

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



BÁO CÁO BÀI TẬP LỚN
LẬP TRÌNH NÂNG CAO - CO2039

WEB/APP SAO KÊ

L01 – HỌC KỲ 241

GVHD: Lê Đình Thuận

SV thực hiện: Đặng Công Sơn - 2212932
Bùi Ngọc Diễm Quỳnh - 2212878
Lưu Quang Khải - 2211549
Huỳnh Quang Thịnh - 2213273
Lê Quang Hiển - 2111207
Nguyễn Lê Anh Xuân - 2420013
Thái Kim Long - 2211888



Mục lục

1	Members and Tasks	2
2	Overview	3
2.1	Project Description	3
2.2	Project Goals	3
2.2.1	User story	3
2.2.2	User benefits	3
3	Requirement	4
3.1	Functional requirements	4
3.2	Non-functional requirements	4
4	Use-case	5
4.1	Tra cứu sao kê theo số tiền gửi	5
4.2	Tra cứu sao kê theo tên người gửi	5
4.3	Tra cứu sao kê theo nội dung	5
5	Activity Diagram	6
5.1	Tra cứu sao kê theo số tiền gửi	6
5.2	Tra cứu sao kê theo tên người gửi	7
5.3	Tra cứu sao kê theo nội dung	7
6	Sơ đồ thành phần cho Module Quan trọng	8
6.1	Mô tả tổng quan	8
6.2	Chi tiết các thành phần	8
6.3	Luồng hoạt động	9
7	Giao diện người dùng cho trang web sao kê	10
7.1	Mô tả giao diện người dùng	10
7.1.1	Đầu trang (Header)	10
7.1.2	Khu vực tìm kiếm	10
7.1.3	Khu vực hiển thị kết quả	10
7.2	Giao diện thực tế	11
8	Backend	12
8.1	Python	12
8.1.1	Giới thiệu về ngôn ngữ Python	12
8.1.2	Đặc điểm nổi bật của Python	12
8.1.3	Lý do chọn Python	12
8.2	Thư viện được sử dụng trong bài tập lớn	12
8.2.1	Pandas	12
8.2.2	Flask	15
8.2.3	Requests	17
9	Screen Flow	19



1 Members and Tasks

MSSV	Tên	Công việc	%
2212932	Đặng Công Sơn	Frontend, tổng hợp code	100
2212878	Bùi Ngọc Diễm Quỳnh	Frontend, báo cáo sơ bộ, slides	100
2211549	Lưu Quang Khải	Tổng hợp báo cáo, phụ Backend	100
2213273	Huỳnh Quang Thịnh	Backend	100
2111207	Lê Quang Hiển	Backend, đóng gói docker	100
2420013	Nguyễn Lê Anh Xuân	Tổng hợp báo cáo, phụ Backend	100
2211888	Thái Kim Long	Backend	100

2 Overview

2.1 Project Description

Hiện nay, việc người dân quyên góp tiền vào các quỹ từ thiện, cứu trợ thiên tai hoặc chương trình xã hội do Mặt trận Tổ quốc Việt Nam tổ chức và quản lý khá phổ biến, kèm theo đó thì bên MTTQ VN cũng phải xây dựng niềm tin và tính minh bạch cho người dân giữa những bất cập hiện nay như :

- Người dân thì khó để tra cứu khoản quyên góp của chính mình.
- Việc thiếu minh bạch trong số tiền của phía quyên góp.
- Việc thiếu minh bạch trong các khoản thu của các tổ chức.

Từ đó, nhóm ra mắt nền tảng “Sao Kê” tra cứu dữ liệu của MTTQ VN như một bước tiến trong việc xây dựng niềm tin và tăng cường giám sát của người dân đối với các hoạt động liên quan đến quỹ từ thiện, cứu trợ thiên tai, và các chương trình an sinh xã hội. Ngoài ra còn thuận tiện cho người dân quan sát, tra cứu một cách thuận tiện và nhanh chóng.

2.2 Project Goals

Mục tiêu nhóm đặt ra là xây dựng một web tra cứu dữ liệu "Sao kê" của MTTQ VN có đầy đủ các chức năng cơ bản giúp người dùng cuối có thể thực hiện việc tra cứu dễ dàng và nhanh chóng.

2.2.1 User story

- Là một người quyên góp, tôi muốn tra cứu các khoản quyên góp của mình.
- Là một người quyên góp, tôi muốn xem tính minh bạch của dữ liệu đã đúng với những gì tôi quyên góp.
- Là một người dân, tôi muốn tra cứu số tiền của một cá nhân hay của một tổ chức quyên góp có đúng với những gì họ đã công bố.
- Là một thành viên của tổ chức, tôi muốn biết tổ chức của mình đã quyên góp bao nhiêu.
- là một thành viên của tổ chức, tôi muốn biết tổ chức của mình đã đóng góp đúng với số tiền mà mọi người cùng tạo.
- Là một người dân, tôi muốn biết những nhà quyên góp thầm lặng đã có công lớn.
- Là một người dân, tôi muốn biết những người đóng góp nhiều.

