BỘ MÔN HỆ THỐNG THÔNG TIN – KHOA CÔNG NGHỆ THÔNG TIN

ĐẠI HỌC KHOA HỌC TỰ NHIÊN THÀNH PHỐ HỒ CHÍ MINH, ĐẠI HỌC QUỐC GIA TP HCM

MÔN ỨNG DỤNG PHÂN TÁN

Giáo viên phụ trách: Phạm Minh Tú

Sinh viên thực hiện: Nhóm 06

ĐỒ ÁN MÔN HỌC - ỨNG DỤNG PHÂN TÁN

HỌC KỲ II – NĂM HỌC 2021-2022

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
227 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM
Điện Thoại: (08) 38.354.266 - Fax:(08) 38.350.096

4.0

BẢNG THÔNG TIN CHI TIẾT NHÓM

Mã nhóm:	N06			
Tên nhóm:	Nhóm 6			
Số lượng:	**3**			
MSSV	**Họ tên**	**Email**	**Điện thoại**	**Hình ảnh**
18120387	Trần Hữu Hoàng	18120387@student.hcmus.edu.vn		
18120401	Mai Khánh Huyền	18120401@student.hcmus.edu.vn		
18120418	Phạm Minh Khoa	18120418@student.hcmus.edu.vn		

Bảng phân công & đánh giá hoàn thành công việc				
Giai đoạn	**Công việc thực hiện**	**Người thực hiện**	**Mức độ hoàn thành**	**Đánh giá của nhóm**
Giai đoạn 1	Tìm chức năng	18120387-Trần Hữu Hoàng	100%	10/10
	Tìm chức năng	18120401-Mai Khánh Huyền	100%	10/10
	Tìm chức năng	18120418-Phạm Minh Khoa	100%	10/10
Giai đoạn 2		18120387-Trần Hữu Hoàng	100%	10/10
	Thiết kế database, thiết kế service	18120401-Mai Khánh Huyền	100%	10/10
	Thiết kế giao diện	18120418-Phạm Minh Khoa	100%	10/10
Giai đoạn 3	Thiết kế service, coding	18120387-Trần Hữu Hoàng	100%	10/10
	Thiết kế giao diện, coding	18120401-Mai Khánh Huyền	100%	10/10
	Thiết kế service, coding	18120418-Phạm Minh Khoa	100%	10/10

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
227 Nguyễn Văn Cừ, Phường 4, Quận 5, TP.HCM
Điện Thoại: (08) 38.354.266 - Fax:(08) 38.350.096

4.0

YÊU CẦU ĐỒ ÁN- BÀI TẬP

Loại bài tập	**□ Lý thuyết þ Thực hành þ Đồ án □ Bài tập □ Seminar**
Ngày bắt đầu	**16/3/2022 (tuần 2)**
Ngày kết thúc	**9/7/2022**

Mục Lục

A. Yêu cầu của Seminar

- Mô tả trong file pdf. Link: https://courses.fit.hcmus.edu.vn/pluginfile.php/175677/mod_resource/content/1/%5BUDPT%5D%20%5BDo%20An%5DDi%20cho%20online.pdf

ĐỒ ÁN MÔN HỌC

Ứng dụng chợ online

1. Giới thiệu dự án

Một công ty đầu tư thương mại ABC muốn xây dựng hệ thống kết nối giữa bên mua và bên bán các mặt hàng thiết yếu. Trong bối cảnh dịch bệnh diễn biến phức tạp, người dùng có nhu cầu mua các mặt hàng sản phẩm thiết yếu như thịt, cá, rau, củ,…. Những thực phẩm này được cung cấp bởi các nhà cung cấp uy tín trên thị trường.

Công ty ABC là bên đầu tư hệ thống phần mềm để quản lý các dịch vụ này, mục tiêu là cung cấp dịch vụ tốt nhất cho người mua và người bán, công ty muốn xây dựng và quản lý hệ thống cung cấp các dịch vụ như đăng kí bán hàng, đăng ký mua hàng, giao hàng, và các dịch vụ khác. Công ty sẽ thu phí dựa trên số lượng hóa đơn phát sinh, doanh thu chủ yếu công ty sẽ đến từ hoạt động chính này. Phần tiếp theo sẽ mô tả các quy trình trong dự án.

2. Quy trình đăng ký bán hàng

Các bên bán hàng mong muốn đăng kí bán hàng cần phải thỏa các tiêu chí sau:

- Giấy phép kinh doanh
- Có chứng nhận an toàn thực phẩm
- Các sản phẩm có nguồn gốc xuất xứ rõ ràng, như nhập từ đâu, thời gian nào

Bên bán phải cung cấp các giấy tờ này và được kiểm tra xét duyệt bởi công ty ABC. Sau khi hoàn thành, bên bán phải trả lời các câu hỏi cam kết liên quan đến tiêu chuẩn bán hàng như: Của hàng bạn phải có nhân viên chăm sóc khách hàng. Cửa hàng bạn phải có thông báo công ty ABC khi có thay đổi địa chỉ kinh doanh,……

Sau khi bên bán hoàn thành tất cả các cam kết, các thông tin sau sẽ được cung cấp đầy đủ như:

- Tên cửa hàng

- Địa chỉ kinh doanh
- Loại mặt hàng: nông sản, thịt-cá,….
- Số điện thoại liên hệ
- Tên nhân viên chăm sóc khách hàng
- Logo của hàng (nếu có)
- Ngày giờ mở cửa
- Ngày thành lập

Sau khi cung cấp thông tin, công ty ABC sẽ kiểm duyệt, thông báo lập hợp đồng và kích hoạt hệ thống để bên bán có thể bán sản phẩm.

3. Quy trình bán hàng

Cửa hàng có thể đăng bán các sản phẩm bằng cách thêm các sản phẩm trên hệ thống như: tên sản phẩm, loại, quy cách, hình ảnh, giá tiền,…….

Cửa hàng sau đó sẽ quản lý các sản phẩm này qua hệ thống như thêm, cập nhật, xem bình luận,…

4. Quy trình mua hàng

Bên mua (khách hàng) có thể xem sản phẩm trên hệ thống, chọn và đặt hàng. Đơn hàng sẽ được chuyển cho cửa hàng kiểm tra và đóng gói cẩn thận. Bất cứ sai sót nào liên quan đến khâu đóng gói, cửa hàng chịu hoàn toàn trách nhiệm. Sau khi đóng gói cửa hàng thông báo bộ phận giao hàng của công ty ABC tiến hành giao hàng. Cứ mỗi đơn hàng giao thành công, công ty ABC sẽ thu phí 5% và khoản này được thanh toán ngay khi đơn hàng được giao hoặc thanh toán theo tháng tùy vào hợp đồng mà cửa hàng kí kết với ABC.

5. Quy trình giao hàng

Bộ phận giao hàng đến cửa hàng sẽ kiểm tra gói hàng bên ngoài xem có gì bất thường hay không, nếu không sẽ tiến hành giao hàng và thông báo trạng thái (đang giao, đã giao, khách hàng từ chối nhận hàng,…..).

6. Chăm sóc khách hàng

Nếu có bất cứ thắc mắc gì về gói hàng, khách hàng liên hệ cửa hàng. Nhân viên chăm sóc khách hàng sẽ giải đáp thắc mắc qua online hoặc điện thoại khách hàng.

7. Yêu cầu chức năng

Mã	UC001

Tên chức năng	Đăng ký thông tin bán hàng
Tóm tắt	Khi khách hàng muốn đăng kí thông tin bán hàng, khách hàng có khả năng đăng kí trên hệ thống.
Lý do	Cửa hàng cần cung cấp hồ sơ để trở thành người bán hàng trên hệ thống
Yêu cầu chi tiết	Khách hàng đăng kí bán hàng phải cung cấp hồ sơ như quy trình 2, các hồ sơ này gửi thông qua hệ thống. Sau đó khách hàng xác nhận thông tin các câu hỏi cam kết, điền thông tin chi tiết cửa hàng. Trạng thái sau khi đăng kí là **chờ xác nhận**. Sau khi xác nhận **thành công** công ty ABC sẽ gửi hợp đồng về địa chỉ của hàng, người có trách nhiệm sẽ kí tên và gửi lại bản cứng cho công ty ABC qua địa chỉ mà công ty ABC cung cấp.

-

Mã	UC002
Tên chức năng	Xét duyệt hồ sơ
Tóm tắt	Xét duyệt hồ sơ đăng kí bán hàng từ khách hàng
Lý do	Công ty ABC cần xét duyệt hồ sơ của cửa hàng để đảm bảo tính pháp lý của hồ sơ, đảm bảo cửa hàng được xác thực của tiêu chuẩn của công ty ABC
Yêu cầu chi tiết	Nhân viên tiếp nhận hồ sơ trên hệ thống, kiểm tra hồ sơ, xác nhận và xuất hợp đồng gửi email về khách hàng, sau khi khách hàng gửi hồ sơ đã kí về công ty, nhân viên tiếp nhận duyệt hồ sơ hoàn thành và phát sinh tài khoản cho cửa hàng.

Mã	UC003
Tên chức năng	Quản lý tài khoản cửa hàng

Tóm tắt	Cửa hàng quản lý thông tin tài khoản
Lý do	Cửa hàng cần cập nhật thông tin nếu có như thay đổi mật khẩu được cấp tự động, thông tin hình ảnh, logo
Yêu cầu chi tiết	Nhân viên cửa hàng có thể dùng hệ thống để thay đổi thông tin cửa hàng, những thông tin không thể chỉnh sửa như địa chỉ cửa hàng, thông tin này chỉ có thể thay đổi khi cửa hàng có giấy xác nhận công ty di chuyển qua địa chỉ mới và phải thông báo cho công ty ABC trước 15 ngày.

Mã	UC004
Tên chức năng	Quản lý sản phẩm
Tóm tắt	Cửa hàng quản lý thông tin sản phẩm bào gồm thêm mới, cập nhật, tìm kiếm và trạng thái tồn tại hay không tồn tại.
Lý do	Cửa hàng cần quản lý những sản phẩm cần bán cho khách hàng, thay đổi thông tin cần thiết và xem bình luận sản phẩm.

