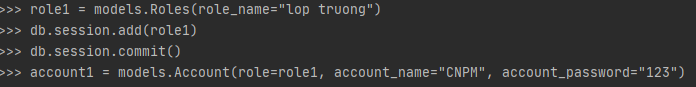
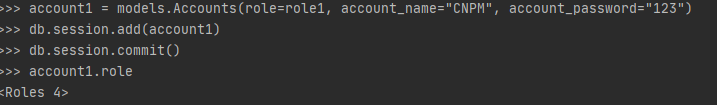
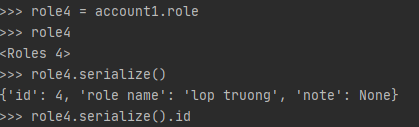
**0/ Kiến thức cơ bản:**

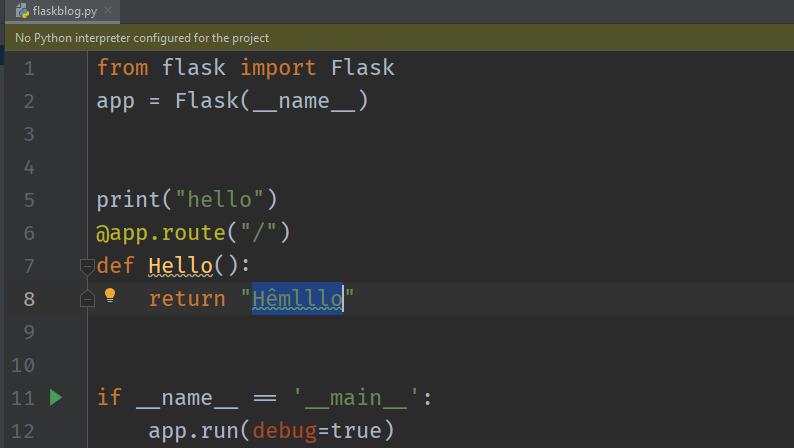
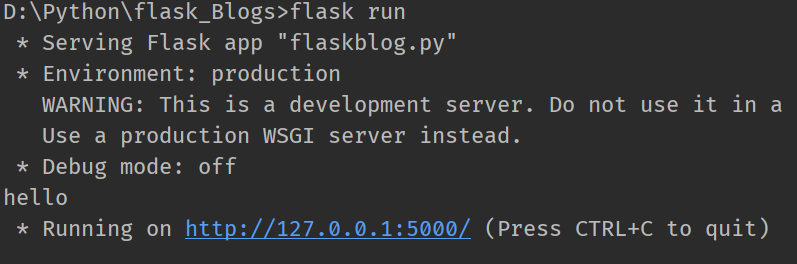
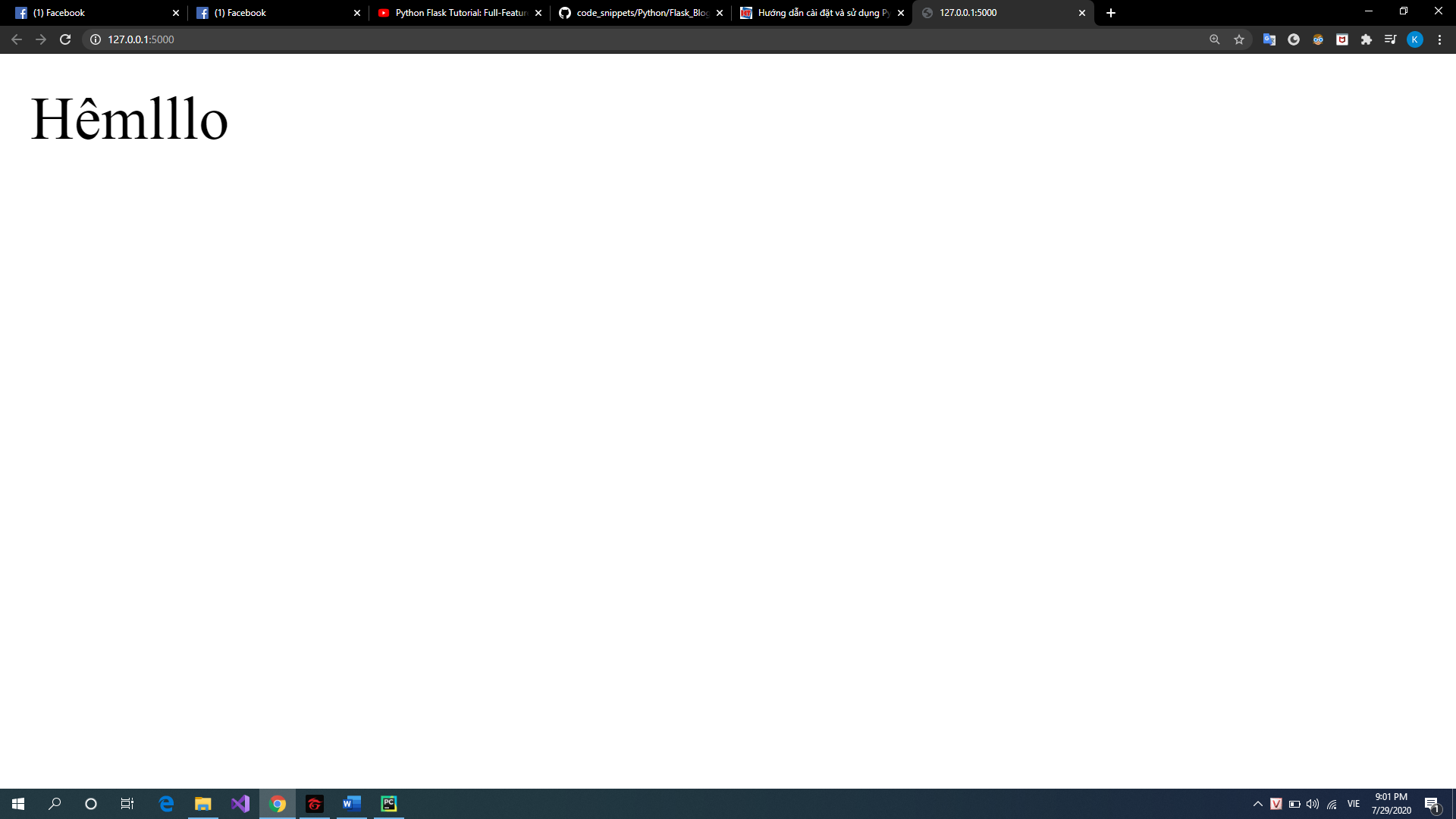
* **JSON data request, response:** 
  + ****
  + **1-Many:**

