**ĐỒ ÁN CUỐI KÌ N2\_CD21CT6**

1. Các thành viên

|  |  |  |
| --- | --- | --- |
| STT | Họ và Tên | MSSV |
| 1 | Trần Thanh Phú Em | 501210175 |
| 2 | Nguyễn Thành Luân | 503210052 |
| 3 | Đàm Minh Hoàng | 501210053 |
| 4 | Trần Ngọc Dương | 501210388 |
| 5 | Phan Kim Ngân | 501210586 |

2. Các link của đồ án

Link Google Drive

<https://drive.google.com/drive/folders/1MDZcG5nRjry7cnMgrkrHT5oWwWHJCU4p?usp=sharing>

Link Github

<https://github.com/T2PhuEm01/DACK_N2_CT6.git>

Link Postman

<https://documenter.getpostman.com/view/25657043/2s93Jxrgsn>

3. Kết luận, nhận xét về kết quả đạt được cũng như giải pháp đề xuất để nâng cấp hệ thống

* Dựa trên các thông số và kết quả thu được từ các Listener sau khi chạy jmeter với cấu hình cho thread group (users) là 10/100/1000, Ram up là 10, loop là 1, ta có thể đưa ra các kết luận và nhận xét như sau:
* View Results Tree:

Thông qua View Results Tree, ta có thể xem lại các request và response khi chạy test. Nếu xuất hiện các lỗi như timeout, lỗi kết nối hay lỗi xử lý dữ liệu thì đây là các điểm cần được chú ý và cần phải tìm hiểu nguyên nhân để khắc phục.

* Graph Results:

Graph Results giúp ta theo dõi sự biến động của các thông số như thời gian phản hồi, số request/second, số thread và tài nguyên sử dụng. Từ đó, ta có thể phát hiện các vấn đề về hiệu năng của hệ thống như quá tải, tài nguyên sử dụng quá mức, thời gian phản hồi quá lâu. Với kết quả này, ta có thể đưa ra giải pháp để nâng cấp hiệu năng của hệ thống.

* Aggregate Report:

Aggregate Report là bảng tổng hợp kết quả chạy test với các thông số như số request thành công, thời gian phản hồi trung bình, số request/second và thời gian xử lý. Từ đây, ta có thể phân tích đánh giá hiệu năng của hệ thống, tìm ra các bottleneck và đưa ra giải pháp cải thiện hiệu năng.

* Summary Report: Cung cấp báo cáo về các giá trị: thời gian phản hồi thấp nhất/cao nhất, số yêu cầu xảy ra lỗi, lưu lượng trung bình.

Để nâng cấp hệ thống, ta có thể áp dụng các giải pháp như:

- Tối ưu hóa cấu hình máy chủ, tăng tài nguyên và tăng số lượng thread để đáp ứng với số lượng người dùng tăng lên.

- Tối ưu hóa mã nguồn để giảm thời gian xử lý và tối đa hóa tài nguyên sử dụng.

- Sử dụng các công nghệ mới để tăng hiệu năng và tăng khả năng mở rộng của hệ thống.

- Tăng số lượng server và phân chia tải để tăng sức chứa của hệ thống.