

TRƯỜNG ĐẠI HỌC HỌC VĂN LANG  
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN MÔN HỌC

PYTHON NÂNG CAO

NGÀNH: CÔNG NGHỆ THÔNG TIN

**Tên Đồ án:**

**XÂY DỰNG TRANG TÍNH TOÁN ĐƠN GIẢN**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ**

**XÂY DỰNG TRANG WEB QUẢN LÝ**

SVTH: TRẦN TẤN PHÁT

MSSV: 2274802010644

LỚP: 241\_71ITSE31003\_01

GVHD: HUỖNH THÁI HỌC

Tp. Hồ Chí Minh – năm 2024

## LỜI CẢM ƠN

Viết một báo cáo đồ án môn học là một trong những việc khó nhất mà chúng em phải hoàn thành trong quá trình học một môn học. Trong quá trình thực hiện đồ án cuối kỳ chúng em đã gặp rất nhiều khó khăn và bối rối. Nếu không có những sự giúp đỡ và lời động viên chân thành của nhiều người có lẽ chúng em khó có thể hoàn thành tốt tiểu luận này. Đầu tiên chúng em xin gửi lời biết ơn chân thành đến thầy Huỳnh Thái Học và cô Nguyễn Thị Mỹ Linh đã là người trực tiếp hướng dẫn chúng em hoàn thành tiểu luận này. Những ý kiến đóng góp của thầy là vô cùng hữu ích, nó giúp chúng em nhận ra các khuyết điểm của đồ án. Cảm ơn thầy, cô và các bạn trường Đại học Văn Lang là những người đã cùng nhóm em sát cánh và trải nghiệm để hoàn thành đồ án môn học.

## NỘI DUNG CHÍNH CỦA ĐỒ ÁN

### ĐỒ ÁN HỌC PHẦN LẬP TRÌNH PYTHON NÂNG CAO

#### NỘI DUNG THỰC HIỆN:

1. XÂY DỰNG ỨNG DỤNG TÍNH TOÁN ĐƠN GIẢN
2. XÂY DỰNG APPLICATION QUẢN LÝ ...
3. XÂY DỰNG WEB QUẢN LÝ ....

## MỤC LỤC

<b>NỘI DUNG CHÍNH ĐỒ ÁN .....</b>	<b>2</b>
<b>CHƯƠNG 1: XÂY DỰNG MÁY TÍNH ĐƠN GIẢN</b>	
1. Giới thiệu: .....	4
2. Phân tích: .....	4
3. Chức năng: .....	5
4. Ưu nhược điểm: .....	8
5. Kết luận: .....	8
<b>CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG QUẢN LÝ</b>	
1. Giới thiệu: .....	9
2. Phân tích: .....	9
3. Chức năng: .....	9
4. Ưu nhược điểm: .....	13
5. Kết luận: .....	13
<b>CHƯƠNG 3: XÂY DỰNG TRANG WEB QUẢN LÝ</b>	
1. Giới thiệu: .....	14
2. Phân tích: .....	14
3. Chức năng: .....	15
4. Ưu nhược điểm: .....	19
5. Kết luận: .....	20
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>20</b>
<b>LINK GITHUB.....</b>	<b>20</b>

# CHƯƠNG 1: XÂY DỰNG MÁY TÍNH ĐƠN GIẢN

## 1. Giới thiệu

Trong lập trình giao diện người dùng (GUI) bằng Python, việc quản lý các thông tin cá nhân và cho phép người dùng tương tác với các chương trình bằng các văn bản được xây dựng bởi thành phần giao diện như Title, Tab, Frame và MessageBox đóng vai trò trong việc tổ chức sắp xếp giao diện. Ngoài ra, báo cáo này giúp nhận diện thông tin người dùng và có một máy tính bên cạnh( cộng, trừ, nhân, chia).

## 2. Phân tích

Các thư viện giao diện người dùng (GUI) mang đến cho Python nhiều tính năng phong phú, hỗ trợ phát triển các ứng dụng một cách trực quan và tương tác. Ngoài ra, thư viện GUI còn cung cấp các yếu tố đồ họa sinh động, phản hồi nhanh chóng, dễ dàng mở rộng nhằm để phục vụ các yêu cầu của người dùng.