****

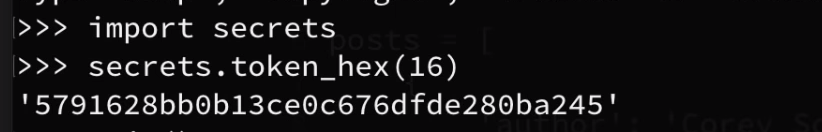
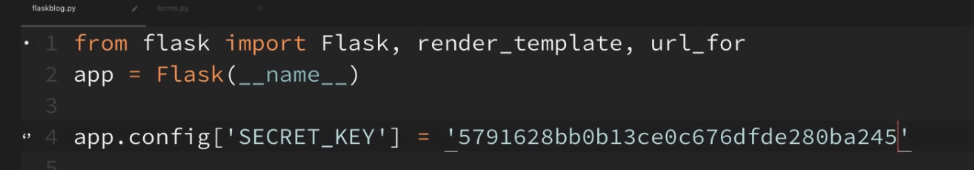
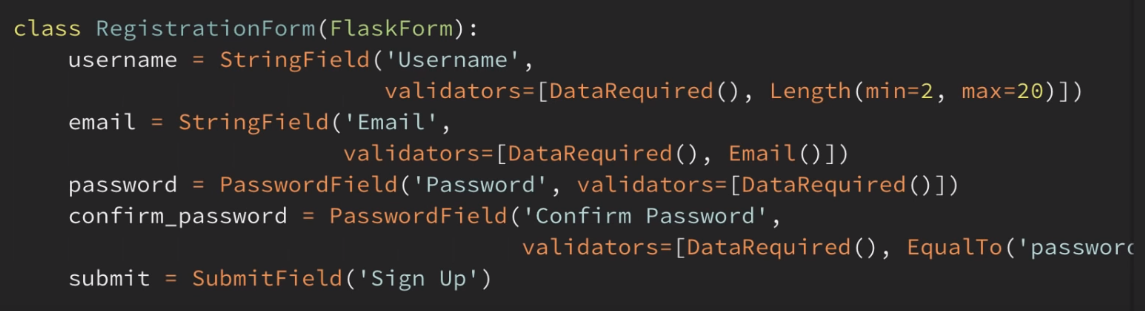
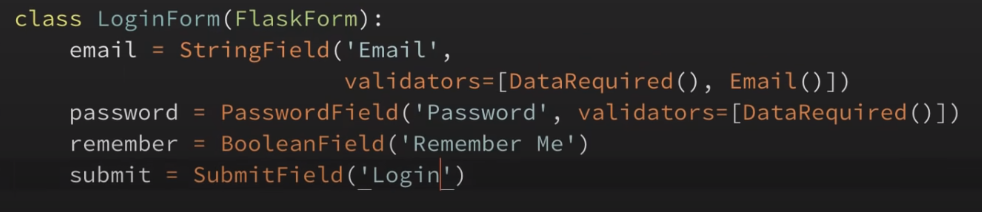
****

