**Báo cáo Bài tập Quản lý tài khoản người dùng và xác thực**

|  |
| --- |
| **HỌC VIÊN KỸ THUẬT MẬT MÃ TP. HCM**  ¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯ |
| HỌC PHẦN  **THƯƠNG MẠI ĐIỆN TỬ** |
| BÁO CÁO BÀI TẬP  **Bài tập Quản lý tài khoản người dùng và xác thực**  MSSV: CT07N0108 Họ và tên: Trần Quang Diệu |
| |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |
| **TP.HCM, 2025** |

**MỤC LỤC**

Nhiệm vụ 1. Tạo chức năng Đăng kí tài khoản người dùng 1

Nhiệm vụ 2. Tạo chức năng Đăng nhập 1

Nhiệm vụ 3. Tạo chức năng Quên mật khẩu 2

Nhiệm vụ 4. Tạo chức năng Thay đổi mật khẩu 2

**\*Tổng quan về kiến trúc áp dụng để thực hiện:**

Hệ thống được thiết kế theo mô hình **MVC (Model – View – Controller)**, đảm bảo tính tách biệt rõ ràng giữa các thành phần:

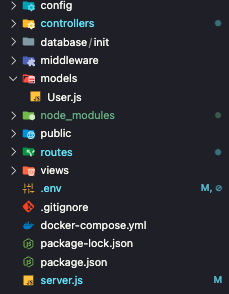
* **Model**: Quản lý dữ liệu, các lớp nghiệp vụ và kết nối với cơ sở dữ liệu.
* **View**: Chịu trách nhiệm hiển thị giao diện cho người dùng.
* **Controller**: Xử lý luồng điều khiển, tiếp nhận request từ người dùng, gọi Model để thao tác dữ liệu và trả kết quả cho View.

Việc áp dụng mô hình MVC không chỉ giúp mã nguồn dễ dàng bảo trì và mở rộng, mà còn góp phần nâng cao tính bảo mật nhờ sự phân chia trách nhiệm rõ ràng.

Bên cạnh đó, hệ thống còn kết hợp **middleware** để:

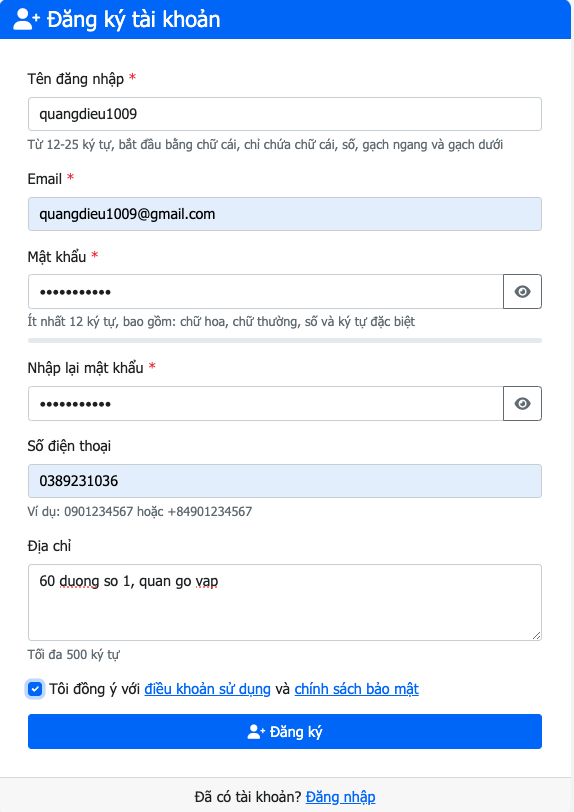
* **Xác thực (Authentication & Authorization)**: kiểm tra quyền truy cập trước khi vào các Controller quan trọng.
* **Validate input** ở cả **3 lớp (Model, Controller, Middleware)** nhằm giảm thiểu nguy cơ tấn công như SQL Injection, XSS, hay truyền dữ liệu không hợp lệ.
* **Logging và giám sát**: ghi nhận hoạt động bất thường để phục vụ việc phát hiện và xử lý sự cố bảo mật.