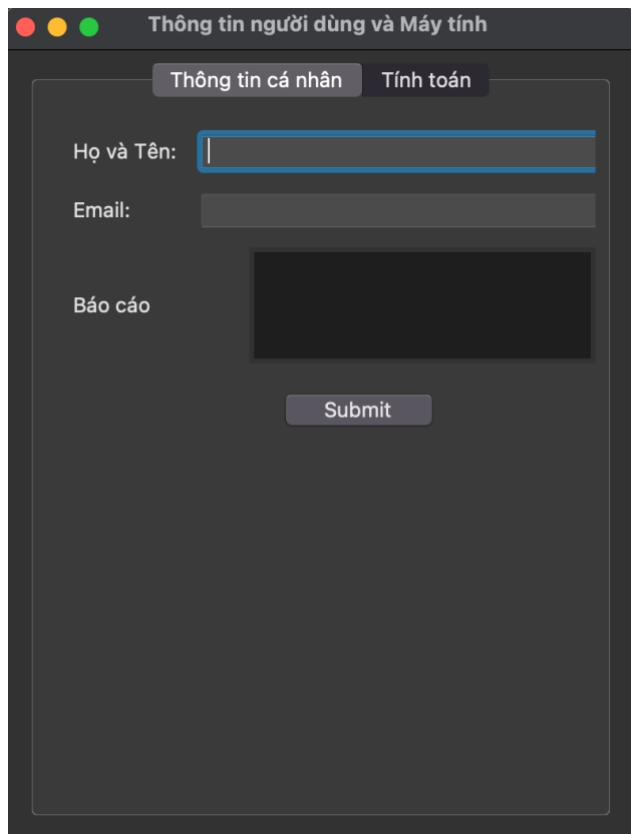
Sau đây là một vài phân tích về đề án bài 1 của tôi:

- Giao diện (Styling): Hiển thị và mô tả những thông tin ứng dụng, thường xuất hiện bên cửa sổ. Đây là thông tin quan trọng giúp người dùng biết giao diện họ sử dụng là gì.
- Widget cơ bản:
  - o Button: Dùng để kích hoạt hành động nào đó xảy ra. Vd: nút “Okay” dùng để đồng ý một hành động trong sự kiện đó.
  - o Label: Hiển thị các văn bản tĩnh, giúp người dùng hiểu rõ hơn nội dung và chức năng.
  - o TextBox: Cho phép người dùng nhập thông tin vào đó. Vd: Hệ thống sử dụng TextBox nhằm lấy ý kiến của người dùng về hệ thống.
- MessageBox: Hiển thị các thông báo, cảnh báo hoặc yêu cầu xác nhận từ người dùng.
- Tabs: Dùng để chia rõ từng chức năng mà người dùng muốn sử dụng khi cần thiết. Vd: Chia ra 2 Tabs là Đăng nhập và Đăng ký cho người dùng.

## 3. Chức năng

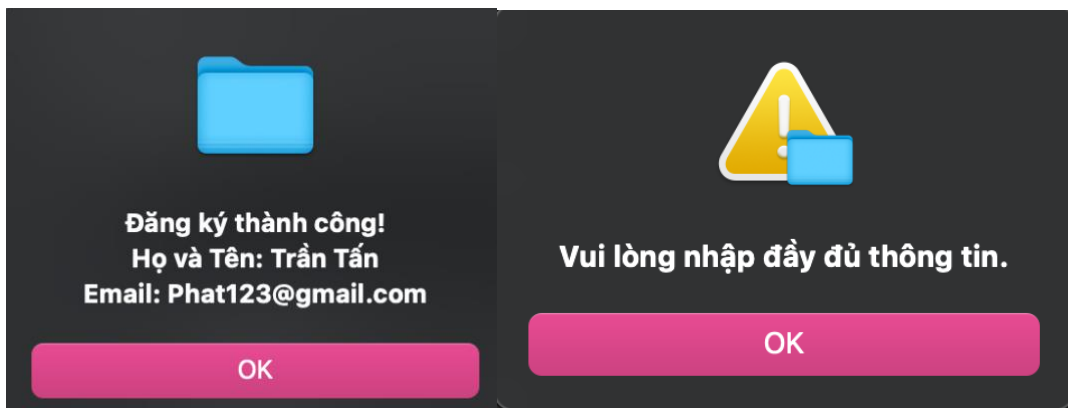
Giúp người dùng có thể nhập thông tin cá nhân, báo cáo lỗi khi sử dụng và có một máy tính có thể tính cộng, trừ nhân chia khi cần thiết. Ngoài ra, các thư viện

hỗ trợ người dùng các thành phần trực quan, dễ thao tác và mở rộng cho các ứng dụng phức tạp. Dưới đây là chi tiết chức năng của đồ án bài 1 trên:



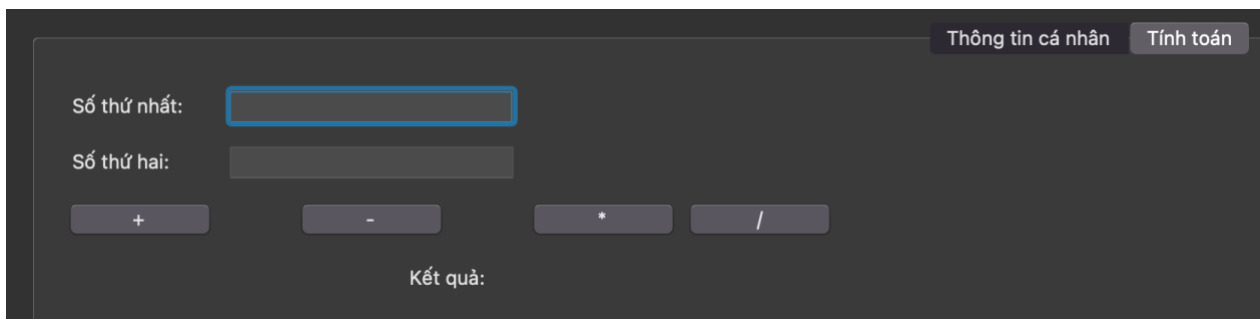
- a. Cửa sổ chính và giao diện cá nhân
  - Cửa sổ chính: Sử dụng giao diện Tkinter (tk.Tk()). Đặt tiêu đề giao diện là “Thông tin cá nhân” và “Máy Tính”
  - Notebook: Sử dụng ttk.Notebook để tạo giao diện tab giúp nhận diện và phân chia ứng dụng thành 2 thành phần chức năng chính
- b. Tab “Thông tin cá nhân”
  - Tab này dùng để thu thập thông tin của người dùng và hiển thị thông báo sau khi đăng ký.
  - Nội dung chính trong Tab:
    - Tk.Entry: Dùng để nhập họ tên và email của người dùng.
    - Báo cáo: tk.Text là hộp văn bản để nhập thông tin báo cáo từ người dùng.
    - Submit: ttk.Button để xác nhận thông báo đã nhập vào

- Kết quả được xuất ra sau khi người dùng nhập thông tin vào.
- Xử lý dữ liệu sự kiện:
  - o On\_submit: Được gọi khi người dùng nhấn nút “Submit”
  - o Kiểm tra email và họ tên người dùng, nếu không nhập thông tin sẽ hiển thị thông báo và cảnh báo.
  - o Sau khi xuất các thông tin người dùng, xóa các nội dung trước đã nhập.



c. Tab “Tính toán”

- Tab này thực hiện các phép tính toán học cơ bản giữa 2 số do người dùng nhập vào.
- Nội dung Widget trong Tab:
  - o Số thứ 1: ttk.Entry để nhập số thứ 1.
  - o Số thứ 2: ttk.Entry để nhập số thứ 2.
  - o Các nút tính toán: Bốn nút “+”, “-“, “\*”, “/” cho phép tính cộng trừ nhân chia
  - o Nhấn kết quả: ttk.Label để hiển thị kết quả tính toán.



Thông tin cá nhân    Tính toán

Số thứ nhất: 8

Số thứ hai: 9

+   -   \*   /

Kết quả: 72.0

- Xử lý sự kiện: Hàm calculate được gọi khi người dùng nhấn vào một nút tính toán
  - Lấy số đầu và số thứ 2 từ các trường hợp nhập bởi người dùng
  - Thực hiện phép tính tương ứng với nút đã nhấn
  - Nếu người dùng nhập sai thông tin dữ liệu, hệ thống sẽ đưa ra thông báo “Lỗi, vui lòng nhập số hợp lệ”
  - Hiển thị phép tính trong result\_label.

Thông tin cá nhân    Tính toán

Số thứ nhất:

Số thứ hai:

+   -   \*   /

Lỗi: Vui lòng nhập số hợp lệ

#### 4. Ưu nhược điểm

- Ưu điểm:
  - Sử dụng giao diện phân tách với Tab
  - Quản lý lỗi khi người dùng nhập thông tin sai
  - Phân tách Logic
- Nhược điểm:
  - Thiếu xác định định dạng email cho người dùng
  - Chưa tối ưu hoá giao diện
  - Thiếu các cơ chế bảo mật thông tin cá nhân cho người dùng
  - Giao diện quá đơn giản, thiếu tính thẩm mỹ.

## 5. Kết luận

- Đoạn code là một ví dụ cơ bản rõ ràng về cách sử dụng thư viện Tkinter trong Python để xây dựng ứng dụng giao diện người dùng (GUI). Ứng dụng này cung cấp 2 chức năng chính: “Thông tin người dùng” và “Máy Tính”, được chia ra độc lập nhằm giúp giao diện dễ sử dụng hơn.
- Mặc dù chức năng ứng dụng cơ bản, đơn giản nhưng đoạn code đã mang đến việc xây dựng GUI như quản lý lỗi, xử lý sự kiện. Tuy nhiên, cần phải thêm các tính năng xác thực dữ liệu nhằm cải thiện chức năng và giao diện trực quan hơn.



## CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG QUẢN LÝ

### 1. Giới thiệu

Ứng dụng trên được sử dụng bằng ngôn ngữ Python với thư viện tkinter để quản lý thông tin sinh viên. Nó cho phép xóa, tải lại danh sách sinh viên và hiển thị dữ liệu trong một bảng. Dữ liệu sinh viên bao gồm thông tin tên, tuổi, giới tính, ngành học. Ứng dụng kết nối với cơ sở dữ liệu PostgreSQL và thực hiện các thao tác trên bảng “info”.

