

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN – ĐHQG HCM
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN 1

Môn: An ninh máy tính

Chủ đề:

Ứng dụng chia sẻ file an toàn

Tp. Hồ Chí Minh – 07/2022

Thành viên

Mã số sinh viên	Họ và tên	Email
19127392	Tô Gia Hào	19127392@student.hcmus.edu.vn
19127525	Nguyễn Thanh Quân (Trưởng nhóm)	19127525@student.hcmus.edu.vn
19127625	Lâm Chí Văn	19127625@student.hcmus.edu.vn

Phân công

Tên	Công việc	Đánh giá
Tô Gia Hào	Task 6: Ký trên tập tin	100%
	Task 7: Xác nhận chữ kí	100%
Nguyễn Thanh Quân	Xây dựng giao diện cho ứng dụng (full responsive)	100%
	Kết nối MongoDB vào dự án	100%
	Task 1: đăng kí tài khoản người dùng	100%
	Task 2: Phát sinh cặp khóa bất đối xứng	100%
	Task 3: Cập nhập thông tin tài khoản	100%
Lâm Chí Văn	Task 4: Mã hóa tập tin	
	Task 5: Giải mã tập tin	

Tổng quan:

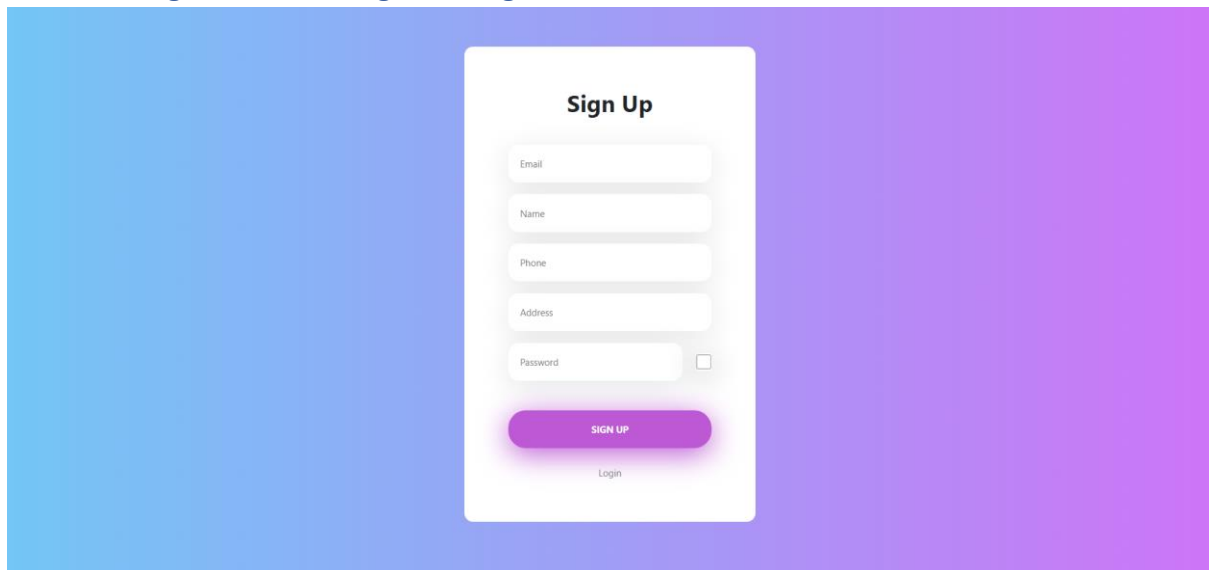
Vì ứng dụng chuyên về Cryptography nên nhóm quyết định dùng Python để viết ứng dụng, vì Python có khá nhiều thư viện hỗ trợ cho Cryptography

Thay vì chỉ làm giao diện Console, nhóm quyết định làm GUI cho dễ tương tác và sinh động vì vậy chọn Flask sẽ làm Framework để viết App