****

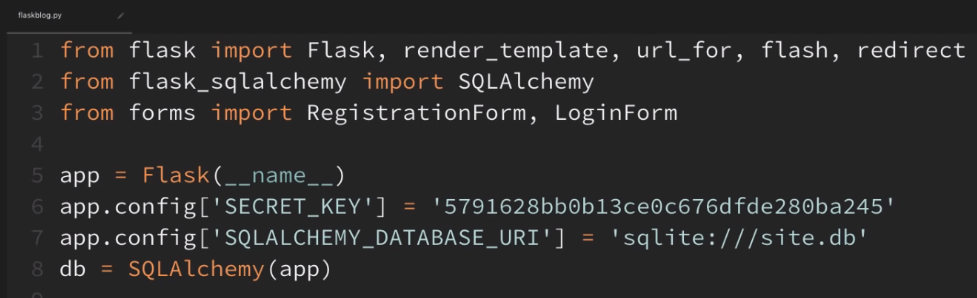
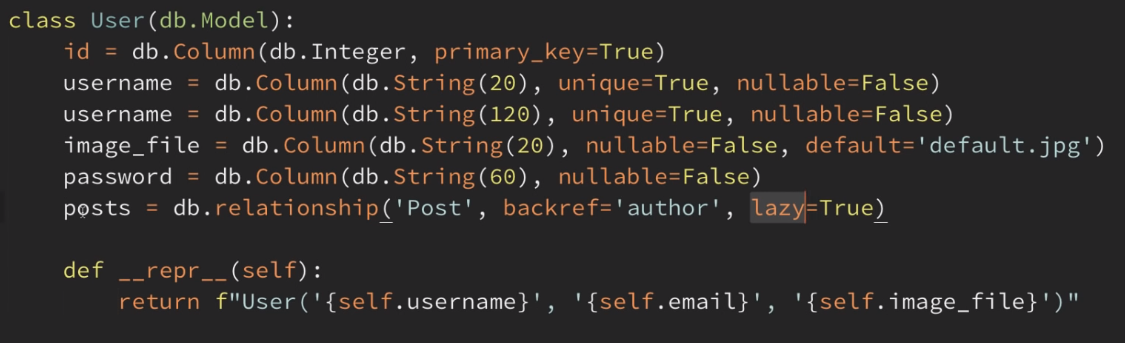
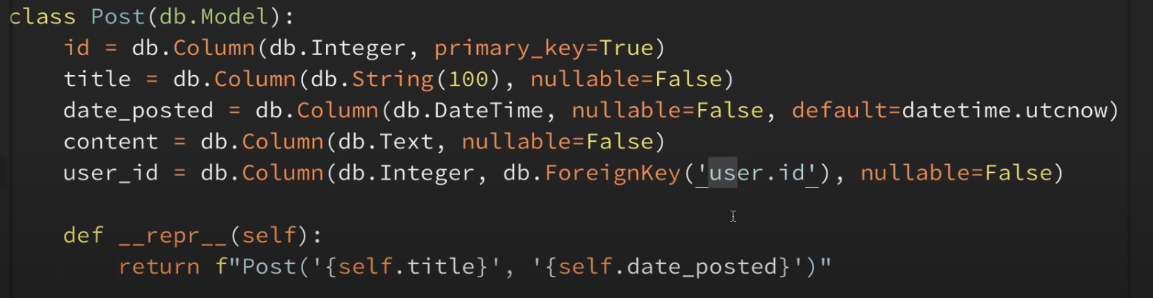
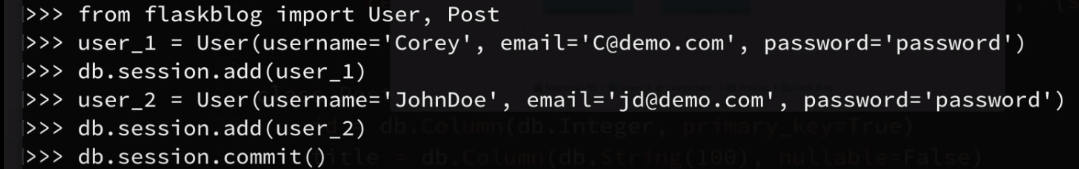
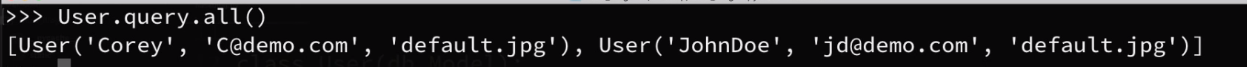
**1/Cài đặt, chạy thử:**

