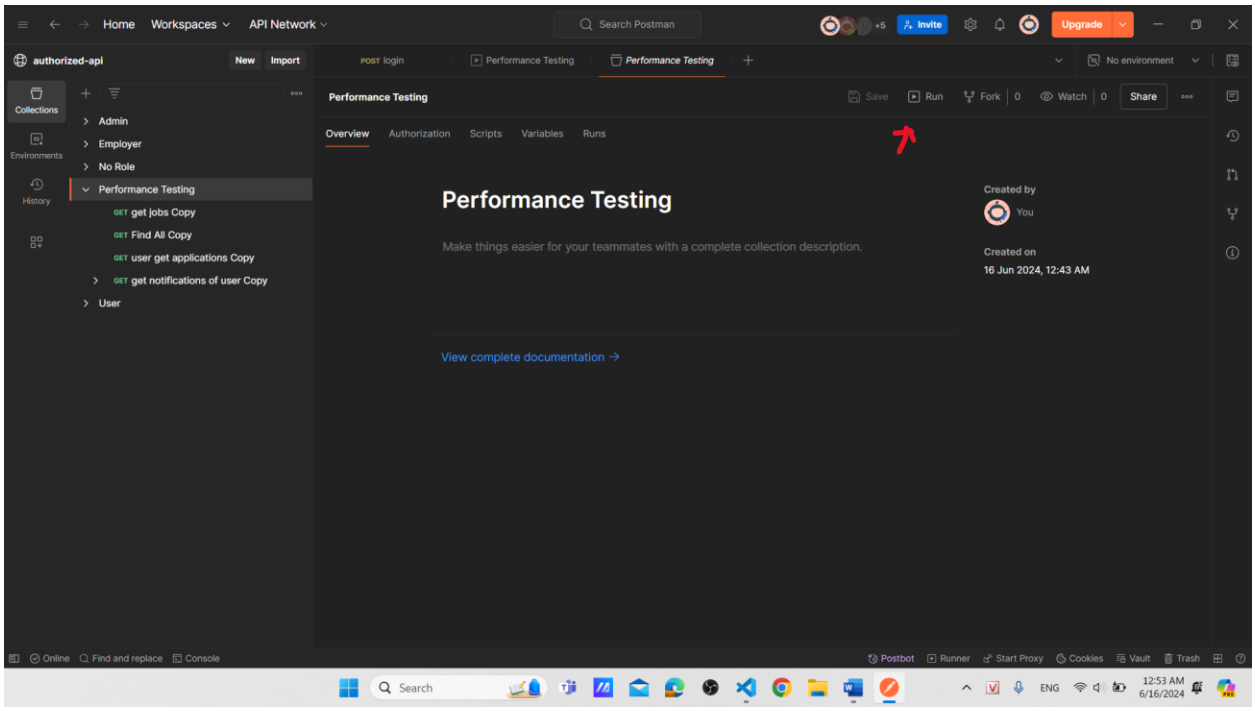
- Bước 1: Tạo folder Performance Test trên postman



