

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN



---

Báo cáo

**ĐỒ ÁN 1**

---

Môn học: An ninh máy tính

CSC15003\_22MMT

Sinh viên:

Lê Hoàng Đạt  
Nguyễn Hồ Đăng Duy  
Phạm Quang Duy

Giảng viên hướng dẫn:

Lê Giang Thanh  
Lê Hà Minh  
Phan Quốc Kỳ

## Mục lục

<b>1</b>	<b>Thông tin sinh viên</b>	<b>2</b>
<b>2</b>	<b>Giới thiệu</b>	<b>2</b>
<b>3</b>	<b>Phân công chi tiết</b>	<b>3</b>
3.1	Planning . . . . .	3
3.2	Implementation . . . . .	3
3.3	Merge . . . . .	4
3.4	Testing - Report - Demo . . . . .	4
<b>4</b>	<b>Kiến trúc hệ thống</b>	<b>5</b>
4.1	Sơ đồ tổng thể . . . . .	5
4.2	Thư mục và các modules chính . . . . .	5
<b>5</b>	<b>Chi tiết chức năng và kỹ thuật bảo mật</b>	<b>6</b>
5.1	Đăng ký tài khoản người dùng . . . . .	7
5.2	Đăng nhập và Xác thực đa yếu tố (MFA) . . . . .	10
5.3	Quản lý khoá RSA cá nhân . . . . .	14
5.4	QR Code Public Key . . . . .	15
5.5	Cập nhật thông tin tài khoản . . . . .	16
5.6	Mã hoá tập tin gửi người khác . . . . .	17
5.7	Giải mã tập tin . . . . .	18
5.8	Ký số tập tin . . . . .	19
5.9	Xác minh chữ ký . . . . .	20
5.10	Phân quyền tài khoản . . . . .	21
5.11	Ghi log bảo mật . . . . .	22
5.12	Chia nhỏ tập tin lớn . . . . .	23
5.13	Kiểm tra trạng thái khoá . . . . .	24
5.14	Tìm kiếm public key . . . . .	25
5.15	Giới hạn đăng nhập . . . . .	26
5.16	Tùy chọn định dạng lưu file . . . . .	27
5.17	Khôi phục tài khoản . . . . .	28
<b>6</b>	<b>Kiểm thử ứng dụng</b>	<b>29</b>
	<b>Tài liệu tham khảo</b>	<b>34</b>

## 1 Thông tin sinh viên

Nhóm gồm có 3 thành viên:

- 22127060 - Lê Hoàng Đạt - 22127060@student.hcmus.edu.vn
- 22127085 - Nguyễn Hồ Đăng Duy - 22127085@student.hcmus.edu.vn
- 22127088 - Phạm Quang Duy - 22127088@student.hcmus.edu.vn

## 2 Giới thiệu

**Secure Vault** là một ứng dụng mô phỏng hệ thống bảo mật theo mô hình thực tế, được xây dựng nhằm mục tiêu bảo vệ dữ liệu cá nhân và hỗ trợ truyền tin an toàn giữa người dùng. Ứng dụng tích hợp nhiều kỹ thuật bảo mật hiện đại như mã hóa RSA/AES, xác thực đa yếu tố (MFA), ký số – xác minh chữ ký, QR Code, cùng với cơ chế phân quyền, kiểm tra trạng thái khóa, ghi log bảo mật,... Người dùng có thể:

- Đăng ký, đăng nhập bảo mật với OTP/TOTP
- Tạo và quản lý cặp khóa RSA cá nhân
- Mã hóa và giải mã tập tin với AES + RSA
- Ký số tập tin và xác minh tính toàn vẹn
- Tạo và quét QR chứa public key để chia sẻ
- Theo dõi trạng thái khóa, phân quyền người dùng (admin/user)
- Khôi phục tài khoản khi quên mật khẩu

Hệ thống được xây dựng với ngôn ngữ `Python`, sử dụng `Flask` cho backend, kết hợp `MySQL/JSON` để lưu trữ dữ liệu, cùng các thư viện bảo mật như `pycryptodome`, `pyotp`, `qrcode`,...  
GitHub Repository: [https://github.com/YuD1405/Secure\\_Vault](https://github.com/YuD1405/Secure_Vault)

### 3 Phân công chi tiết

#### 3.1 Planning

Giai đoạn 1: Tìm hiểu và lên kế hoạch		
Thành viên	Nhiệm vụ	Thời hạn
Phạm Quang Duy	Tóm tắt thông tin đồ án, chia thành các modules	16/06
Phạm Quang Duy	Tạo repository github và chốt công nghệ	16/06
Phạm Quang Duy	Chia task	16/06

#### 3.2 Implementation

QUẢN LÝ NGƯỜI DÙNG			
Thành viên	Yêu cầu	Nhiệm vụ	Thời hạn
Nguyễn Hồ Đăng Duy	1	Đăng kí người dùng	17/06 - 19/06
Nguyễn Hồ Đăng Duy	2	Đăng nhập và xác thực MFA	18/06 - 20/06
Phạm Quang Duy	5	Cập nhật thông tin tài khoản	
Nguyễn Hồ Đăng Duy	10	Phân quyền Admin	20/06 - 23/06
Nguyễn Hồ Đăng Duy	15	Giới hạn login	19/06 - 20/06
Phạm Quang Duy	17	Khôi phục tài khoản	17/06 - 23/06
Nguyễn Hồ Đăng Duy	UI	Frontend: Form accounts, menu - navigation	29/01 - 09/03

MÃ HÓA VÀ GIẢI MÃ			
Thành viên	Yêu cầu	Nhiệm vụ	Thời hạn
Lê Hoàng Đạt	3	Quản lí khóa	
Lê Hoàng Đạt	4	Tạo và quét QR cho pub key	
Lê Hoàng Đạt	6	Mã hóa tập tin gửi	
Lê Hoàng Đạt	7	Giải mã tập tin	
Lê Hoàng Đạt	12	Chia nhỏ tập tin khi mã hóa	
Lê Hoàng Đạt	16	Tùy chọn định dạng lưu	23/06 - 23/06
Phạm Quang Duy	UI	Pop-up: QR, chọn file, loader khi mã hóa / giải mã	

