

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

NGUYỄN DUY QUANG – 20120360

BÁO CÁO BÀI TẬP TUẦN 03
MÔN HỌC ỨNG DỤNG THỊ GIÁC MÁY TÍNH
LỚP 20_23

GIÁO VIÊN HƯỚNG DẪN

Võ Hoài Việt

Tp. Hồ Chí Minh, tháng 10/2023

Mục lục

Mục lục	2
I. Giới thiệu	3
II. Thực hiện	4
III. Nội dung chính:	5
1. Cấu trúc ứng dụng:	5
a. Kiến trúc hệ thống:	5
b. Kiến trúc phần mềm:	5
c. Sơ đồ Use Case:	5
d. Flowchart	8
2. Kết quả	8
IV. Nhận xét	13
V. Tham khảo:	14

I. Giới thiệu

Ứng dụng cung cấp một giao diện căn bản cho người dùng thực hiện các chức năng:

- Đăng ký một tài khoản trên ứng dụng. Tài khoản chỉ yêu cầu hai thông số: Tên tài khoản và mật khẩu.
- Đăng nhập một tài khoản đã được đăng ký.
- Tải lên một ảnh. Bởi vì ảnh này được sử dụng như là ảnh đại diện nên mỗi tài khoản chỉ kết nối với một ảnh.
- Thực hiện phát hiện mặt người (face detection) trong ảnh. Khu vực sau khi được phát hiện sẽ được đóng khung.

II. Thực hiện

Ứng dụng sử dụng Django, một framework dành cho Python để xây dựng các ứng dụng web. Django hỗ trợ rất nhiều các công cụ và khuôn mẫu, giúp người lập trình không phải tự tạo lại hay phụ thuộc quá nhiều vào các công cụ thứ ba.

Ngoài ra, một số thư viện khác được dùng để thiết kế ứng dụng là:

- PostgreSQL: PostgreSQL là một hệ thống database mạnh mẽ, có thể phục vụ cho cả dữ liệu SQL và noSQL. PostgreSQL nổi tiếng về sự đáng tin cậy, hiệu suất và khả năng mở rộng. Đi cùng với PostgreSQL là Psycopg2, một bộ chuyển đổi cơ sở dữ liệu (database adapter) của PostgreSQL dành cho Python.
- Django-environ: Là một gói chứa các module giúp Python có thể xử lý file .env, giúp tách biệt môi trường khỏi codebase. Django-environ giúp ứng dụng có thể đi theo lối thiết kế 12-factor app.
- Pillow: Pillow là một thư viện Python dành cho xử lý ảnh căn bản. Trong ứng dụng này, Pillow được dùng kết hợp với PostgreSQL để lưu trữ ảnh trên database.
- OpenCV: Là một trong những thư viện hàng đầu trong xử lý ảnh và thị giác máy tính.

III. Nội dung chính:

1. Cấu trúc ứng dụng:

a. Kiến trúc hệ thống:

Ứng dụng được chia làm ba thành phần chính:

- Giao diện người dùng (user interface): Được chứa trong thư mục frontend. Thành phần này chủ yếu giúp hiển thị giao diện người dùng và quản lý các tương tác của người dùng với giao diện.
- Business logic: Được chứa trong thư mục backend. Thành phần này chứa các chức năng chính của ứng dụng, bao gồm xử lý các yêu cầu từ người dùng (mà cụ thể là từ user interface) và giao tiếp với thành phần truy cập dữ liệu.
- Truy cập dữ liệu (Data access): Được chứa một phần trong thư mục backend và thư mục media. Thành phần này nhận trách nhiệm tương tác với database để lưu trữ và nhận dữ liệu, với database được lưu ở 2 vị trí: Dữ liệu bảng được lưu tùy vào môi trường, và dữ liệu files được lưu trong thư mục media.

b. Kiến trúc phần mềm:

Các ứng dụng sử dụng framework Django tuân theo cấu trúc MVT. Cấu trúc MVT khá tương đồng với cấu trúc MVC thường gặp trong các ứng dụng khác.