* Cài đặt: import flask
* Cách chạy cơ bản :
  + 
  + 
  + 
  + 
  + 

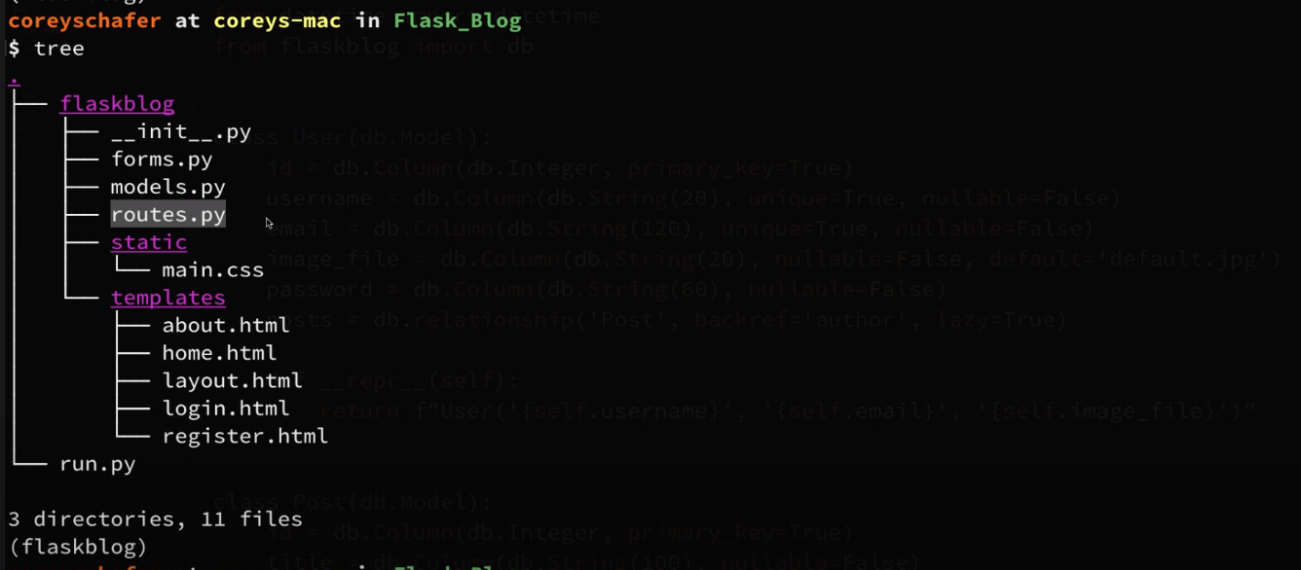
**2/LÀM VIỆC VỚI FORM, INPUT:**

* Cách lấy **SECRECT KEY** trong python: ****
* Set app config: ****
* Khởi tạo 2 class RegistrationForm and LoginForm: trong module FORMS
  + ****
  + ****
* Tạo 2 route với 2 chức năng “Register” và “Login”:

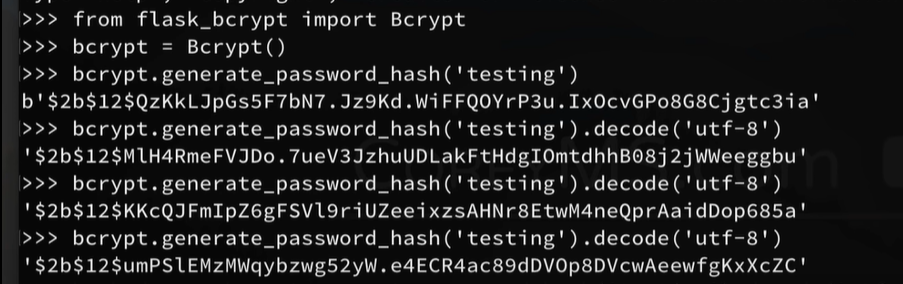
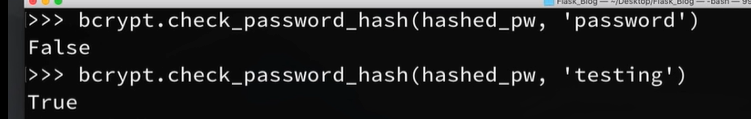
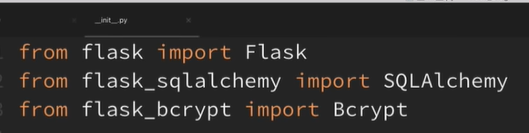
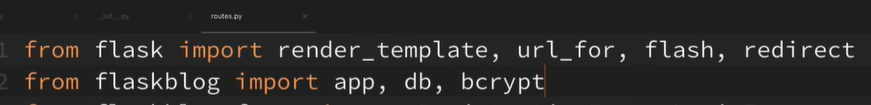
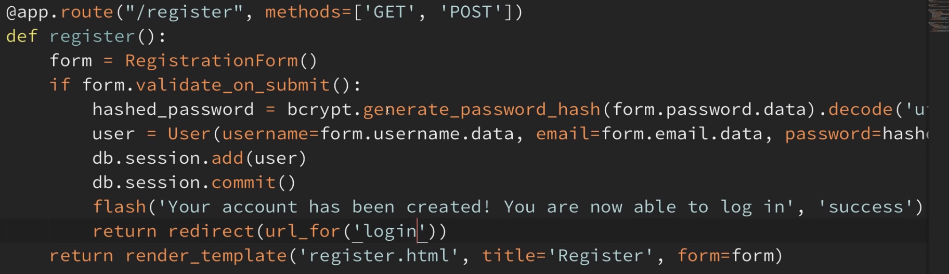
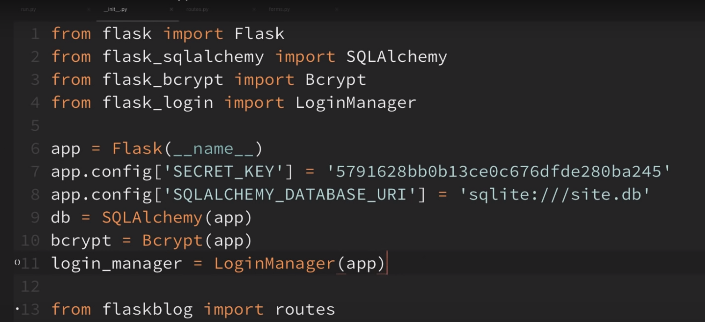
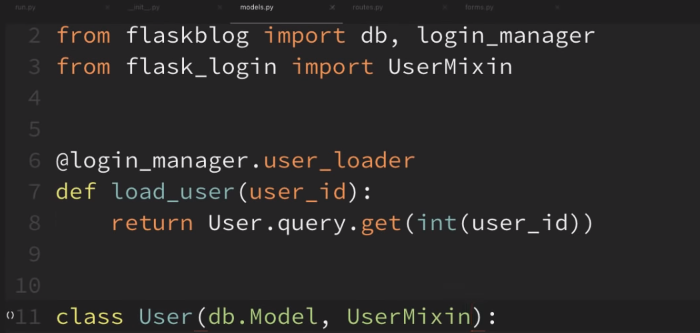
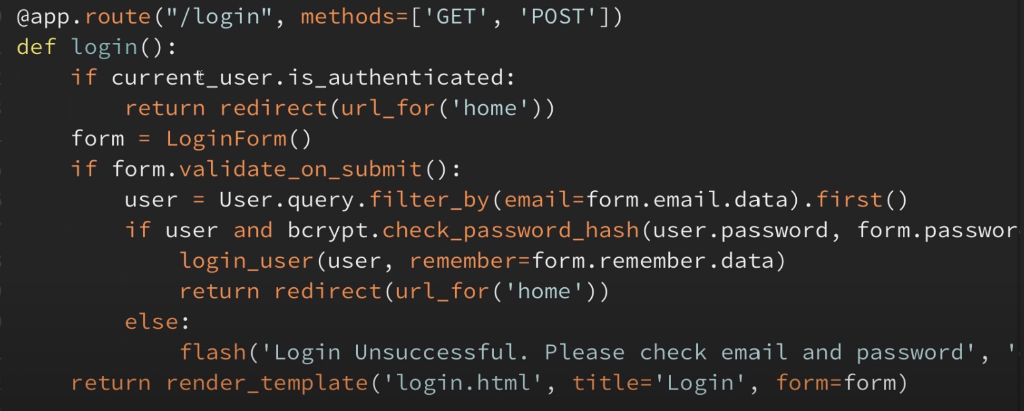
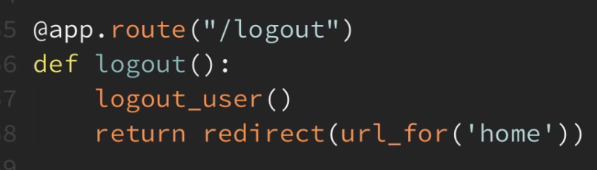
**3/ Database with Flask-SQLAlchemy:** <https://flask-sqlalchemy.palletsprojects.com/en/2.x/>

* Cài đặt: 
* Import:  => Sẽ làm việc với biến “db” này (.create\_all()).
* Khai báo 1 table trong SQLALCHEMY:
  + Ví dụ table USER:  => Relationship: 1 USER thì có NHIỀU bài POSTs (1:n)
  + Ví dụ table POST: => ForeignKey là khóa ngoại trỏ tới bảng user.id
* Tạo database từ các models mình vừa tạo:
  + Vào cửa sổ PYTHON hoặc cửa sổ TERMINAL của project:
    - Import biến db từ trong project “flaskblog” của mình vào cửa sổ Python: 
    - Tạo file .db database từ các models trên: 
    - Thêm (insert data) từ python hoặc terminal của project:  => commit là save xuống database
    - Lấy (read) tất cả dữ liệu:

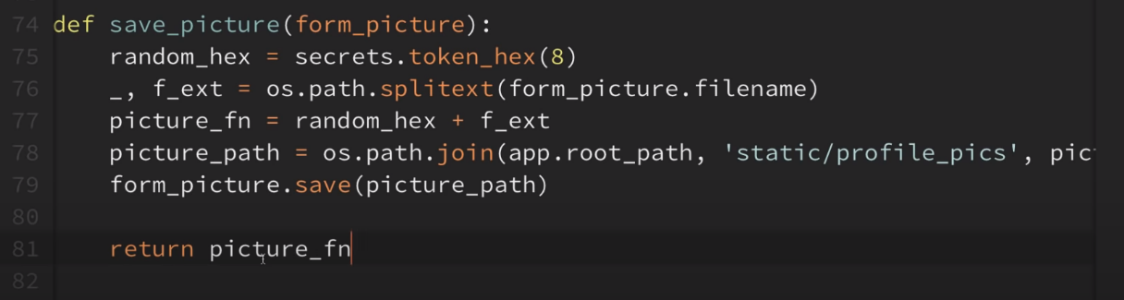
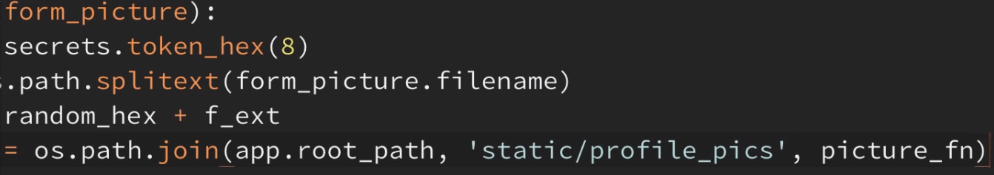
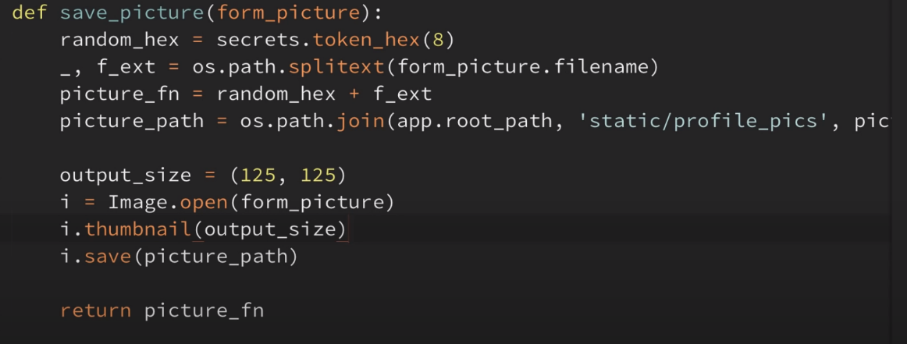
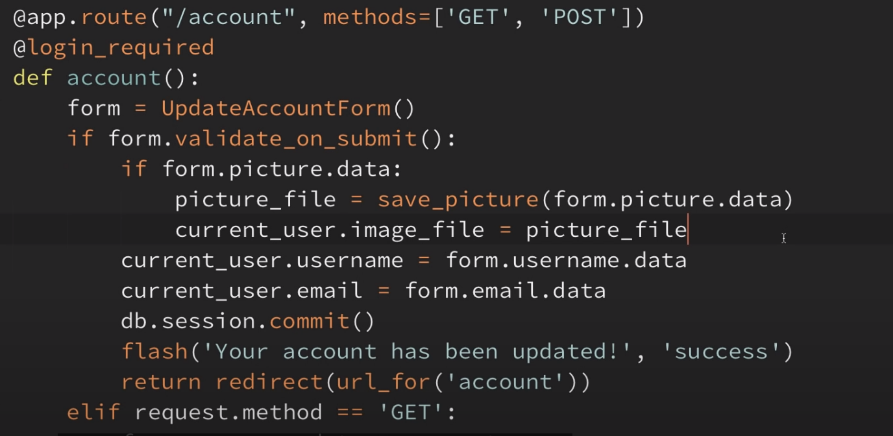
**4/Package Structure:**