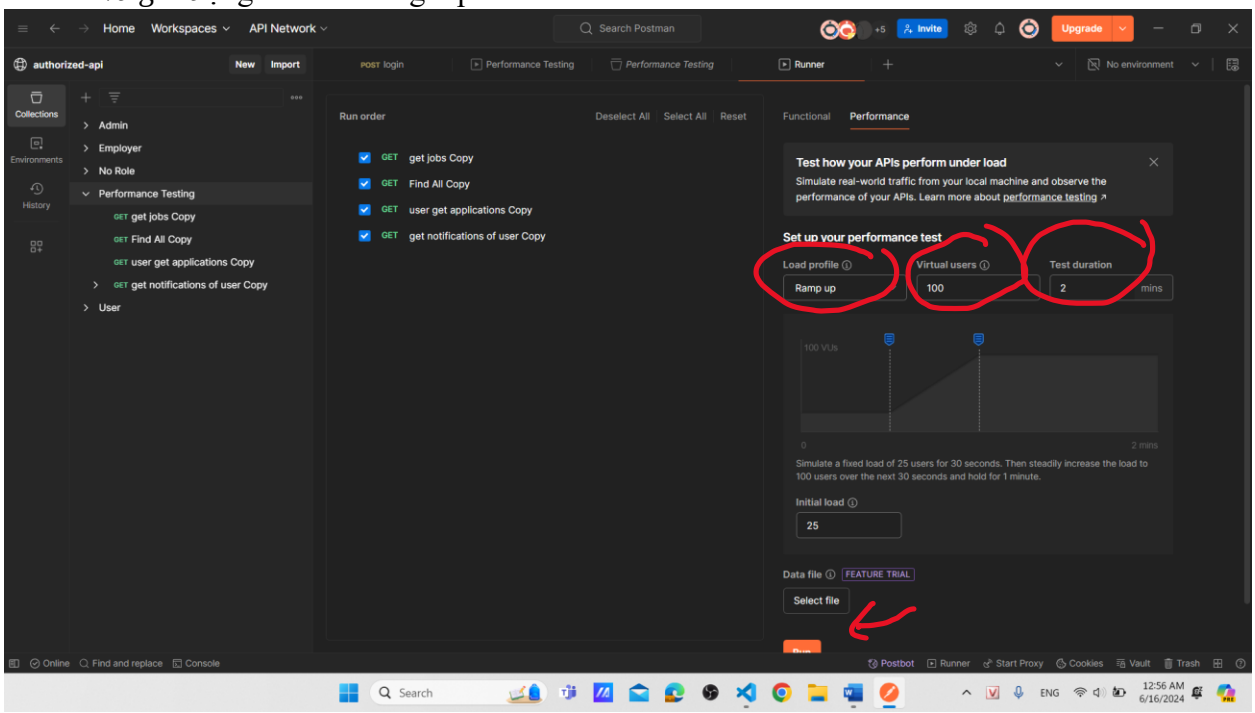
❖ Bước 2: Tạo hoặc copy các api cần test vào folder Performance Testing