- M - Model: Đại diện cho dữ liệu của ứng dụng. Model là các đối tượng trong ứng dụng, được ứng dụng sử dụng để xử lý và lưu trữ.
- V - Views: Chịu trách nhiệm trong việc xử lý các yêu cầu chính của ứng dụng. View thường được thể hiện dưới dạng các hàm nhận request và trả response, và cũng được sử dụng để trả về giao diện.
- T - Templates: Chịu trách nhiệm trong việc định nghĩa bố cục giao diện. Nói chung, giao diện của các ứng dụng Django sẽ dựa theo các file templates, thông qua chỉnh sửa của view mà trả về dưới dạng HTML.

c. Sơ đồ Use Case:

Yêu cầu: Người dùng tạo tài khoản.

- Actor: Người dùng.
- Goal: Người dùng tạo một tài khoản trên hệ thống.
- Preconditions:
 - o Người dùng truy cập vào được ứng dụng.
 - o Người dùng chưa có tài khoản hoặc muốn có một tài khoản mới.
 - o Người dùng đang ở trang chủ.
- Postconditions:
 - o Người dùng có một tài khoản có thể dùng để đăng nhập.
- Basic flow:

- Người dùng nhấp chọn nút "Register" trên trang chủ.
- Hệ thống chuyển đến trang đăng ký, đồng thời hiển thị form đăng ký gồm 3 mục: Tên tài khoản, mật khẩu và xác nhận mật khẩu.
- Người dùng điền thông tin vào form, sau đó nhấp chọn đăng ký.
- Hệ thống đánh giá (valid) thông tin người dùng đã nhập, rồi khởi tạo tài khoản (và thêm vào database) dựa trên thông tin người dùng đã nhập.
- Hệ thống sẽ chuyển hướng người dùng sang trang đăng nhập.
- Alternate flow:
 - Thông tin trong form của người dùng không phù hợp. => Hệ thống không khởi tạo tài khoản, đồng thời không chuyển hướng trang của người dùng.

Yêu cầu: Người dùng đăng nhập tài khoản.

- Actor: Người dùng.
- Goal: Người dùng đăng nhập vào tài khoản trên hệ thống.
- Preconditions:
 - Người dùng truy cập vào được ứng dụng.
 - Người dùng đã có tài khoản trên hệ thống.
 - Người dùng đang ở trang chủ.
- Postconditions:
 - Người dùng đăng nhập vào hệ thống và có thể truy cập các tính năng yêu cầu đăng nhập.
- Basic flow:
 - Người dùng nhấp chọn nút "Login" trên trang chủ.
 - Hệ thống chuyển đến trang đăng nhập, đồng thời hiển thị form đăng ký gồm 2 mục: Tên tài khoản và mật khẩu.
 - Người dùng điền thông tin vào form, sau đó nhấp chọn đăng nhập.
 - Hệ thống tìm kiếm thông tin tài khoản mà người dùng đã nhập, rồi đăng nhập vào tài khoản đó.
 - Hệ thống sẽ chuyển hướng người dùng sang trang chủ.
- Alternate flow:
 - Người dùng được chuyển hướng từ trang đăng ký.
 - Người dùng nhập thông tin vào form không chính xác. => Hệ thống không đăng nhập, đồng thời không chuyển hướng trang người dùng.

Yêu cầu: Người dùng tải ảnh lên.

- Actor: Người dùng.
- Goal: Người dùng tải lên một ảnh như ảnh đại diện của tài khoản người dùng.
- Preconditions:
 - Người dùng truy cập vào được ứng dụng.
 - Người dùng đã đăng nhập vào tài khoản.
 - Người dùng đang ở trang chủ.
- Postconditions:

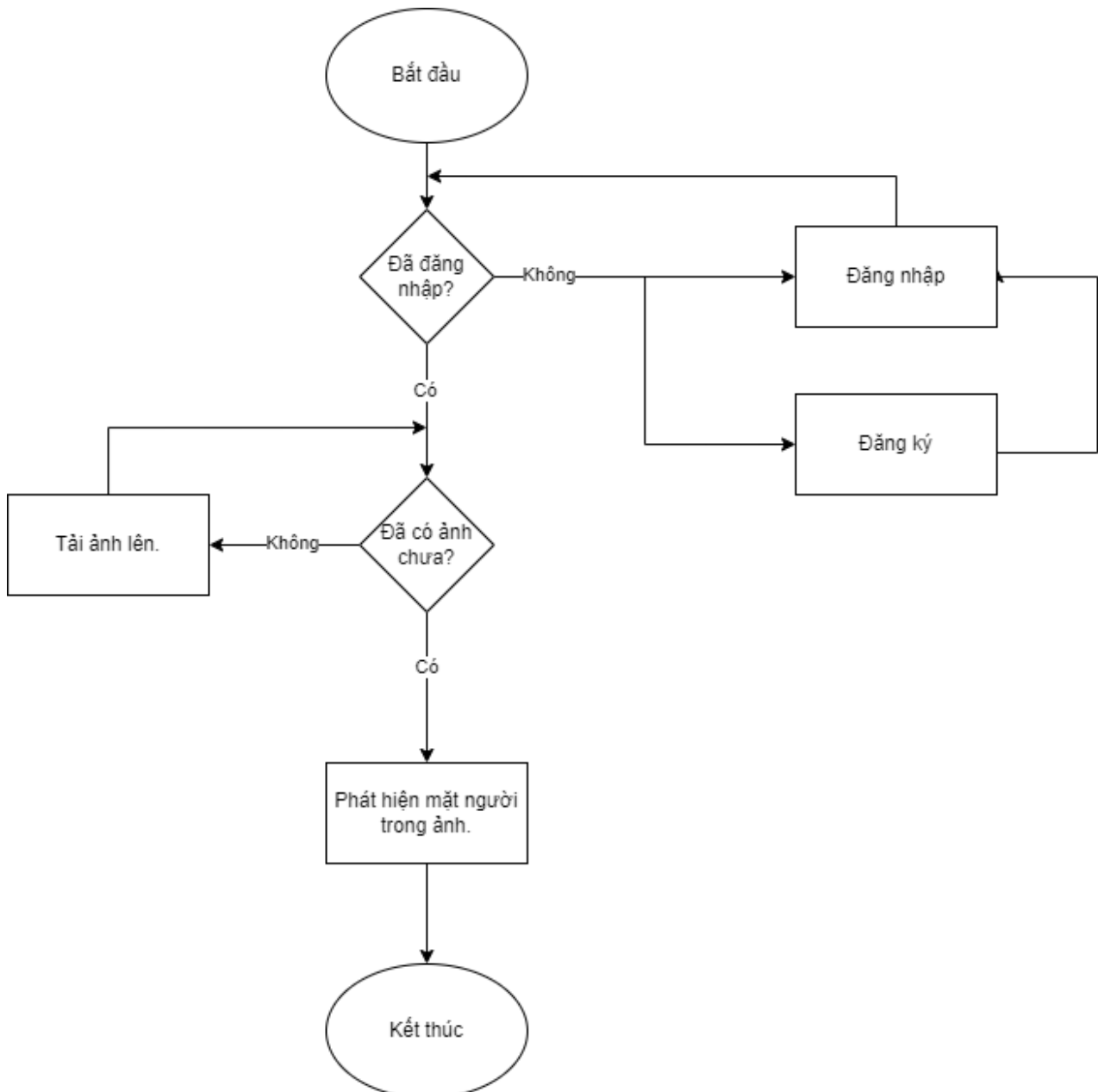
- Người dùng tải được ảnh của mình lên tài khoản, và ảnh sẽ được hiển thị ở trang chủ.
- Basic flow:
 - Người dùng nhấp chọn nút "Choose file" trên trang chủ.
 - Hệ thống mở một bảng chọn để người dùng chọn file ảnh.
 - Người dùng tiến hành chọn file ảnh rồi tải lên.
 - Hệ thống sẽ tiến hành tải ảnh lên hệ thống file, rồi tiến hành cập nhật lại người dùng. Cụ thể, thuộc tính file đại diện của người dùng sẽ là đường dẫn tới hình ảnh đó.
 - Hệ thống sẽ hiển thị hình ảnh lên trang chủ người dùng.
- Alternate flow:
 - Người dùng tải file không phải ảnh. => Hệ thống vẫn sẽ lưu file, nhưng sẽ không hiển thị được ảnh. (Exception flow)
 - Người dùng đã có ảnh. => Hệ thống vẫn sẽ lưu file mới, đồng thời ghi đè lên thuộc tính file đại diện của người dùng.

Yêu cầu: Người dùng phát hiện mặt người trên ảnh.