Nhờ cơ chế này, hệ thống vừa đảm bảo **tính đúng đắn của dữ liệu**, vừa nâng cao khả năng **phòng chống tấn công** và tăng độ **an toàn bảo mật** cho người dùng.



Cấu trúc dự án

1. **Tạo chức năng Đăng kí tài khoản người dùng**



***Mô tả chức năng***

* Chức năng đăng ký cho phép người dùng tạo tài khoản mới với các thông tin cơ bản như sau:
* Tên đăng nhập (12-25 ký tự, bắt đầu bằng chữ cái, số gạch gang và gạch dưới)
* Email hợp lệ
* Mật khẩu mạnh (tối thiểu 12 ký tự bao gồm: chữ hoa, chữ thường, số và ký tự đặc bieịt)
* Số điện thoại (tùy chọn)
* Địa chỉ (tùy chọn)

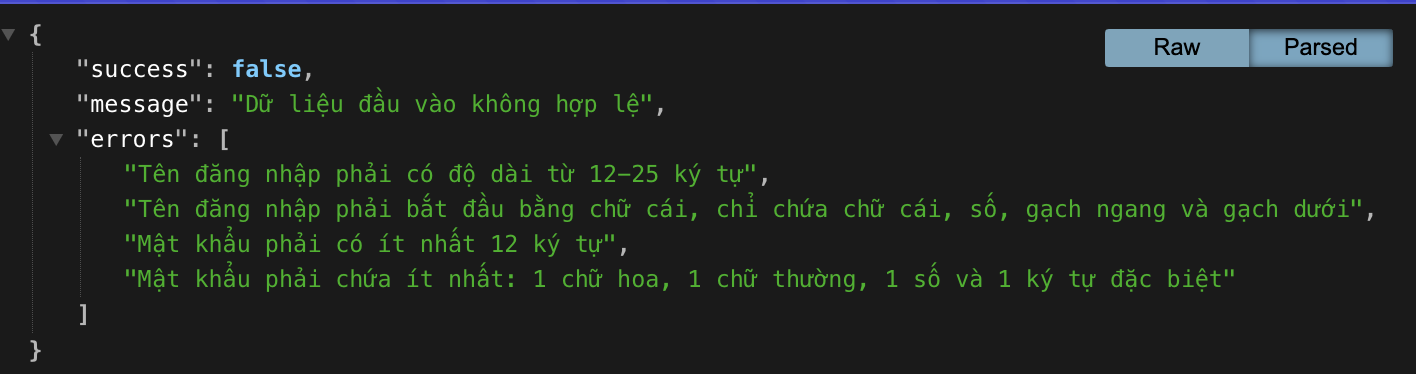
***Validation được áp dụng***

* Kiểm tra tên đăng nhập không trùng lặp và không sử dụng danh từ trong hệ thống
* Validation định dạng email input
* Chính sách yêu cầu nhập mật khẩu mạnh