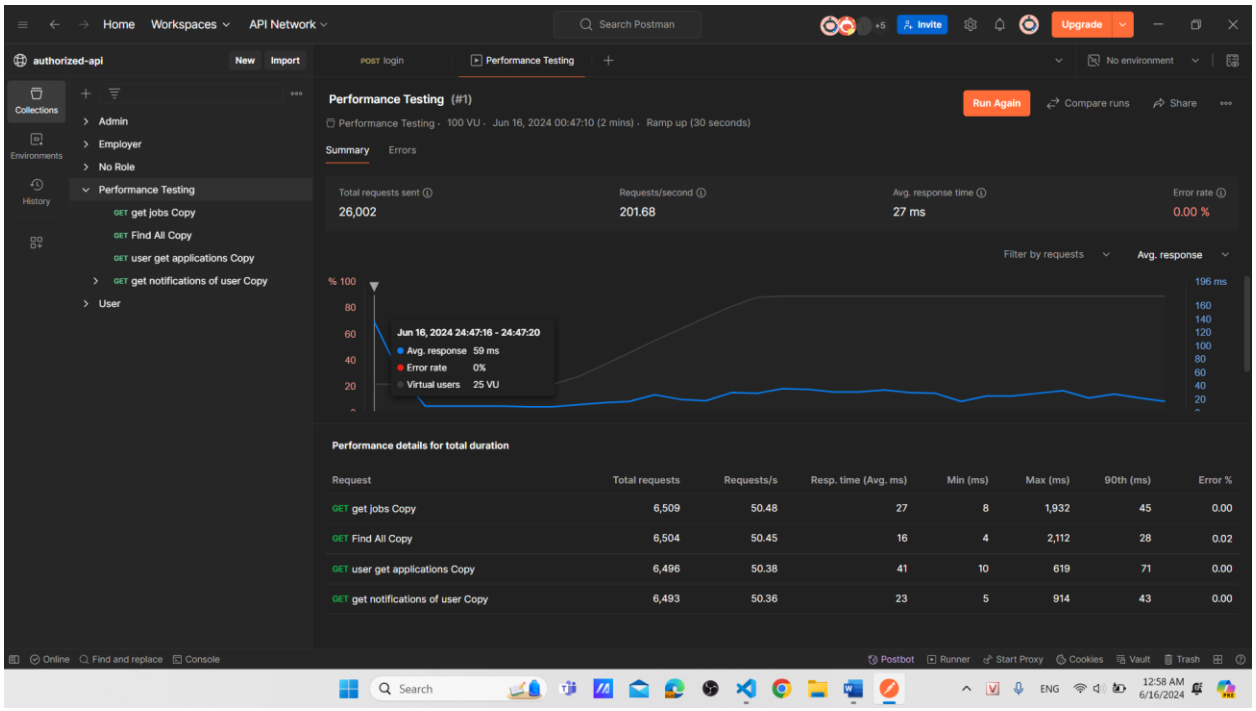
❖ Bước 3: Nhấn vào Run



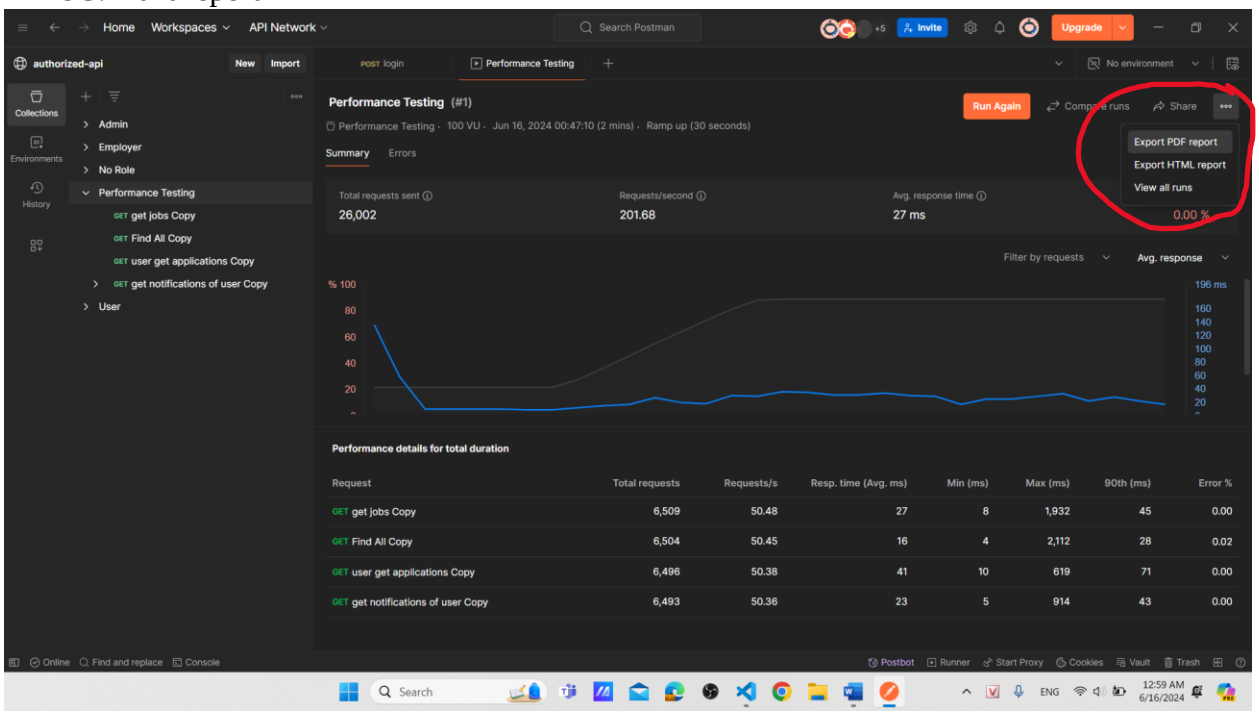
- Bước 4: Cài đặt thông số
  - Load profile: Ramp up (số lượng user truy cập tăng dần cho đến tối đa), ở đây khởi đầu với 25 user trong 30s và tăng dần đến 100 user trong 30s kế tiếp và giữ trạng thái đó trong 1 phút cuối



Màn hình sau khi hoàn thành load test:



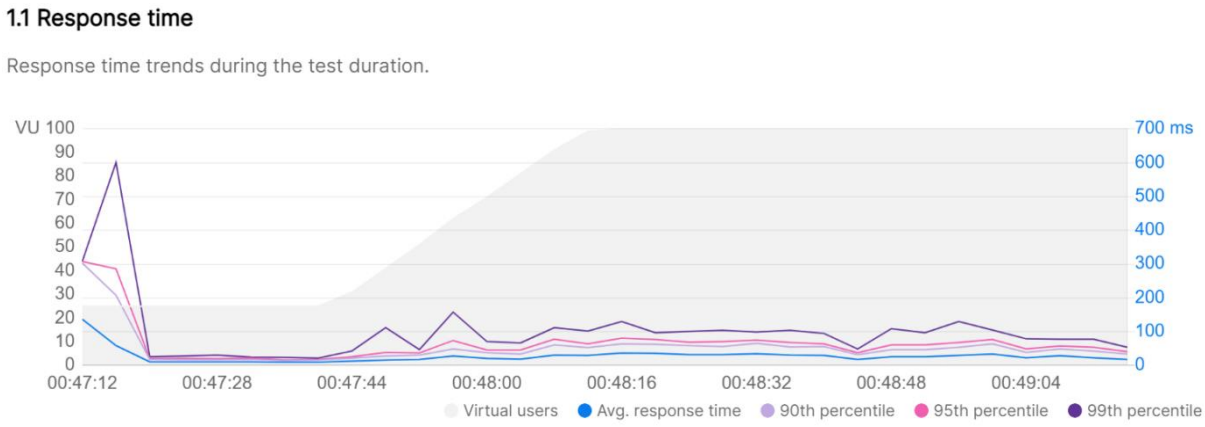
❖ Bước 5: Xuất report



c) Kết quả thu được

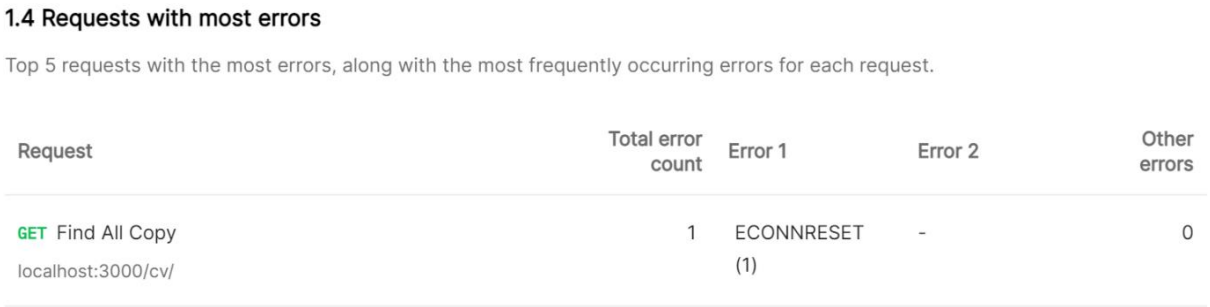
Với số lượng user là 100 và thời gian test là 2 phút thì web vẫn đáp ứng tốt , không xảy ra các tình trạng lỗi hay error , tốc độ phản hồi nhanh (27ms)

❖ Thời gian phản hồi



Thời gian phản hồi ổn định, khi số lượng request tăng thì web vẫn ổn định, không xảy ra tình trạng sập hay phản hồi chậm

- Lỗi



Chỉ gặp 1 lỗi duy nhất là ECONNRESET đối với CV service

Tổng kết lại, đối với số lượng user là 100 với số lượng request là 26000 thì hệ thống vẫn đáp ứng được với độ phản hồi nhanh và không xày ra lỗi nghiêm trọng