Yêu cầu chi tiết	Nhân viên cửa hàng upload danh sách sản phẩm mà cửa hàng cần bán, danh sách sản phẩm được lưu file excel với template có sẵn tại hệ thống. Hệ thống cũng cho phép nhập thông tin từng sản phẩm. Khi cần update thông tin sản phẩm, nhân viên có thể thay đổi danh sách trong excel và upload lại, dữ liệu mới sẽ ghi đè lên dữ liệu cũ.

Mã	UC005

Tên chức năng	Quản lý đơn hàng
Tóm tắt	Khách hàng là người có yêu cầu mua hàng từ hệ thống
Lý do	Khách hàng đi chợ online được hệ thống hỗ trợ mua hàng online, đặt hàng và xem trạng thái giao hàng
Yêu cầu chi tiết	Khách hàng có thể xem sản phẩm, tìm kiếm, chọn và lưu vào giỏ hàng, nhập các thông tin cần thiết như họ tên, địa chỉ giao, số điện thoại được xác minh,… Sau đó đơn hàng được lập và gửi đến cửa hàng. Nhân viên cửa hàng kiểm tra sản phẩm, đóng gói và chuyển đến bộ phận giao hàng. Bộ phận giao hàng kiểm tra địa chỉ, liên hệ khách hàng và tiến hành giao hàng. Các trạng thái đơn hàng được cập nhật đầy đủ như: tiếp nhận, đóng gói, chuẩn bị giao hàng, đang giao hàng, giao hàng thành công,…

8. Yêu cầu

Các mốc quan trọng

A. Các nhóm dựa trên mô tả quy trình đồ án, các chức năng gợi ý , phát triển thêm các yêu cầu chức năng và yêu cầu phi chức năng **(19/03 -> 16/04)**
B. Thiết kế CSDL, front-end, API **(17/04 -> 17/05)**
C. Tích hợp các chức năng và kiểm thử, vấn đáp đồ án thực hành **(18/05 -> 20/06)**

Yêu cầu website: PHP, HTML, JS, CSS dùng để xây dựng trang web với các chức năng cơ bản như: login, logout, thông tin liên hệ,…

Yêu cầu API: Java framework dùng để xây dựng API, trả dữ liệu về bên website, giao tiếp với API theo hướng đồng bộ. Các API cũng trao đổi message với nhau theo hướng bất đồng bộ phụ thuộc vào chức năng. Cần có API gateway để quản lý nhiều API khác trong trường hợp một chức năng gọi trên 3 API.

B. Kết quả

1. Giai đoạn 1

1.1. Chức năng

Mã	UC001
Phân hệ	Cửa hàng
Tên chức năng	Đăng ký thông tin bán hàng
Tóm tắt	Khi khách hàng muốn đăng kí thông tin bán hàng, khách hàng có khả năng đăng kí trên hệ thống.
Lý do	Cửa hàng cần cung cấp hồ sơ để trở thành người bán hàng trên hệ thống
Yêu cầu chi tiết	Khách hàng đăng kí bán hàng phải cung cấp hồ sơ như quy trình 2, các hồ sơ này gửi thông qua hệ thống. Sau đó khách hàng xác nhận thông tin các câu hỏi cam kết, điền thông tin chi tiết cửa hàng. Trạng thái sau khi đăng kí là **chờ xác nhận**. Sau khi xác nhận **thành công** công ty ABC sẽ gửi hợp đồng về địa chỉ của hàng, người có trách nhiệm sẽ kí tên và gửi lại bản cứng cho công ty ABC qua địa chỉ mà công ty ABC cung cấp.

Mã	UC002
Phân hệ	Admin
Tên chức năng	Xét duyệt hồ sơ
Tóm tắt	Xét duyệt hồ sơ đăng kí bán hàng từ khách hàng
Lý do	Công ty ABC cần xét duyệt hồ sơ của cửa hàng để đảm bảo tính pháp lý của hồ sơ, đảm bảo cửa hàng được xác thực của tiêu chuẩn của công ty ABC
Yêu cầu chi tiết	Nhân viên tiếp nhận hồ sơ trên hệ thống, kiểm tra hồ sơ, xác nhận và xuất hợp đồng gửi email về khách hàng, sau khi khách hàng gửi hồ sơ đã kí về công ty, nhân viên tiếp nhận duyệt hồ sơ hoàn thành và phát sinh tài khoản cho cửa hàng.

Mã	UC003
Phân hệ	Cửa hàng
Tên chức năng	Quản lý tài khoản cửa hàng
Tóm tắt	Cửa hàng quản lý thông tin tài khoản
Lý do	Cửa hàng cần cập nhật thông tin nếu có như thay đổi mật khẩu được cấp tự động, thông tin hình ảnh, logo
Yêu cầu chi tiết	Nhân viên cửa hàng có thể dùng hệ thống để thay đổi thông tin cửa hàng, những thông tin không thể chỉnh sửa như địa chỉ cửa hàng, thông tin này chỉ có thể

	thay đổi khi cửa hàng có giấy xác nhận công ty di chuyển qua địa chỉ mới và phải thông báo cho công ty ABC trước 15 ngày.

Mã	UC004
Phân hệ	Cửa hàng
Tên chức năng	Quản lý sản phẩm
Tóm tắt	Cửa hàng quản lý thông tin sản phẩm bao gồm thêm mới, cập nhật, tìm kiếm và trạng thái tồn tại hay không tồn tại.
Lý do	Cửa hàng cần quản lý những sản phẩm cần bán cho khách hàng, thay đổi thông tin cần thiết và xem bình luận sản phẩm.
Yêu cầu chi tiết	Nhân viên cửa hàng upload danh sách sản phẩm mà cửa hàng cần bán, danh sách sản phẩm được lưu file excel với template có sẵn tại hệ thống. Hệ thống cũng cho phép nhập thông tin từng sản phẩm. Khi cần update thông tin sản phẩm, nhân viên có thể thay đổi danh sách trong excel và upload lại, dữ liệu mới sẽ ghi đè lên dữ liệu cũ.

Mã	UC005
Phân hệ	Khách hàng
Tên chức năng	Quản lý đơn hàng
Tóm tắt	Khách hàng là người có yêu cầu mua hàng từ hệ thống
Lý do	Khách hàng đi chợ online được hệ thống hỗ trợ mua hàng online, đặt hàng và xem trạng thái giao hàng
Yêu cầu chi tiết	Khách hàng có thể xem sản phẩm, tìm kiếm, chọn và lưu vào giỏ hàng, nhập các thông tin cần thiết như họ tên, địa chỉ giao, số điện thoại được xác minh,… Sau đó đơn hàng được lập và gửi đến cửa hàng. Nhân viên cửa hàng kiểm tra sản phẩm, đóng gói và chuyển đến bộ phận giao hàng. Bộ phận giao hàng kiểm tra địa chỉ, liên hệ khách hàng và tiến hành giao hàng. Các trạng thái đơn hàng được cập nhật đầy đủ như: tiếp nhận, đóng gói, chuẩn bị giao hàng, đang giao hàng, giao hàng thành công,…

Mã	UC006
Phân hệ	Khách hàng
Tên chức năng	Đăng ký tài khoản
Tóm tắt	Khách hàng cần có tài khoản để sử dụng ứng dụng
Lý do	Khách hàng cần cung cấp các thông tin cá nhân để có thể mua hàng trên hệ thống.
Yêu cầu chi tiết	Khách hàng đăng kí bán hàng phải cung cấp Họ tên, Số

	điện thoại, địa chỉ. Sau đó khách hàng xác nhận thông tin các câu hỏi cam kết, sẽ trở thành khách hàng của hệ thống. Khách hàng cần phải cung cấp thông tin đầu đủ và chính xác để có thể giao hàng nhanh và chính xác nhất.

Mã	UC007
Phân hệ	Khách hàng
Tên chức năng	Khiếu nại, hỗ trợ
Tóm tắt	- Khi khách hàng muốn khiếu nại đơn hàng có thể liên hệ trực tiếp qua hệ thống. - Khi khách hàng cần hỗ trợ có thể liên hệ trực tiếp qua hệ thống.
Lý do	
Yêu cầu chi tiết	- Tạo một chat box trên ứng dụng. Chat box do nhân viên trò truyện hoặc chat tự động. - Tạo phone call trên ứng dụng. Sẽ có nhân viên đảm nhận khi có khách hàng liên lạc. Hệ thống chỉ hỗ trợ 5h và 23h.

Mã	UC008
Phân hệ	Bộ phận giao hàng
Tên chức năng	Giao hàng
Tóm tắt	Khi có yêu cầu vận chuyển đơn hàng, Bộ phận giao hàng sẽ tiếp nhận và giao hàng cho khách hàng.
Lý do	Sau khi khách hàng đặt hàng cần có Bộ phận giao hàng đến cửa hàng lấy hàng và giao hàng cho khách hàng.
Yêu cầu chi tiết	Khi khách hàng đặt đơn hàng thì cửa hàng sẽ gửi yêu cầu cho bên bộ phận giao hàng. Bộ phận giao hàng sẽ kiểm tra đơn hàng đã được đóng gói đúng yêu cầu chưa, có tem niêm phong chưa. Nếu đơn hàng oke, Bộ phận giao hàng sẽ cập nhật trạng thái đơn hàng là đang giao và tiến hàng giao cho khách hàng. Khi giao xong dơn hàng thì cập nhật trạng thái đã giao.

Mã	UC009
Phân hệ	Khách hàng
Tên chức năng	Tìm cửa hàng gần nhất
Tóm tắt	Khách hàng sẽ tìm danh sách những cửa hàng gần với mình nhất.
Lý do	- Nhanh chóng có được đơn hàng sau khi đặt hàng. - Đảm bảo chất lượng sản phẩm. - Tiết kiệm chi phí giao hàng.

Yêu cầu chi tiết	Khi khách hàng tìm những cửa hàng gần nhất sẽ liệt kê danh sách các cửa hàng gần nhất. Tối đa là 5 cửa hàng trong một lần tìm (trả về theo dạng phân trang). Khi khách hàng chọn một cửa hàng thì sẽ vào xem được danh sách các sản phẩm của cửa hàng và chọn những món hàng họ muốn.