Giao diện GUI: HTML, CSS, JavaScript và Bootstrap 5

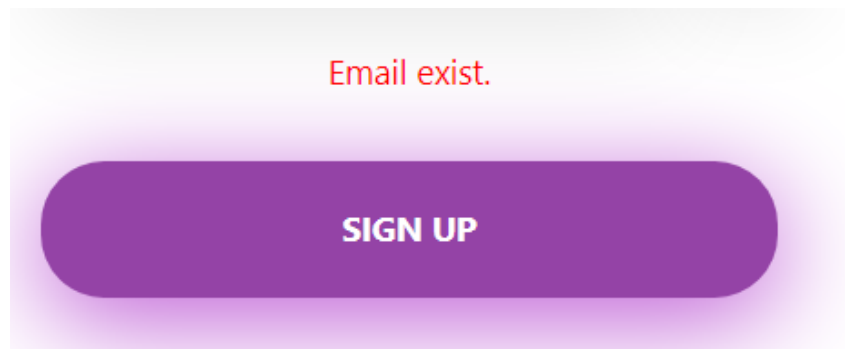
Cơ sở dữ liệu: MongoDB

Task 1: Đăng kí tài khoản người dùng

A sign-up form titled "Sign Up" is centered on a background with a blue-to-purple gradient. The form is white with rounded corners and contains several input fields: "Email", "Name", "Phone", "Address", and "Password". The "Password" field has a small square checkbox to its right. Below the input fields is a purple button with the text "SIGN UP" in white. At the bottom of the form, there is a link that says "Login".

Hình 1: Giao diện đăng kí tài khoản

Khi đăng kí, chương trình bắt buộc nhập đầy đủ thông tin vào các trường có trong form Đăng kí
Với trường hợp đăng kí bằng gmail đã đăng kí trước đó, chương trình sẽ thông báo lỗi



Hình 2: Lỗi đăng kí trùng tài khoản đã tồn tại

Mật khẩu được lưu trữ dưới hash có kết hợp salt nên mật khẩu sẽ có dạng

$\text{Salt} = \text{random}(32 \text{ byte})$

$\text{Hash_password} = \text{hash}(\text{password} + \text{salt})$

Mật khẩu được lưu vào database sẽ có dạng: $\text{passphrase} = \text{salt} + \text{hash_password}$

Vì mật khẩu được lưu vào database với cấu trúc như trên nên không cần phải lưu salt vào database.

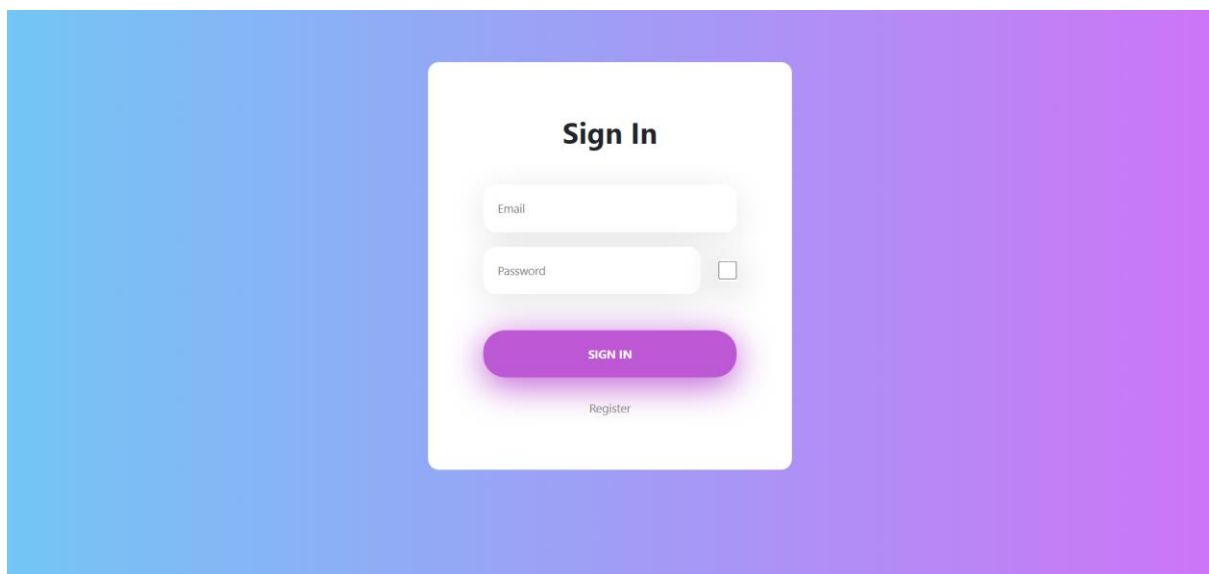
Khi authen chỉ cần tách passphrase ra thì ta sẽ có được salt và hash_password

```
salt = passphrase[:32]
```

```
hash_password = passphrase[32:]
```

