

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



Assignment

ADVANCED PROGRAMMING

Đề tài:

Web sao kê

GVHD: Lê Đình Thuận
Lớp: L01

SV thực hiện: Hoàng Mạnh Đức - 2210787
Nguyễn Phương Hoài Vũ - 2213999
Thang Khiết Dương - 2210623
Huỳnh Hữu Kha - 2211412
Nguyễn Duy Ngọc - 2333012
Phạm Đình Chương - 2210400

TP. HỒ CHÍ MINH, THÁNG 12/2024

Mục lục

I.	Mở đầu	3
II.	Giới thiệu hệ thống và các tính năng của hệ thống	3
1	Giới thiệu hệ thống	3
2	Các tính năng của hệ thống	3
III.	Công nghệ cho dự án	3
1	Git	3
2	ReactJS	4
3	Vite	5
4	Tailwin CSS	5
5	Flask	6
6	Visual Studio Code	6
IV.	Cấu trúc dữ liệu và giải thuật tìm kiếm	7
1	List và Dictionary	7
2	Inverted Index	8
3	Bucketing	8
4	Combined search	10
V.	Cách chạy	10
VI.	Kết luận	11
VII.	Link sourcecode và Slide thuyết trình	12
VIII.	Tư liệu tham khảo	12
1	Học sử dụng HTML, CSS	12
2	Học cách sử dụng Vite và TailwinCSS	12
3	Học sử dụng ReactJS	12
4	Học sử dụng Flask	12



DANH SÁCH THÀNH VIÊN - TIẾN ĐỘ CÔNG VIỆC

No.	Fullname	Student ID	Percentage of work
1	Hoàng Mạnh Đức	2210787	100%
2	Nguyễn Phương Hoài Vũ	2213999	100%
3	Thang Khiết Dương	2210623	100%
4	Huỳnh Hữu Kha	2211412	100%
5	Nguyễn Duy Ngọc	2333012	100%
6	Phạm Đình Chương	2210400	100%

I Mở đầu

Lập trình nâng cao (CO2039) là môn học giúp sinh viên tiếp cận chuyên sâu và nâng cao kiến thức và kỹ năng cần thiết để trở thành những nhà phát triển phần mềm vượt trội.

Trên nền tảng của những kiến thức cơ bản về lập trình, môn học này đưa sinh viên vào những khía cạnh phức tạp và thú vị hơn của ngôn ngữ lập trình. Từ việc làm quen với cấu trúc dữ liệu và giải thuật đến việc phát triển phần mềm hoàn chỉnh, sinh viên sẽ được đào tạo để làm chủ những kỹ thuật lập trình tiên tiến.

II Giới thiệu hệ thống và các tính năng của hệ thống

1 Giới thiệu hệ thống

Web Sao Kê là một hệ thống trực tuyến hiện đại, được thiết kế để cung cấp giải pháp quản lý tài chính và minh bạch hóa các giao dịch một cách hiệu quả và tiện lợi. Hệ thống này cho phép người dùng dễ dàng theo dõi, kiểm tra, và quản lý các sao kê tài khoản, giúp tiết kiệm thời gian và nâng cao trải nghiệm trong việc xử lý thông tin tài chính cá nhân hoặc doanh nghiệp.

2 Các tính năng của hệ thống

Với giao diện thân thiện và trực quan, Web Sao Kê mang đến các tính năng nổi bật như:

- Tìm kiếm và đối chiếu thông tin nhanh chóng, giảm thiểu sai sót.
- Hỗ trợ báo cáo tài chính tùy chỉnh phù hợp với nhu cầu cá nhân hoặc tổ chức.