Mã	UC010
Phân hệ	Bộ phận giao hàng, khách hàng, admin, cửa hàng
Tên chức năng	Đăng nhập
Tóm tắt	- Người dùng bắt buộc phải đăng nhập ứng để sử dụng ứng dụng.
Lý do	- Biết được thông tin người đang đang sử dụng hệ thống. - Phục vụ mục đích sau này như phân tích dữ liệu, xây dựng thêm hệ thống đề xuất xản phẩm cho khách hàng. - Nắm được thông tin khi có tấn công hủy hoại hệ thống. Truy cứu trách nhiệm nhân viên gian lận. - Lưu giữ thông tin riêng cho mỗi khách hàng, cửa hàng.
Yêu cầu chi tiết	- Người dùng cần phải đăng nhập trước khi sử dụng ứng dụng. - Đăng nhập gồm có tên đăng nhập, mật khẩu hoặc hỗ trợ đăng nhập với tài khoản google, facebook.

Mã	UC011
Phân hệ	Bộ phận giao hàng, khách hàng, admin, cửa hàng
Tên chức năng	Đăng xuất
Tóm tắt	- Người dùng muốn thoát khỏi ứng dụng có thể sử dụng chức năng đăng xuất.
Lý do	- Bảo vệ được tài khoản khi không còn hoạt động trên thiết bị. Tránh người dùng sau vẫn còn đăng nhập vào tài khoản.
Yêu cầu chi tiết	- Ứng dụng sẽ cung cấp chức năng đăng xuất. Sau khi đăng xuất thì sẽ yêu cầu đăng nhập lại nếu muốn đăng nhập lại.

Mã	UC012 (Huyền)
Phân hệ	Khách hàng
Tên chức năng	Thêm vào danh sách cửa hàng yêu thích
Tóm tắt	Khách hàng sẽ thêm những cửa hàng vào danh sách yêu thích của mình.
Lý do	- Tiết kiệm thời gian hơn so với tìm kiếm.

	- Khách hàng hay mua hàng những nơi quen thuộc, yêu thích.
Yêu cầu chi tiết	Sẽ có chức năng thêm vào danh sách cửa hàng yêu thích đối với từng cửa hàng. Khách hàng có thể thêm vào danh sách yêu thích của mình. Khách hàng cũng có thể xóa cửa hàng khỏi danh sách yêu thích.

Mã	UC013
Phân hệ	Khách hàng
Tên chức năng	Thanh toán
Tóm tắt	Khách hàng có thể thanh toán trả trước (thanh toán online) hoặc trả sau (thanh toán tiền mặt).
Lý do	- Thu được tiền của khách hàng. - Có nhiều phương thức thanh toán cho khách hàng. - Thanh toán online giúp tiếp cận nguồn tiền nhanh và tiện lợi hơn.
Yêu cầu chi tiết	- Đối với thanh toán tiền mặt bộ phận giao hàng sẽ thu tiền. Sau đó sẽ chiếc khấu 5% cho công ty và lập hóa đơn chuyển phần còn lại cho cửa hàng. - Đối với thanh toán online công ty sẽ tích hợp những ví điện tử như momo, paypal. Sau đó sẽ giao tiền lại cho cửa hàng. - Sẽ thu phí theo ngày, hoặc tháng tùy theo hợp đồng cửa hàng ký kết với công ty.

1.1. Phi chức năng

STT	Trọng số	Mô Tả
1	5/5	Các mặt hàng được đăng bán trên hệ thống cần được đảm bảo về an toàn thực phẩm, nguồn gốc rõ ràng.
2	3/5	Tốc độ load dữ liệu không quá 2 giây.
3	4/5	Hệ thống dễ dàng bảo trì, cải tiến, mở rộng (thêm các dịch vụ mới).
4	5/5	Thông tin cá nhân quan trọng của khách hàng được bảo mật.

2. Giai đoạn 2

2.1. Thiết kế service

- Như thế nào là asynchronous: https://qr.ae/pv4cnX. Có nghĩa là send command message không thôi và không chờ response từ command. Response sẽ gửi cho action khác kèm theo định danh gì đó (ví dụ như oderId) để check là từ cái command nào rồi truy vấn db xử lý.

Usecase	Service	Operation	Collaborators	Mô tả
UC001 - Đăng ký thông tin bán hàng	shopService	registerShop()	- acceptRegisterShop() - sendMailContractShop() - activateRegisterShop()	- Vì các operation này phải chờ xác nhận hồ sơ, duyệt thủ công nên không thể xử lý bất đồng bộ
		getStatusRegisterShop()	- getStatusRegisterShop()	
UC002 - Xét duyệt hồ sơ	shopService	acceptRegisterShop()	- acceptRegisterShop() - sendMailContractShop()	-
		activateRegisterShop()	- activateRegisterShop()	-
UC003 - Quản lý tài khoản cửa hàng	shopService	manageShop()	- resetPasswordShop() - modifyLogoShop() - modifyInfoShop()	
UC004 - Quản lý sản phẩm	shopService	postProductList()	- getIdShop() - postProductList()	- getIdShop() lấy trực tiếp trong request khi shop đã đăng nhập
		updateProductList()	- getIdShop() - updateProductList()	- Ghi đè lên dữ liệu cũ
		postProduct()	- getIdShop() - postProduct()	Đăng 1 sản phẩm
		updateProduct()	- getIdShop() - updateProductList()	
		getCommentProduct()	- getIdShop() - getCommentProduct()	
UC005 - Quản lý đơn hàng	customerService	verifyOrder()	- verifyCustomerOrder()	- Nếu như có xử lý login session thì verify có thể sẽ không cần tạo. - Trong buyProdcut chúng ta có thể giảm số lương đơn hàng rồi check nó không âm thì cho đặt hàng. - payOrder(): có thể pending nó tạo async. Tùy vào phương thức thanh toán sẽ gọi các operation khác nhau. Ví dụ trên là thanh toán online.
	productService	buyProduct()	- checkProductOrder() - buyProduct()	
	shopService	createOrder()	- receiveCustomerOrder() - approveCustomerOrder()	
	deliveryService	deliveryOrder()	- receiveDeliveryOrder() - acceptDeliveryOrder()	
	orderService	completeOrder()	- payOrder() - authPaymentOrder()	
UC006 - Đăng ký tài khoản	customerService	registerCustomer()	- getInforCustomer() - verifyCustomer()	

Usecase	Service	Operation	Collaborators	Mô tả
UC007 - Khiếu nại, hỗ trợ	customerService	Complain()	- Chat() - CallShop()	-
UC008 - Giao hàng	OrderService	sendShippingRequest()	- sendShippingRequest()	
	deliveryService	deliveryOrder()	- checkOrder() - updateStatus() - confirmDeliverySuccess()	
UC009- Tìm cửa hàng gần nhất	customerService	getCustomerAddress()	- getCustomerAddress()	
	shopService	findNearestStore()	- getCustomerAddress() - findNearestStore())	
UC010 – Đăng nhập	customerService( )	Login()	- getCustomerInfor() - Login()	
UC010 – Đăng xuất	customerService( )	Logout()	- Logout()	
Danh sách cửa hàng yêu thích (UC012)	customerService	PreferShop()	- getPreferShopByID() - AddPreferShop()	

2.1. Thiết kế csdl

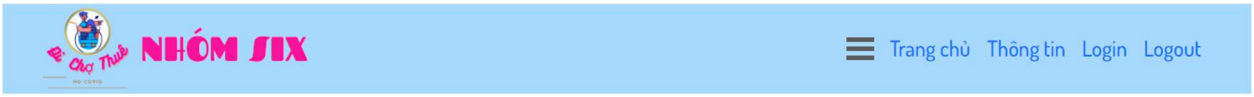
Service	Bảng	Cấu trúc	Mô tả
shopService	Stores	ID Name Phone Email Password DayStart timeOpen timeClose productCategory logo agreeTeam address	- Port shopService của express: 3005 - Port reporter của metrics của shopNode: 3035
productService	Product	IdProduct IdStore productName productPrice Unit InventoryNumber sold	
	productCategory	ID categoryName Description	
orderService	order	ID customerID orderID productList[{productID, productName, price, quantity}] Status Store{storeID, storeName}	

Service	Bảng	Cấu trúc	Mô tả
customerService	Customer	ID customerName DoB Gender Phone Address	- Port reporter của metrics của customerNode1: 3036
	PreferStore	Store[{storeID, storeName}] customerID	-
cartService	Cart	cartID CustomerID, Product[{productID, productName, quantity, price}], Status Store{storeID, storeName}	

2.1. Thiết kế frontend

2.1.1. Thiết kế logo

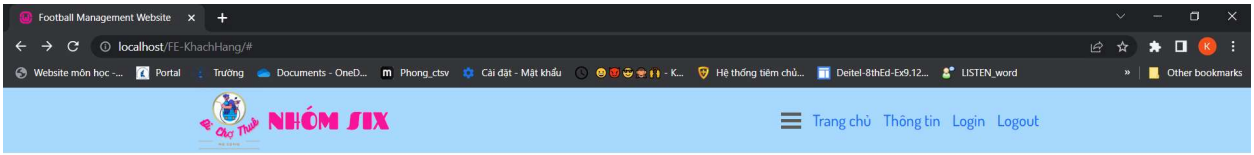
2.1.1. Thiết kế header

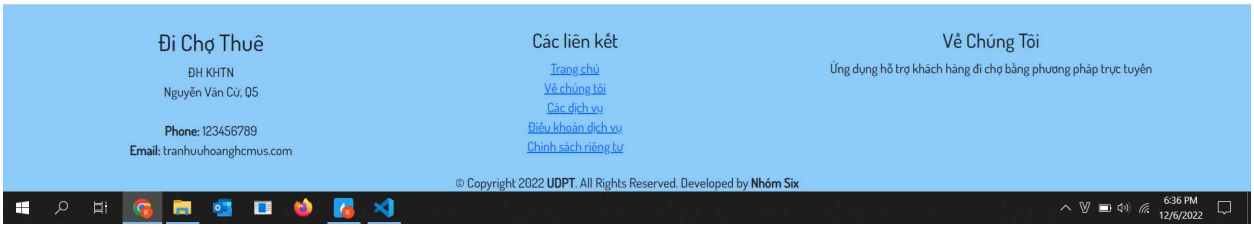


2.1.1. Thiết kế footer

2.1.1. Banner

2.1.1. Trang chủ





2.1.1.