* 

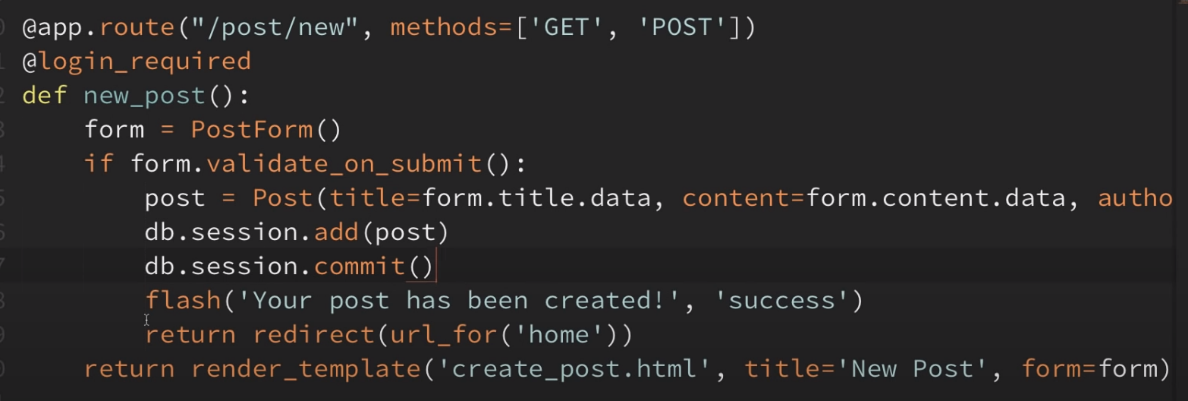
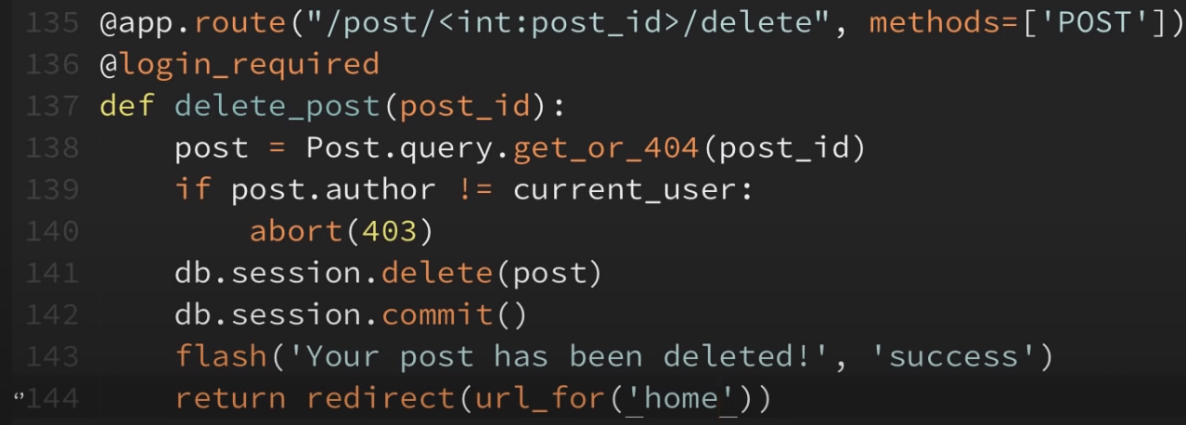
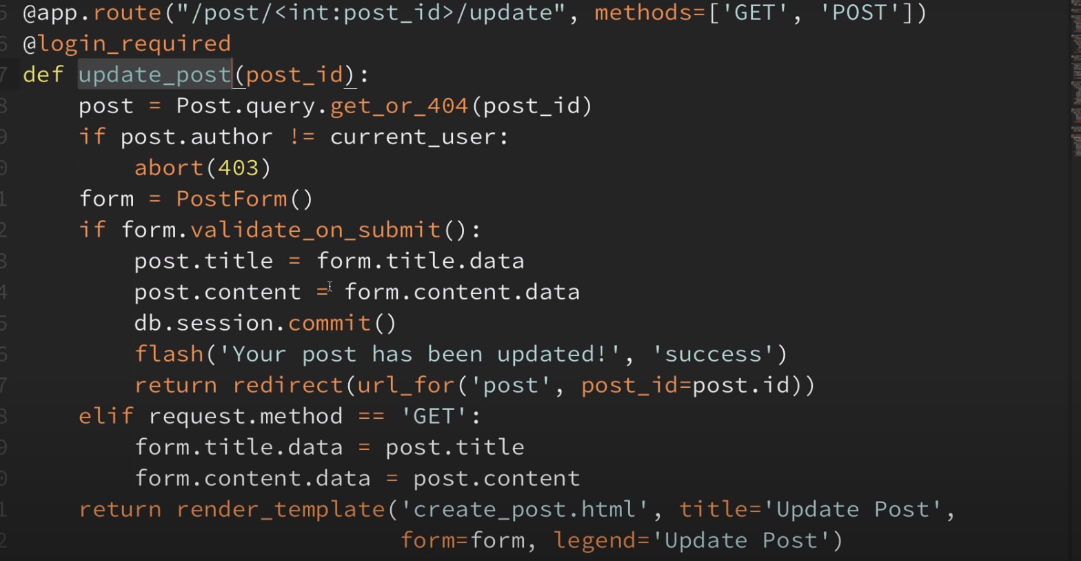
**6/User Authentication**:

* Import và sử dụng Bcrypt để mã hóa:
  + 
* Kiểm tra hashed password với input password từ user:
  + 
  + 
* Áp dụng project:
  +  => import
  +  => những package khác muốn sử dụng biến bcrypt này thì sẽ import nó vào từ module này
  +  => import bcrypt mà ta đã tạo ở trên.
  +  => hash password khi ta đăng ký.
  +  => hàm kiểm tra xem username đã tồn tại trong database chưa!
* Chức năng Login:
  + Cài đặt:
    - 
    -  => line 4, 11: import flask-login, tạo instance ở file init project.
    - 
    -  => import current\_user
    -  => User router. “Current\_user” là biến lưu giữ thông tin khi mình dùng hàm “login\_user(user, remember=…) [”. Khi dùng ‘@Login\_required()](mailto:)’ sẽ dựa vào biến Current\_User để kiểm tra.
* Chức năng Logout():
  +  ‘