d) Video demo

Link youtube: https://youtu.be/cxP5xPsfxZE

III. Stress test

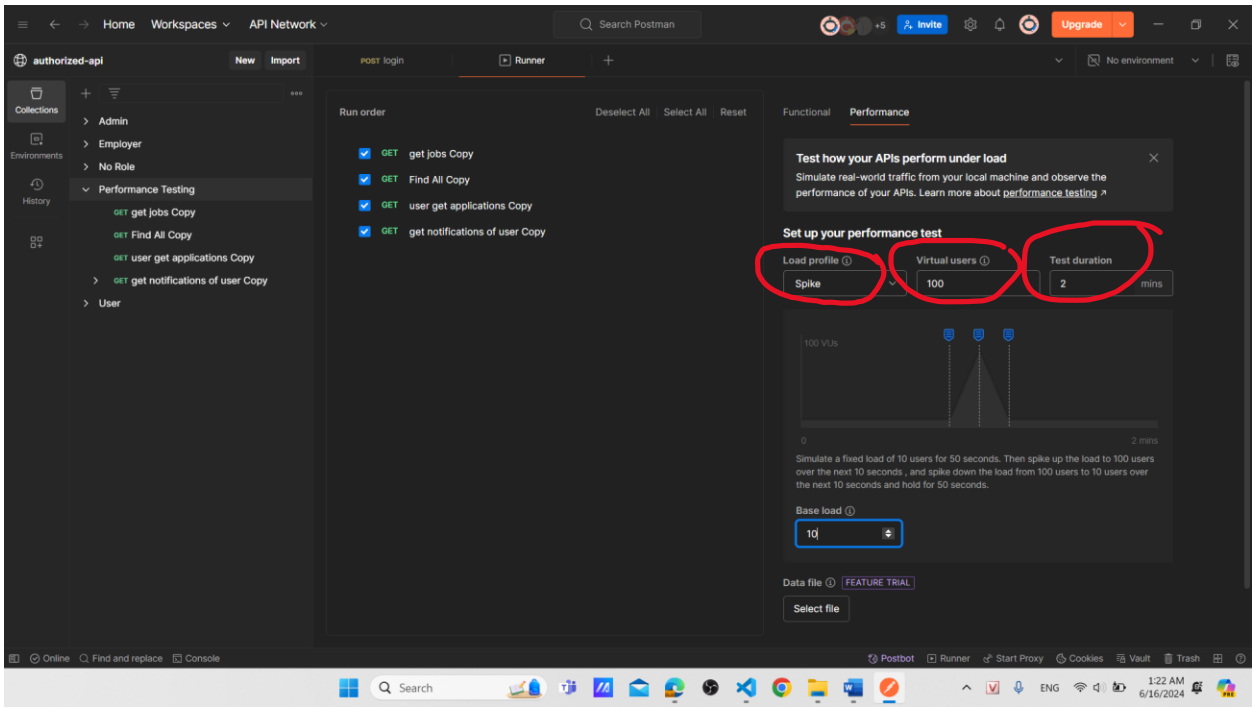
a) Đăng ký và cài đặt

Sử dụng postman để stress test

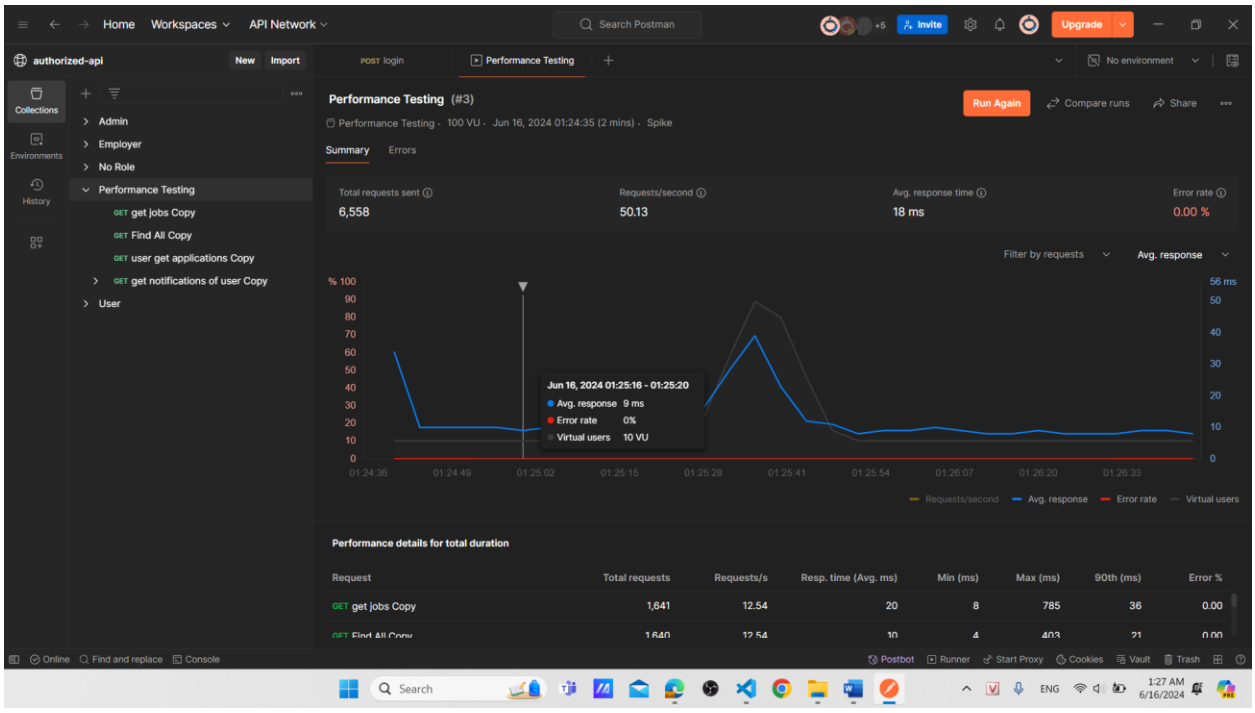
Link collection: https://www.postman.com/martian-star-52215/workspace/authorized-api