2.2.2 User benefits

- Người dân: Hệ thống giúp người dân dễ dàng truy cập và tra cứu thông tin tài chính, từ đó tăng cường sự tin tưởng và ủng hộ đối với MTTQ VN.
- Nhân viên MTTQ VN: Giúp giảm bớt công việc giấy tờ và tăng hiệu quả quản lý thông tin tài chính, đồng thời đảm bảo dữ liệu luôn được cập nhật và chính xác.
- Cơ quan kiểm toán và giám sát: Tạo điều kiện thuận lợi cho việc kiểm tra và giám sát, đảm bảo các hoạt động tài chính được thực hiện đúng quy định và minh bạch

3 Requirement

3.1 Functional requirements

- Người dùng có thể tra cứu thông tin sao kê tài chính theo từng khoảng thời gian.
- Người dùng có thể tìm kiếm thông tin theo các tiêu chí như tên người gửi, khoản tiền gửi, ngày tháng, và nội dung chuyển khoản.
- Hệ thống cung cấp báo cáo tài chính tổng hợp cho người dùng.

3.2 Non-functional requirements

- Hệ thống phải đảm bảo tính bảo mật cao, chỉ những người được phân quyền mới có thể truy cập và chỉnh sửa dữ liệu.
- Giao diện người dùng phải thân thiện, dễ sử dụng và hỗ trợ đa ngôn ngữ.
- Hệ thống phải có khả năng xử lý nhanh chóng các yêu cầu tra cứu và cập nhật dữ liệu.
- Dữ liệu phải được sao lưu định kỳ để đảm bảo an toàn và khôi phục khi cần thiết.
- Hệ thống phải tương thích với nhiều loại thiết bị và trình duyệt khác nhau.
- Hệ thống hoạt động ổn định, có khả năng mở rộng, bảo trì và phục hồi.
- Hệ thống được vận hành trên web và hoạt động 24/7.

4 Use-case

4.1 Tra cứu sao kê theo số tiền gửi

Tên Use Case	Tra cứu theo số tiền gửi
Mô tả	Người dùng có thể tra cứu sao kê tài chính theo số tiền gửi.
Tiền điều kiện	Người dùng đã truy cập vào hệ thống.
Hậu điều kiện	Hệ thống trả về danh sách các giao dịch tương ứng với số tiền gửi đã chọn.
Luồng sự kiện chính	1. Người dùng truy cập vào hệ thống. 2. Người dùng nhập khoảng giá trị tiền gửi muốn tìm kiếm. 3. Hệ thống hiển thị danh sách kết quả phù hợp.
Ngoại lệ	Hệ thống hiển thị thông báo lỗi nếu: - Không có dữ liệu phù hợp, hệ thống sẽ hiển thị thông báo lỗi. - Người dùng nhập khoảng giá trị không hợp lệ (ví dụ: số âm), hệ thống yêu cầu nhập lại.

4.2 Tra cứu sao kê theo tên người gửi

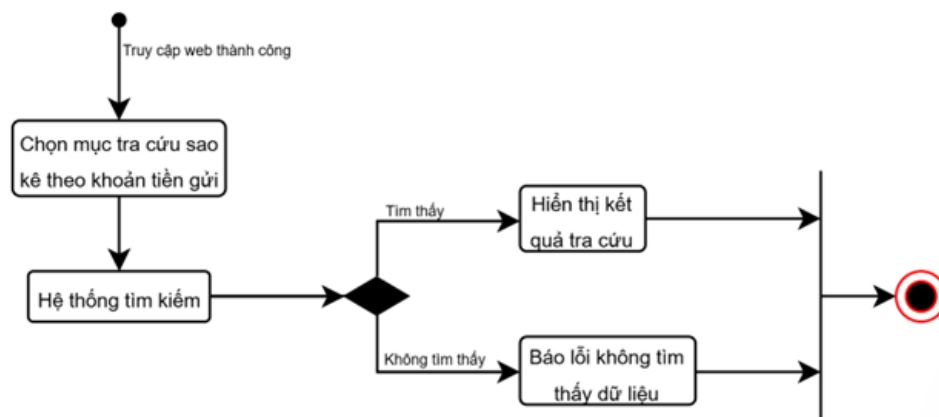
Tên Use Case	Tra cứu sao kê theo tên người gửi
Mô tả	Người dùng tra cứu sao kê tài chính theo tên người gửi
Tiền điều kiện	Người dùng đã truy cập vào hệ thống.
Hậu điều kiện	Hệ thống trả về danh sách các giao dịch tương ứng với tên người gửi đã nhập
Luồng sự kiện chính	1. Người dùng truy cập vào hệ thống. 2. Người dùng nhập tên người gửi muốn tìm kiếm. 3. Hệ thống sẽ tìm kiếm trong detail xem có tên người gửi phù hợp không. 4. Hệ thống hiển thị danh sách kết quả phù hợp.
Ngoại lệ	Hệ thống hiển thị thông báo lỗi nếu: - Không có dữ liệu phù hợp, hệ thống sẽ hiển thị thông báo lỗi.

4.3 Tra cứu sao kê theo nội dung

Tên Use Case	Tra cứu sao kê theo nội dung
Mô tả	Người dùng tra cứu sao kê tài chính theo nội dung
Tiền điều kiện	Người dùng đã truy cập vào hệ thống.
Hậu điều kiện	Hệ thống trả về danh sách giao dịch tương ứng với nội dung đã nhập.
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Người dùng truy cập vào hệ thống. 2. Người dùng nhập nội dung chuyển tiền muốn tìm kiếm. 3. Hệ thống sẽ tìm kiếm nội dung thích ứng trong phần detail. 4. Hệ thống hiển thị danh sách kết quả phù hợp.
Ngoại lệ	Hệ thống hiển thị thông báo lỗi nếu: - Nếu không có dữ liệu phù hợp, hệ thống sẽ hiển thị thông báo lỗi.