III Công nghệ cho dự án

1 Git

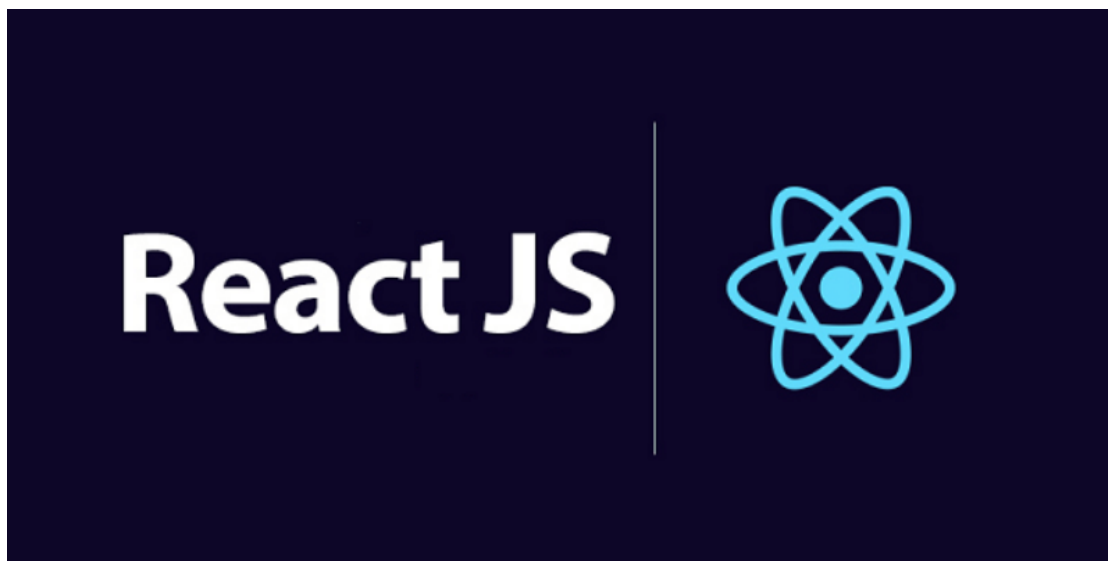


Hình 1: Sử dụng công nghệ github

GitHub là một nền tảng phổ biến để lưu trữ và hợp tác trên các dự án mã nguồn mở bằng cách sử dụng hệ thống quản lý phiên bản Git. Nó được sử dụng rộng rãi bởi các nhà phát triển và các nhóm để quản lý các kho lưu trữ mã nguồn, theo dõi các thay đổi và phối hợp công việc

trên các dự án phần mềm. GitHub cung cấp các tính năng như theo dõi vấn đề, yêu cầu kéo (pull requests), bảng dự án và wikis, làm cho nó trở thành một công cụ linh hoạt cho cả các nhà phát triển cá nhân và các tổ chức lớn. Nó cũng là một trung tâm cho các dự án mã nguồn mở, nơi các nhà phát triển có thể chia sẻ mã nguồn của họ với cộng đồng và hợp tác với nhau để cải thiện mã nguồn.

2 ReactJS



Hình 2: Sử dụng công nghệ ReactJs

Dự án này sử dụng ReactJS làm công cụ chính để xây dựng giao diện người dùng của ứng dụng web. Bằng cách sử dụng ReactJS, đã tạo ra các thành phần UI linh hoạt và tái sử dụng, giúp tăng tính linh hoạt và hiệu quả trong quá trình phát triển.

Trong toàn bộ dự án, đã sử dụng các khái niệm như "component-based architecture" để phân chia giao diện người dùng thành các thành phần nhỏ và độc lập, mỗi thành phần đảm nhận một phần của giao diện. Điều này giúp dễ dàng quản lý và tái sử dụng các thành phần, cũng như tạo ra một cấu trúc code rõ ràng và dễ bảo trì.

Ngoài ra, tận dụng các tính năng của ReactJS như quản lý trạng thái của ứng dụng bằng cách sử dụng các state và props của các thành phần, cũng như sử dụng các thư viện bổ sung như React Router và Redux để quản lý định tuyến và trạng thái toàn cục của ứng dụng. Điều này giúp xây dựng các ứng dụng web phức tạp một cách dễ dàng và hiệu quả.

3 Vite



Hình 3: Sử dụng Vite để xây dựng dự án

Vite là một công cụ build hiện đại được thiết kế để tối ưu hóa quá trình phát triển và triển khai ứng dụng web. Ra đời nhằm khắc phục những hạn chế của các công cụ truyền thống như Webpack, Vite sử dụng mô hình ESModules để tăng tốc độ khởi động dự án và cải thiện hiệu năng trong quá trình phát triển. Với tính năng nổi bật như Hot Module Replacement (HMR), Vite cho phép cập nhật giao diện ngay tức thì khi bạn thay đổi mã nguồn mà không cần tải lại toàn bộ trang. Ngoài ra, Vite tích hợp hoàn hảo với các framework phổ biến như React, Vue, và Svelte, đồng thời cung cấp khả năng build sản phẩm tối ưu với Rollup. Nhẹ, nhanh và linh hoạt, Vite không chỉ phù hợp với các dự án nhỏ mà còn là giải pháp mạnh mẽ cho các ứng dụng lớn cần hiệu năng cao và cấu hình dễ dàng.