2.1.1. .

3. Giai đoạn 3 – Chọn framework Moleculer

Thực hiện cài đặt các service

Service	Operation	Collaborators
authenticationService	login()	Service authenticationService::login ()
	register()	Service authenticationService::register ()
orderService	checkOut()	Service orderService::checkOut()
	addDeliveryInfo()	Service orderService:: addDeliveryInfo()
	addOrderTicket()	Service orderService:: addOrderTicket()
	addOrderTicket()	Service orderService:: addOrderTicket()
homeService	addProductToCart()	Service homeService:: addProductToCart()
	showAllProduct()	Service homeService:: showAllProduct()
	showProductByType()	Service homeService:: showProductByType()
	showProductByCategory()	Service homeService:: showProductByCategory()
	searchProduct()	Service homeService:: searchProduct()
userService	showUserInfo()	Service userService:: showUserInfo()
	showOrderList()	Service userService:: showOrderList()

	showCart()	Service userService:: showCart()
	updateUserInfo()	Service userService:: updateUserInfo()
	addOrderReview()	Service userService:: addOrderReview()
adminService	addProduct()	Service userService:: addOrderReview()
	deleteProduct()	Service userService:: deleteProduct()
	updateProduct()	Service userService:: updateProduct()
	productList()	Service userService:: productList()

3.1. Những nội dung chính của moleculer (pronounciation)

a. Service (dịch vụ)

- Dịch vụ là một mô-đun JavaScript đơn giản chứa một phần của ứng dụng phức tạp. Nó bị cô lập và khép kín, có nghĩa là ngay cả khi nó đi ngoại tuyến hoặc gặp sự cố, các ***dịch vụ (service)*** còn lại sẽ không bị ảnh hưởng.
- Ở đồ án này là các service đã thiết kế ví dụ như: orderService, productService, customerService, adminService. Mỗi service phục vụ một số chức năng riêng biệt độc lập mà hệ thống phân tán cần.

b. Node (nút)

- Một nút là một quy trình HĐH đơn giản chạy trên mạng cục bộ hoặc bên ngoài. Một instance duy nhất của một nút có thể lưu trữ một hoặc nhiều dịch vụ.

c. Local service (dịch vụ địa phương)

- Hai (hoặc nhiều) dịch vụ chạy trên một nút duy nhất được coi là dịch vụ cục bộ. Họ chia sẻ tài nguyên phần cứng và sử dụng local bus để liên lạc với nhau, không có độ trễ mạng (không sử dụng ***trình vận chuyển (transporter)***).

d. Remote service (dịch vụ từ xa)

- Các dịch vụ được phân phối trên nhiều nút được coi là từ xa. Trong trường hợp này, giao tiếp được thực hiện thông qua vận chuyển.

e. Service Broker (Dịch vụ môi giới)

- Dịch vụ môi giới (service broker) là trái tim của phân tử (Moleculer). Nó chịu trách nhiệm quản lý và giao tiếp giữa các dịch vụ (địa phương và từ xa). Mỗi nút phải có một thể hiện của nhà môi giới dịch vụ.

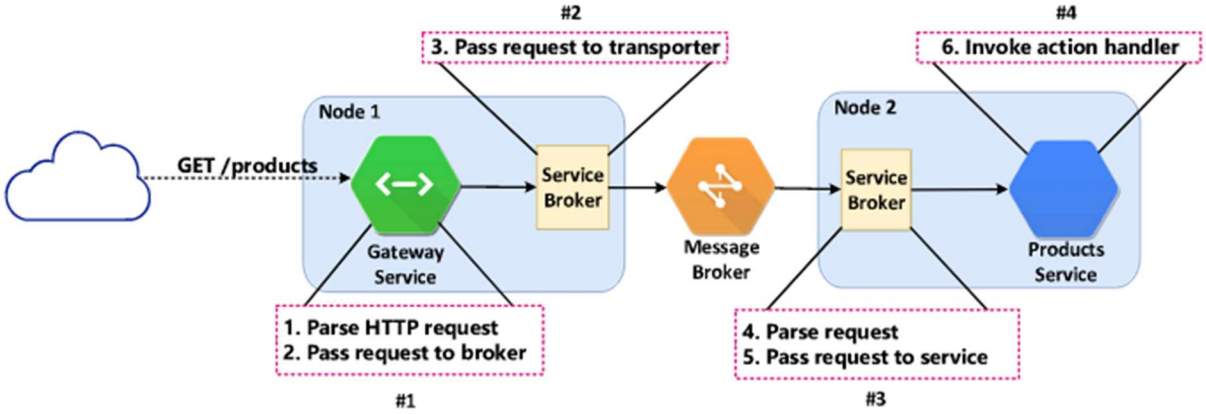
f. Transporter

- Transporter là một tuyến (bus) truyền thông mà các dịch vụ sử dụng để trao đổi tin nhắn. Nó chuyển các sự kiện, yêu cầu và phản hồi.

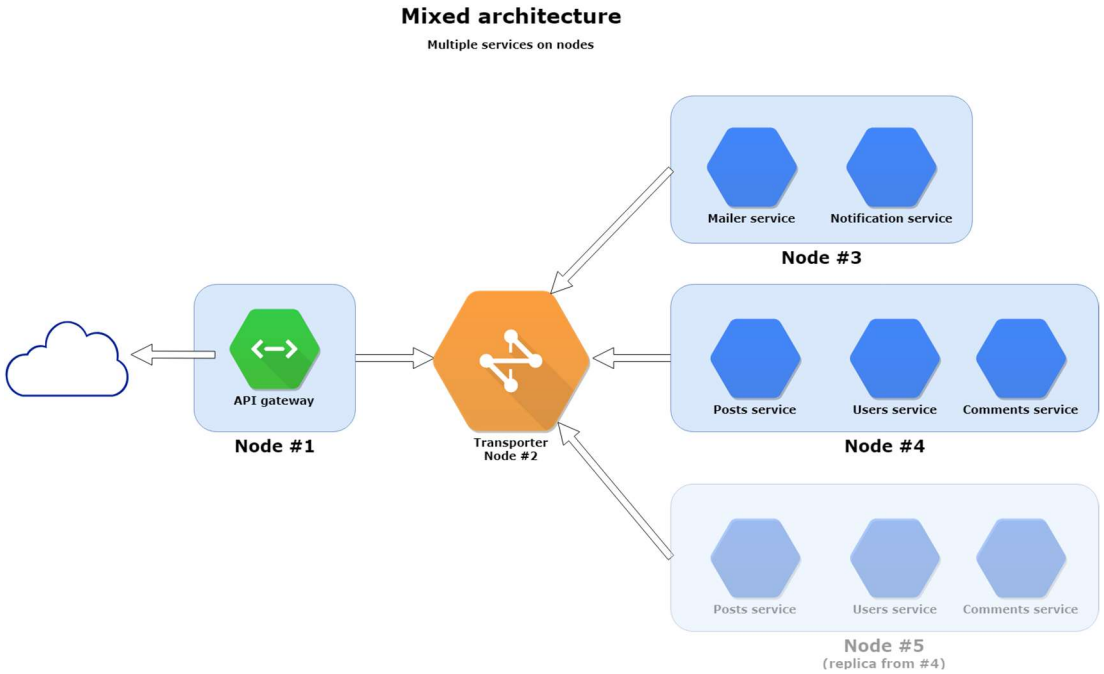
g. Gateway

- API Gateway đưa ra các dịch vụ Moleculor cho người dùng cuối. Gateway là một dịch vụ Moleculer thông thường chạy một máy chủ (HTTP, WebSockets, v.v.). Nó xử lý các yêu cầu đến, ánh xạ chúng thành các cuộc gọi dịch vụ và sau đó trả về các phản hồi phù hợp.

h. Overall View



hình giao tiếp giữa 2 node



Hình mixed architechture

- **Continueous Link:**
  https://moleculer.services/docs/0.14/concepts.html#:~:text=returns%20appropriate%20responses.-,Overall%20View,-There%E2%80%99s%20nothing%20better

3.2. Các module trong moleculer

- https://moleculer.services/modules.html

3.3. Transporter trong moleculer microservice nodejs framework (link)

Transporter là một mô-đun quan trọng nếu bạn đang chạy các dịch vụ trên nhiều nút. Transporter giao tiếp với các nút khác. Nó chuyển các sự kiện, gọi các yêu cầu và xử lý phản hồi, v.v. Nếu một dịch vụ chạy trên nhiều instance trên các nút khác nhau, các yêu cầu sẽ được cân bằng tải giữa các nút trực tiếp.

Moleculer hỗ trợ các loại sau:

- TCP transporter
  -
- NATS Transporter
- Redis Transporter
- MQTT Transporter
- AMQP Transporter
- Kafka Transporter
- NATS Streaming (STAN) Transporter
- Custom transporter

3.4. Moleculer runner

Thứ tự ưu tiên load moleculer config cho ServiceBroker (node hay instance của node): https://moleculer.services/docs/0.14/runner.html#Configuration-loading-logic

- Đầu tiên là lấy giá trị trong lúc tạo service broker. Dưới đây, logger sẽ được set là false. Các properties khác sẽ mặc định và thiết lập sau:

```
const broker = new ServiceBroker({ logger: false });
```

- Tiếp theo là tới biến môi trường trong file .env.
- Tiếp theo là biến môi trường lúc set ở CLI
- Tiếp theo là trong file moleculer.config.js
- Tiếp theo là trong file moleculer.config.json

3.5. Configuration

Những cấu hình mặc định cho ServiceBroker trong file moleculer.config.js: link

Những option cần lưu ý

- nodeID: định danh của một node và phải duy nhất trong namespace
- transporter: giao tiếp giữa các node (ServiceBroker). Mặc định là null, cần sửa tải 1 message broker như rabbitmq để sử dụng giao tiếp giữa các node.
- disableBalancer: Mặc định balancer được xử lý bởi Moleculer. Để sử dụng balancer của rabbitmq thì cần thiết lập disableBalancer thành true (dòng 111 trong file moleculer.config.js).