Với việc tách như vậy, khi authen chỉ cần so sánh input_passphrase với passphrase (tương đương với email input) trong database

Với input_passphrase = hash(input_password + salt) (salt này mới lấy từ passphrase lưu trong database ở trên)



Hình 3: Giao diện đăng nhập

Task 2: Phát sinh cặp khóa bất đối xứng

phát sinh một cặp khóa(Kpublic, Kprivate)có độ dài là 2048 bit

Vì vậy sử dụng thư viện Crypto để generate cặp khóa có độ dài 2048 bit và lấy cặp khóa dưới dạng PEM

```
34 def gen_RSA_key_pem():
35     key = RSA.generate(2048)
36     pub_key_pem = key.publickey().exportKey().decode()
37     priv_key_pem = key.exportKey().decode()
38     return pub_key_pem, priv_key_pem
```

Hình 4: hàm phát sinh cặp khóa RSA

Sau khi có được cặp khóa RSA dưới dạng PEM, tiến hành mã hóa Khóa bí mật bằng thuật toán AES với mode CBC với nội dung là Khóa bí mật và Ksecret trong thuật toán AES là passphrase của người dùng

```
43 encrypted_priv_key = AES_encrypt(priv_key_pem, passphrase)
```

Hình 5: Mã hóa Khóa bí mật RSA bằng thuật toán AES

Sau khi có tạo cặp khóa và mã hóa khóa bí mật, ta lưu khóa công khai (PEM) và khóa bí mật đã được mã hóa vào database dưới dạng byte

```
41 def gen_user_RSA_key_pem(passphrase):
42     pub_key_pem, priv_key_pem = gen_RSA_key_pem()
43     encrypted_priv_key = AES_encrypt(priv_key_pem, passphrase)
44
45     return pub_key_pem.encode(), encrypted_priv_key
```

Hình 6: hàm xử lý cặp khóa RSA vừa phát sinh ở Hình 3

Task 3: cập nhật thông tin tài khoản

Sau khi đăng nhập thành công, chương trình sẽ lưu thông tin người dùng vào Session sau đó sẽ tự động redirect về trang chủ

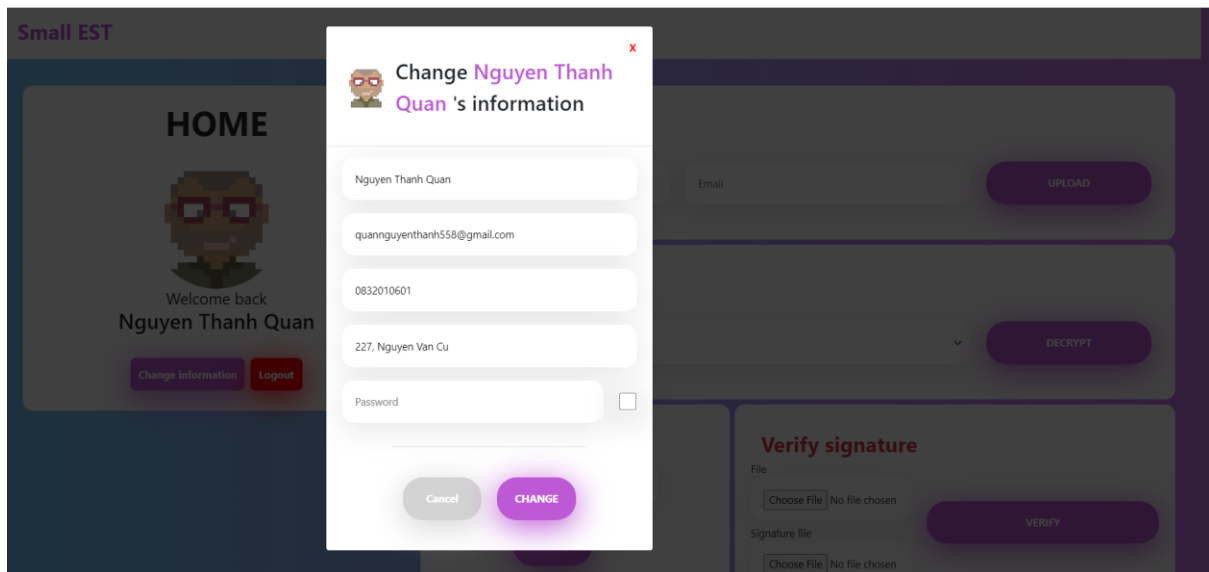
Việc lưu thông tin người dùng vào Session sẽ phục vụ cho việc Authorize (nếu có session user -> người dùng đã đăng nhập thành công, ngược lại: người dùng chưa đăng nhập)