4 Tailwin CSS



Hình 4: Sử dụng TailwinCSS để tạo giao diện

Tailwind CSS là một framework CSS theo hướng utility-first, được thiết kế để giúp lập trình viên tạo giao diện nhanh chóng và hiệu quả. Khác với các framework truyền thống như Bootstrap, Tailwind không cung cấp các thành phần giao diện dựng sẵn mà thay vào đó cung cấp một bộ class utility phong phú, linh hoạt, cho phép bạn tùy chỉnh thiết kế trực tiếp trong HTML mà không cần viết CSS thủ công. Với khả năng tùy chỉnh cao thông qua file cấu hình, hỗ trợ thiết kế responsive, và tích hợp sẵn các công cụ tối ưu hóa như loại bỏ CSS không sử dụng (purge CSS), Tailwind CSS giúp cải thiện hiệu suất và tính thẩm mỹ cho dự án. Đây là lựa chọn lý tưởng để xây dựng giao diện hiện đại, đẹp mắt, và dễ dàng mở rộng.

5 Flask



Hình 5: Sử dụng Flask để phát triển web

Flask là một microframework mạnh mẽ và linh hoạt dành cho phát triển web trong Python. Được thiết kế với triết lý tối giản, Flask cung cấp các công cụ cần thiết để xây dựng ứng dụng web mà không áp đặt quá nhiều ràng buộc về cấu trúc. Nhờ sử dụng WSGI (Web Server Gateway Interface) và Jinja2 (một công cụ template mạnh mẽ), Flask cho phép lập trình viên dễ dàng xử lý routing, quản lý request-response, và tích hợp với các thư viện hoặc cơ sở dữ liệu bên ngoài. Điểm đặc biệt của Flask là tính tùy biến cao, phù hợp với cả dự án nhỏ và lớn, đồng thời có cộng đồng hỗ trợ rộng rãi, giúp người dùng nhanh chóng xây dựng và triển khai ứng dụng web hiệu quả.

6 Visual Studio Code



Hình 6: Sử dụng tool Visual Studio Code để lập trình

Visual Studio Code (VS Code) là một trình biên tập mã nguồn mở và miễn phí được phát

triển bởi Microsoft. Nó là một trong những trình biên tập mã nguồn phổ biến nhất được sử dụng bởi các nhà phát triển trên toàn thế giới.

VS Code cung cấp một loạt các tính năng hữu ích để giúp các nhà phát triển làm việc hiệu quả, bao gồm:

Hỗ trợ nhiều ngôn ngữ lập trình: VS Code hỗ trợ nhiều ngôn ngữ lập trình phổ biến như JavaScript, Python, HTML/CSS, và nhiều ngôn ngữ khác.

Cú pháp kiểm tra và gợi ý: Nó cung cấp tính năng gợi ý cú pháp và kiểm tra lỗi giúp làm việc nhanh chóng và chính xác hơn.

Mở rộng và tích hợp: VS Code cho phép cài đặt các tiện ích mở rộng (extensions) để mở rộng chức năng của nó. Các tiện ích mở rộng có thể bao gồm hỗ trợ ngôn ngữ mới, tích hợp với các công cụ phát triển khác như Git, Docker, hay hỗ trợ cho các framework và thư viện như ReactJS, Angular, và Vue.js.

Debugging: Nó cung cấp tính năng debugging cho nhiều ngôn ngữ lập trình, giúp người phát triển dễ dàng gỡ lỗi mã nguồn.

Control Versioning: VS Code tích hợp với hệ thống quản lý phiên bản như Git, cho phép bạn quản lý mã nguồn của dự án một cách thuận tiện.

Tổng thể, Visual Studio Code là một công cụ phổ biến và mạnh mẽ, được sử dụng rộng rãi trong cộng đồng phát triển phần mềm.

IV Cấu trúc dữ liệu và giải thuật tìm kiếm

1 List và Dictionary

List là một cấu trúc dữ liệu tuyến tính, lưu trữ các phần tử theo thứ tự và cho phép truy cập bằng chỉ số (index), dict (dictionary) là cấu trúc dữ liệu ánh xạ (key-value), cho phép truy xuất nhanh dựa trên key.