- hotReload: khởi động lại service khi nó thay đổi.
- cacher: cache dữ liệu, mặt định là false, không kích hoạt. Nếu dùng có những option như, true-> Memory, LRU, Redis. link
- preferLocal: Đầu tiên ServiceBroker sẽ thử gọi local instance của service để giảm độ trễ mạng. Do đó cần thiết lập thuộc tính preferLocal: false dưới registry key (trong moleculer.config.js) để gọi được tới remote node (chứa service giống với service local node). Nếu trong node không có service cần thì gọi remote service.
- strategy: "RoundRobin". Là chiến lược loadbalancing, mặc định là RoundRobin. Trong object registry.

3.6. Service

- Link: https://moleculer.services/docs/0.14/services.html

3.6.1. Load service khi thay đổi service

- Sử dụng module nodemon cài ở global trong nodejs (cài global là dùng để chạy mà khỏi phải cài mỗi lần vô project khi có project mới, nodejs tự hiểu và lấy source nodemon ở global).
- Đọc thêm về moleculer runner: https://moleculer.services/docs/0.14/services.html#Hot-Reloading-Services

3.6.2. Schema

3.6.3.

3.6.4. Name

3.6.5. Action

- Link: https://moleculer.services/docs/0.14/services.html#Actions
- Định nghĩa các phương thức của service.
- Được gọi bởi broker.call, ctx.call

3.6.6. actione

3.7. Action

3.7.1. Call action

```
const res = await broker.call(actionName, params, opts);
```

3.7.2. Stream (gửi file)

- Link: https://moleculer.services/docs/0.14/actions.html#Streaming

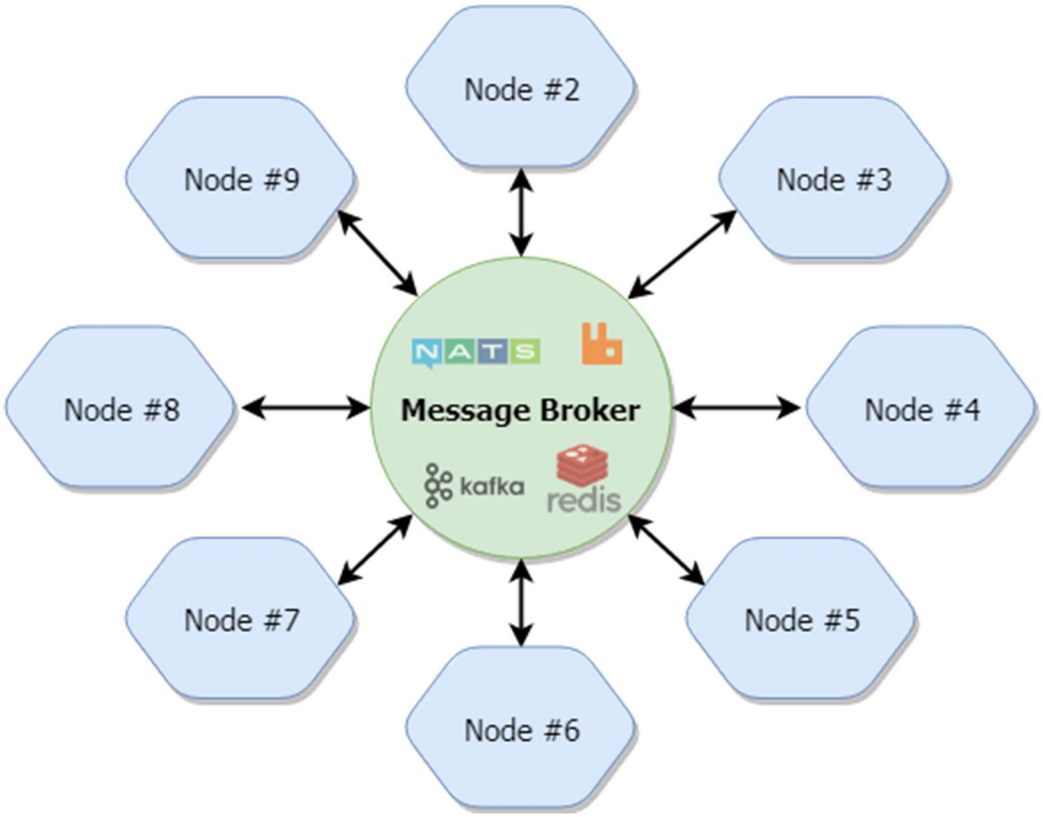
3.7.3. .

3.7.4. Hook (chèn mã vào trong 1 hoặc nhiều action)

- Ví dụ chèn mã vào trong action của service: https://moleculer.services/docs/0.13/actions.html#Action-hooks

- Ví dụ chèn mã trong moleculer db: https://moleculer.services/docs/0.13/moleculer-db.html#Data-Manipulation
- Define action hooks to wrap certain actions coming from mixins. There are before, after and error hooks. Assign it to a specified action or all actions (*) in service. The hook can be a Function or a String. The latter must be a local service method name.

3.8. Networking



giao tiếp giữa các node thông qua message broker

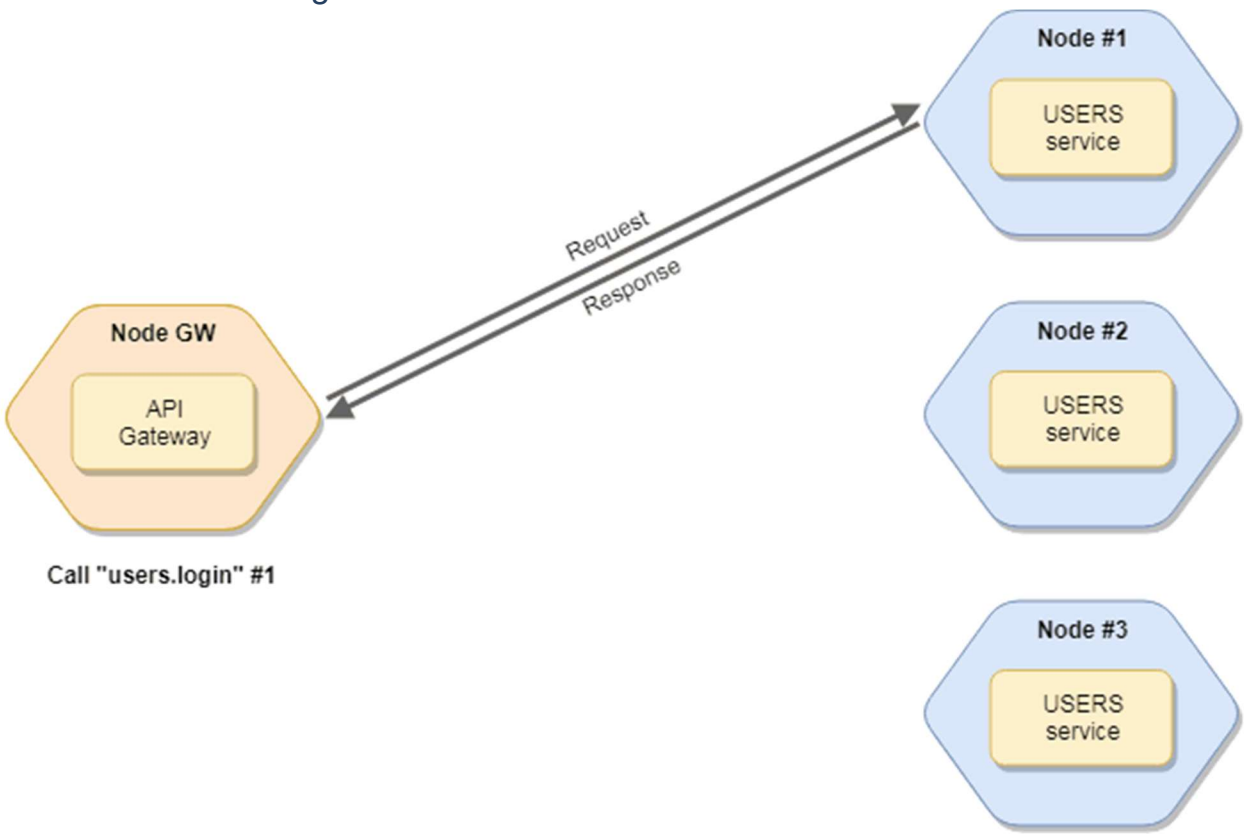
- Giao tiếp giữa các node (local node và các remote node): https://moleculer.services/docs/0.14/networking.html#:~:text=server%2D2%22%0A%5D-,Serviceless%20node,-Please%20note%2C%20you
- Sử dụng AMQP 0.9 (rabbitmq). Ngoài ra còn có kafka, redis. Sau đó phải thiết lập disableBalancer:true để sử dụng NATS.
  https://moleculer.services/docs/0.14/networking.html#AMQP-0-9-Transporter
  https://moleculer.services/docs/0.14/networking.html#Disabled-balancer
- Tải, cài đặt rabbitmq và erlang theo hướng dẫn trên trang chủ của rabbitmq.
- Mặc định balancer được xử lý bởi Moleculer. Để sử dụng balancer của rabbitmq thì cần thiết lập disableBalancer thành true (dòng 111 trong file moleculer.config.js).