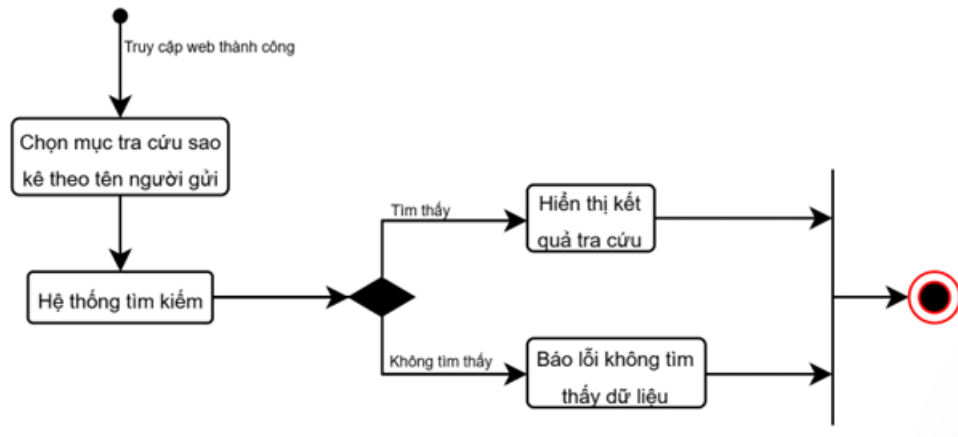
5 Activity Diagram

5.1 Tra cứu sao kê theo số tiền gửi



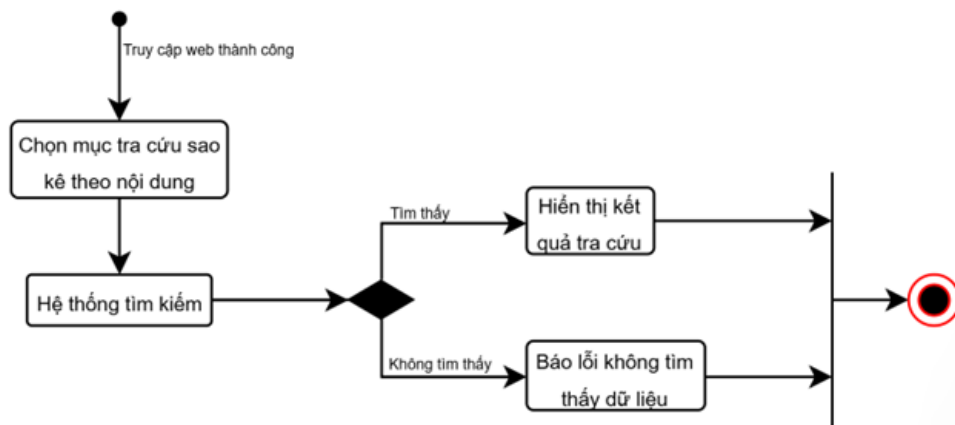
Hình 1: Tra cứu theo số tiền gửi.

5.2 Tra cứu sao kê theo tên người gửi



Hình 2: Tra cứu theo tên người gửi

5.3 Tra cứu sao kê theo nội dung



Hình 3: Tra cứu theo nội dung

6 Sơ đồ thành phần cho Module Quan trọng

6.1 Mô tả tổng quan

Module quan trọng của hệ thống được thiết kế để cung cấp chức năng tra cứu sao kê. Sơ đồ thành phần dưới đây trình bày mối liên kết giữa các thành phần chính, bao gồm:

- **Giao diện người dùng (Frontend):** Cho phép người dùng nhập các tiêu chí tìm kiếm và hiển thị kết quả.
- **Controller (Backend):** Xử lý các yêu cầu từ giao diện người dùng, gọi đến lớp Service để xử lý logic nghiệp vụ.
- **Service:** Thực hiện các logic xử lý dữ liệu và áp dụng các bộ lọc trên tập dữ liệu sao kê.

6.2 Chi tiết các thành phần

- **Controller (sao_ke_blueprint):**
 - Nhận yêu cầu GET từ giao diện người dùng thông qua endpoint `/search`.
 - Trích xuất các tham số từ yêu cầu, bao gồm:
 - * **min_credit, max_credit:** Giá trị tối thiểu và tối đa của cột `credit`.
 - * **min_debit, max_debit:** Giá trị tối thiểu và tối đa của cột `debit`.
 - * **detail:** Chuỗi mô tả tìm kiếm.
 - Gọi đến `SaoKeService` để áp dụng bộ lọc dựa trên các tham số đầu vào.
 - Trả về kết quả lọc hoặc thông báo lỗi nếu không tìm thấy dữ liệu phù hợp.
- **Service (SaoKeService):**
 - **Tải dữ liệu sao kê:**
 - * Tải tệp CSV từ URL `https://s.thuanle.me/chuyen_khoan.csv`.
 - * Chuyển đổi dữ liệu thành một `DataFrame` của `Pandas` để dễ dàng xử lý.
 - **Lọc dữ liệu:**
 - * Lọc dựa trên các giá trị cột `credit, debit`.
 - * Tìm kiếm chuỗi trong cột `detail`.
 - * Trả về `DataFrame` chứa kết quả lọc.
- **Giao diện người dùng (Frontend):**
 - Hiển thị các trường nhập liệu cho người dùng để nhập tiêu chí tìm kiếm, bao gồm:
 - * Khoảng giá trị cho số tiền ghi có (`credit`) và số tiền ghi nợ (`debit`).
 - * Chuỗi mô tả tìm kiếm (`detail`).
 - Nút **“Tìm kiếm”**: Gửi yêu cầu GET đến backend thông qua endpoint `/search`.
 - Hiển thị kết quả tìm kiếm dưới dạng bảng, với các cột: `credit, date_time, debit, detail, trans_no`.