**Chụp ảnh màn hình khi nhập và kết quả sau khi nhập dữ liệu đầy đủ, hợp lệ để đăng ký tài khoản người dùng:**



**Chụp ảnh kết quả trả về khi nhập dữ liệu vi phạm chính sách an toàn vào form đăng kí và dán vào dưới đây.**



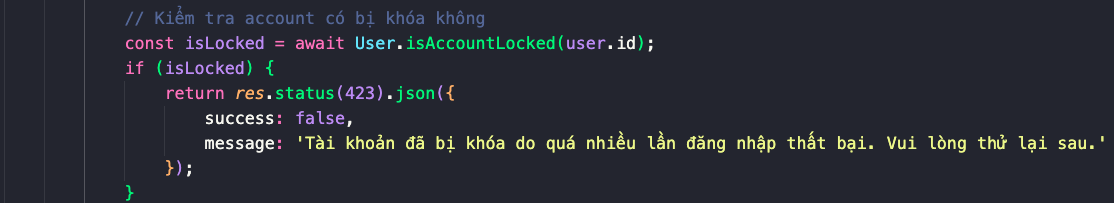
2. **Tạo chức năng Đăng nhập**



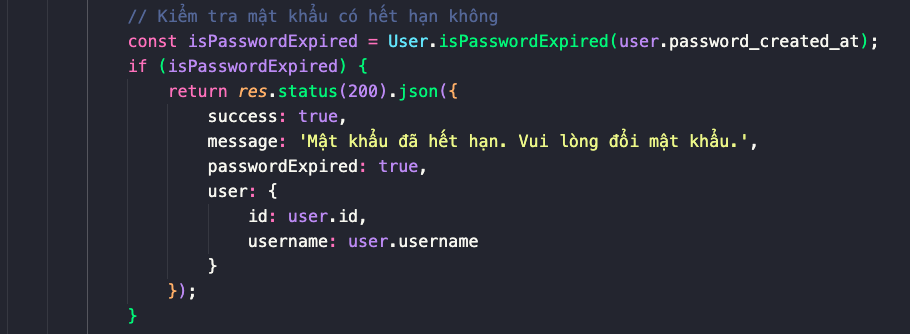
***Mô tả chức năng***

Chức năng đăng nhập xác thực người dùng. Hệ thống áp dụng:

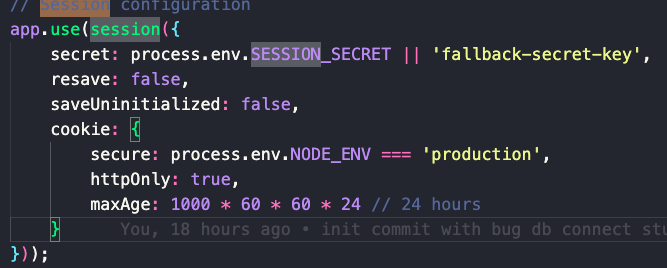
* Rate limiting theo địa chỉ ip để chống brute force



* Set policy sử dụng mật khẩu trong 60 ngày



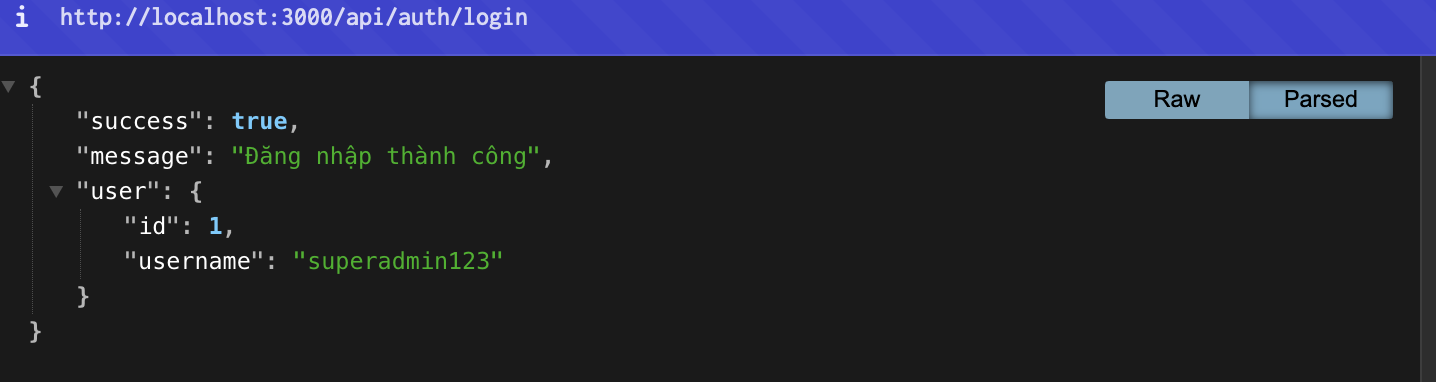
* Sử dụng session để duy trì đăng nhập của người dùng trong vòng 1 ngày



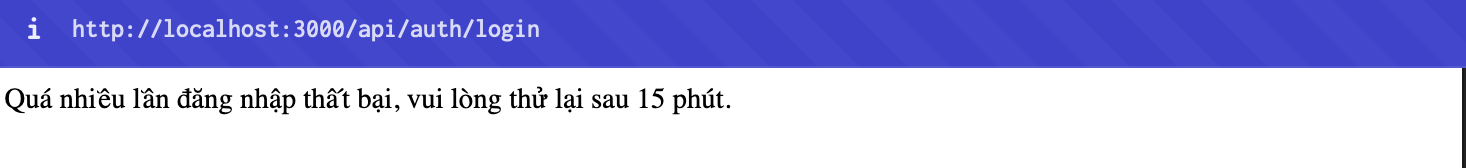
***Cơ chế bảo mật***

* Giới hạn chỉ 5 lần đăng nhập và 15 phút cho mỗi ip gửi request

Chụp ảnh kết quả khi đăng nhập thành công và dán vào dưới đây.