3.9. Discovery & Registry

- https://moleculer.services/docs/0.14/registry.html

- Một nút không cần biết bất cứ điều gì về các nút khác trong thời gian start. Khi nó bắt đầu, nó sẽ thông báo rằng nó có mặt cho tất cả các nút khác để mỗi nút có thể xây dựng local service registry của riêng mình. Trong trường hợp xảy ra sự cố nút (hoặc dừng), các nút khác sẽ phát hiện nó và xóa các dịch vụ bị ảnh hưởng khỏi sổ đăng ký của họ. Bằng cách này, các yêu cầu sau đây sẽ được chuyển đến các nút trực tiếp.
- Sử dụng local (default option)là khả thi cho đồ án này. Nếu số lượng node (nơi chứa các service) hơn 100 thì bộ nhớ ngoài như redis,…

3.10. Load balancing



Load balancing

- https://moleculer.services/docs/0.14/balancing.html
- Nếu một service đang chạy trên nhiều node instancce, ServiceRegistry sử dụng strategies (vd: RoundRobin) để chon 1 node trong những node khả dụng
- RoundRobin là một strategy mà ở đó mỗi tiến trình sẽ gán một thời gian xử lý nhất định.Sau khi hết thời gian đó sẽ đến lược tiến trình khác. Lặp lại cho tới khi nào hoàn thành công việc của tiến trình.
- Đầu tiên ServiceBroker sẽ thử gọi local instance của service để giảm độ trễ mạng. Do đó cần thiết lập thuộc tính preferLocal: false dưới registry key (trong moleculer.config.js) để gọi được tới remote node (chứa service giống với service local node)

3.11. Fault Tolerance (ServiceBroker)

- Liên kết: https://moleculer.services/docs/0.14/fault-tolerance.html
- Xử lý lỗi service, node, timeout

3.11.1. Bulkhead

- https://moleculer.services/docs/0.14/fault-tolerance.html#Bulkhead
- Bulkhead được triển khai trong Moleculer để kiểm soát việc xử lý request đồng thời của những action.

```
const broker = new ServiceBroker({
   bulkhead: {
      enabled: true, //kích hoạt true để sử dụng
      concurrency: 3, // Số request được xử lý đồng thời
      maxQueueSize: 10, //Số request tối đa nhận vào
   }
});
```

- Được cấu hình mặc định trong file moleculer.config.js. Nếu muốn thiết lập lại, bạn có thể ghi đè trong file cấu hình service (servicename.service.js thường nằm trong thư mục services)

3.12. Caching

- Link web moleculer: https://moleculer.services/docs/0.13/caching.html
- vd cache ở action:

```
actions: {
        list: {
            // Enable caching to this action
            cache: true,
            handler(ctx) {
                this.logger.info("Handler called!");
                return [
                    { id: 1, name: "John" },
                    { id: 2, name: "Jane" }
                ]
            }
        }
    }
```

3.13. API Gateway

- https://moleculer.services/docs/0.14/moleculer-web.html#Aliases
- Dùng để khai báo các api của các service đăng ký vào.
- Có thể dùng rest mặc định cho các restful api service: list, get, create, update and remove actions (link)

3.13.1. Nhận request body, paramas và mặc định merge cả 2 thành ctx.params

- https://moleculer.services/docs/0.13/moleculer-web.html#Parameters
- Có thể set tách riêng mergeParams: false

```
broker.createService({
```

```
    mixins: [ApiService],
    settings: {
        routes: [{
            path: "/",
            mergeParams: false
        }]
    }
});
```

-

3.13.2. Response status, message, content type, header, redirect location

- https://moleculer.services/docs/0.13/moleculer-web.html#Response-type-amp-status-code

  - `ctx.meta.$statusCode` - set `res.statusCode`.
  - `ctx.meta.$statusMessage` - set `res.statusMessage`.
  - `ctx.meta.$responseType` - set `Content-Type` in header.
  - `ctx.meta.$responseHeaders` - set all keys in header.
  - `ctx.meta.$location` - set `Location` key in header for redirects.

```
module.exports = {
    name: "export",
    actions: {
        // Download response as a file in the browser
        downloadCSV(ctx) {
            ctx.meta.$responseType = "text/csv";
            ctx.meta.$responseHeaders = {
                "Content-Disposition": `attachment; filename="data-${ctx.params.id}.csv"`
            };

            return csvFileStream;
        }

        // Redirect the request
        redirectSample(ctx) {
            ctx.meta.$statusCode = 302;
            ctx.meta.$location = "/login";

            return;
        }
    }
}
```

-

3.13.3. CORS

- https://moleculer.services/docs/0.13/moleculer-web.html#CORS-headers

3.13.4. Middlewares (sử dụng các module của nodejs)

https://moleculer.services/docs/0.14/moleculer-web.html#Middlewares

- Ví dụ như sử dụng passportjs.

3.13.5. Chú ý khi register những service tương đương (service name giống nhau) và những action của service

- Phải giống nhau về action trong các nhân bản của service.
- Ví dụ: remote service khác action là cho api gateway chỉ chọn một trong hai node (đều tất nhiên). Nhưng khi call một service mà không có action hoặc action nó khác thì tất nhiên là fail. Vì cơ chế loadbalancing là sẽ xoay tua load node (gồm nhiều service).

3.13.6. .

3.14. Validating (Kiểm tra params truyền vào, validate defining schema)

- Dùng fastest-validator: https://github.com/icebob/fastest-validator#properties-9
- Link moleculer: https://moleculer.services/docs/0.13/validating.html
- Strict validation: https://github.com/icebob/fastest-validator#strict-validation
  Khi có param ngoài những param define thì lỗi hoặc remove chúng để sử dụng ctx.params luôn cho việc create đối tượng mới mà khỏi phải kiểm tra hoặc liệt kê từng param khi create.
  To remove the additional fields in the object, set $$strict: "remove".

```
const schema = {
    name: { type: "string" }, // required
    $$strict: true // no additional properties allowed
}

const check = v.compile(schema);

check({ name: "John" }); // Valid
check({ name: "John", age: 42 }); // Fail
```

-

3.15. Moleculer Database

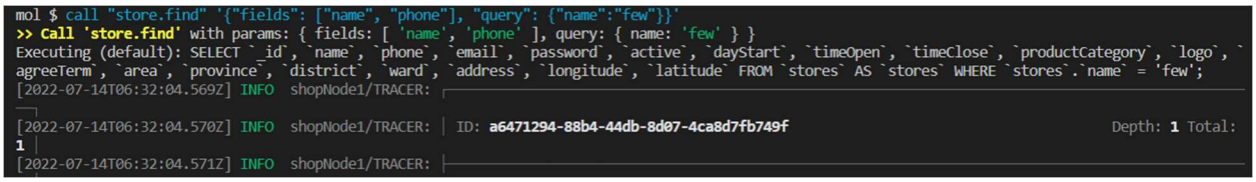
- Sử dụng để kết nối tới db. Sau đó dùng mixin merge chúng vào service. Link: https://moleculer.services/docs/0.13/moleculer-db.html#Connect-to-multiple-DBs
- Code mẫu: https://github.com/moleculerjs/moleculer-db/blob/master/packages/moleculer-db-adapter-sequelize/examples/integration/index.js
  https://github.com/moleculerjs/moleculer-db/blob/master/packages/moleculer-db-adapter-sequelize/examples/simple/index.js
- update statement: https://github.com/moleculerjs/moleculer-db/blob/master/packages/moleculer-db-adapter-sequelize/examples/simple/index.js#:~:text=%7D)%3B-,//%20Update%20a%20posts,-checker.add
- Những action cache dữ liệu mặc định: find, count, list, get

3.16. Moleculer repl

3.16.1. Call action từ repl

- link ví dụ: https://moleculer.services/docs/0.13/moleculer-repl.html#Call-an-action

- vd:
  mol $ call "math.add" '{"a": 5, "b": "Bob", "c": true, "d": false, "e": { "f": "hello" } }'
  Params will be `{ a: 5, b: 'Bob', c: true, d: false, e: { f: 'hello' } }`



  call "store.find" '{"fields": ["name", "phone"], "query": {"name":"few"}}'
-

3.17. Va

3.18. .

4. Giai đoạn 3 – Thực hiện cài đặt

4.1. Cài đặt moleculer project

- Mở vs code cli lên.
- Lệnh cài đặt global moleculer cli: npm i -g moleculer-cli
- Lệnh khởi tạo: moleculer init project project-folder-name
- Chọn Yes npm install. Sau đó trả lời các câu hỏi. Để thực hiện có cài chúng không
  - Thường nếu muốn tạo api gateway thì Yes.
  - Nếu muốn giao tiếp giữa nhiều node chọn yes và sau đó transporter (message broker) là AMQP (tức là rabbitmq).
  - Yes metrics để log thông báo số liệu ra console
  - Yes tracer để giám sát gì đấy
  - Mấy cái khác như tạo db, tạo docker file thì không cần nếu không cần chúng.
-
- cd ./project-folder-name: di chuyển tới moleculer project mới tạo
- Sau khi chạy sẽ có một repl (read eval print loop) và chúng ta có thể thực hiện những lệnh như load, loadFolder để load template rồi tạo 1 service mới trong node hiện hành… repl có dạng như sau, nếu không thấy hãy enter để chúng xuất hiện.

```
mol $ nodes
```

4.2. Tạo nhiều node trong moleculer như nào

- Để tạo node mới chúng ta cũng tạo project giống như trên. Sau đó config lại file moleculer.config.js để tránh bị lỗi và dễ quản lý. Một số lưu ý:
  - **nodeID**: định danh của một node và phải duy nhất trong namespace
  - transporter: giao tiếp giữa các node (ServiceBroker). Mặc định là null. Để sủ dụng, phải tải 1 message broker như rabbitmq để sử dụng giao tiếp giữa các node. Phải tải package của message broker đó nữa.

- **disableBalancer**: Mặc định balancer được xử lý bởi Moleculer. Để sử dụng balancer của rabbitmq thì cần thiết lập disableBalancer thành true (dòng 111 trong file moleculer.config.js).
- **hotReload**: khởi động lại service khi file template của nó thay đổi.
- cacher: cache dữ liệu, mặt định là false, không kích hoạt. Nếu dùng có những option như, true-> Memory, LRU, Redis. link
- **preferLocal**: Đầu tiên ServiceBroker sẽ thử gọi local instance của service để giảm độ trễ mạng. Do đó cần thiết lập thuộc tính preferLocal: false dưới registry key (trong moleculer.config.js) để gọi được tới remote node (chứa service giống với service local node). Nếu trong node không có service cần thì gọi remote service.
- **strategy**: "RoundRobin". Là chiến lược loadbalancing, mặc định là RoundRobin. Trong object registry.
- **metrics**: dùng để report, đầu ra mặc định là console. port phải khác với port của các node khác.

- Trong project có thể thêm các file template định nghĩa service với cấu trúc tên có đuôi là {name}.service.js. Khi chạy project với lệnh npm run dev (nên cài thêm nodemon ở global) thì xem trong file package.json, phần script sẽ có key dev với lệnh sau:

```
"scripts": {
    "dev": "moleculer-runner --repl --hot services/**/*.service.js",
    "start": "moleculer-runner",
    "cli": "moleculer connect AMQP",
    "ci": "jest --watch",
    "test": "jest --coverage"
  },
```

  Trong đó moleculer-runner là một package giúp chúng ta chạy project load các template service và tự động luôn các service trong đường dẫn cụ thể là folder services. Tất cả các file trong folder và kể cả subfolder sẽ được gọi và load service nếu có.
  Lệnh --repl là sau khi project chạy sẽ hiện repl này với ký dòng mod $ và sẽ có các lệnh như load service,…
  option --hot dùng để check xem file service nếu thay đổi thì sẽ reload lại service.
- Giờ chỉ việc định nghĩa các template service và load chúng vào node sau khi chạy poject.