b) Phương pháp thực thi

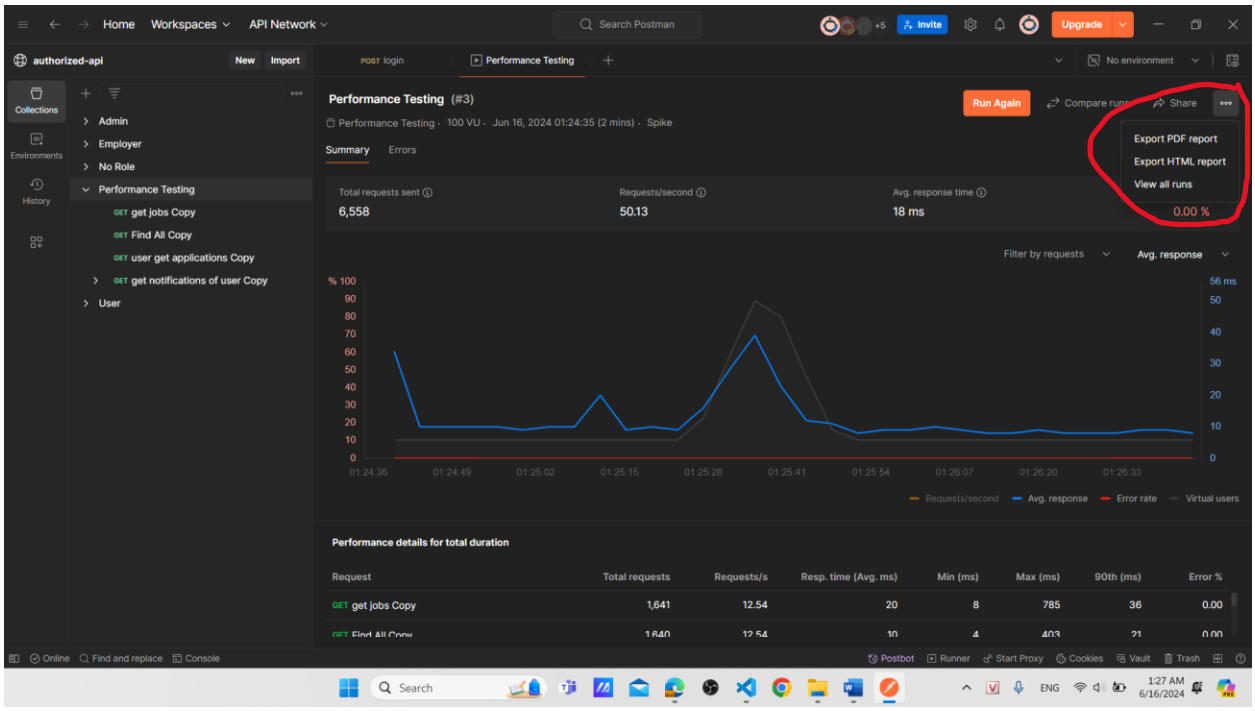
- Bước 1: Do chúng ta đã có sẵn folder Performance Testing nên kế tiếp chúng ta sẽ run với thông số như sau:
  - Load profile: Spike (số lượng user tăng và giảm đột ngột), ở đây với khởi đầu có 10 user truy cập và đột ngột tăng lên 100 user trong 10s và giảm về 10 user trong 10s kế tiếp , duy trì trong 50s cuối cùng