6.3 Luồng hoạt động

- Người dùng nhập các tiêu chí tìm kiếm và nhấn nút **“Tìm kiếm”**.
- Yêu cầu được gửi đến Controller qua endpoint `/search`.
- Controller chuyển các tham số đến lớp Service để thực hiện logic lọc dữ liệu.
- Service tải dữ liệu từ tệp CSV, áp dụng bộ lọc và trả về kết quả phù hợp.
- Controller trả về kết quả tìm kiếm cho giao diện người dùng để hiển thị.

7 Giao diện người dùng cho trang web sao kê

7.1 Mô tả giao diện người dùng

Giao diện người dùng của trang web sao kê được thiết kế với các thành phần chính như sau:

7.1.1 Đầu trang (Header)

- Hiển thị logo và tiêu đề **Mặt trận Tổ quốc Việt Nam**, với slogan “Đoàn kết - Dân chủ - Hợp tác - Phát triển” ngay bên dưới.
- Màu sắc chủ đạo là đỏ, kết hợp với biểu tượng logo Mặt trận Tổ quốc Việt Nam.

7.1.2 Khu vực tìm kiếm

- Các trường tìm kiếm:
 - **Số tiền ghi có (Credit)**: Cho phép người dùng nhập khoảng giá trị từ một số cụ thể đến một số khác.
 - **Số tiền ghi nợ (Debit)**: Cung cấp chức năng tương tự như trường Credit.
- **Nút “Tìm kiếm”**: nằm ngay dưới trường nhập liệu, giúp người dùng thực hiện thao tác nhanh chóng.

7.1.3 Khu vực hiển thị kết quả

- Dòng thông báo tên người dùng, ví dụ: “**NGUYEN THI MAO**”.
- Kết quả trả về:
 - Số lượng giao dịch được tìm thấy, ví dụ: “**Kết quả: 2 giao dịch**”.
 - Một bảng hiển thị chi tiết giao dịch với các cột:
 - * **Credit**: Số tiền ghi có.
 - * **Date_Time**: Ngày giờ giao dịch.
 - * **Debit**: Số tiền ghi nợ.
 - * **Detail**: Chi tiết giao dịch, bao gồm mô tả như tên giao dịch và mục đích.
 - * **Trans_No**: Số giao dịch (Transaction Number).



7.2 Giao diện thực tế

**MẶT TRẬN TỔ QUỐC VIỆT NAM**
Đoàn kết - Dân chủ - Hợp tác - Phát triển

Tra Cứu Sao Kê

Số tiền ghi có (Credit)

Số tiền ghi nợ (Debit)

Từ

Đến

Từ

Đến

Chi tiết giao dịch

NGUYEN THI MAO

Tìm kiếm

Kết quả: 2 giao dịch

credit	date_time	debit	detail	trans_no
3000	01/09/2024_5215.97152	0	267515.010924.122904.NGUYEN THI MAO Chuyen tien	1
369000	10/09/2024_5217.13167	0	137545.100924.193056.NGUYEN THI MAO gui tinh yeu thuong va nang luong binh an den dong bao bi anh huong bao co 3 ET24254137778036	128555