4.3. Cài đặt Rabbitmq

- Cài trên máy windows
- Trước tiên cài Erlang: https://www.rabbitmq.com/which-erlang.html
- Link download: https://www.rabbitmq.com/install-windows.html#installer
- Nơi gần download: https://www.rabbitmq.com/install-windows.html#downloads. Link trực tiếp
- Sau đó mở cli của rabbitmq lên chạy lệnh sau: rabbitmq-plugins enable rabbitmq_management
- Nếu bị lỗi Erlang environment thì mở variable environment trên window để set chúng.
- Tiếp theo truy cập url sau trên trình duyệt:

go to http://localhost:15672

username: guest
passowrd: guest

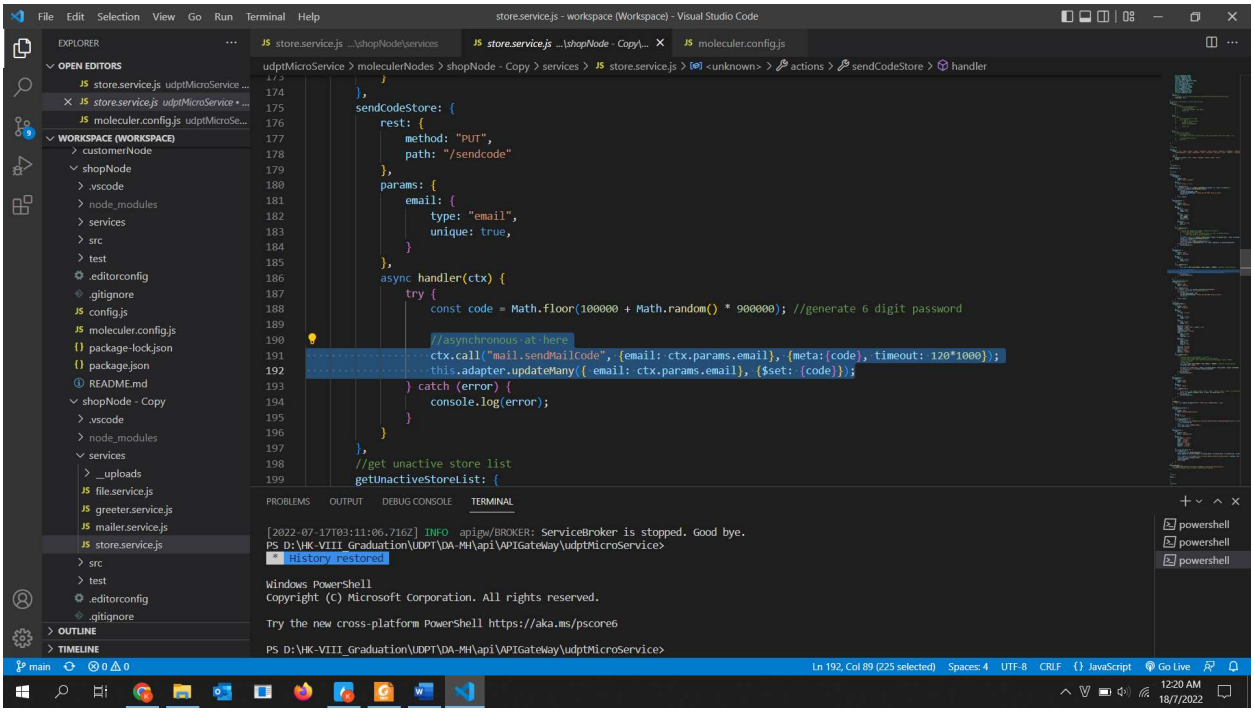
4.4. Giao tiếp giữa các service của internal node hoặc remote node.

- Như thế nào là giao tiếp asynchronous giữa các service: https://qr.ae/pv4cnX. Có nghĩa là send command message không thôi và không chờ response từ command. Response sẽ gửi cho action khác kèm theo định danh gì đó (ví dụ như oderId) để check là từ cái command nào rồi truy vấn db xử lý. Tức là chỉ có this.broker.call, không có await trước và biến gán để lấy data response. Link xem thêm: https://qr.ae/pvkFWF

Slide communication in microservice architecture

- Interprocess communication in a microservice architecture. Interaction styles:
  - One-to-one—Each client request is processed by exactly one service
  - One-to-many—Each request is processed by multiple services
  - Synchronous—The client expects a timely response from the service and might even block while it waits.
  - Asynchronous—The client doesn't block, and the response, if any, isn't necessarily sent immediately..

- Communicating using the Asynchronous messaging
  - Implementing one-way notifications is straightforward using asynchronous messaging. The client sends a message, typically a command message, to a point-to-point channel owned by the service. The service subscribes to the channel and processes the message. **It doesn't send back a reply**.



- Implementing publish/subscribe: Messaging has built-in support for the publish/subscribe style of interaction. A client publishes a message to a publish-subscribe channel that is read by multiple consumers. Này chắc sử dụng multiple calling action của moleculer: https://moleculer.services/docs/0.13/actions.html#Multiple-calls

-

- Link: https://moleculer.services/docs/0.13/actions.html
- Xem phần networking sẽ thấy các node giao tiếp đồng bộ hoặc bất đồng bộ qua message broker.
- Call remote action: sử dụng ctx.call() trong (trong hàm handler(ctx)). https://moleculer.services/docs/0.13/actions.html#Metadata
  const res = await broker.call(actionName, params, opts);
- Trong trường hợp đồng bộ hóa dữ liệu database giữa các action của service có cùng chức năng thì bạn cần thêm option là nodeID để xác định chính xác action của service của node nào. Điều này tránh gọi về chính nó.
- Call internal action: https://moleculer.services/docs/0.13/actions.html#:~:text=%7D%0A%20%20%20%20%7D%0A%7D)%3B-,When%20making,-internal%20calls%20to

```
communicateRemoteNode: {
            rest: {
                method: "GET",
                path: "/communicate"
            },
```

```
            /** @param {Context} ctx  */
            async handler(ctx) {
                //calling remote action
                const resultRemoteHelloAction = await ctx.call("greeter.hello");

                //calling internal action
(https://moleculer.services/docs/0.13/actions.html#:~:text=%7D%0A%20%20%20%20%7D%
0A%7D)%3B-,When%20making,-internal%20calls%20to)
                const resultInternalHelloAction = await this.actions.hello();


                return `Here is a service communication on multible nodes,
                    remote: ${resultRemoteHelloAction}, internal:
${resultInternalHelloAction}`;
            }
        }
```

- Có thể call bằng this.broker.call trong handler: https://medium.com/swlh/moleculer-develop-ecommerce-features-using-microservice-architecture-37a4a0d48788

```
broker.createService({
 name: "mod",
 actions: {
  hello(ctx) {
   console.log(ctx.meta);
   // Prints: { user: 'John' }
   ctx.meta.age = 123
   return this.actions.subHello(ctx.params, { parentCtx: ctx });
  },

  subHello(ctx) {
   console.log("meta from subHello:", ctx.meta);
   // Prints: { user: 'John', age: 123 }
   return "hi!";
  }
 }
});
```

-

4.5. shopService

4.5.1. api

- Tạo một project
- Sử dụng mysql2 dependency cho nodejs để kết nối tới mysql

- Link: https://github.com/sidorares/node-mysql2
-

4.5.2. upload file trong moleculer web

- https://github.com/moleculerjs/moleculer-web/issues/54
- link demo github: https://github.com/moleculerjs/moleculer-web/blob/master/examples/full/index.js#L59
- file upload service: https://github.com/moleculerjs/moleculer-web/blob/master/examples/file.service.js
- form upload: https://github.com/moleculerjs/moleculer-web/blob/master/examples/full/assets/upload.html#L12
- Ajax upload: https://github.com/moleculerjs/moleculer-web/pull/85
- Action gửi file: https://moleculer.services/docs/0.14/actions.html#Streaming
-

- Chúng ta sẽ lưu trực tiếp file ảnh logo trong mysql. Muốn render ra tag image của html thì cần chuyển sang base64. Link: https://stackoverflow.com/questions/34111390/displaying-blob-image-from-mysql-database-into-dynamic-div-in-html
  echo '';
  ⇨ kết quả:
  <div style="background-color:black; text-align:center; padding: 5px;">
  <img
  src="">
  </div>
- stackoverflow upload file, ctx.meta.socket = socket: https://stackoverflow.com/questions/66172801/getting-undefined-value-even-after-attaching-socket-to-ctx-in-beforecall