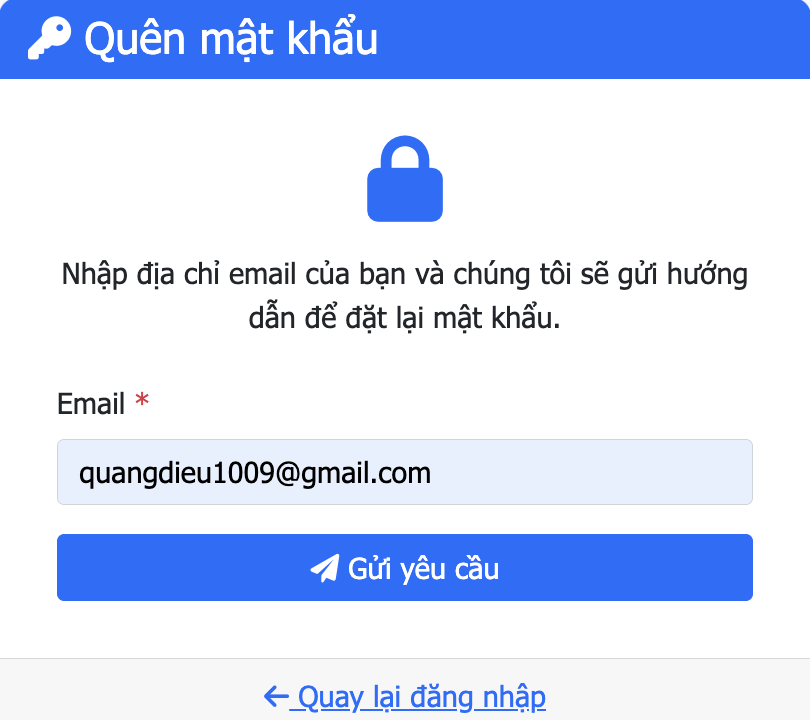
Chụp ảnh kết quả khi đăng nhập không thành công và dán vào dưới đây.



3. **Tạo chức năng Quên mật khẩu**

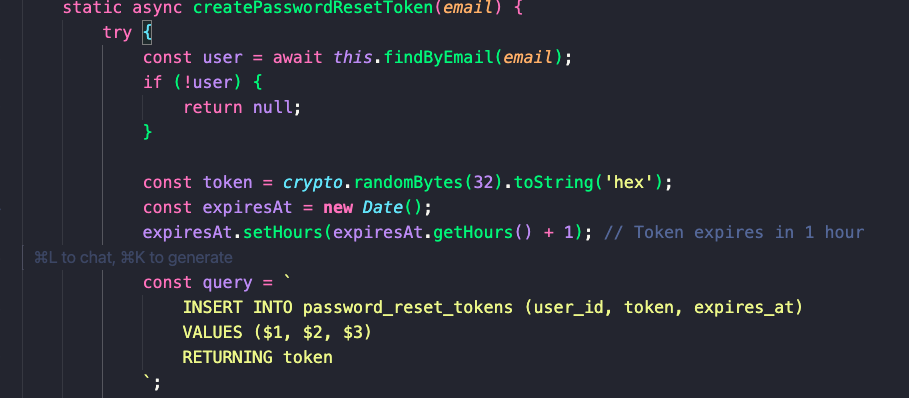
***Mô tả chức năng***

Chức năng quên mật khẩu cho phép người dùng đặt lại mật khẩu thông qua Email OTP.