### 2. Phân tích

- Kết nối cơ sở dữ liệu (connect\_db):
  - o Hàm connect\_db thiết lập kết nối với cơ sở dữ liệu PostgreSQL, tạo bảng “info” nếu vẫn chưa tồn tại.
  - o Bảng “info” có chứa các thông tin như id, name ,age, gender, major.
- Chức năng thêm sinh viên (add\_student):
  - o Hàm này lấy dữ liệu từ các trường hợp được nhập (name, age, gender, major).
  - o Kiểm tra tuổi người dùng, sau đó thêm sinh viên vào bảng “info” trong cơ sở dữ liệu.
  - o Sau khi thêm, các ô thêm thông tin được làm trống và danh sách sinh viên được làm mới để cập nhật.
- Chức năng xóa sinh viên (delete\_student):
  - o Lấy sinh viên được chọn từ dashboard, xác nhận qua hộp thoại, và sau đó xóa sinh viên khỏi cơ sở dữ liệu.
  - o Danh sách sẽ tự động cập nhật sau khi xóa.
- Giao diện người dùng:
  - o Top Frame: Chứa các trường nhập thông tin như : Name, age , gender, major.
  - o Middle Frame: Chứa các chức năng như: Thêm sinh viên, Xóa sinh viên, Tải lại danh sách.
  - o Bottom Frame: Chứa các danh sách sinh viên được cập nhật.

### 3. Chức năng

- Connect\_db: kết nối với cơ sở dữ liệu PostgreSQL bằng thông tin đăng nhập
- Widget Treeview: Hiển thị danh sách sinh viên với cột id, tên , tuổi , giới tính, ngành học.

```
def connect_db():
    try:
        conn = psycopg2.connect(
            database="postgres",
            user="postgres",
            password="phattan112",
            host="localhost",
            port="5432"
        )

        cursor = conn.cursor()
```

*Hệ thống kết nối vào cơ sở dữ liệu PostgreSQL*

- Tạo bảng “info” gồm các thông tin:
  - Id: khoá chính, kiểu SERIAL
  - Name: tên sinh viên, kiểu VARCHAR(100)
  - Age: tuổi sinh viên, kiểu INTEGER
  - Gender: giới tính sinh viên, kiểu VARCHAR(10)
  - Major: ngành học, kiểu VARCHAR(100)

```
cursor = conn.cursor()
cursor.execute("""
    CREATE TABLE IF NOT EXISTS info (
        id SERIAL PRIMARY KEY,
        name VARCHAR(100) NOT NULL,
        age INTEGER,
        gender VARCHAR(10),
        major VARCHAR(100)
    )
""")
conn.commit()
cursor.close()
return conn
except Exception as e:
    print("Error connecting to the database:", e)
    return None
```

*Tạo bảng tên “info” gồm các thông tin*

- Add\_student: Thu thập dữ liệu từ các trường nhập gồm name, age, gender, major của sinh viên.
- Kiểm tra tính hợp lệ của dữ liệu:
  - Trường name và age là bắt buộc
  - Kiểm tra số tuổi trên age có phải số nguyên không
- Nếu hợp lệ sẽ thực hiện lệnh INSERT để thêm sinh viên vào “info”

- Sau khi thực hiện xong, làm trống các trường nhập lại.

Quản Lý Thông Tin Sinh Viên

Tên: Trần Tấn Phát

Tuổi: 19

Giới tính: Nam

Ngành học: CNTT

Thêm sinh viên

Xóa sinh viên

Tải lại danh sách

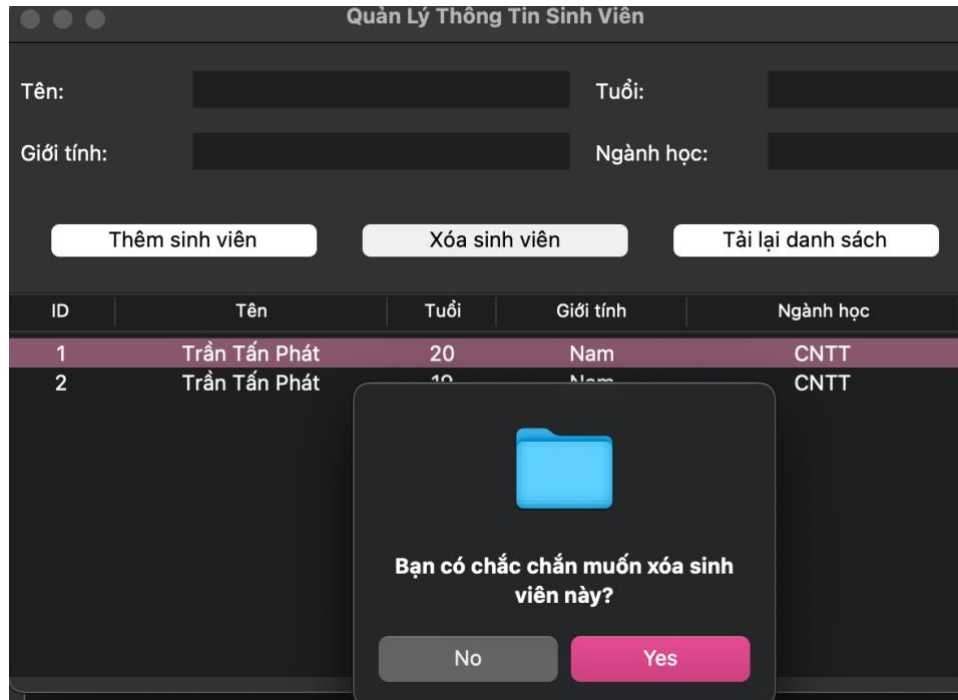
ID	Tên	Tuổi	Giới tính	Ngành học
1	Trần Tấn Phát	20	Nam	CNTT