SIGNING & LOGGING & UTILS			
Thành viên	Yêu cầu	Nhiệm vụ	Thời hạn
Phạm Quang Duy	8	Ký số	17/06 - 18/06
Phạm Quang Duy	9	Xác minh ký số	19/06 - 19/06
Phạm Quang Duy	11	Log bảo mật	20/06 - 20/06
Lê Hoàng Đạt	13	Kiểm tra trạng thái khóa	21/06 - 21/06
Phạm Quang Duy	14	Tìm pub Key	22/06 - 22/06
Nguyễn Hồ Đăng Duy	UI	Logging interface	23/06 - 27/06

### 3.3 Merge

Giai đoạn 3: Hoàn thành các chức năng và merge code		
Thành viên	Nhiệm vụ	Thời hạn
Nguyễn Hồ Đăng Duy	Kết nối: Group 1: User management → main UI	
Lê Hoàng Đạt	Kết nối: Group 2: Encrypt và Decrypt → main UI	
Phạm Quang Duy	Kết nối: Group 3: Logging và Utils → main UI	
Phạm Quang Duy	Kết nối toàn bộ chức năng (flow giữa các group chức năng)	

### 3.4 Testing - Report - Demo

Giai đoạn 4: Test chức năng, quay video và hoàn thiện report		
Thành viên	Nhiệm vụ	Thời hạn
Nguyễn Hồ Đăng Duy	Test group 1 + group 2	
Lê Hoàng Đạt	Test group 1 + group 3	
Phạm Quang Duy	Test group 2 + group 3	
Phạm Quang Duy	Test toàn bộ flow chương trình	
Nguyễn Hồ Đăng Duy	Report	
Nguyễn Hồ Đăng Duy	README.md	
Phạm Quang Duy	Record video	

## 4 Kiến trúc hệ thống

### 4.1 Sơ đồ tổng thể

### 4.2 Thư mục và các modules chính

Đồ án được tổ chức rõ ràng theo mô hình **Flask MVC** và phân chia modules theo chức năng bảo mật:

- `main.py` - Điểm khởi chạy của ứng dụng Flask
- `.env` - Tập tin cấu hình (thông tin DB, email, secret key,...)
- `data/` - Chứa dữ liệu hệ thống
  - `key_manage/` -
  - `keys/` -
  - `qr/`
- `flaskapi/` - Backend API
  - `app.py` - Cấu hình Flask app và đăng ký route
  - `routes/` - Chứa các tệp định nghĩa route theo nhóm chức năng
- `frontend/` - Các file HTML render giao diện với Jinja2
- `log/` - Ghi toàn bộ hành vi quan trọng: đăng nhập, mã hóa, ký số,... kèm thời gian, email, trạng thái
- `modules/` - Các modules chức năng chính
  - `auth/` - Hash passphrase, sinh OTP, xác thực MFA
  - `crypto/` - Mã hóa RSA/AES, ký số và xác minh chữ ký
  - `utils/` - Hàm tiện ích chung: log sự kiện, mail, QR,...
- `mySQL/` - Các chương trình để tạo database, table cho cơ sở dữ liệu MySQL
- `report/` - Báo cáo và video demo
- `README.md` - Hướng dẫn cài đặt và chạy thử đồ án. Mô tả các chức năng và công nghệ sử dụng.