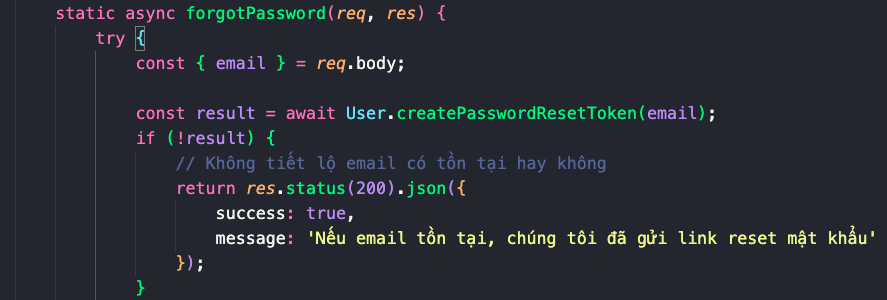


**Cơ chế bảo mật**

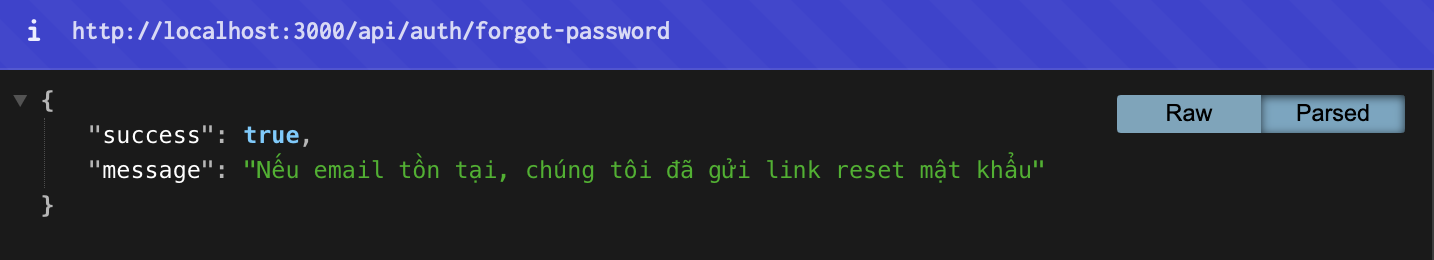
* Set hạn cho token đổi pass



* Không trả về kết quả email có tồn tại hay không để tránh thăm dò tài khoản email



Chụp ảnh kết quả khi thực hiện chức năng Quên mật khẩu thành công và dán vào dưới đây.

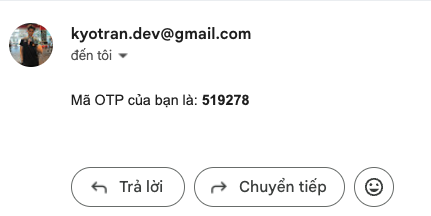


4. **Tạo chức năng Thay đổi mật khẩu**

***Mô tả chức năng***

Chức năng thay đổi mật khẩu cho phép người dùng đã đăng nhập cập nhật mật khẩu mới