Trong code, dữ liệu từ file csv được nạp vào dưới dạng list các dict:

```
def load_data(file_path):  
    data = []  
    with open(file_path, 'r', encoding='utf-8-sig') as file:  
        reader = csv.DictReader(file)  
        for row in reader:  
            row['credit'] = float(row['credit']) if row['credit'] else 0  
            row['debit'] = float(row['debit']) if row['debit'] else 0  
            # Add a new 'transaction_amount' column as the max of 'credit' and 'debit' (since its  
            row['transaction_amount'] = max(row['credit'], row['debit'])  
            data.append(row)  
    return data  
  
bank_data = load_data(file_path)
```

Thao tác này chỉ chạy 1 lần khi khởi động chương trình, nên mỗi lần query không cần phải mở file csv lên và dò toàn bộ file.

Thêm nữa, 2 giải thuật tìm kiếm Inverted Index và Bucketing dưới đây đều dùng list và dict để query.

2 Inverted Index

Inverted index là một cấu trúc dữ liệu thường được dùng trong các hệ thống tìm kiếm văn bản. Nó giúp truy xuất các mục liên quan nhanh chóng bằng cách ánh xạ từ khóa đến vị trí chứa các từ khóa đó.

Ví dụ:

- row[0]: "Hoai Vu loves coding"
- row[1]: "Hoai enjoys music"
- row[2]: "Vu plays the guitar"

"Hoai" sẽ được ánh xạ đến [0, 1], "Vu" sẽ được ánh xạ đến [2]. Khi query đến các từ khóa này, nó sẽ tự động ánh xạ đến các dòng thích hợp với thời gian thực $O(1)$.

Trong code, sẽ ánh xạ mỗi từ trong cột 'detail' từ dữ liệu nguồn đến danh sách các chỉ số của các dòng chứa các từ đó:

```
inverted_index = defaultdict(list)
for i, row in enumerate(bank_data):
    words = re.findall(r'\b\w+\b', row['detail'].lower())    # Split by all special character
    for word in words:
        inverted_index[word].append(i)
```

Sau đó, khi query `search_term`, chương trình sẽ chia query thành các từ ("Bach Khoa" -> ["Bach", "Khoa"]) rồi tìm các dòng dữ liệu chứa mỗi từ này dùng `Inverted Index`, sau đó giao các tập hợp lại với nhau:

```
if search_term:
    words = re.findall(r'\b\w+\b', search_term.lower())
    term_matches = None    # Start with no matches

    for word in words:
        if word in inverted_index:
            word_matches = set(inverted_index[word])
            if term_matches is None:
                term_matches = word_matches    # Initialize with first word's matches
            else:
                term_matches &= word_matches    # AND
        else:
            term_matches = set()    # If a word has no matches, the result is empty (cuz AND)
            break
```

Mỗi query dùng `Inverted Index` sẽ có thời gian thực $O(K + R)$, với K là số từ khóa cần tìm và R là số kết quả tìm được.

3 Bucketing

Bucketing là một kỹ thuật chia tập dữ liệu thành các nhóm nhỏ (bucket) dựa trên một giá trị khóa. Mỗi bucket đại diện cho một khoảng giá trị liên tiếp, qua đó giảm số lượng dữ liệu cần xử lý bằng cách chỉ xét các bucket liên quan đến điều kiện lọc.

Ví dụ: Nếu dữ liệu là các giao dịch tài chính và giá trị giao dịch nằm trong khoảng [0, 50,000], ta có thể chia thành các bucket như:

- Bucket 0: [0–4,999]
- Bucket 1: [5,000–9,999]
- Bucket 2: [10,000–14,999]
- ...

Nếu cần tìm giao dịch từ 7,000–12,000, chỉ xét bucket 1 và bucket 2.
Chương trình sẽ chia nhỏ các dữ liệu dựa trên số tiền được chuyển:

```
bucket_size = 100000
buckets = defaultdict(list)
for i, row in enumerate(bank_data):
    bucket_id = int(row['transaction_amount'] // bucket_size)
    buckets[bucket_id].append((row['transaction_amount'], i))  # Store amount and index
```

Dữ liệu trong các bucket là tuple gồm số tiền được chuyển và index. Lý do lưu lại số tiền chuyển để có thể sort các bucket, sau đó có thể dùng Binary search để giảm thời gian query.

```
for bucket_id in buckets:
    buckets[bucket_id].sort(key=lambda x: x[0])  # Sort by transaction_amounts
```