Hình 4: Giao diện nhập liệu tra cứu kết quả sao kê

Kết quả: 4691 giao dịch				
credit	date_time	debit	detail	trans_no
3000	01/09/2024_5215.97152	0	267515.010924.122904.NGUYEN THI MAO Chuyen tien	1
10000	01/09/2024_5219.24714	0	018806.010924.213139.chuc mung ngay 29 nuoc cong hoa xa hoi chu nghia Viet Nam	2
10000	02/09/2024_5212.22965	0	055464.020924.064157.Ung Ho Nha Nuooc Viet Nam Va Giup do Nguoi dan (by TPBank ChatPay)	3
10000	02/09/2024_5245.21394	0	MBVCB.6924605040.gia dinh Dung Thuy Giang chuc to quoc khoe manh, binh an, hanh phuc, tinh vuong, cam on.CT tu 1018144916 HOANG THANH THUY toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW	4
2000	02/09/2024_5216.8149	0	524322.020924.175952.2091945	6
10000	03/09/2024_5218.45892	0	881384.030924.070324.Ung Ho Nha Nuooc Viet Nam Va Giup do Nguoi dan	8
9000	03/09/2024_5017.43849	0	888828.030924.121121.NGUYEN THI KIEU OANH chuyen tien	13
10000	03/09/2024_5243.88658	0	MBVCB.6934364247.gia dinh Dung Thuy Giang chuc to quoc khoe manh, binh an, hanh phuc, cam on.CT tu 1018144916 HOANG THANH THUY toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW	14
6000	04/09/2024_5214.46619	0	268515.040924.081538.Ha Duy Phuong	20
10000	04/09/2024_5245.19623	0	MBVCB.6939182690.gia dinh Dung Thuy Giang chuc to quoc khoe manh, tri tue, tu bi, cam on.CT tu 1018144916 HOANG THANH THUY toi 0011001932418 MAT TRAN TO QUOC VN - BAN CUU TRO TW	21
6000	04/09/2024_5215.70219	0	563774.040924.092651.PHAM THI THAO chuyen tien	22
3000	04/09/2024_5220.4615	0	429224.040924.100236.NGUYEN THI NGOC LE chuyen tien	23
6000	04/09/2024_5390.37909	0	020097040509041008052024HEV1019980.37909 .100752.Vietcombank0011001932418:TRAN THI ANH NGUYET chuyen tien	25
3000	04/09/2024_5215.56651	0	786355.040924.101558.PHAM DUY AN chuyen tien	26
3000	04/09/2024_5218.4587	0	535425.040924.102817.NGUYEN THI NGOC LE chuyen tien	27

Hình 5: Giao diện kết quả các giao dịch ứng với dữ liệu truy xuất

8 Backend

8.1 Python

Ngôn ngữ lập trình được sử dụng: Python.

8.1.1 Giới thiệu về ngôn ngữ Python

Python là một ngôn ngữ lập trình bậc cao, đa năng, được thiết kế để đơn giản hóa quá trình phát triển phần mềm. Ra mắt lần đầu năm 1991 bởi Guido van Rossum, Python sở hữu cú pháp dễ học, dễ đọc, và gần gũi với ngôn ngữ tự nhiên. Python được sử dụng rộng rãi trong các lĩnh vực như phát triển web, trí tuệ nhân tạo, xử lý dữ liệu, và nhiều ứng dụng khác.

8.1.2 Đặc điểm nổi bật của Python

- Dễ đọc, dễ hiểu: Gần gũi với ngôn ngữ tự nhiên, giúp các nhà phát triển dễ dàng đọc và hiểu các chương trình, tăng cường khả năng hợp tác và bảo trì mã nguồn.
- Năng suất cao làm việc: Cung cấp một loạt các bộ mã đa dạng, giúp tăng cường năng suất và giảm thời gian phát triển ứng dụng.
- Thư viện phong phú: Có hàng ngàn thư viện hỗ trợ từ xử lý dữ liệu, AI, đến phát triển ứng dụng web.
- Cộng đồng lập trình viên đông đảo.
- Tính năng cấp cao: Khả năng quản lý bộ nhớ và cấu trúc dữ liệu hiệu quả.

8.1.3 Lý do chọn Python

- Cú pháp đơn giản, dễ đọc, triển khai nhanh: phù hợp cho việc thực hiện bài tập lớn trong thời gian ngắn.
- Nhiều Framework phù hợp, mạnh mẽ, dễ sử dụng: giúp đơn giản hóa việc xây dựng và quản lý ứng dụng trong đó có Flask, một micro-framework nhẹ, phù hợp cho bài tập lớn với quy mô ứng dụng nhỏ và đơn giản.
- Thư viện phong phú: Pandas, Requests, NumPy giúp thao tác với dữ liệu sao kê và xử lý HTTP dễ dàng.
- Python có một cộng đồng lớn và tài liệu phong phú, giúp dễ dàng tìm kiếm giải pháp.

8.2 Thư viện được sử dụng trong bài tập lớn

8.2.1 Pandas

8.2.1.1 Giới thiệu:

Pandas là một thư viện mã nguồn mở trong Python, được thiết kế dành riêng cho xử lý và phân tích dữ liệu. Được xây dựng trên NumPy, Pandas cung cấp các cấu trúc dữ liệu linh hoạt và hiệu quả, đặc biệt hữu ích khi làm việc với dữ liệu dạng bảng (CSV, EXCEL, SQL).

8.2.1.2 Cài đặt:

Để cài đặt thư viện Pandas, bạn có thể làm theo một số cách khác nhau:

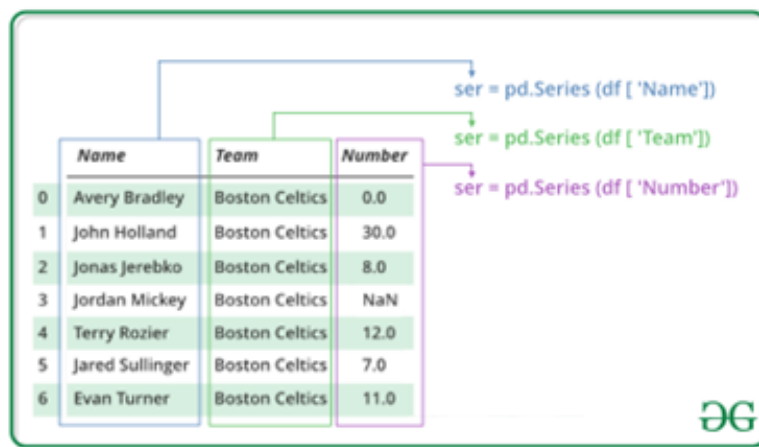
- Sử dụng pip: `pip install pandas`
- Sử dụng conda: `conda install pandas`

8.2.1.3 Cấu trúc dữ liệu chính trong Pandas

Series

Series là mảng một chiều được gắn label mà có thể lưu trữ bất kì kiểu dữ liệu nào (integer, string, float...). Mỗi giá trị sẽ có một label trở đến chỉ mục(index) của nó, bạn có thể đặt tên cho label với đối số index. Sử dụng label để có thể truy cập đến một giá trị cụ thể.