4.5.3. Gửi mã kích hoạt qua mail

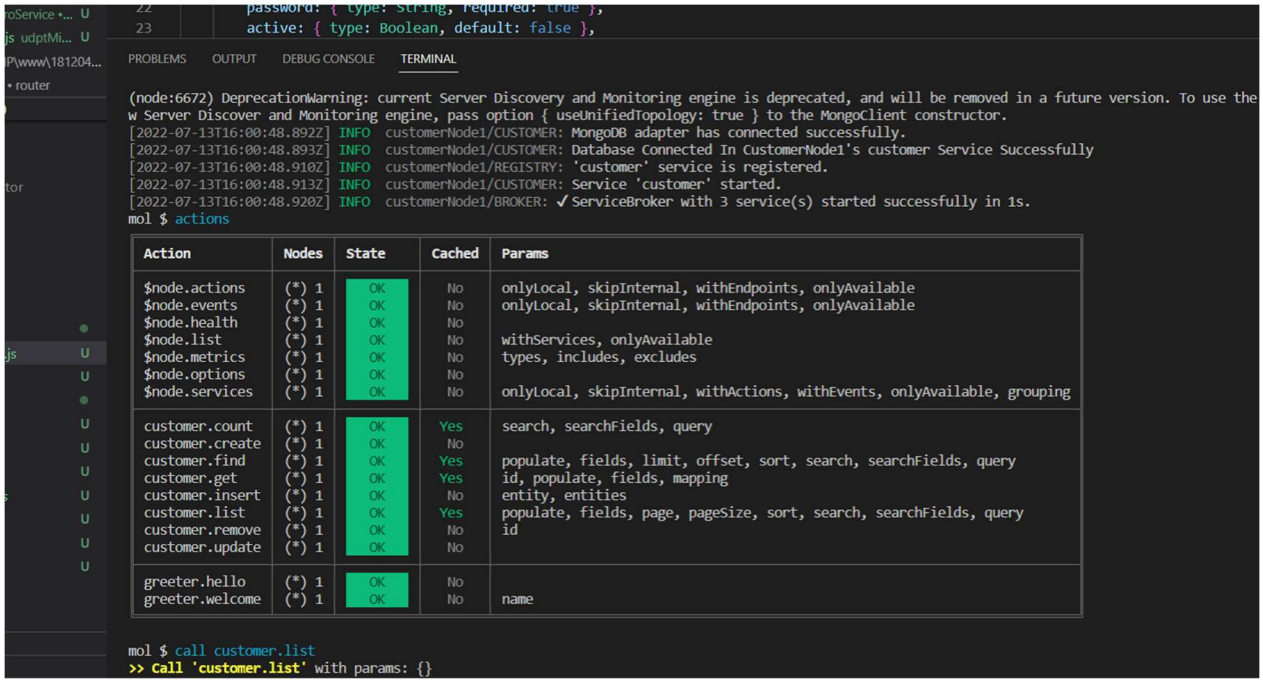
- link mailer trong moleculer: https://github.com/moleculerjs/moleculer-addons/tree/master/packages/moleculer-mail#readme

4.5.4. .

4.6. customerService

4.6.1. Cài đặt mongoose adapter cho service

- Link github moleculer-db-adapter-mongoose: https://github.com/moleculerjs/moleculer-db/tree/master/packages/moleculer-db-adapter-mongoose
- Link code mẫu: https://github.com/moleculerjs/moleculer-db/blob/master/packages/moleculer-db-adapter-mongoose/examples/simple/index.js
- Link moleculer docx lý thuyết hướng dẫn cài đặt, trong này có mấy action mặc định là find, delete, create, list,…: https://moleculer.services/docs/0.13/moleculer-db.html#Mongoose-Adapter
- Chú ý là cài mongoose phiên bản nhỏ hơn 6 chú không bị lỗi (npm install mongoose@5.8.11)
- Chạy lệnh cd vô thư mục customerNode (không thì chuột phải vào project folder, chọn integaret terminal), npm i xong chạy npm run dev, enter để thấy chữ mod $ là oke, rồi chạy actions sẽ liệt kê list action. Xong chạy call customer.find để xem có bao nhiêu customer trong db. Để call được api thì chạy thêm node chính ngoài project nữa. chuột phải udptMicroservice rồi chọn Terminal xong hiện console thì nhập npm run dev. Mở chrome lên với port 3000. Danh sách các api sẽ hiện ra.



4.7. .

-

5. Triển khai các service dưới dạng service windows (để chạy nhẹ nhanh hơn khi chạy trên IDE)

a. Nodejs

https://www.youtube.com/watch?v=obXr1NtSEOU&ab_channel=HarveyWilliams

b. Java

https://www.youtube.com/watch?v=JeXl60PbXXw&ab_channel=JavaTechie

6. Các lỗi thường gặp

6.1. Moleculer

1. Lỗi trùng port của metrics trong moleculer.config.js khi sử dụng nó làm khuôn mẫu để tạo nhiều node (ServiceBroker) mà quên cấu hình lại:

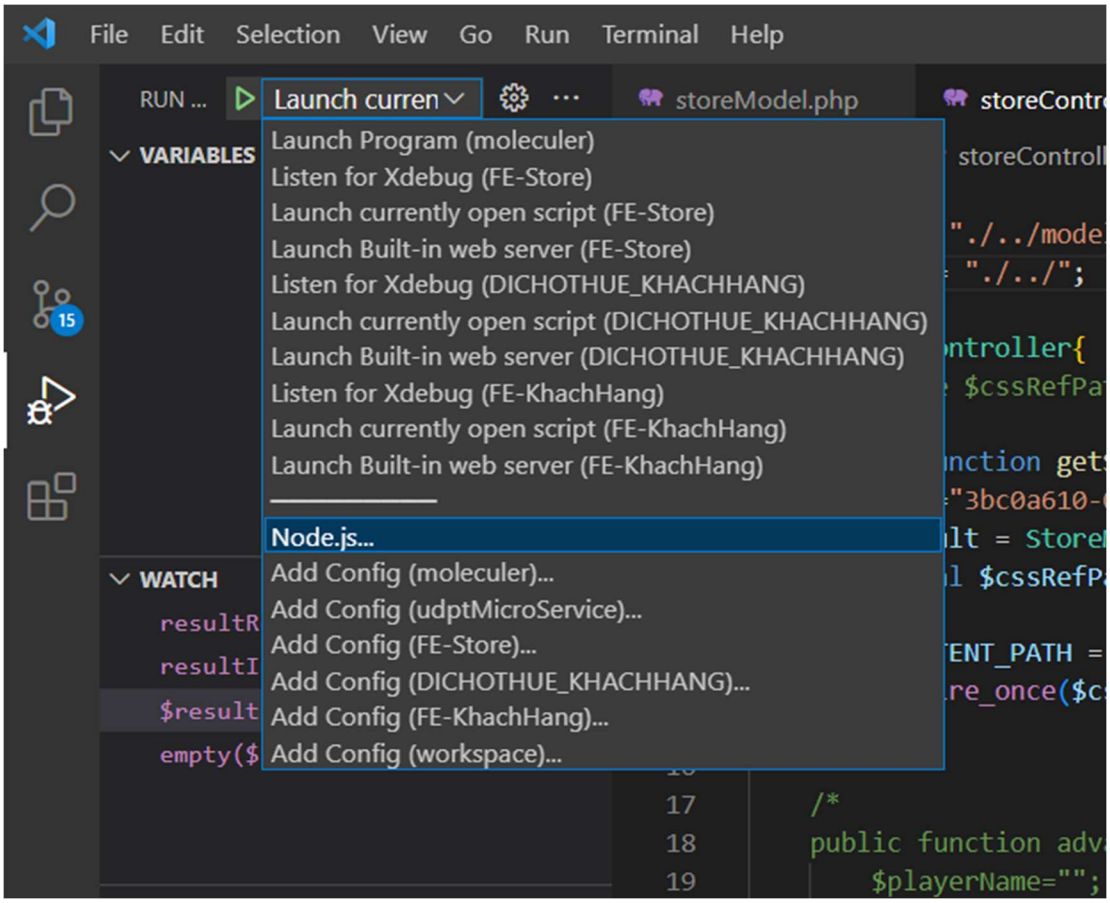
```
[2022-07-10T10:02:54.312Z] INFO  customer-node2/BROKER: Transporter: AmqpTransporter
events.js:377
      throw er; // Unhandled 'error' event
      ^

Error: listen EADDRINUSE: address already in use :::3030
    at Server.setupListenHandle [as _listen2] (net.js:1331:16)
    at listenInCluster (net.js:1379:12)
    at Server.listen (net.js:1465:7)
    at PrometheusReporter.init (C:\Users\Admin\AppData\Roaming\npm\node_modules\moleculer-
cli\node_modules\moleculer\src\metrics\reporters\prometheus.js:69:15)
```

2.

6.2. Debug trong php

6.2.1. chạy debug php trong vs code dùng launch curent open script

- Trước tiên cài các extension debug trong vs code cho php như php server, php debug (Xdebug).
- Trong vs code của tôi chỉ chạy được phần launch script.
- Tiếp theo phải add config



- Chạy Launch curent open script (script muốn chạy, script đầu nguồn gọi các scipt khác chạy. Thường sẽ là index.php hoặc file ở router).

6.2.2. Launch Build-in web server vs code - php

- Sẽ mở trình duyệt của bạn lên sau khi start debug. Thường sẽ ở một port random. Chú ý sửa lại url vì url này nằm trong warmserver hay xampp cho nên chúng không đúng. Hãy xóa tên folder trong url đi là được

6.2.3. .

6.3. Jquery

- KHông truyền data vào PUT trong ajax được:
  https://stackoverflow.com/questions/8032938/jquery-ajax-put-with-parameters

```
$.ajax({
  url: '/echo/html/',
  type: 'PUT',
  data: "name=John&location=Boston",
  success: function(data) {
    alert('Load was performed.');
  }
});

contentType: 'application/json',
```

-

6.4. .

7. Tham khảo

1. File pdf hướng dẫn của giảng viên.
2. Link demo hướng dẫn microservice: https://www.youtube.com/watch?v=RB1I3jWAIns&list=PLnKoBOgxxO68WAD4RsdfkrhZDpd5a4cy0&index=2
3. Thiết kế logo: https://www.fiverr.com/?source=top_nav,
4. Video moleculer nodejs: https://www.youtube.com/watch?v=Hjd22ZmXcS0
5. NATS một transporter (message broker) trong moleculer: https://docs.nats.io/nats-concepts/overview
6. Các transporter trong moleculer: https://moleculer.services/docs/0.12/transporters.html
7. Load service từ npm run dev: https://moleculer.services/docs/0.14/runner.html
8. Tham khảo demo sơ khai moleculer: https://medium.com/moleculer/moleculer-a-modern-microservices-framework-for-nodejs-bc4065e6b7ba
9. Sử dụng repl (read eval print loop) console khi chạy project trên vs code: https://moleculer.services/docs/0.14/moleculer-repl.html
10. video hướng dẫn cài rabbitmq: https://www.youtube.com/watch?v=V9DWKbalbWQ
11. Giải thích như thế nào là async trong giao tiếp của microservice: https://qr.ae/pv4chr https://qr.ae/pv4cnX
12. curl để call api trong php: https://viblo.asia/p/curl-va-cach-su-dung-trong-php-naQZRAXdKvx
13. Global, Constant, Session variable in php: https://stackoverflow.com/questions/39443875/how-can-i-define-a-variable-available-to-all-php-fileslike-superglobals
14. d