**BÁO CÁO THỰC TẬP TUẦN 2  
VNPT-IT3 – PYTHON**

1. Nhiệm vụ tuần 2:

|  |  |
| --- | --- |
| Thứ 2 | Tìm hiểu về Python |
| Thứ 3 | Tìm hiểu môi trường ảo pipenv, thao tác, cài thư viện (suggest sử dụng vscode) |
| Thứ tư | Thứ 4: Python thao tác SQLite, NoSQL vs SQL khác nhau như thế nào, ưu nhược của mỗi loại, khi nào áp dụng. |
| Thứ 5 + 6 | Tìm hiểu về Git:  Git cơ bản và nâng cao, các rule khi commit, thao tác trên trang quản lý, xử lý conflict khi merge code |

1. Báo cáo nội dung thực hiện:
2. Tìm hiểu về Python.  
   Do em đã và đang lập trình với Python, nên ở nhiệm vụ này em sẽ tìm kiếm các thông tin trên Internet để tìm hiểu để hiểu thêm về các tính năng của ngôn ngữ Python, những lợi ích khi sử dụng Python và biết được Python lập trình cho những mảng nào là tốt nhất…
3. Môi trường ảo Pipenv
4. Tìm hiểu về Pipenv

Pipenv là công cụ quản lý gói kết hợp chức năng của Pip và Virtualenv, cùngvớicác tính năng tốt nhất của các công cụ đóng gói từ các ngôn ngữ hác như Bundler và NPM. Điều này dẫn đến một quy trình công việc đơn giản hóa để cài đặt cácgói vàquản lý môi trường ảo.

Các vấn đề mà Pipenv cần giải quyết:

* Không còn cần phải sử dụng pip và virtualenv riêng biệt.
* Việc quản lý tệp requirements.txt có thể có vấn đề, vì vậy PipenvsửdụngPipfile và Pipfile.lock để tách các khai báo phụ thuộc trừu tượng khỏi kết hợp được thử nghiệm cuối cùng.
* Hash được sử dụng ở mọi nơi, mọi lúc. An toàn
* Khuyến khích mạnh mẽ việc sử dụng các phiên bản phụ thuộc mới nhất đểgiảm thiểu rủi ro bảo mật phát sinh từ các thành phần lỗi thời.

1. Các lệnh để sử dụng Pipenv

* Ban đầu, ta tạo thư mục bằng lệnh “mkdir <tên thư mục>” và trỏ đến thư mục bằng “cd <tên thư mục>”
* Pip install pipenv: Cài đặt môi trường ảo pipenv
* Pipenv install: Tạo môi trường ảo pipenv cho mỗi thư mục
* pipenv --python 3.10: Tạo ra virtual environment và 2 file Pipfile và Pipfile.lock khi bạn khởi tạo project mới
* pipenv shell: Để khởi chạy terminal trong virtual environment
* pipenv requirements > requirements.txt: xuất file chứa thông tin về thư viện.
* pipenv lock -r > requirements.txt :cập nhật file requirements

1. Tìm hiểu về SQLite.
2. Giới thiệu về SQLite:

SQLite là một hệ quản trị cơ sở dữ liệu hay còn gọi là hệ thống cơ sở dữ liệu quan hệ nhỏ gọn, khác với các hệ quản trị khác như MySQL, SQL Server, Ocracle, PostgreSQL… SQLite là một thư viện phần mềm mà triển khai một SQL Database Engine truyền thống, không cần mô hình client-server nên rất nhỏ gọn. SQLite được sử dụng vào rất nhiều chương trình từ desktop đến mobile hay là website.

1. Các thao tác cơ bản trên SQLite.  
   SQLite là một hệ quản trị cơ sở dữ liệu hay còn gọi là hệ thống cơ sở dữ liệu quan hệ nhỏ gọn, khác với các hệ quản trị khác như MySQL, SQL Server, Ocracle, PostgreSQL… SQLite là một thư viện phần mềm mà triển khai một SQL Database Engine truyền thống, không cần mô hình client-server nên rất nhỏ gọn. SQLite được sử dụng vào rất nhiều chương trình từ desktop đến mobile hay là website.
2. So sánh SQL và NoSQL:

* NoSQL database chỉ là một kiểu database có cách lưu trữ, truy

vân dữ liệu hoàn toàn khác so với RDBMS và SQL.

* NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suấtnhanh và khả năng mở rộng (scalability). Chính những ưu điêm trên mànNoSQL được sử dụng nhiều trong các dự án Big Data, các dự án Real-time, số lượng dữ liệu nhiều.

|  |  |  |
| --- | --- | --- |
| Tính năng | SQL | NoSQL |
| Hiệu suất | Kém hơn NoSQL, vì khi truy vấn nó phải tính toán, kiểm tra và xử lý các mối quan hệ các bảng. | Tốt hơn SQL vì nó bỏ qua các ràng buộc |
| Mở rộng theo chiều ngang | CÓ thể thực hiện được nhưng quá trình mở rộng sẽ rất phức tạp nếu đã tồn tại dữ liệu trong database. | Mở rộng dễ dàng |
| Tốc độ read/write | Kém hơn NoSQL vì phải đảm bảo tính ràng buộc dữ liệu giữa các bảng.  Nếu sử dụng nhiều server thì phải bảo toàn tính nhất quán về dữ liệu ở các server với nhau. | Tốc độ nhanh hơn SQL vì nó bỏ qua các cơ chế ràng buộc của các bảng.  Vì dữ liệu được lưu trong RAM, sau đó mới đẩy xuống HDD và có tính nhất quan cuối. |
| Phần cứng | Đòi hỏi phần cứng cao | Không đòi hỏi quá cao về phần cứng |
| Thay đổi số node trong hệ thống | Vì tính nhất quán về dữ liệu nên khi thêm hay xóa 1 node cần phải shutdown hệ thống trong khoảng 1 thời gian | Vì tính nhất quán cuối nên sẽ không cần phải shutdown hệ thống |
| Truy vấn và báo cáo | Dễ dàng sử dụng ngôn ngữ SQL query để truy vẫn trực tiếp dữ liệu từ database hoặc dùng công cụ hỗ trợ để lấy báo cáo | Việc lấy báo cáo dữ liệu trực tiếp từ NoSQL chưa được hỗ trợ tốt, thực hiện chủ yếu thông qua giao diện ứng dụng. |
| Mở rộng dữ liệu | Khi muốn bổ sung thêm cột cho một bảng, chúng ta phải khai báo trước. | Không cần khai báo trước. |
| Ứng dụng | Sử dụng để xây dựng những hệ thống có quan hệ chặt chẽ và cần tính đồng nhất về dữ liệu như: tài chính, ngân hàng, chứng khoán…. | Sử dụng xây dựng những hệ thống lưu trữ thông tin lớn, không quá quan trọng về vấn đề đồng nhất dữ liệu trong 1 thời gian nhất định. |

1. Git.
2. Git là gì?

Git là hệ thống kiểm soát phiên bản phân tán mà nguồn mở. Các dự án thực tế thường có nhiều nhà phát triển làm việc song song. Vì vậy, một hệ thống kiểm soát phiên bản như Git là cần thiết để đảm bảo không có xung đột mã giữa các nhà phát triển. Ngoài ra, các yêu cầu trong dự án thay đổi thường xuyên. Vì vậy, cần một hệ thống cho phép nhà phát triển quay lại phiên bản cũ hơn của mã.

Repository là nơi chứa tất cả mã nguồn cho một dự án được tạo bởi Git. Hoặc có thể hiểu Repository chính khai báo thư mục chứa dự án của bạn trên local hoặc remote. Một repository sẽ có hai cấu trúc dữ liệu chính đó là Object store và Index được lưu trữ ẩn trong thư mục .git.

Có hai loại repository là:local repository và remote repository:

Local repository: là repository được cài đặt trên máy tính của lập trình viên, repository này sẽ đồng bộ hóa với remote bằng các lệnh của Git.

Remote repository: là repository được cài đặt trên server chuyên dụng, điển hình hiện nay là Github.  Branch là những phân nhánh ghi lại luồng thay đổi của lịch sử, các hoạt động trên mỗi branch sẽ không ảnh hưởng lên các branch khác nên có thể tiến hành nhiều thay đổi đồng thời trên một repository giúp giải quyết nhiều nhiệm vụ cùng lúc. Khi tạo một repository thì Git sẽ thiết lập branch mặc định là master.

1. Một số câu lệnh cơ bản của Git.

-Cấu hình tên tài khoản trên local: git config –global user.name [username].

-Cấu hình email trên local: git config -global user.email [youremail].

-Tải source về local đang gọi : git clone [url].

-Tạo kết nối trên local: git init.

-Thêm các file cần tải lên vào index: git add [namefile].

-Tạo commit từ local với sever: git commit -m "message".

-Tạo nhánh mới: git branch [namebranch].

-Truy cập vào nhánh: git checkout [namebranch].

-Gộp nhánh: git merge -Tải soure: git pull -Tạo kết nối với server: git remote add origin [url].