1. Gửi email chứ OTP để cập nhật lại mật khẩu

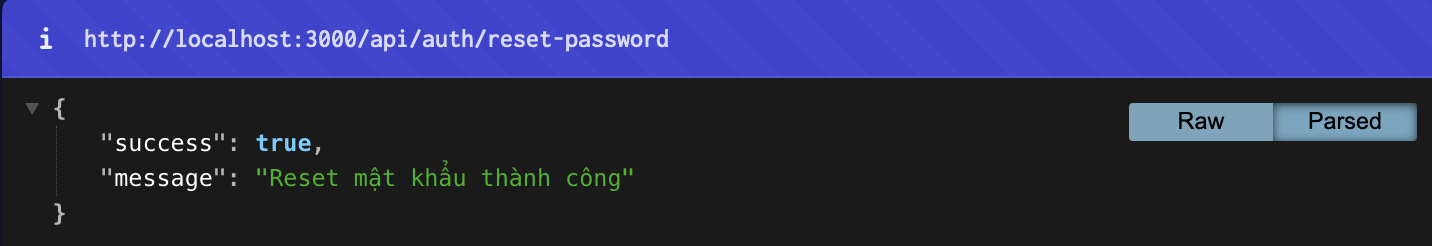


***Cơ chế bảo mật***

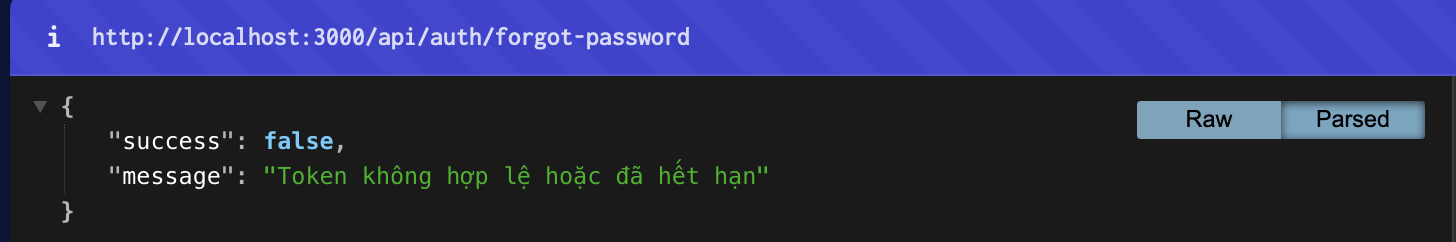
Tương tự đặt mật khẩu khi đăng ký, hash password với thuật toán **bcrypt** và kết hợp với salt tránh để tránh việc lộ thông tin khi database bị rò rỉ. Bằng cách này, ngay cả khi hacker lấy được dữ liệu, họ cũng khó có thể khôi phục lại mật khẩu gốc.



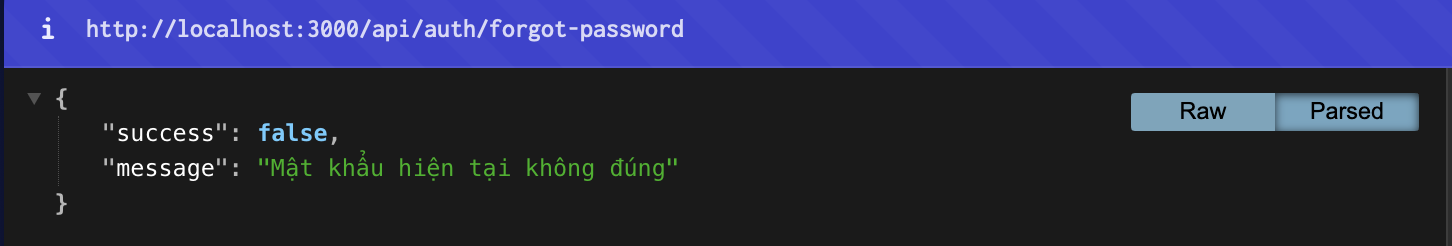
Chụp ảnh kết quả khi thay đổi mật khẩu thành công và dán vào dưới đây.



Chụp ảnh kết quả khi nhập thông tin mật khẩu mới không tuân theo chính sách an toàn và dán vào dưới đây.



Chụp ảnh kết quả khi nhập thông tin mật khẩu mới và mật khẩu nhập lại không trùng khớp và dán vào dưới đây.



**TỰ CHẤM ĐIỂM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tiêu chí đánh giá** | **Trọng số** | **Điểm tự chấm** | **Ghi chú / Chi tiết** |
| 1 | Chức năng đã cài đặt | 70% | 60% | - Đăng ký tài khoản với validation đầy đủ ở 3 lớp  - Duy trì đăng nhập với qua phiên đăng nhập(session)  - Quên mật khẩu với email OTP  -Thay đổi mật khẩu thành công  - Các chính sách bảo mật (ip limiting, lockout) |
| 2 | Thiết kế: Logic, dễ sử dụng, đẹp | 10% | 3% | - Có giao diện để nhập dữ liệu input |
| 3 | Tổ chức mã: MVC, Clean Architecture | 10% | 8% | - Tách biệt theo mô hình MVC - Clean Architecture chứa các layer tách biệt rõ ràng |
| 4 | Phong cách lập trình | 10% | 9% | - Nhất quán việc đặt tên biến,hàm theo naming convention.  - Code formatting và commenting đầy đủ .  - Handle try/catch lỗi đầy đủ và log hệ thống đầy đủ.  -Sử dụng docker để host DB và sử dụng git để dễ dàng phối hợp khi làm nhóm. |

**Chú ý: nếu không thực hiện đúng theo yêu cầu thì coi như không làm**

*Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.*