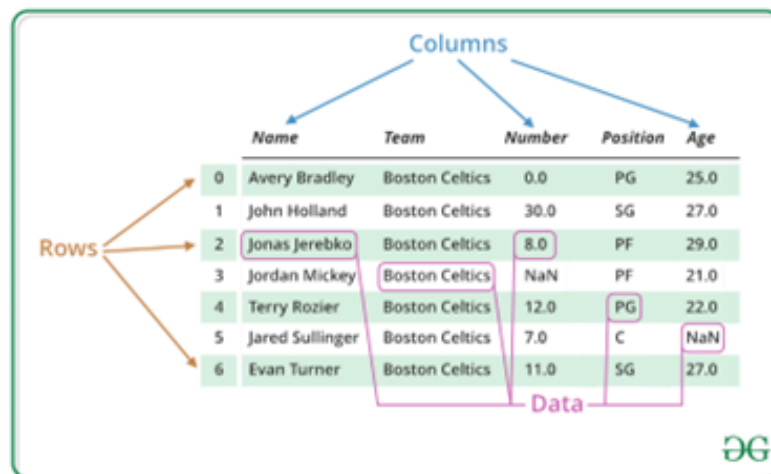
Một Series có thể được tạo bằng các đầu vào khác nhau như: Array, Dict, Scalar value hoặc constant. Hoặc tạo ra bằng cách lấy dữ liệu từ các kho lưu trữ dữ liệu, đó có thể là SQL Database, file CSV, file Excel...



DataFrame

DataFrame là cấu trúc dữ liệu bảng không đồng nhất, có 2 chiều và kích thước có thể thay đổi. DataFrame cũng có các labels tương ứng với các hàng và cột. Pandas DataFrame bao gồm 3 thành phần chính là : data, rows và columns.

Một DataFrame có thể được tạo bằng các đầu vào khác nhau như: Lists, Dict, Series, Numpy ndarrays, DataFrame khác. Hoặc có thể được tạo bằng cách lấy dữ liệu từ các kho dữ liệu như SQL Database, CSV file, Excel file



8.2.1.4 Tính năng chính

Pandas cung cấp các hàm và câu lệnh hỗ trợ cho việc xử lý dữ liệu trên cấu trúc dữ liệu của thư viện có thể kể đến như:

- Nhập và xuất dữ liệu
 - Đọc và ghi dữ liệu từ/đến các định dạng phổ biến:
CSV: `pd.read_csv()`, `df.to_csv()`
Excel: `pd.read_excel()`, `df.to_excel()`
JSON: `pd.read_json()`, `df.to_json()`
SQL: `pd.read_sql()`, `df.to_sql()`
- Lựa chọn và truy cập dữ liệu:
 - Truy cập dữ liệu
Theo nhãn: `.loc[]`
Theo vị trí: `.iloc[]`
 - Lọc dữ liệu dựa trên điều kiện
- Thao tác dữ liệu
 - Thêm/xóa cột hoặc hàng: `df['new_column'] = values`, `df.drop()`
 - Sắp xếp dữ liệu: `df.sort_values()`, `df.sort_index()`
 - Thay đổi kiểu dữ liệu: `df.astype()`
- Tổng hợp và thống kê
 - Các hàm tính toán cơ bản: `mean()`, `sum()`, `std()`, `min()`, `max()`
 - Nhóm dữ liệu: `df.groupby('column_name').agg()`
 - Tóm tắt dữ liệu: `df.describe()`
- Kết hợp dữ liệu
 - Ghép nối (`concat`), hợp nhất (`merge`), và thêm dữ liệu (`join`)
- Xử lý chuỗi ký tự
 - Thao tác chuỗi: `str.upper()`, `str.contains()`, `str.replace()`

8.2.1.5 Lý do sử dụng thư viện

Vì dữ liệu từ Datasource có đuôi `.csv` và dạng theo bảng 2 chiều, phù hợp cho cấu trúc dữ liệu DataFrame của thư viện. Cùng với đó thư viện cung cấp các hàm hỗ trợ mạnh mẽ cho việc nhập, xuất tệp và thao tác trên các dữ liệu của tệp. Vì thế Flask phù hợp mục tiêu của bài tập lớn.

8.2.1.6 Áp dụng trong bài tập lớn

- Đọc file CSV
`self.data = pd.read_csv(csv_data)`
Đọc dữ liệu từ file CSV và lưu dưới dạng DataFrame.
- Lọc dữ liệu dựa trên tín dụng (`credit`)

```
def search_by_credit(self, min_credit, max_credit):  
    return self.data[(self.data['credit'] >= min_credit)  
                      & (self.data['credit'] <= max_credit)]  
def search_by_debit(self, min_debit, max_debit):  
    return self.data[(self.data['debit'] >= min_debit)  
                      & (self.data['debit'] <= max_debit)]
```

Lọc dữ liệu trực tiếp dựa trên các điều kiện.