- Actor: Người dùng.
- Goal: Người dùng tiến hành phát hiện mặt người trên ảnh đại diện của người dùng.
- Preconditions:
 - Người dùng truy cập vào được ứng dụng.
 - Người dùng đã đăng nhập vào tài khoản.
 - Người dùng đã có ảnh đại diện.
 - Người dùng đang ở trang chủ.
- Postconditions:
 - Ảnh đại diện của người dùng được phát hiện mặt người, thể hiện bằng một bounding box tại mặt người.
- Basic flow:
 - Người dùng nhấp chọn nút "Detect face" trên trang chủ.
 - Hệ thống chuyển hướng đến trang phát hiện mặt người, nơi có một nút "Face detection".
 - Người dùng nhấp chọn nút "Face detection".
 - Hệ thống sẽ tiến hành thực hiện phát hiện mặt người trên đó. Từ vị trí được phát hiện đó, hệ thống sẽ 'đóng khung' lại rồi cập nhật lên ảnh đại diện.
 - Hệ thống chuyển hướng người dùng về trang chủ, cùng với ảnh đã được cập nhật.
- Alternate flow:
 - Người dùng chưa có ảnh đại diện. => Hệ thống không hiển thị nút "Detect face" trên trang chủ.
 - Người dùng truy cập vào trang khi chưa có ảnh. => Hệ thống trả về trang lỗi. (Exception flow)
 - Ảnh đại diện của người dùng không phải là file ảnh. => Hệ thống trả về trang lỗi. (Exception flow)

- Ảnh đại diện của người dùng không có mặt người. => Hình ảnh sẽ không có khung.
- Ảnh đại diện của người dùng có nhiều khuôn mặt. => Hình ảnh sẽ đóng khung nhiều khuôn mặt.
- Ảnh đại diện của người dùng đã phát hiện mặt người rồi. => Phát hiện trên chính hình đó. (Corner case?)