ID	Tên	Tuổi	Giới tính	Ngành học
1	Trần Tấn Phát	20	Nam	CNTT
2	Trần Tấn Phát	19	Nam	CNTT

*Tạo thêm sinh viên, sau khi thêm thành công sẽ cập nhật lại thông tin*

- Delete\_student:

- Lấy mục sinh viên đã chọn từ Treeview để xoá.
- Hiển thị hộp thoại xác nhận điều trên.
- Nếu người dùng đồng ý sẽ kết nối vào cơ sở dữ liệu, thực hiện lệnh DELETE để xoá sinh viên có id tương ứng.
- Gọi lại hành load\_data sau khi xoá.



*Ứng dụng hiển thị hộp thoại xác nhận yêu cầu người dùng*

ID	Tên	Tuổi	Giới tính	Ngành học
2	Trần Tấn Phát	19	Nam	CNTT

*Sau khi xoá thông tin sẽ cập nhật lại ứng dụng*

#### 4. Ưu nhược điểm

- Ưu điểm:
  - Giao diện trực quan: Sử dụng thư viện tkinter để xây dựng giao diện GUI với các trường nhập liệu, nút điều khiển và widget Treeview để hiển thị danh sách sinh viên.
  - Hiển thị và tạo mới tự động: Sau khi thêm hoặc xóa người dùng, danh sách hiển thị sẽ tự động cập nhật, đảm bảo tính đồng bộ với cơ sở dữ liệu hiển thị.
  - Kiểm tra tính hợp lệ dữ liệu: Trường “name” và “age” được kiểm tra nhằm đảm bảo người dùng không bỏ trống.
- Nhược điểm:
  - Thiếu cơ chế bảo mật: Hàm connect\_db trả về với đối tượng kết nối nhưng đóng kết nối không an toàn
  - Thiếu cơ chế cập nhật thông tin sinh viên gây hạn chế tính linh hoạt của ứng dụng.
  - Giới hạn nhập liệu: Các trường không có giới hạn nhập liệu, dẫn đến khả năng nhập sai kiểu dữ liệu.

#### 5. Kết luận

- Đoạn code về ứng dụng quản lý trên là một ứng dụng quản lý sinh viên đơn giản, được xây bằng ngôn ngữ Python và thư viện tkinter để tạo giao diện người dùng (GUI). Mục tiêu chính nhằm quản lý thông tin sinh viên gồm tên, tuổi, giới tính, ngành học thông qua một giao diện trực quan.
- Ứng dụng này là nền tảng cơ bản tốt cho việc quản lý thông tin và có thể mở rộng thêm nhiều chức năng mới và phức tạp hơn nếu cần.

## CHƯƠNG 3: XÂY DỰNG TRANG WEB QUẢN LÝ

### 1. Giới thiệu

Trong Python nói chung và các thư viện riêng lẻ của Flask nói riêng, các thư viện như Flask, Flask\_SQLAlchemy đã được kết hợp với nhau để tạo nên một ứng dụng Web có khả năng lưu trữ thông tin người dùng vào cơ sở dữ liệu, đăng xuất, đăng nhập, và quản lý session một cách dễ dàng. Ngoài ra, các thư viện trên còn kết nối với các ngôn ngữ có liên quan đến xây dựng trang web và thiết kế đồ họa giao diện như HTML, CSS,...