Khi query min_amount và max_amount chương trình sẽ tìm khoảng bucket cần tìm, skip qua các bucket ngoài khoảng ấy, và dùng Binary search để tìm các dòng dữ liệu thỏa mãn query:

```
from bisect import bisect_left, bisect_right

if min_amount is not None or max_amount is not None:
    amount_matches = set()

    # Bucket ID search range
    min_bucket_id = int(min_amount // bucket_size) if min_amount is not None else None
    max_bucket_id = int(max_amount // bucket_size) if max_amount is not None else None

    for bucket_id, records in buckets.items():
        # Skip buckets outside the search range
        if (min_bucket_id is not None and bucket_id < min_bucket_id) or \
            (max_bucket_id is not None and bucket_id > max_bucket_id):
            continue

        # Perform binary search to find the range of relevant data
        lower_idx = bisect_left(records, (min_amount, -float('inf'))) if min_amount is not None else 0
        upper_idx = bisect_right(records, (max_amount, float('inf'))) if max_amount is not None else len(records)

        # Add the relevant indices to the matches
        amount_matches.update(record[1] for record in records[lower_idx:upper_idx])  # indices
```

Thời gian thực của giải thuật tìm kiếm này là xấp xỉ $O(B \cdot \log S + R)$, trong đó B là số các bucket trong khoảng cần tìm, S là size của mỗi bucket, R là số các kết quả tìm được

4 Combined search

Query từ khóa dùng Inverted Index và query miền giá trị tiền dùng bucketing. Sau đó, giao hai tập hợp thỏa mãn những điều kiện tương ứng trên lại với nhau, qua đó lấy được kết quả cuối.

```
matching_indices = set(range(len(bank_data)))

# Inverted index search
matching_indices &= term_matches

# Bucketing search
matching_indices &= amount_matches

return [bank_data[i] for i in matching_indices]
```

V Cách chạy

Đầu tiên, cần cài các library cần thiết:

```
pip install Flask
npm install
```

Chú ý: Nếu dùng Windows thì cần phải chạy command sau trước khi install npm:

```
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
```

Sau đó, mở 2 terminal, 1 terminal sẽ di chuyển đến thư mục backend và chạy file main.py

```
python main.py
```

Terminal còn lại sẽ di chuyển đến thư mục frontend rồi chạy:

```
npm run dev
```

Terminal 2 sẽ hiện lên link đến <http://localhost:5173/>, click link đó để tương tác với web.

VI Kết luận

Dự án Web Sao Kê đã được thiết kế và xây dựng với mục tiêu mang lại một giải pháp quản lý tài chính trực quan, hiệu quả và bảo mật cao cho người dùng. Việc sử dụng các công nghệ hiện đại như Vite, Tailwind CSS và Flask không chỉ đảm bảo tốc độ phát triển nhanh chóng mà còn tối ưu hóa hiệu suất, mang đến trải nghiệm tốt nhất cho người dùng.

Hệ thống hỗ trợ trích xuất sao kê tự động và cung cấp báo cáo tài chính chính xác, góp phần tăng tính minh bạch và thuận tiện trong quản lý tài chính cá nhân hoặc doanh nghiệp. Đồng thời, giao diện thân thiện và khả năng tùy chỉnh linh hoạt giúp dự án đáp ứng được đa dạng nhu cầu của người dùng.

Kết quả của dự án không chỉ là một sản phẩm hoàn thiện về mặt kỹ thuật, mà còn khẳng định tầm quan trọng của việc áp dụng các công nghệ tiên tiến vào thực tế.



VII Link sourcecode và Slide thuyết trình

Nguồn code có thể truy cập ở đây [sourcecode](#)
Slide thuyết trình [slide](#)

VIII Tư liệu tham khảo

1 Học sử dụng HTML, CSS

- [Học sử dụng HTML, CSS \(YouTube Playlist\)](#)
- [Học sử dụng HTML \(W3Schools\)](#)
- [Học sử dụng CSS \(W3Schools\)](#)

2 Học cách sử dụng Vite và TailwinCSS

- [Học cách sử dụng Vite \(YouTube Playlist\)](#)
- [Học cách sử dụng Tailwin CSS \(YouTube Playlist\)](#)

3 Học sử dụng ReactJS

- [Học sử dụng ReactJS \(YouTube Playlist\)](#)

4 Học sử dụng Flask

- [Python Website Full Tutorial - Flask, Authentication, Databases](#) [More](#)