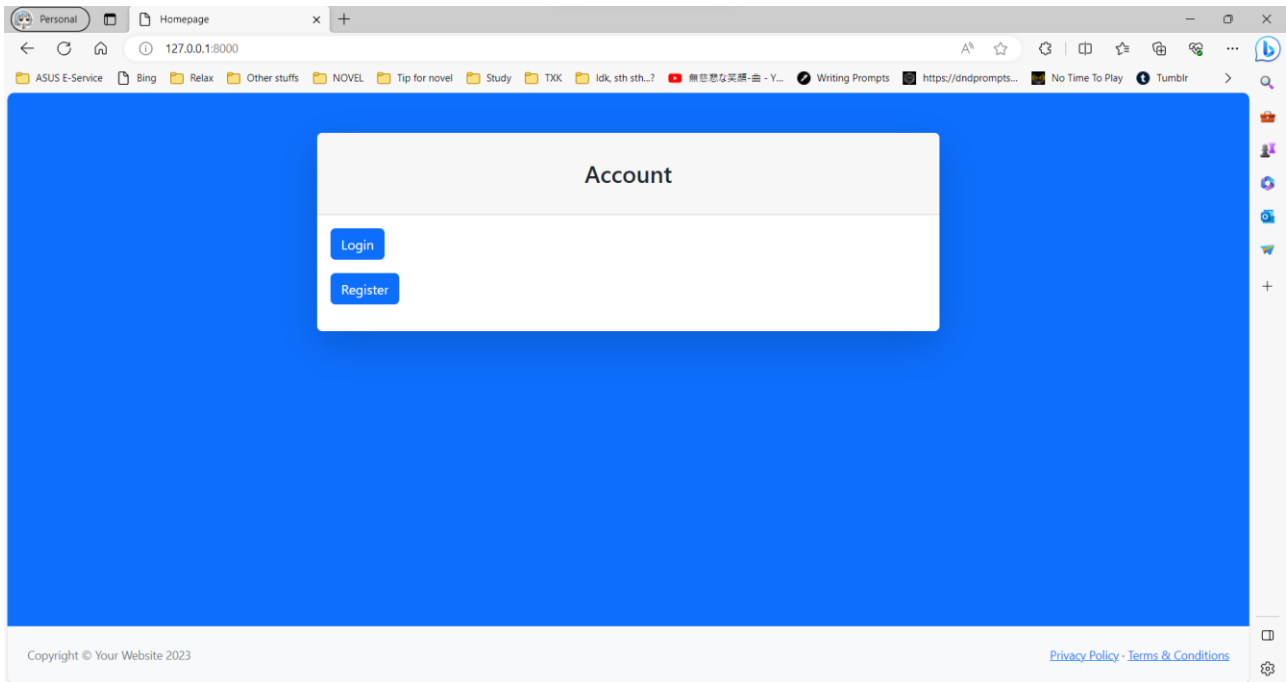
d. Flowchart



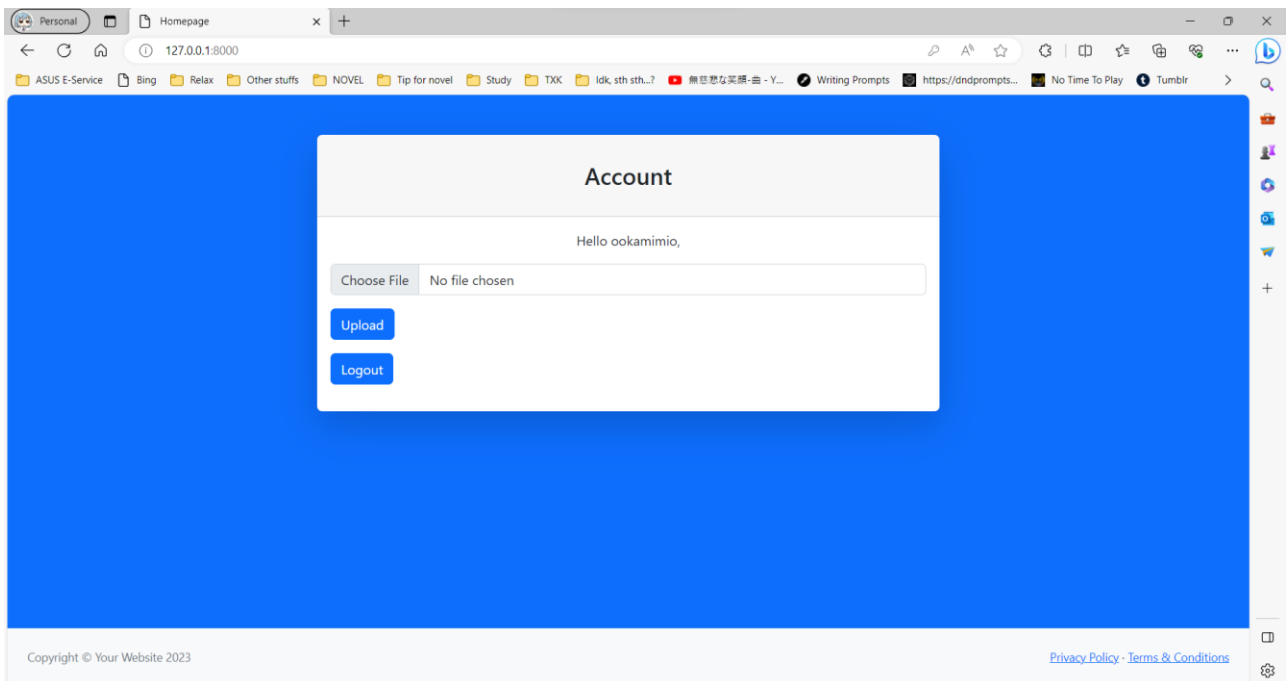
2. Kết quả

Danh sách các màn hình:

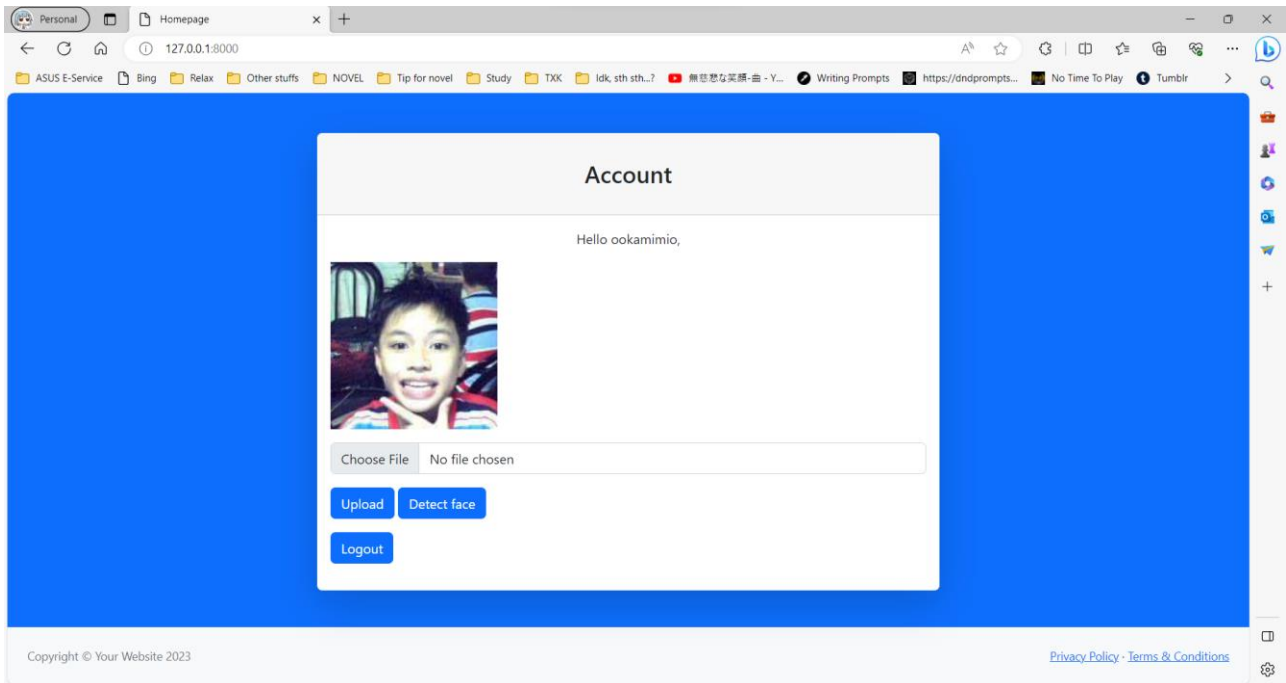
- Màn hình trang chủ:
 - o Chưa đăng nhập:



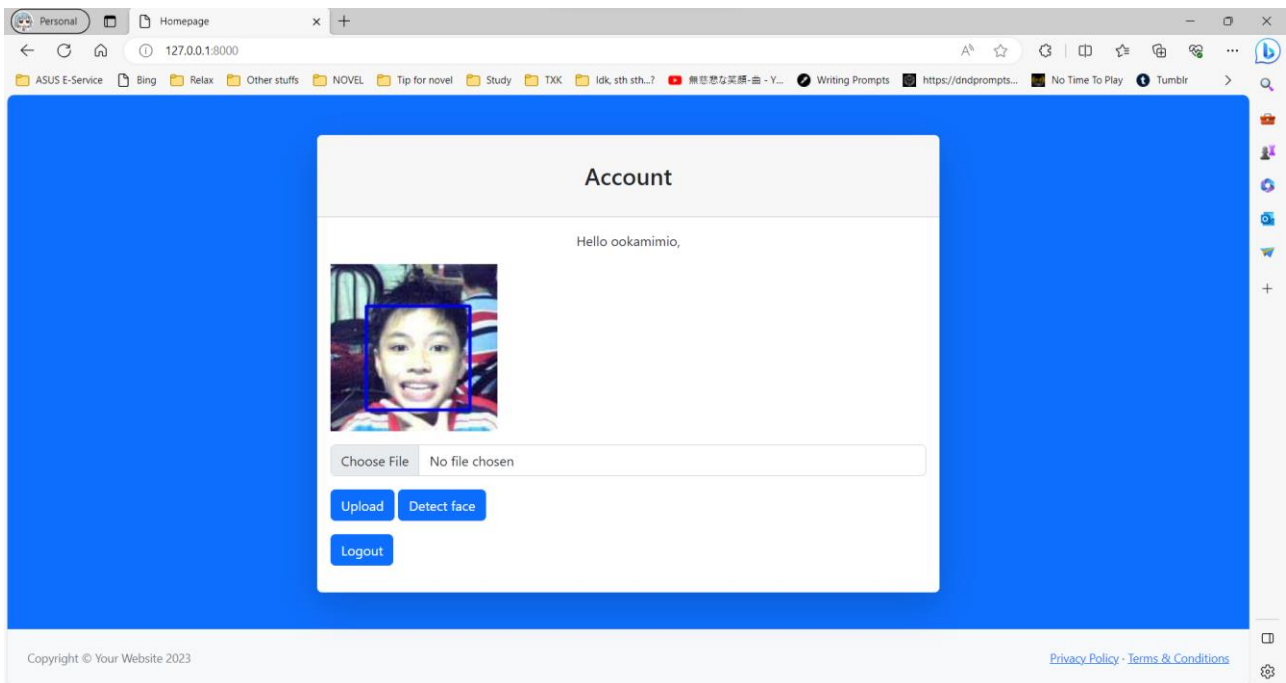
- o Đã đăng nhập/Chưa có ảnh:



- o Đã đăng nhập/Đã có ảnh/Chưa phát hiện mặt người:



- Đã đăng nhập/Đã có ảnh/Đã phát hiện mặt người:



- Màn hình đăng ký:

Personal Register - SB Admin x +

127.0.0.1:8000/register/

ASUS E-Service Bing Relax Other stuffs NOVEL Tip for novel Study TXK Idk, sth sth...? 無慈悲な英雄 - Y... Writing Prompts https://dndprompts... No Time To Play Tumblr

Create Account

Username

Password Confirm Password

Create Account

[Have an account? Go to login](#)

Copyright © Your Website 2023 [Privacy Policy](#) [Terms & Conditions](#)

- Màn hình đăng nhập:

Personal Login - SB Admin x +

127.0.0.1:8000/login/

ASUS E-Service Bing Relax Other stuffs NOVEL Tip for novel Study TXK Idk, sth sth...? 無慈悲な英雄 - Y... Writing Prompts https://dndprompts... No Time To Play Tumblr

Login

Username

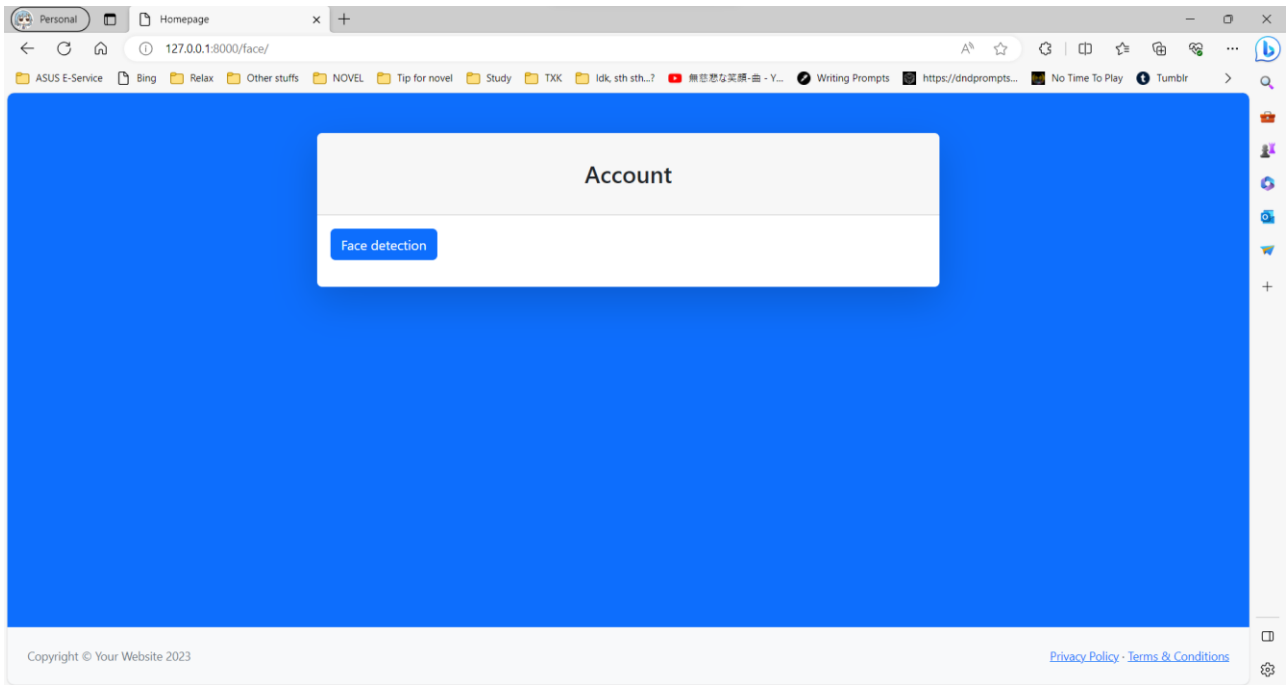
Password

Login

[Need an account? Signup!](#)

Copyright © Your Website 2023 [Privacy Policy](#) [Terms & Conditions](#)

- Màn hình phát hiện mặt người:



IV. Nhận xét

Ở bài tập tuần lần này, đi theo gợi ý của thầy là sử dụng các framework Python, em quyết định chọn Django để tự học và làm thử. Một số cảm nhận của em về Django là:

- Django là một framework rất mạnh. Đa số hầu hết các tính năng cần thiết của một web app đều đã có sẵn trong Django, và có thể chỉnh sửa tương đối dễ dàng.
- Django là một framework có cấu trúc. Mô hình MVT của Django được dựa theo mô hình MVC thường thấy, và được cài đặt sẵn phần nào trong app của Django. Vì vậy, cài đặt theo framework Django sẽ giúp người thiết kế có cấu trúc hơn.
- Django phức tạp. Do hỗ trợ rất nhiều tính năng nên bản thân Django cồng kềnh, rất dễ rối rắm.

Kết luận: Với các ứng dụng nhỏ và đơn giản, việc sử dụng các framework đơn giản sẽ có tiện lợi hơn. Nhưng với các ứng dụng lớn và phức tạp, có thể cân nhắc sử dụng Django. Ngoài ra, việc thiết kế giao diện (frontend), mặc dù được Django hỗ trợ, vẫn không tiện lợi như các framework của Node.js.