### 2. Phân tích

- Các thư viện kết hợp trong Python mang đến cho người dùng nhiều chức năng phong phú, phát triển nhiều hệ thống. Ngoài ra, các ngôn ngữ hỗ trợ khác như HTML và CSS cung cấp các yếu tố đồ họa sinh động, xây dựng trang web giúp phục vụ được các yêu cầu thiết yếu của người dùng nhằm có trải nghiệm tốt nhất về hệ thống. Các thư viện được kết hợp và tạo ra một ứng dụng, trang web đơn giản trên:
- Flask:
  - redirect: Chuyển hướng người dùng đến 1 route khác.
  - url\_for: Dùng để xây dựng các URL cho route, giúp ứng dụng dễ bảo trì hơn.
  - render\_template: Được dùng để render các template HTML từ các thư mục.
  - request: Dùng để lấy dữ liệu từ yêu cầu HTTP.
  - flash: Tạo thông báo tạm thời cho người dùng. Các thông báo này sẽ biến mất khi người dùng chuyển quan một trang khác.
  - SQLAlchemy: Quản lý cơ sở dữ liệu.
  - Session: Lưu thông tin người dùng đăng nhập trong một khoảng thời gian nhất định.
- HTML, CSS:
  - Dùng các thư viện đơn giản nhằm xây dựng một trang web đơn giản, dễ sử dụng cho người dùng. Ngoài ra, HTML còn kết hợp với CSS nhằm đưa trang web đó trở nên có sắc và sinh động hơn.

### 3. Chức năng

- Cấu hình ứng dụng Flask và kết nối cơ sở dữ liệu

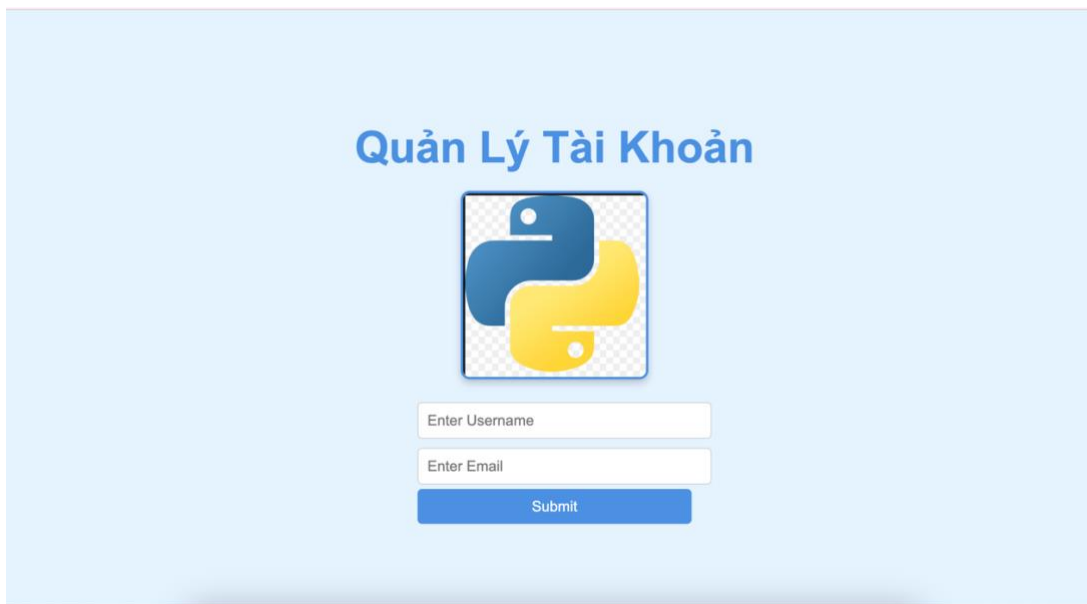
```

app = Flask(__name__)
app.config["SECRET_KEY"] = "phattan112"
app.config["SQLALCHEMY_DATABASE_URI"] = "sqlite:///database.db"
app.config["SQLALCHEMY_TRACK_MODIFICATIONS"] = False
app.permanent_session_lifetime = timedelta(minutes=5)

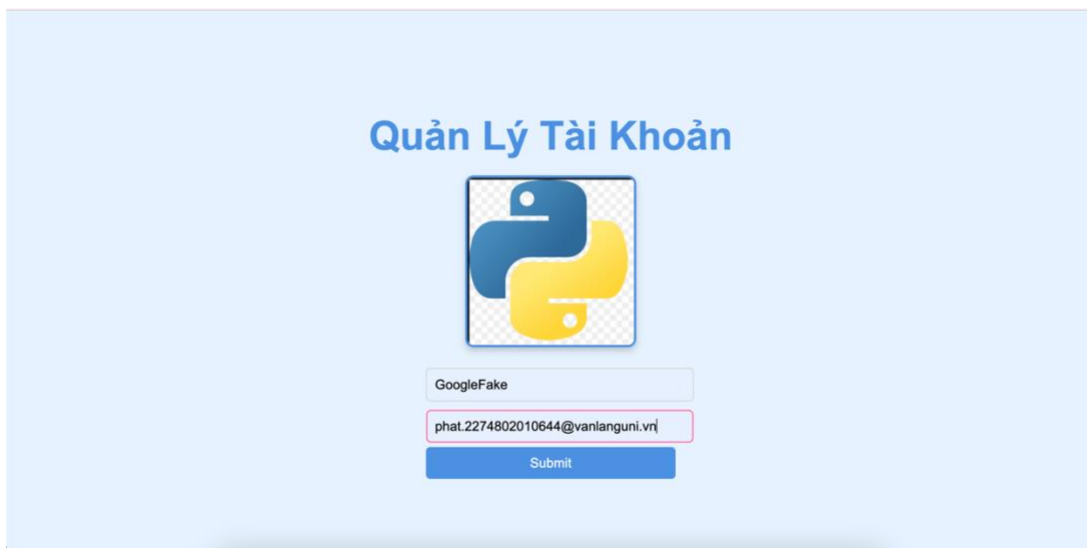
db = SQLAlchemy(app)