## 5 Chi tiết chức năng và kỹ thuật bảo mật

### 5.1 Đăng ký tài khoản người dùng

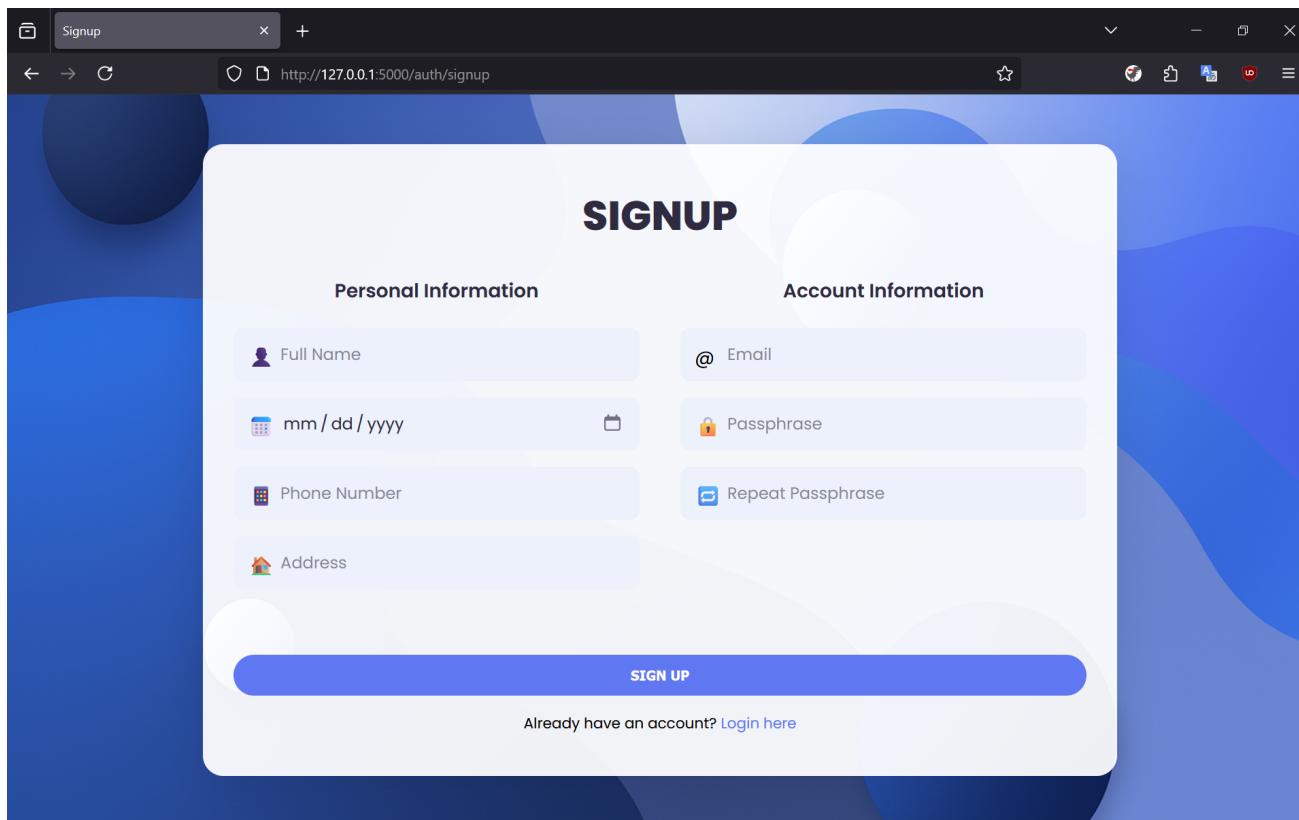
#### Mục tiêu

Cho phép người dùng tạo tài khoản bằng cách điền đầy đủ thông tin cá nhân và **passphrase** (đảm bảo đủ an toàn). Thông tin sau đó được kiểm tra tính hợp lệ, hash **passphrase** và lưu vào cơ sở dữ liệu. Đồng thời tạo **recovery key** và cung cấp cho người dùng.

#### Giao diện

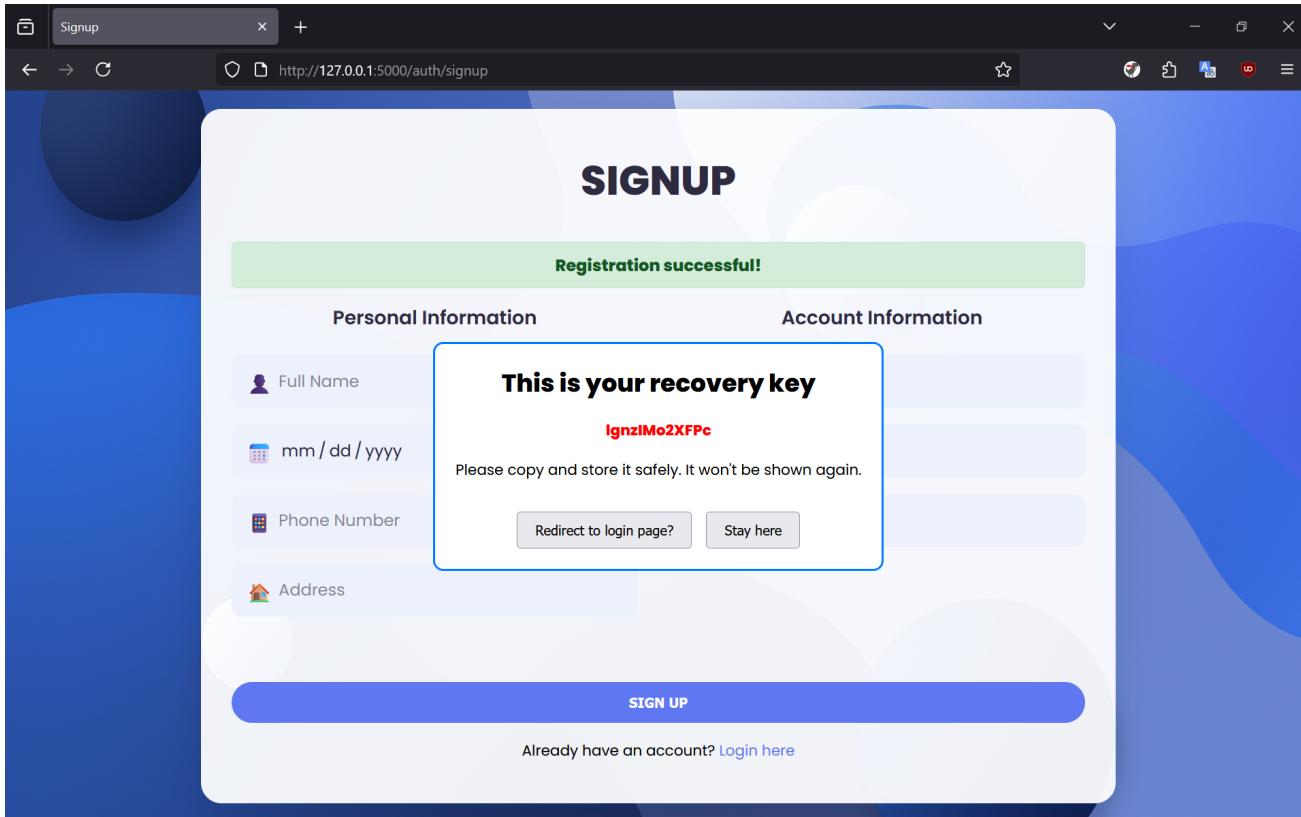
Giao diện `/auth/signup` là form HTML bao gồm các trường:

- Email
- Họ tên
- Ngày sinh
- Số điện thoại
- Địa chỉ
- Passphrase và xác nhận passphrase



Hình 1: Giao diện Signup

Khi đăng ký thành công, hệ thống sẽ hiển thị mã khôi phục tài khoản, yêu cầu người dùng lưu lại mã này để phục hồi nếu mất mật khẩu.