Với bài tập tuần này, em đã thực hiện được các yêu cầu căn bản, đồng thời có thể áp dụng một template UI (em sử dụng bootstrap) để trang trí thêm. Tuy nhiên, những thứ em chưa làm được là:

- Yêu cầu bổ sung: Em đã có ý tưởng làm cụ thể: Thêm một thuộc tính đại diện cho khuôn mặt người dùng, thêm lựa chọn cho trang đăng nhập... nhưng em không thể cài đặt vì vấn đề thời gian.
- Về database, đặc biệt là file system vẫn còn chưa tối ưu. File được tải lên sẽ nằm trên file system, và kể cả khi đã thay đổi ảnh đại diện/thực hiện phát hiện khuôn mặt thì file cũ vẫn không bị xóa. Em không thể cài đặt kịp thời chức năng tự xóa file sau khi cập nhật.
- Cài đặt thêm chức năng: Mục tiêu ban đầu của em là khi vào trang phát hiện khuôn mặt, sau khi thực hiện khuôn mặt, trang web sẽ hiển thị bức ảnh trước và sau khi phát hiện khuôn mặt bên cạnh nhau cho người dùng so sánh trước khi nhấn xác nhận. Tuy nhiên, vì vấn đề thời gian nên em không thể cài đặt kịp thời chức năng này.
- Thực hiện 12-factor app, mà cụ thể:
 - o Backing service: Phần data access và database của em vẫn chưa thật sự tách rời với hệ thống. Cụ thể, database của em đã bị code cứng là PostgreSQL (dù người thiết kế có thể thay đổi bằng file môi trường), và file system của em cũng nằm trong thư mục project của ứng dụng.
 - o Build, release, run.
 - o Logs: Bởi vì việc thiết kế các phần trước quá lâu nên em không có đủ thời gian để cài đặt việc ghi nhật ký vào.
- ...
- Mã nguồn của em vẫn chưa 'sạch' và gọn gàng.

Nhìn chung, nếu tự đánh giá, mặc dù có sự cải thiện so với các bài tập trước, bài tập này của em vẫn chưa đạt yêu cầu.

V. Tham khảo:

There are no sources in the current document.