```

- Khởi tạo Flask bằng `app=Flask(__name__)` khởi tạo ứng dụng Flask.
- Cấu hình bảo mật và session:
  - Sử dụng “SECRET\_KEY” để bảo vệ dữ liệu của session.
  - SQLALCHEMY\_DATABASE\_URL: Định nghĩa địa chỉ cơ sở dữ liệu, nơi lưu trữ thông tin người dùng.
  - SQLALCHEMY\_TRACK\_MODIFICATIONS: Tắt tính năng theo dõi các thay đổi đối tượng.
- SQLAlchemy: Sử dụng “`db= SQLAlchemy(app)`” để khởi tạo kết nối với cơ sở dữ liệu.
- Chức năng Route Login/Logout
  - Route cho trang đăng nhập và đăng xuất. Người dùng nhập biểu mẫu, nếu hợp lệ thì ứng dụng sẽ lưu vào session và kiểm tra xem người dùng có tồn tại trong dữ liệu không.
    - Nếu người dùng tồn tại, email của họ sẽ lưu vào trong session.
    - Nếu người dùng chưa tồn tại trong cơ sở dữ liệu, họ sẽ được yêu cầu đăng nhập.
    - Khi người dùng đăng nhập thành công, ứng dụng sẽ chuyển người dùng sang trang dashboard.
  - Khi người dùng muốn đăng xuất tài khoản, hệ thống ứng dụng sẽ đăng xuất người dùng bằng cách xóa thông tin người dùng và email khỏi session.



*Trang đăng nhập mà ứng dụng yêu cầu người dùng nhập vào*



*Chức năng đăng nhập tài khoản người dùng vào ứng dụng*



- Chức năng Dashboard:

- Hiển thị danh sách người dùng: Hiển thị toàn bộ danh sách người dùng đã đăng ký trong hệ thống, bao gồm các thông tin như tên và email của từng người dùng.
- Thêm người dùng (add\_user): Dashboard cung cấp hoặc liên kết để chuyển người dùng đến trang thêm thông tin người dùng mới. Tại đây, người dùng có thể nhập thông tin người dùng để lưu vào cơ sở dữ liệu.
- Chỉnh sửa thông tin người dùng (edit\_user): Tùy chọn chuyển hướng đến trang chỉnh sửa, nơi đây có thể thay đổi thông tin người dùng và lưu lại.
- Xóa thông tin người dùng (delete\_user): Có thể chọn người dùng trong danh sách cũng có thể tùy chọn để xóa, khi chọn chức năng này thì người dùng được chọn sẽ bị xóa.
- Thông báo Flash: Thông báo người dùng nếu các thao tác thành công hoặc bị lỗi khi người dùng thêm xóa sửa thông tin người dùng.

Home Home Login Logout Dashboard				Search	Search
Dashboard - User Data					
User ID	Name	Email	Actions		
9	Trần Tấn Phát	trantanphat08112004@gmail.com	Edit	Delete	
15	tcr	123@gmail.com	Edit	Delete	
16	ka	889@gmail.com	Edit	Delete	
17	Phat	123@gmail.com	Edit	Delete	
18	phat	tranngocphu02012001@gmail.com	Edit	Delete	
19	abc	tranngocphu02012001@gmail.com	Edit	Delete	
20	admin	888@gmail.com	Edit	Delete	
22	zzzz	12334@gmail.com	Edit	Delete	

*Ứng dụng sẽ vào trang quản lý thông tin khi người dùng hoàn thành đăng nhập*

The screenshot shows an SQL editor window titled 'SQL editor 1'. It displays a table with 13 rows of user data. The table has three columns: 'user\_id', 'name', and 'email'. The data is as follows:

user_id	name	email
9	Trần Tấn Phát	trantanphat08112004@gmail.com
15	tcr	123@gmail.com
16	ka	889@gmail.com
17	Phat	123@gmail.com
18	phat	trangocphu02012001@gmail.com
19	abc	trangocphu02012001@gmail.com
20	admin	888@gmail.com
22	zzzz	12334@gmail.com
23	123456	89000@gmail.com
24	GoogleFake	trantanphat08112004@gmail.com
25	123	trantanphat08112004@gmail.com
26	80123	trantanphat08112004@gmail.com
27	kotaana	812312321@gmail.com

The interface includes a toolbar at the top with various icons, and tabs for 'Query', 'Results', and 'History'. Below the table, there are buttons for 'Grid view' and 'Form view'. A status bar at the bottom indicates 'Total rows loaded: 13'.