Hình 2: Giao diện popup recovery key

### Quy trình thực hiện

1. Người dùng nhập thông tin và submit form → POST `/signup`
2. Flask gọi hàm `register_user(request.form)` trong `logic.py`
3. Thông tin được kiểm tra và chuẩn hóa:
  - Email hợp lệ (`is_valid_email`)
  - Ngày sinh đúng định dạng (`is_valid_date`)
  - Số điện thoại gồm đúng 10 chữ số (`is_valid_phone`)
  - Passphrase đủ mạnh (`is_strong_passphrase`)
4. Nếu hợp lệ:
  - Sinh `salt` ngẫu nhiên
  - Băm `passphrase` kết hợp với salt bằng `SHA-256`
  - Sinh `recovery_code` để dùng cho khôi phục

- Sinh `mfa_secret` phục vụ TOTP trong MFA
5. Thực hiện `INSERT` bản ghi vào bảng `users` trong CSDL
  6. Ghi log hành động: `log_user_action(email, "Register", "Success")`
  7. Trả về kết quả thành công và hiển thị mã khôi phục

## Chi tiết kỹ thuật và thư viện bảo mật

### 1. Hashing passphrase với Salt

Thư viện: `hashlib, os`

Kỹ thuật:

- Salt được sinh bằng `os.urandom(16).hex()` → đảm bảo tính ngẫu nhiên mạnh.
- `passphrase` được hash bằng `SHA-256(passphrase + salt)` trước khi lưu xuống DB.

### 2. Validator kiểm tra đầu vào

Thư viện: `re, datetime`

Kỹ thuật:

- Regex kiểm tra định dạng email, số điện thoại.
- Kiểm tra `passphrase` phải ít nhất 8 ký tự, chứa chữ hoa, số, ký hiệu.
- Loại bỏ ký tự nguy hiểm khỏi input (`sanitize_input()`) → giảm rủi ro XSS / SQLi / Code Injection.

### 3. Tạo mã khôi phục

Thư viện: `random, string`

Kỹ thuật:

- Sinh chuỗi ngẫu nhiên gồm chữ + số, dùng 1 lần.
- Người dùng được yêu cầu lưu lại → dùng khi mất `passphrase`.
- Được lưu trong DB (`users.recovery_code`) và xóa sau khi reset thành công.

## 5.2 Đăng nhập và Xác thực đa yếu tố (MFA)

### Mục tiêu

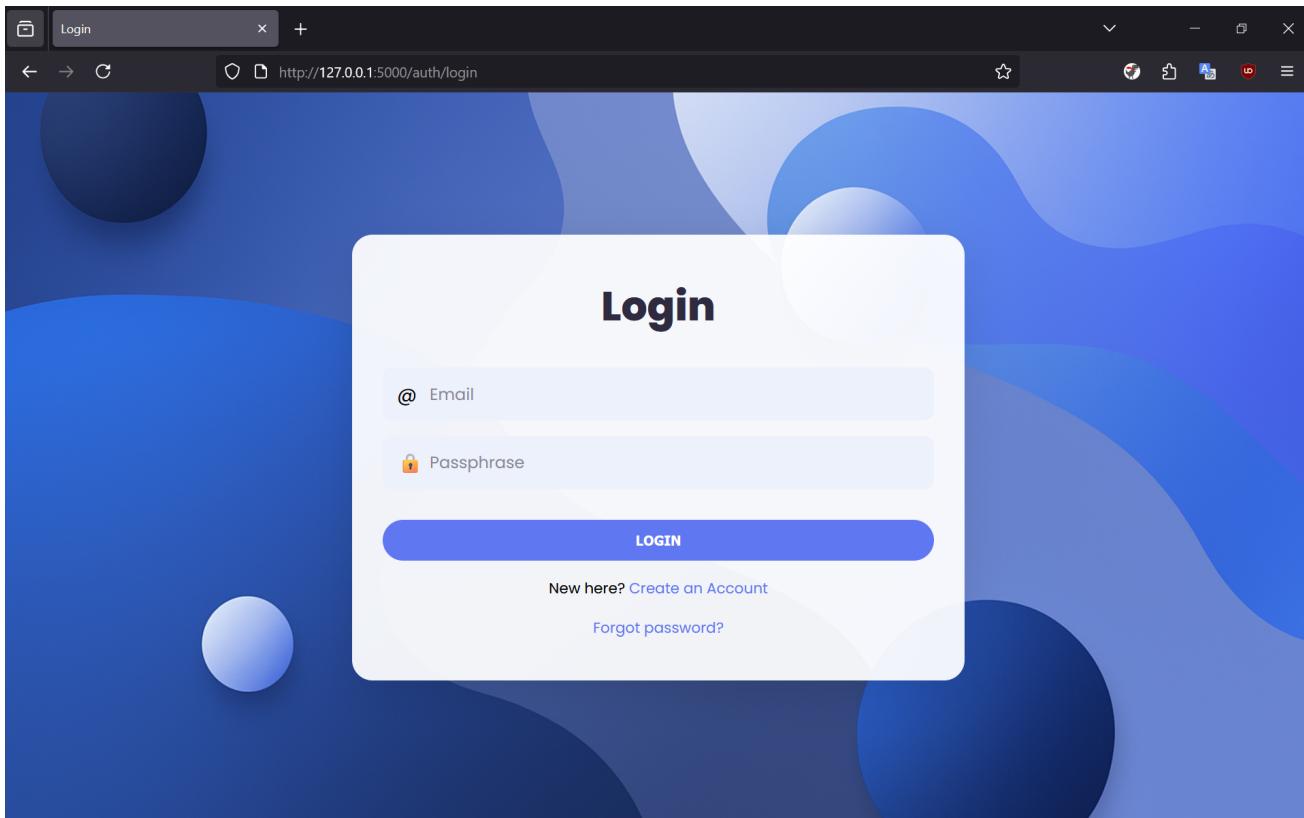
Cho phép người dùng đăng nhập bằng cách nhập `email` và `passphrase`. Nếu thông tin hợp lệ, hệ thống sẽ yêu cầu xác minh OTP qua email hoặc mã TOTP (Google Authenticator). Hệ thống hỗ trợ các biện pháp bảo mật nâng cao:

- Bảo vệ bằng xác thực 2 yếu tố
- Tự động khóa tài khoản 5 phút sau 5 lần đăng nhập sai
- Cho phép admin khóa tài khoản thủ công
- Ghi log toàn bộ hành vi đăng nhập

### Giao diện

Giao diện `/auth/login` là form HTML bao gồm 2 trường:

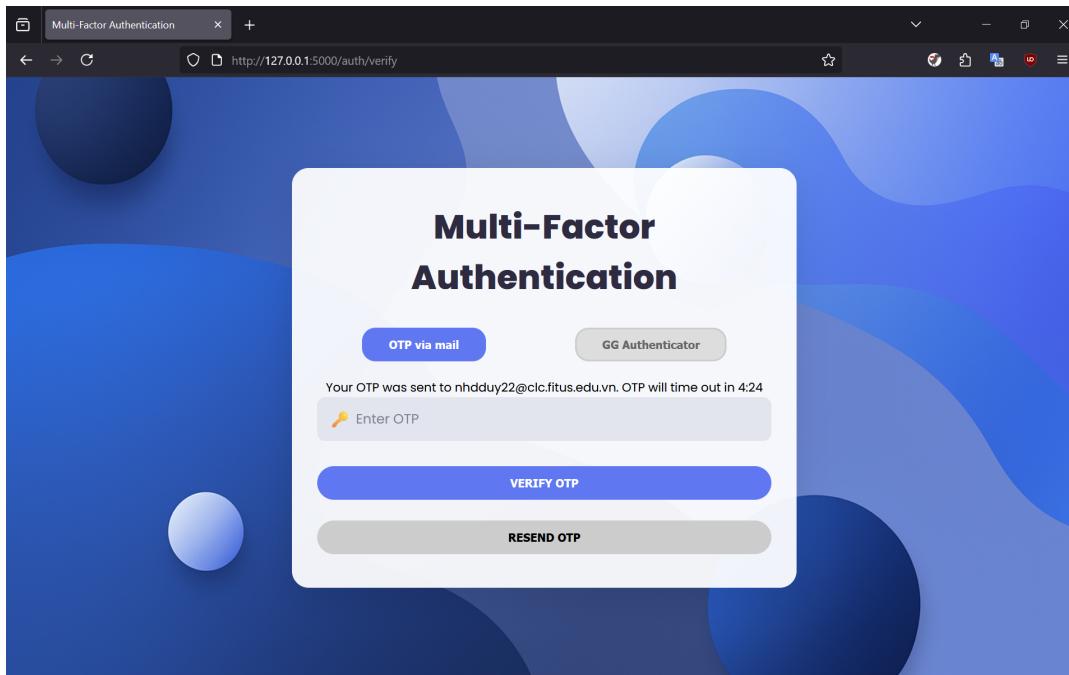
- Email
- Passphrase



Hình 3: Giao diện Login

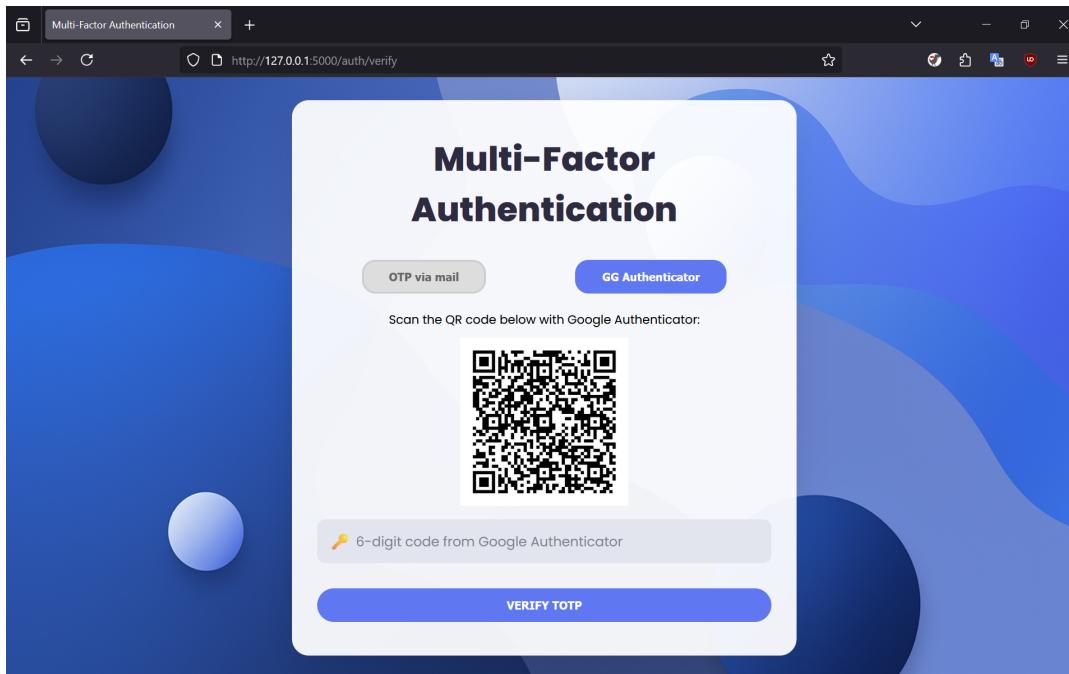
Sau khi đăng nhập thành công, người dùng được chuyển đến `/auth/verify` để xác thực MFA bằng 1 trong 2 phương thức:

## 1. OTP gửi qua email (mặc định)



Hình 4: Xác thực bằng OTP gửi qua mail

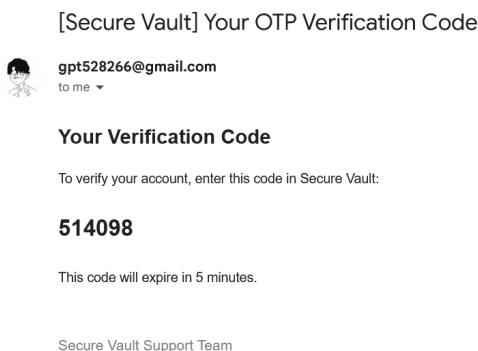
## 2. Mã TOTP (QR code cho Google Authenticator)



Hình 5: Xác thực bằng TOTP qua QR code

## Quy trình thực hiện

1. Người dùng submit form `/auth/login` với `email` và `passphrase`.
2. Flask gọi hàm `process_login(email, passphrase)` trong `logic.py`
3. Chương trình thực hiện:
  - Kiểm tra người dùng tồn tại hay không.
  - So sánh `hash(passphrase + salt)`.
  - Nếu sai, tăng `failed_attempts` và cập nhật `last_failed_login`.
  - Nếu sai hơn 5 lần trong 2 phút → Gán `is_locked = 1`, trả về trạng thái **locked**.
  - Nếu tài khoản bị khóa bởi admin (`last_failed_login = null` và `is_locked = 1`) → trả về **locked by admin**.
  - Nếu các thông tin đăng nhập đúng → Reset `failed_attempts`, ghi log, chuyển đến bước xác thực MFA.
4. Người dùng được chuyển đến `/auth/verify`, chọn xác thực OTP qua email hoặc quét QR TOTP (mặc định là gửi OTP qua email đã đăng ký).
5. Nếu chọn OTP:
  - Gọi `generate_and_send_otp(email)` → sinh mã 6 chữ số, gửi qua email, lưu DB kèm thời gian hết hạn.



Hình 6: Cấu trúc mail

- Kiểm tra bằng `verify_otp_code(email, input_code)`.
6. Nếu chọn TOTP:
  - Sinh mã QR từ `generate_qr_code(email)` → dùng cho ứng dụng Google Authenticator.
  - Kiểm tra bằng `verify_totp_code(email, input_code)`.
7. Nếu đúng → Lưu `session['user_id']` và chuyển đến `auth/dashboard`

## Chi tiết kỹ thuật và thư viện bảo mật

### 1. Hashing passphrase với Salt

Thư viện: `hashlib`

Kỹ thuật:

- Salt đã được sinh bằng `os.urandom(16).hex()` →.
- Kiểm tra `passphrase` được người dùng nhập vào sau khi hash có giống với chuỗi được lưu trong DB hay không.

### 2. Gửi mã OTP qua email

Thư viện: `random, datetime, smtplib, email`

Kỹ thuật:

- Sinh mã 6 chữ số ngẫu nhiên.
- Lưu `expires_at = now + 5 minutes`
- Dùng `email.mime.text` và `email.mime.multipart` để format tin nhắn gửi email.
- Tạo tài khoản gmail đã xác thực 2 yếu tố và tiến hành tạo `app password` và lưu vào `.env` với cấu trúc như sau

<sup>1</sup> `SMTP_USER=<sender mail>`

<sup>2</sup> `SMTP_PASS=<app password>`

- OTP được tạo sẽ lưu vào DB và gửi bằng SMTP → người dùng cần kiểm tra email để nhận OTP.
- Người dùng nhập mã OTP → Server sẽ lấy bản `otp_code` mới nhất và kiểm tra xem mã có đúng và còn hạn hay không → Nếu hợp lệ, người dùng xác thực thành công

### 3. TOTP

Thư viện: `pyotp, qrcode, base64`

Kỹ thuật:

- Mỗi tài khoản có một `mfa_secret` (chuỗi `base32`) sinh ngẫu nhiên và được lưu trong bảng `users`.
- Dùng `pyotp.totp.TOTP(mfa_secret).provisioning_uri()` để tạo ra URI định dạng chuẩn TOTP.
- Sinh ảnh QR code từ URI trên.
- Người dùng sử dụng Google Authenticator để quét mã trên.
- Người dùng nhập 6 số do app tạo ra mỗi 30 giây.
- Server sẽ dùng lại `mfa_secret` từ DB để sinh lại mã đúng tại thời điểm đó. Nếu mã số giống nhau thì người dùng xác thực thành công.

### 5.3 Quản lý khoá RSA cá nhân

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.4 QR Code Public Key

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.5 Cập nhật thông tin tài khoản

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.6 Mã hoá tập tin gửi người khác

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.7 Giải mã tập tin

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.8 Ký số tập tin

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.9 Xác minh chữ ký

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.10 Phân quyền tài khoản

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.11 Ghi log bảo mật

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.12 Chia nhỏ tập tin lớn

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

### 5.13 Kiểm tra trạng thái khoá

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.14 Tìm kiếm public key

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.15 Giới hạn đăng nhập

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.16 Tùy chọn định dạng lưu file

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 5.17 Khôi phục tài khoản

Mục tiêu

Giao diện

Quy trình thực hiện

Chi tiết kỹ thuật và thư viện bảo mật

## 6 Kiểm thử ứng dụng

Tất cả hình ảnh kết quả kiểm thử được cập nhật đầy đủ trong drive sau

### 6.1 Đăng ký tài khoản người dùng (/signup)

Mô tả kiểm thử	Input	Kết quả mong đợi
Đăng ký thành công	Email hợp lệ, Passphrase mạnh, các trường đầy đủ	Thông báo "Registration successful", pop-up <code>recovery_code</code>
Email sai định dạng	<code>abc@.com</code>	Báo lỗi "Invalid email format"
Passphrase yếu	<code>abc12345</code>	Báo lỗi "Passphrase too weak"
Trùng email đã đăng ký	Email đã có trong DB	Báo lỗi "Account already exists"
XSS/SQL injection	<code>&lt;script&gt;, " OR "1"="1</code>	Bị sanitize, không lỗi hệ thống