- Tìm kiếm chuỗi

```
def search_by_detail(self, detail_content):  
    return self.data[self.data['detail'].str.contains(detail_content,  
                                                         case=False, na=False)]
```

Sử dụng `.str.contains` để tìm kiếm chuỗi trong một cột, với tùy chọn bỏ qua phân biệt hoa thường (`case=False`).

8.2.2 Flask

8.2.2.1 Giới thiệu

Python Flask là loại framework web siêu nhẹ dựa trên ngôn ngữ lập trình Python, nó được sử dụng để xây dựng các ứng dụng web. Flask sở hữu lối thiết kế đơn giản, linh hoạt và dễ sử dụng với mục tiêu cung cấp phương án phát triển ứng dụng web hiệu quả.

Flask cung cấp các tính năng cơ bản như định tuyến URL, tạo giao diện người dùng, xử lý yêu cầu và phản hồi HTTP. Ngoài ra, nó cũng hỗ trợ các tính năng mở rộng thông qua các thư viện mở rộng Flask (Flask extensions) để thêm các chức năng như xác thực người dùng, kết nối cơ sở dữ liệu, quản lý phiên và nhiều tính năng khác.

8.2.2.2 Cài đặt

Để cài đặt thư viện Flask, bạn có thể sử dụng command sau đây để cài Flask:

- `pip install Flask`

8.2.2.3 Tính năng chính

- Routing: Flask cho phép bạn xác định các URL và hành động tương ứng thông qua decorators như `@app.route('/')`.
- HTTP Methods: Flask hỗ trợ các phương thức HTTP như GET, POST, PUT, DELETE và nhiều phương thức khác thông qua decorators và request methods.
- Testing: Flask hỗ trợ việc viết các bài kiểm tra cho ứng dụng của bạn, đảm bảo rằng mã của bạn hoạt động như mong đợi.
- Debugging: Trong quá trình phát triển, Flask cung cấp các công cụ debugging như thông báo lỗi chi tiết và môi trường debug để bạn dễ dàng xác định và khắc phục lỗi.
- Cấu trúc ứng dụng: Dễ dàng tạo và tổ chức ứng dụng với Flask thông qua cấu trúc thư mục tiêu chuẩn. Bạn có thể phân chia ứng dụng thành các module, view functions, templates, và static files.

8.2.2.4 Lý do sử dụng thư viện

Vì Flask là một framework nhẹ và linh hoạt, phù hợp cho việc xây dựng ứng dụng web nhỏ theo đề bài. Flask cho phép khởi tạo ứng dụng nhanh chóng với cú pháp đơn giản, dễ hiểu, đồng thời hỗ trợ tổ chức code theo mô-đun thông qua Blueprints, giúp chia ứng dụng web thành các phần khác nhau. Tích hợp các công cụ xử lý HTTP requests, query parameters, và phản hồi JSON, Flask giúp việc giao tiếp với frontend dễ dàng hơn. Vì thế Flask phù hợp mục tiêu của bài tập lớn.

8.2.2.5 Áp dụng trong bài tập lớn

- Tạo ứng dụng Flask

```
app = Flask(__name__)
```

Tạo một ứng dụng web cơ bản làm nền tảng xử lý các yêu cầu HTTP.

- Sử dụng Blueprint để tổ chức ứng dụng

```
sao_ke_blueprint = Blueprint('sao_ke', __name__)
```

Quản lý endpoint theo module sao kê, giúp tăng tính rõ ràng và khả năng mở rộng.

- Định nghĩa API endpoint

```
@sao_ke_blueprint.route('/search', methods=['GET'])
```

Định nghĩa một endpoint `/api/search` với HTTP method GET. Endpoint này tiếp nhận các tham số truy vấn (`min_credit`, `max_credit`, `min_debit`, `max_debit`, `detail`) từ query string.

- Xử lý request và tham số

```
min_credit = float(request.args.get('min_credit', 0))
max_credit = float(request.args.get('max_credit', float('inf')))
min_debit = float(request.args.get('min_debit', 0))
max_debit = float(request.args.get('max_debit', float('inf')))
detail = request.args.get('detail', '')
```

Sử dụng `request.args` giúp lấy các tham số mà client gửi qua URL. Tham số được chuyển đổi về kiểu số thực (float) và có giá trị mặc định khi không được truyền.

- Xử lý response

```
return jsonify(results.to_dict(orient="records"))
```

Dữ liệu được chuyển đổi từ DataFrame (Pandas) thành danh sách từ điển (`to_dict`), sau đó đóng gói dưới dạng JSON bằng `jsonify`.

- Kích hoạt ứng dụng

```
if __name__ == '__main__':
    app.run(debug=True)
```

Flask được cấu hình để chạy ứng dụng trong môi trường phát triển (`debug=True`), cho phép ứng dụng sẵn sàng xử lý các yêu cầu HTTP từ client.

