TRƯỜNG ĐẠI HỌC ĐÀ NẴNG
ĐẠI HỌC BÁCH KHOA

TRUNG TÂM VNPT-IT KHU VỰC 3

BÁO CÁO THỰC TẬP TUẦN 2

Thời gian: Từ 04/07/2022 đến 08/07/2022

I. Nhiệm vụ:

Sinh viên thực hiện: Lê Văn Mẫn

Bảng 1.1: Phân bố nhiệm vụ trong tuần

Thời gian	Nhiệm vụ
Thứ hai (04/07/2022)	Ôn tập kiến thức cơ bản về Python
Thứ ba (05/07/2022)	Tìm hiểu môi trường ảo pipevn và các thao tác thực hiện.
Thứ tư (06/07/2022)	Các thao tác với SQLite bằng Python. Tìm hiểu sự khác biệt giữa NoSQL và SQL.
Thứ năm (07/07/2022)	Tìm hiểu về Git
Thứ sáu (08/07/2022)	Tìm hiểu về Git

II. Nội dung báo cáo

1. Kiến thức cơ bản về Python

Ôn tập các kiến thức cơ bản của Python và thực hiện một số ví dụ liên quan.

Một số ví dụ về bài tập Python được mô tả ở **đây**

2. Pipenv

2.1. Giới thiệu Pipevn

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

Các vấn đề mà Pipenv tìm cách 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 Pipenv sử dụng Pipfile 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 để giảm thiểu rủi ro bảo mật phát sinh từ các thành phần lỗi thời.

- Hợp lý hóa quy trình phát triển bằng cách tải các tệp *.env*.

2.2. Một số thao tác thực hiện với Pipenv

- Cài đặt pipenv: $ `pip install pipenv`
- Tạo một shell môi trường ảo: $ `pipenv shell`
- Cài đặt flash phiên bản phù hợp: $ `pipevn install flask == 0.12.1`
- Cài đặt thư viện numpy : $ `pipevn install numpy`
- Cài đặt requests kiểm soát phiên bản: $ `pipevn install -e git+https://github.com/requests/requests.git#egg=requests`
- Khóa môi trường: $ `pipevn lock`
- Tìm môi trường ảo: $ `pipenv --venv`
- Tìm project: $ `pipevn --where`

2.3. Ví dụ

Một số ví dụ thực hiện thao tác với Pipevn được mô tả ở **đây**

3. SQL

3.1. Giới thiệu 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

3.2. Một số thao tác cơ bản với SQLite

- Tạo ra một bảng (table) trong cơ sở dữ liệu.
- Nạp dữ liệu cơ sở dữ liệu từ các bản ghi dữ liệu.
- Lấy dữ liệu từ cơ sở dữ liệu SQLite.
- Update bản ghi dữ liệu trong cơ sở dữ liệu SQLite.
- Delete bản ghi dữ liệu trong cơ sở dữ liệu SQLite.

Để sử dụng SQLite, ta phải **import sqlite3** vào project.

3.3. Ví dụ

Một số ví dụ thực hiện thao tác với SQLite được mô tả ở **đây**

3.4. SQL và NoSQL

SQL là hệ cơ sở dữ liệu được lưu trữ theo dạng bảng, có quan hệ rõ ràng và yêu cao về phần cứng.

NoSQL là hệ cơ sở dữ liệu phân tán, không ràng buộc, có thể chứa hàng petabytes, độ chịu tải cao và không yêu cầu về tài nguyên phần cứng.

Bảng 3.1: So sánh NoSQL và SQL dựa vào tính năng của mỗi loại

Tính năng	SQL	NoSQL
Hiệu suất	Kém hơn NoSQL vì khi truy vấn phải tính toán kiểm tra và xử lý các mối quan hệ bảng	Tốt hơn SQL vì bỏ qua các ràng buộc
Mở rộng theo chiều ngang	Qúa trình mở rộng sẽ phức tạp nếu đã tồn tại dữ liệu trong database	Mở rộng dễ dàng
Tốc độ read/write	Kém hơn NoSQL vì tính ràng buộc dữ liệu. Nếu sử dụng quá nhiều server phải bảo toàn tính nhất quán về dữ liệu ở các server với nhau	Nhanh hơn SQL vì bỏ qua các cơ chế ràng buộc của các bảng. dữ liệu được lưu ở RAM, đẩy xuống HDD và có tính nhất quán cuối
Phần cứng	Đòi hỏi phần cứng cao	Không đòi hỏi quá cao về phần cứng
Thay đổi số node trong hệ thống	Khi thêm hoặc xóa 1 node cần phải shutdown hệ thống một khoảng thời gian vì tính nhất quán về dữ liệu.	Vì tính nhất quán cuối nên không cần shutdown hệ thống
Truy vấn và báo cáo	Dễ dàng sử dụng ngôn ngữ SQL query để truy vấn 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 dữ liệu cột cho bảng, 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 đồng nhất dữ liệu trong một thời gian nhất định như báo chí, mạng xã hội, diễn đàn,…

4. Git

4.1. Giới thiệu Git

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.

4.2. Một số thao tác với Git

Cấu hình tên cho tài khoản git trên máy: git config –global user.name "[tên người dùng]"

Cấu hình email cho tài khoản git trên máy: git config –global user.email [email]

Tải source về vị trí folder đang gọi terminal: git clone [url]

Commit: git commit -m "commit message"

git push, git pull, git diff, git status, git log

Tạo nhánh mới ở local từ nhánh hiện tại: git branch [name]

Merge nhánh khác vào nhánh hiện tại: git merge [branchname]

Lấy các thay đổi của lần stash gần nhấtgit stash pop

Tên nhánh ở local: git checkout [branchname]

Merge commit từ nhánh khác qua nhánh hiện tại: git cherry-pick [commit]

Revert file code đang ở trạng thái staging chờ push code: git reset HEAD ~1

4.3. Ví dụ

Một số ví dụ thực hiện thao tác với SQLite được mô tả ở **đây**