Khi nhấn vào nút Logout, chương trình sẽ tự động xóa session user, để đảm bảo tránh trường hợp không đăng nhập vẫn vào được trang chủ

Small EST

Hình 7: Giao diện trang chủ

Nhấn nút Change Information để thay đổi thông tin người dùng



Hình 8: Popup thay đổi thông tin chi tiết

Khi thay đổi các thông tin (trừ password) ứng dụng sẽ cập nhật thẳng lên database mà không cần xử lý

Khi thay đổi password, chương trình sẽ:

Tạo lại passphrase mới như Task 1 với $\text{new_passphrase} = \text{new_salt} + \text{hash_new_password}$

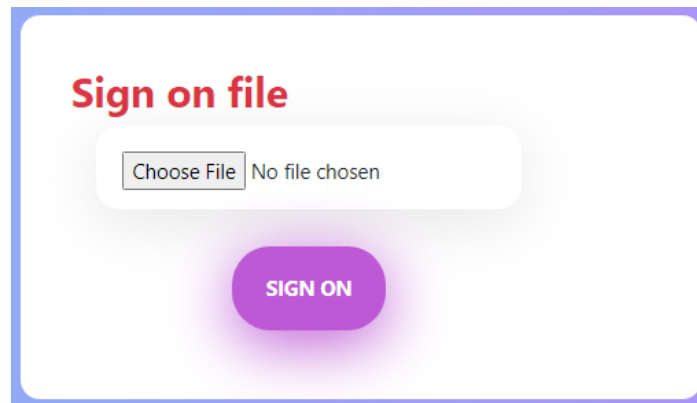
Giải mã khóa bí mật (đã được mã hóa) với khóa Ksecret là passphrase cũ (line 14)

Sau đó mã hóa lại khóa bí mật bằng thuật toán AES với Ksecret là passphrase vừa tạo

```
10 # change password
11 new_passphrase = authen.salt_hash256(new_info["passphrase"])
12
13 # change Private_key to ensure Integrity
14 origin_priv_key = cryptography.AES_decrypt(current_info["private_key"], current_info["passphrase"])
15
16 new_priv_key = cryptography.AES_encrypt(origin_priv_key, new_passphrase)
```

Hình 9: Quá trình xử lý, thay đổi passphrase mới

Task 6:



The interface for 'Sign on file' features a red title 'Sign on file' at the top. Below it is a white rounded rectangle containing a 'Choose File' button and the text 'No file chosen'. At the bottom center is a large, rounded purple button with the text 'SIGN ON' in white.

Hình 10: Giao diện Sign on File

Task 7:



The interface for 'Verify signature' has a red title 'Verify signature'. Below the title are two file selection sections. The first section, labeled 'File', contains a 'Choose File' button and 'No file chosen' text. The second section, labeled 'Signature file', also contains a 'Choose File' button and 'No file chosen' text. To the right of these sections is a large, rounded purple button with the text 'VERIFY' in white.

Hình 11: Giao diện Verify signature