### 6.2 Đăng nhập và xác thực đa yếu tố (/login → /verify)

Mô tả kiểm thử	Input	Kết quả mong đợi
Đăng nhập đúng pass → chờ xác thực OTP	Email và passphrase đúng	Chuyển hướng tới trang <code>/verify</code> để xác thực OTP hoặc TOTP
Sai pass < 5 lần	Email đúng, passphrase sai	Báo lỗi "Wrong email or password" và tăng <code>failed_attempts</code>
Sai pass 5 lần	Nhập sai liên tiếp 5 lần	Khóa tài khoản 5 phút, thông báo thời gian chờ
OTP hợp lệ trong thời gian	Mã OTP 6 chữ số từ email (trong vòng 5 phút)	Xác thực thành công, chuyển đến <code>/dashboard</code> hoặc <code>/admin_dashboard</code>
OTP sai hoặc hết hạn	Mã OTP sai hoặc quá hạn 5 phút	Báo lỗi "Invalid or expired OTP"
TOTP đúng từ Google Authenticator	Mã 6 chữ số từ ứng dụng	Xác thực thành công, chuyển trang chính
TOTP sai	Nhập sai mã TOTP	Báo lỗi "Invalid TOTP code"

### 6.3 Quản lý khóa RSA cá nhân (/manage\_keys)

Mô tả kiểm thử	Input	Kết quả mong đợi
Tạo cặp khóa thành công	Bấm nút "Tạo khóa RSA" sau khi đăng nhập	Sinh file private và public '.pem', lưu DB với ngày tạo và hạn 90 ngày
Kiểm tra trạng thái khóa	Tài khoản đã có khóa	Hiển thị trạng thái: Còn hạn / Gần hết hạn / Hết hạn
Giải mã private key thành công	Passphrase đúng	Mở được private key để ký / giải mã
Giải mã private key thất bại	Passphrase sai	Báo lỗi "Unable to decrypt private key"

## 6.4 QR Code Public Key (/utils/qr)

Mô tả kiểm thử	Input	Kết quả mong đợi
Tạo QR thành công	Email có public key	Tạo mã QR base64, lưu file PNG
Quét QR thành công	File ảnh QR đúng định dạng	Hiển thị: email, public key, ngày tạo
Quét file ảnh sai định dạng	PNG không chứa mã QR hoặc bị lỗi	Thông báo "Không thể đọc mã QR"

## 6.5 Cập nhật thông tin tài khoản (/update\_account)

Mô tả kiểm thử	Input	Kết quả mong đợi
Cập nhật thông tin thành công	Họ tên, địa chỉ, SĐT đúng định dạng	Lưu thay đổi thành công, reload dữ liệu
Đổi passphrase thành công	Pass cũ đúng, pass mới đủ mạnh	Passphrase thay đổi, AES key tự cập nhật
Pass cũ sai	Nhập sai passphrase hiện tại	Báo lỗi "Passphrase hiện tại không đúng"
Pass mới yếu	Mới <8 ký tự hoặc không đủ yêu cầu	Báo lỗi "Passphrase mới quá yếu"

## 6.6 Mã hoá tập tin gửi người khác (/crypto/encrypt)

Mô tả kiểm thử	Input	Kết quả mong đợi
Mã hoá thành công (gộp file)	Chọn file + email người nhận có public key	Tạo file .enc chứa dữ liệu mã hoá và metadata
Mã hoá thành công (tách file)	Chọn lưu dạng tách	Sinh file .enc và .key riêng biệt
Người nhận không có public key	Nhập email chưa có key lưu trữ	Báo lỗi "Không tìm thấy public key của người nhận"
Metadata đúng định dạng	Sau khi mã hoá	Metadata gồm người gửi, thời gian, thuật toán, định dạng

## 6.7 Giải mã tập tin (/crypto/decrypt)

Mô tả kiểm thử	Input	Kết quả mong đợi
Giải mã thành công (file gộp)	File .enc gộp và passphrase đúng	Giải mã thành công, hiển thị file gốc và metadata
Giải mã thành công (file tách)	.enc + .key + passphrase đúng	Khôi phục file gốc đúng nội dung
Sai passphrase / khóa sai	Passphrase không đúng hoặc thiếu .key	Báo lỗi "Decryption failed" hoặc "Unable to decrypt key"
Tự động nhận dạng định dạng file	Dù là gộp hay tách	Tự động phân tích đúng định dạng và giải mã phù hợp

## 6.8 Ký số tập tin (/crypto/sign)

Mô tả kiểm thử	Input	Kết quả mong đợi
Ký số thành công	Chọn file bất kỳ + passphrase đúng	Tạo file chữ ký .sig theo định dạng SHA-256 + RSA
Thiếu private key hoặc passphrase sai	Không giải mã được private key	Báo lỗi "Không thể ký tập tin"
Log ký số đúng	Sau ký thành công	Ghi log: email người ký, thời gian ký, file đã ký

## 6.9 Xác minh chữ ký (/crypto/verify)

Mô tả kiểm thử	Input	Kết quả mong đợi
Xác minh đúng chữ ký	File gốc + file .sig đúng	Hiển thị email người ký, ngày ký, thông báo hợp lệ
Chữ ký sai hoặc không khớp	File bị thay đổi hoặc .sig giả mạo	Báo lỗi "Invalid signature" hoặc "Verification failed"
Không có public key người ký	Người ký chưa có key trong danh sách	Báo lỗi "Không tìm thấy public key phù hợp"

## 6.10 Phân quyền tài khoản

Mô tả kiểm thử	Input	Kết quả mong đợi
Admin truy cập trang quản lý	Người dùng có role = admin	Hiển thị danh sách tài khoản, log, nút khóa/mở tài khoản
Người thường truy cập trang admin	role = user	Báo lỗi "Access denied" và chuyển hướng về login
Khoá / mở tài khoản người dùng	Bấm nút khóa/mở trong giao diện admin	Cập nhật trạng thái khóa, lưu log thao tác