*Các dữ liệu trên trang web được lưu về database ở SQLite*

The screenshot shows a web application interface. At the top, there is a form titled 'Edit User'. The form has two input fields: 'Name' with the value 'Trần Tấn Phát' and 'Email' with the value 'trantanphat08112004@gmail.com'. Below the fields are two buttons: 'Save Changes' (in blue) and 'Cancel' (in grey).

Below the form, there is a confirmation dialog box with a light pink background. It contains the text '127.0.0.1:5000 cho biết' and 'Are you sure you want to delete this user?'. At the bottom right of the dialog are two buttons: 'Huỷ' (Cancel) in a red outline and 'OK' in a solid red button.

A green rectangular button with the text "Add New User" in white, followed by a light blue rectangular button.

*Ứng dụng tạo nút cho phép thêm tài khoản , chỉnh sửa thông tin hoặc xoá tài khoản*

#### 4. Ưu nhược điểm

##### - Ưu điểm:

- Sử dụng Flask Framework: Flask là một framework nhẹ và dễ vận hành, giúp xây dựng và vận hành trang web nhanh chóng và hiệu quả.
- Chức năng người dùng cơ bản: Dễ dàng mở rộng để thêm các tính năng khác như quản lý mật khẩu, xác thực email,...
- Thông báo qua flash: Hiển thị thông báo cho người dùng bằng cách cung cấp phản hồi ngay lập tức.
- Dễ mở rộng chức năng theo yêu cầu người dùng.
- Tạo cơ sở dữ liệu tự động: Tự động kiểm tra sự tồn tại của cơ sở dữ liệu và tạo cơ sở dữ liệu mới nếu chưa có.
- Có khả năng tạo mới, chỉnh sửa và xoá thông tin quản lý.

##### - Nhược điểm:

- Thiếu xác thực người dùng và mật khẩu: Mã không sử dụng mật khẩu để xác thực người dùng, dễ gây ra rủi ro bảo mật lớn vì bất kì ai cũng có thể đăng nhập bằng tên người dùng.
- Hiệu suất yếu: Cách lưu trữ thông tin người dùng trong session như hiện tại không phù hợp cho các ứng dụng có số lượng người dùng lớn.
- Thiếu bảo mật bổ sung: Thiếu các lớp bảo mật bổ sung, hạn chế số lần thử đăng nhập.
- Cần cải thiện khi gặp phải các vấn đề cụ thể
- Cấu trúc mã chưa tối ưu hoá.

## 5. Kết luận

- Đoạn code xây dựng trang web quản lý này đã sử dụng các thư viện cơ bản như Flask và cơ sở dữ liệu để quản lý thông tin người dùng với các chức năng tạo, đọc, cập nhật và xoá thông tin. Ứng dụng đảm bảo trạng thái đăng nhập của người dùng, đảm bảo được sử dụng sau khi đăng nhập.
- Qua đó đã cung cấp một ứng dụng web quản lý người dùng đơn giản và trực quan, tuy nhiên cần có các cập nhật cải tiến trang web liên tục nhằm bảo đảm về tính bảo mật và tính logic sao cho phù hợp với môi trường sản xuất hoặc các ứng dụng thực tế.

## TÀI LIỆU THAM KHẢO

1. Thành IT: <https://www.youtube.com/watch?v=FuMj3lbf00> (Youtube)
2. E-Learning: [Flask Web Development\\_ Developing Web Applications with Python - PDF Room.pdf](#) (E-book)
3. W3s: [https://www.w3schools.com/python/python\\_mysql\\_getstarted.asp](https://www.w3schools.com/python/python_mysql_getstarted.asp) (W3s)
4. Webcode: [https://www.youtube.com/watch?v=TZDMX4GbQyQ&list=PL4P8sY6zvjk5im\\_dA5SB3u45xQRLFd9CZ](https://www.youtube.com/watch?v=TZDMX4GbQyQ&list=PL4P8sY6zvjk5im_dA5SB3u45xQRLFd9CZ) (Youtube)
5. Các trang hỗ trợ code: ChatGPT, W3s, Coursera,...

### LINK GITHUB:

[https://github.com/trantanphat0811/TranTanPhat\\_2274802010644\\_BaoCaoDoAn\\_PythonNangCao\\_241\\_71ITSE31003\\_01](https://github.com/trantanphat0811/TranTanPhat_2274802010644_BaoCaoDoAn_PythonNangCao_241_71ITSE31003_01) (Github)