8.2.3 Requests

8.2.3.1 Giới thiệu

Requests là một thư viện Python mạnh mẽ và dễ sử dụng, được phát triển để làm việc với HTTP, giúp người dùng gửi các yêu cầu HTTP một cách đơn giản và linh hoạt. Đây là một trong những thư viện phổ biến nhất trong Python cho các tác vụ giao tiếp với API hoặc các trang web.

8.2.3.2 Cài đặt

Để cài đặt thư viện Requests, bạn có thể sử dụng command sau đây để cài Requests:

- `pip install requests`

8.2.3.3 Tính năng chính

- Gửi yêu cầu HTTP đơn giản
 - Hỗ trợ các phương thức HTTP phổ biến như:
GET: `requests.get()`
POST: `requests.post()`
PUT: `requests.put()`
DELETE: `requests.delete()`
- Xử lý tham số truy vấn và dữ liệu gửi đi
 - Gửi tham số query trong URL và dữ liệu form hoặc JSON trong yêu cầu :
`requests.get(url, params='key': 'value')`
`requests.post(url, json='key': 'value')`
- Quản lý Cookies và Session
 - Lưu trữ và gửi cookies qua các yêu cầu HTTP
`requests.get(url, cookies='key': 'value')`
 - Tạo session để giữ trạng thái giữa các yêu cầu
`requests.Session()`
- Xử lý nội dung phản hồi
 - Hỗ trợ truy xuất nội dung phản hồi dưới dạng
Text: `response.text`
JSON: `response.json`
Binary: `response.content`

8.2.3.4 Lý do sử dụng thư viện

Thư viện Requests được áp dụng để thực hiện việc tải dữ liệu từ một nguồn bên ngoài thông qua giao thức HTTP. Cụ thể, Requests giúp gửi một yêu cầu HTTP GET đến một URL chứa file CSV, từ đó lấy dữ liệu sao kê về. Việc sử dụng Requests đơn giản hóa quá trình giao tiếp với các dịch vụ web khác, giúp ứng dụng web dễ dàng nhận và xử lý dữ liệu từ các nguồn trực tuyến mà không cần phải lưu trữ dữ liệu cục bộ.

8.2.3.5 Áp dụng trong bài tập lớn

- Gửi yêu cầu HTTP GET

```
response = requests.get(file_url, headers=headers)
```

`requests.get(file_url)` là yêu cầu HTTP GET để tải dữ liệu từ `file_url` (mặc định là `https://s.thuanle.me/chuyen_khoan.csv`). `headers` là một từ điển chứa User-Agent, giúp yêu cầu HTTP giống như một trình duyệt thông thường. Điều này giúp tránh bị chặn hoặc hạn chế truy cập từ server.

- Kiểm tra mã trạng thái HTTP

```
if response.status_code == 200:
    csv_data = StringIO(response.text)
    self.data = pd.read_csv(csv_data)
else:
    raise Exception(f"Lỗi khi tải dữ liệu. Mã lỗi: {response.status_code}")
```

Sau khi gửi yêu cầu, mã trạng thái HTTP được kiểm tra để xác định xem yêu cầu có thành công không. Mã 200 đại diện cho một yêu cầu thành công. Nếu không, trả về "Lỗi khi tải dữ liệu. Mã lỗi: `response.status_code`"

9 Screen Flow

Quy trình tìm kiếm trên website bao gồm ba bước chính, được mô tả như sau:

Bước 1 - Nhập Tiêu Chí Tìm Kiếm: Người dùng nhập các tiêu chí tìm kiếm vào các ô nhập liệu, bao gồm khoảng số tiền và từ khóa chi tiết.



Hình 6: Màn hình nhập tiêu chí tìm kiếm.

Bước 2 - Thực Hiện Tra Cứu: Người dùng nhấn nút "Tìm Kiếm", ứng dụng sẽ gửi yêu cầu tìm kiếm và xử lý ngay lập tức.



Hình 7: Màn hình thực hiện tra cứu sau khi nhấn nút tìm kiếm.

Bước 3 - Xem Kết Quả: Sau khi tra cứu xong, kết quả sẽ được hiển thị trên màn hình.



TRA CỨU SAO KẾ

Số tiền truy vấn từ thẻ tín dụng (Credit) Số tiền truy vấn từ thẻ ghi nợ (Debit)

Từ: Đến: Từ: Đến:

Nội dung giao dịch Tên người gửi

Nhập nội dung giao dịch NGUYEN QUYNH ANH

Tim kiếm

Kết quả: 92 giao dịch

credit	date_time	debit	detail	trans_no
200000	09/09/2024_5212.91413	0	753508.090924.153136.18FT NGUYEN QUYNH ANH chuyển tiền	2213
2000000	09/09/2024_5218.3069	0	405511.090924.185926.Nguyen Quynh Anh chuyển tiền ung ho dong bao khu vuc bao lu	3271
3000000	09/09/2024_5389.63102	0	020097042209092146512024G90X756399.63102_214652.NGUYEN QUYNH ANH chuyển tiền ho tro dong bao sau bao	6306
100000	10/09/2024_5213.69323	0	848592.090924.225548.NGUYEN QUYNH ANH chuyển tiền den MAT TRAN TO QUOC VN - BAN CUU TRO TW - 0011001932418	7226
300000	10/09/2024_5017.3492	0	401547.090924.232549.Nguyen Quynh Anh chuyển tiền	8195

Hình 8: Màn hình hiển thị kết quả tìm kiếm.