## 6.11 Ghi log bảo mật (security.log hoặc bảng log\_activity)

Mô tả kiểm thử	Input	Kết quả mong đợi
Ghi log đăng nhập thành công	Email + passphrase đúng	Log sự kiện "Login success" với trạng thái "Pending MFA"
Ghi log đăng nhập sai	Nhập sai passphrase	Ghi vào log với thông tin thất bại, timestamp, email
Ghi log ký số / cập nhật / mã hoá	Thao tác thành công các chức năng	Ghi đúng hành vi người dùng vào log theo chuẩn đã định

## 6.12 Chia nhỏ tập tin lớn khi mã hóa (>5MB)

Mô tả kiểm thử	Input	Kết quả mong đợi
File >5MB được chia nhỏ	Upload file >5MB	Hệ thống chia thành block 1MB, mã hóa từng block bằng AES-GCM
File nhỏ <5MB không chia	Upload file 2MB	Hệ thống mã hóa nguyên khối, không chia block
Kiểm tra toàn vẹn sau khi gộp lại	Giải mã file đã chia	Nội dung khôi phục đúng, không mất dữ liệu

## 6.13 Kiểm tra trạng thái khóa (/manage\_keys)

Mô tả kiểm thử	Input	Kết quả mong đợi
Khóa còn hạn	Ngày tạo < 60 ngày	Hiển thị trạng thái “Còn hạn”
Khóa gần hết hạn	Ngày tạo 80–89 ngày	Hiển thị “Gần hết hạn”
Khóa hết hạn	Ngày tạo > 90 ngày	Hiển thị “Hết hạn”, cho phép gia hạn hoặc tạo mới

## 6.14 Tìm kiếm public key (/keys/search)

Mô tả kiểm thử	Input	Kết quả mong đợi
Tìm thấy public key	Nhập email có public key trong DB	Hiển thị: email, QR code, ngày tạo, hạn dùng
Không tìm thấy public key	Email không tồn tại trong DB	Thông báo “Không tìm thấy khóa công khai”

## 6.15 Giới hạn số lần đăng nhập (/login)

Mô tả kiểm thử	Input	Kết quả mong đợi
Sai 5 lần liên tiếp	Nhập sai passphrase 5 lần trong 2 phút	Khoá tài khoản trong 5 phút, hiển thị thông báo thời gian chờ
Sau 5 phút thử lại	Đợi hết thời gian khoá	Tài khoản được mở lại và đăng nhập thành công nếu đúng pass
Log mỗi lần sai	Nhập sai liên tiếp	Ghi log với timestamp, lý do và email liên quan

## 6.16 Tùy chọn định dạng lưu file (/crypto/encrypt)

Mô tả kiểm thử	Input	Kết quả mong đợi
Lưu dạng gộp file	Chọn tùy chọn "Gộp .enc" khi mã hoá	Sinh 1 file duy nhất chứa toàn bộ dữ liệu và metadata
Lưu dạng tách file	Chọn "Tách .enc và .key" khi mã hoá	Sinh 2 file riêng biệt: dữ liệu mã hoá và AES key
Tự động nhận dạng khi giải mã	Upload file (gộp hoặc tách) khi giải mã	Tự động phân tích định dạng và xử lý phù hợp

## 6.17 Khôi phục tài khoản (/recover\_account, /verify\_recovery)

Mô tả kiểm thử	Input	Kết quả mong đợi
Khôi phục thành công	Nhập đúng recovery code hiển thị sau đăng ký	Cho phép đổi passphrase mới
Mã khôi phục sai hoặc hết hiệu lực	Mã recovery không đúng hoặc đã dùng	Báo lỗi "Invalid recovery code"
Đổi pass thành công sau xác minh	Pass mới đủ mạnh	Lưu pass mới, cập nhật salt mới

## Tài liệu

- [1] Lanchec, S. (2024, June 5). User Authentication: a guide for developers. Forest Admin Blog. <https://www.forestadmin.com/blog/user-authentication-a-guide-for-developers/>
- [2] The five different types of authentication — WorkOS. (n.d.). WorkOS. <https://workos.com/blog/the-five-different-types-of-authentication>
- [3] What is Authentication? Definition and uses - Auth0. (n.d.). Auth0. <https://auth0.com/intro-to-iam/what-is-authentication>
- [4] GeeksforGeeks. (2024, June 6). What is User Authentication, and Why is it Important? GeeksforGeeks. <https://www.geeksforgeeks.org/what-is-user-authentication-and-why-is-it-important/>
- [5] Frontegg. (2025, January 9). Authentication: What It is and How It Works | Frontegg. Frontegg. [https://frontegg.com/blog/authentication#API\\_Authentication\\_Methods](https://frontegg.com/blog/authentication#API_Authentication_Methods)
- [6] Sakshyam Shah. (2022, February 25). 6 Authentication best practices. Teleport. <https://goteleport.com/blog/authentication-best-practices/>
- [7] Intel® AMT SDK implementation and Reference Guide. (n.d.). [https://software.intel.com/sites/manageability/AMT\\_Implementation\\_and\\_Reference\\_Guide/default.htm?url=WordDocuments%2Fintroductiontokerberosauthentication.htm](https://software.intel.com/sites/manageability/AMT_Implementation_and_Reference_Guide/default.htm?url=WordDocuments%2Fintroductiontokerberosauthentication.htm)
- [8] Author, V., & Author, V. (2024, October 8). What is Kerberos? Kerberos Authentication Explained. VAADATA - Ethical Hacking Services. <https://www.vaadata.com/blog/what-is-kerberos-kerberos-authentication-explained/>
- [9] Wikipedia contributors. (2025, February 8). Kerberos (protocol). Wikipedia. [https://en.wikipedia.org/wiki/Kerberos\\_\(protocol\)](https://en.wikipedia.org/wiki/Kerberos_(protocol))