TRƯỜNG ĐẠI HỌC ĐÀ NẴNG
ĐẠI HỌC BÁCH KHOA

TRUNG TÂM VNPT-IT KHU VỰC 3

BÁO CÁO THỰC TẬP TUẦN 3

Thời gian: Từ 11/07/2022 đến 15/07/2022

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 (11/07/2022)	Tìm hiểu, áp dụng, chạy thử framework Flask
Thứ ba (12/07/2022)	Tìm hiểu, áp dụng, chạy thử framework Flask
Thứ tư (13/07/2022)	Viết API (sử dụng Flask) gửi email bằng Pyhon (email được nhập từ dữ liệu đầu vào)
Thứ năm (14/07/2022)	Tìm hiểu và chạy thử dự án nhỏ có kết hợp giữ Flask và SQLite
Thứ sáu (15/07/2022)	Tìm hiểu và chạy thử dự án nhỏ có kết hợp giữ Flask và SQLite

II. Nội dung báo cáo

1.Tìm hiểu về Framework Flask

Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask cho phép bạn xây dựng các ứng dụng web từ
đơn giản tới phức tạp. Nó có thể xây dựng các api nhỏ, ứng dụng web chẳng hạn như các trang web, blog, trang wiki hoặc một website dựa theo thời gian hay
thậm chí là một trang web thương mại. Flask cung cấp cho bạn công cụ, các thư viện và các công nghệ hỗ trợ bạn làm những công việc trên.
Flask là một micro-framework. Điều này có nghĩa Flask là một môi trường độc lập, ít sử dụng các thư viện khác bên ngoài. Do vậy, Flask có ưu điểm là
nhẹ, có rất ít lỗi do ít bị phụ thuộc cũng như dễ dàng phát hiện và xử lý các lỗi bảo mật.

- Ưu điểm và Nhược điểm của Flask
  + Như đã nêu trước đó, Flask được phân loại là Web Framework siêu nhỏ, nhẹ. Thông thường, một framework vi mô là một framework tối giản hoặc không phụ thuộc vào thư viện bên ngoài.
  Và trong mọi trường hợp, khi một lập trình viên sử bất kỳ framework nào, nó đều có ưu điểm và nhược điểm, flask cũng vậy:

  - Tốc độ
  - Hỗ trợ cho NoQuery

- Độ phức tạp tối thiểu
- Chủ nghĩa tối giản tuyệt đối
- Không có ORM, dễ dàng kết nối với tiện ích mở rộng
- Trình gỡ lỗi được nhúng trong trình duyệt
- Mã ngắn và đơn giản trong số các bộ xương Python khác

Điểm nổi bật khi sử dụng Flask để lập trình web là bạn sẽ rất ít bị phụ thuộc bên thứ 3, do đó đề phòng được các lỗi bảo mật.
Bạn có thể kiểm soát mọi thứ khi sử dụng Flask. Và quan trọng, học Flask giúp bạn hiểu các cơ chế bên trong các Framework khác. Đây là tiền đề tốt để bạn có thể nắm giữ nhiều công nghệ hơn.
Mặc dù nhược điểm của việc sử dụng Flask là đôi khi bạn phải tự mình làm nhiều việc hơn hoặc cần tự mình gọi thêm các tiện ích mở rộng.
Flask dựa trên Werkzeug (một thư viện tiện ích WSGI) và Jinja2 (template engine).

1.1.Thư viện có liên quan

SQLAlchemy
SQLAlchemy là một bộ công cụ SQL mã nguồn mở và ORM sử dụng trong ngôn ngữ lập trình Python, giúp hỗ trợ việc quản lý và thao tác với cơ sở dữ liệu.
SQLAlchemy cung cấp cho người dùng một ORM sử dụng mô hình thiết kế Data Mapper.
Được ra lò vào năm 2006, SQLAlchemy nhanh chóng khẳng định vị thế của mình trong cộng đồng lập trình viên Python, được sử dụng rất rộng rãi bên cạnh
Django's ORM. SQLAlchemy được sử dụng phổ biến ở cả những "ông lớn" trong ngành công nghệ như Yelp!, Reddit, Dropbox, …

Marshmallow
Là một thư viện cho phép người dùng có thế thực hiện các thao tác liên quan đến chuyển đổi dữ liệu từ cơ sở dữ liệu thành các kiểu dữ liệu khác nhau
đặc biệt là file Json.

1.2.Cài đặt Flask và các thư viện liên quan

Chúng ta có thể cài đặt Flask bằng Pipenv : *$pipenv install flask*
Cài đặt SQLAlchemy để quản lý cơ sở dữ liệu: *$pipenv install flask_sqlalchemy*
Cài đặt Marshmallow để chuyển đổi cơ sở dữ liệu sang Json: *$pipenv install flask_marshamallow*
Cài đặt thư viện liên quan đến thao tác email: *$pipenv install flask_mail*

1.3.Hoạt động của Flask

Một ví dụ cơ bản:
*from flask import Flask app = Flask(**name**) @app.route('/') def hello_world(): return 'Hello, World!' if **name** == '**main**': app.run()*

ở ví dụ trên thì để khởi tạo mội app thì ta dùng: *app = Flask(**name**)*
Khởi tạo URL: *@app.route('/')*

Để thực hiện các thao tác của đường dẫn ta sử dụng function ở ngay dưới đường đãn được khởi tạo.
Để thực hiện khởi chạy thì sử dụng: *app.run()*

2.Viết API để gửi email từ python

Cài đặt thư viện liên quan đến thao tác email: *$pipenv install flask_mail*.
Sau đó chúng ta có thể khai báo địa chỉ email và mật khẩu ứng dụng liên quan đến email nguồn.
Lấy địa chỉ email của người nhận.
Tiến hành gửi email theo nội dung và người nhận đã được nhập vào ở trên.

- *Code:*

```
# Flask- Mail Tutorial
# File & Directories (Creating vitual environment is optional)
# Frontend HTML File Setup (Press "! + Tab" to get boilerplate HTML)
# This block of code í setting up Flask-Mail to send emails (mail server, port, sender's email, etc)
from flask import Flask, render_template, request
from flask_mail import Mail, Message
import os

app = Flask(__name__)

app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = 'levanman1801@gmail.com'
app.config['MAIL_PASSWORD'] = '**************' # 'Nhập mật khẩu ứng dụng'
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True

mail = Mail(app)



@app.route('/home', methods= ['GET', 'POST'])
@app.route('/', methods= ['GET', 'POST'])
def home():
    if request.method == 'POST':
        msg = Message("Báo Cáo Ngày Thứ 4", sender='levanman1801@gmail.com', recipients=['leman9604@gmail.com','vutruong5438@gmail.com','thopn95@gmail.com'])
```

```
❖          msg.body = "Xin chào hai anh!(Em gửi bằng Flask) (Lê Văn Mẫn)"
❖          mail.send(msg)
❖          return "Sent email"
❖      return render_template('index.html')
❖
❖  if __name__ == '__main__':
❖      app.run(debug=True)
```