Màn hình sau khi stress test hoàn thành:

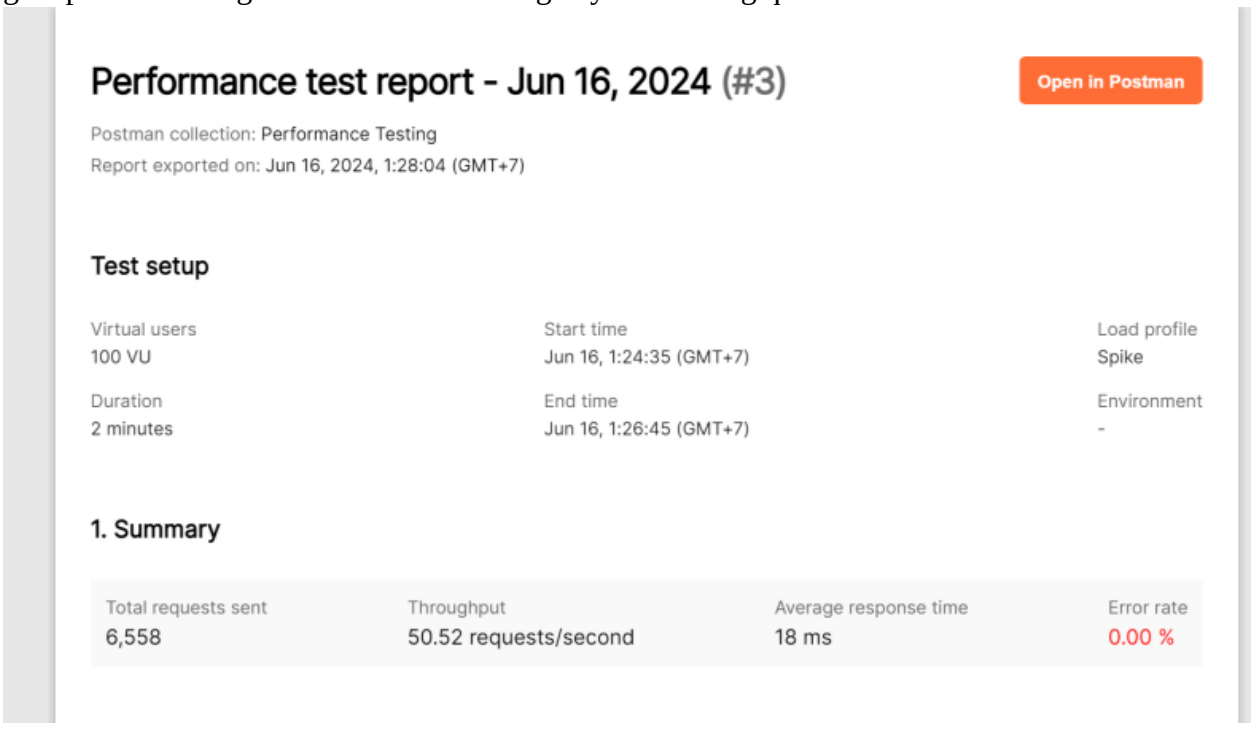


❖ Bước 2: Xuất ra report

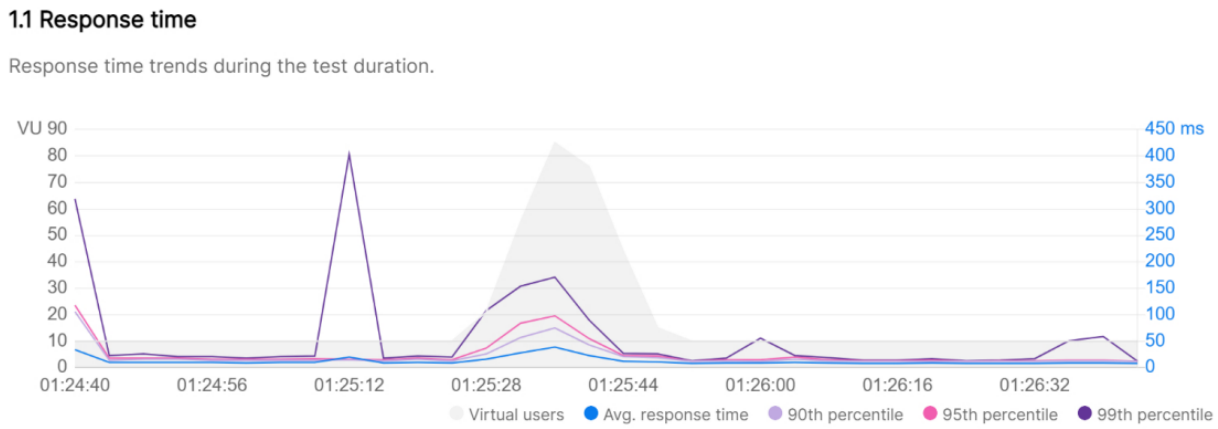


c) Kết quả thu được

Với số lượng user là 100 và thời gian test là 2 phút thì hệ thống vẫn đáp ứng tốt với thời gian phản hồi trung bình là 18ms và không xảy ra lỗi trong quá trình test

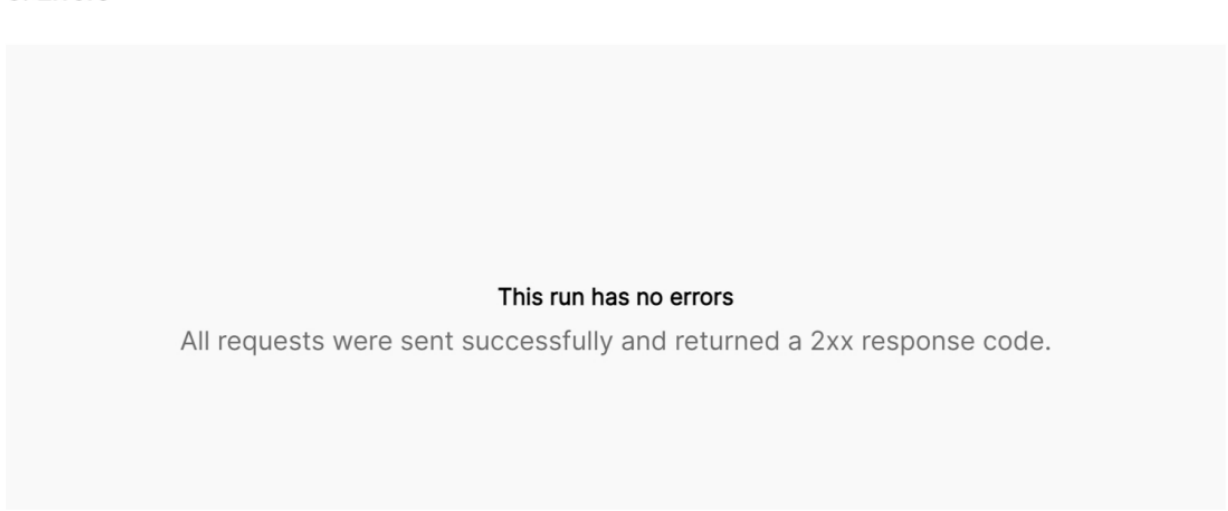


❖ Thời gian phản hồi



Thời gian phản hồi tăng nhẹ khi số lượng request tăng đột xuất, nhưng xuất hiện sự tăng mạnh bất thường trước lúc số lượng user tăng lên, có thể do áp lực hệ thống hoặc xuất hiện 1 sự kiện bất thường

❖ **Lỗi**



Không xuất hiện lỗi

Tổng kết lại, đối với 100 user , khi số lượng request tăng đột ngột vào 1 thời điểm thì hệ thống vẫn đáp ứng được với thời gian phản hồi nhanh và ổn định

d) Video demo

Link youtube: https://youtu.be/JNJ7JZaNs_8