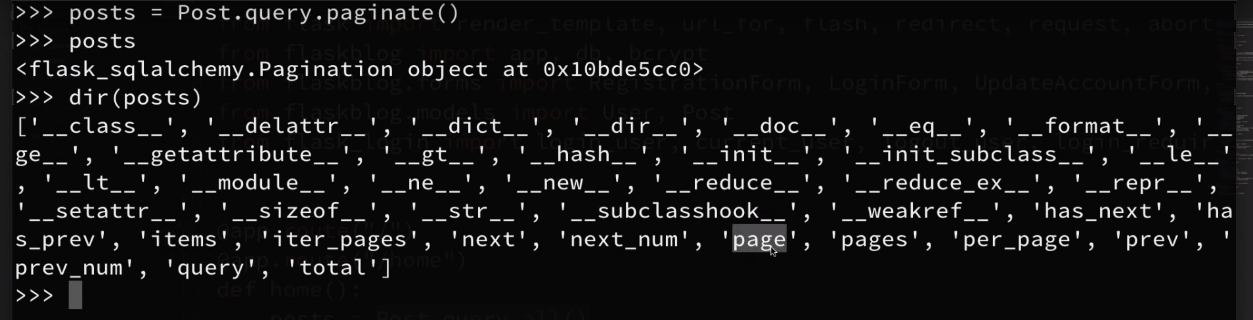
**7/ User Account and Picture Profile:**

* Chức năng SAVE PICTURE xuống database sau khi ảnh gửi lên từ FE:
  + 
  + 
    - Import secrets.
    - “Form\_picture” là ảnh gửi lên từ FE.
    - “Picture\_fn” là FileName.
    - Line 79: Save bức hình từ FE gửi xuống địa chỉ “Picture\_path”.
  + Mình có thể Resize lại ảnh cho tiết kiệm bộ nhớ, tăng Performance: 
* Chức năng Cập nhật lại ảnh:
  + 

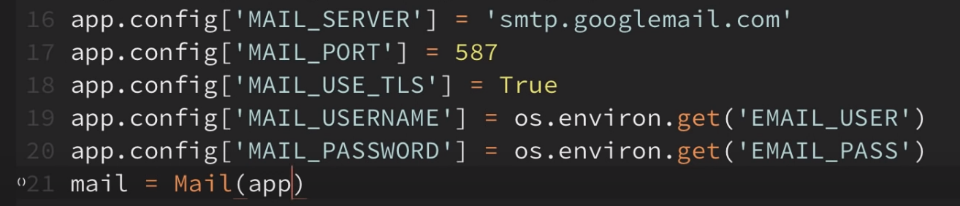
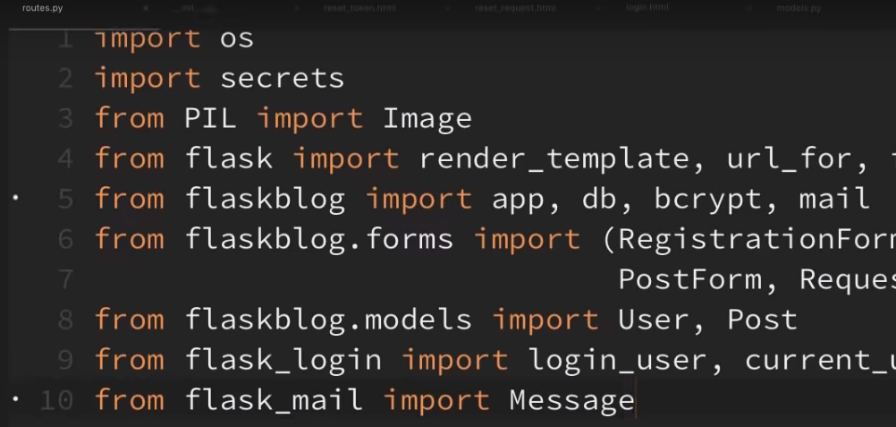
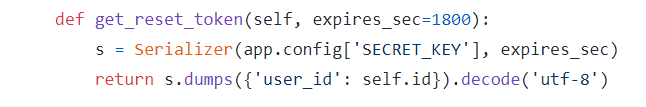
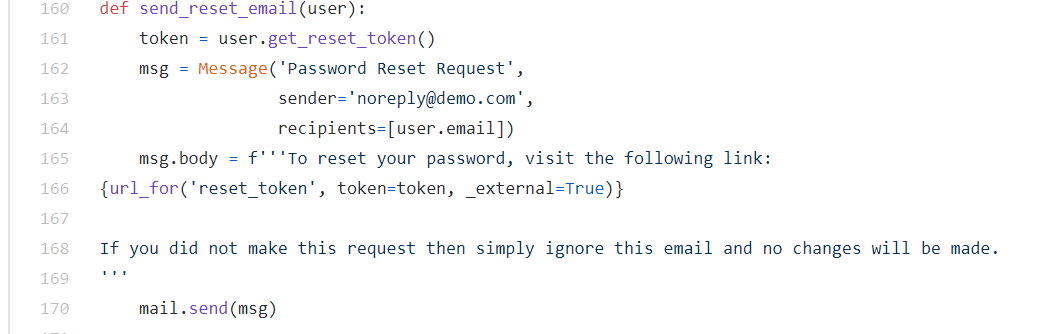
**8/ Create, Update, and Delete Posts:**

* **CREATE: **
* **DELETE: **
* **UPDATE: **

**9/ Pagination: Phân trang**

* **** 
  + Có các thuộc tính như “has\_next”, “perpage”, “page”…

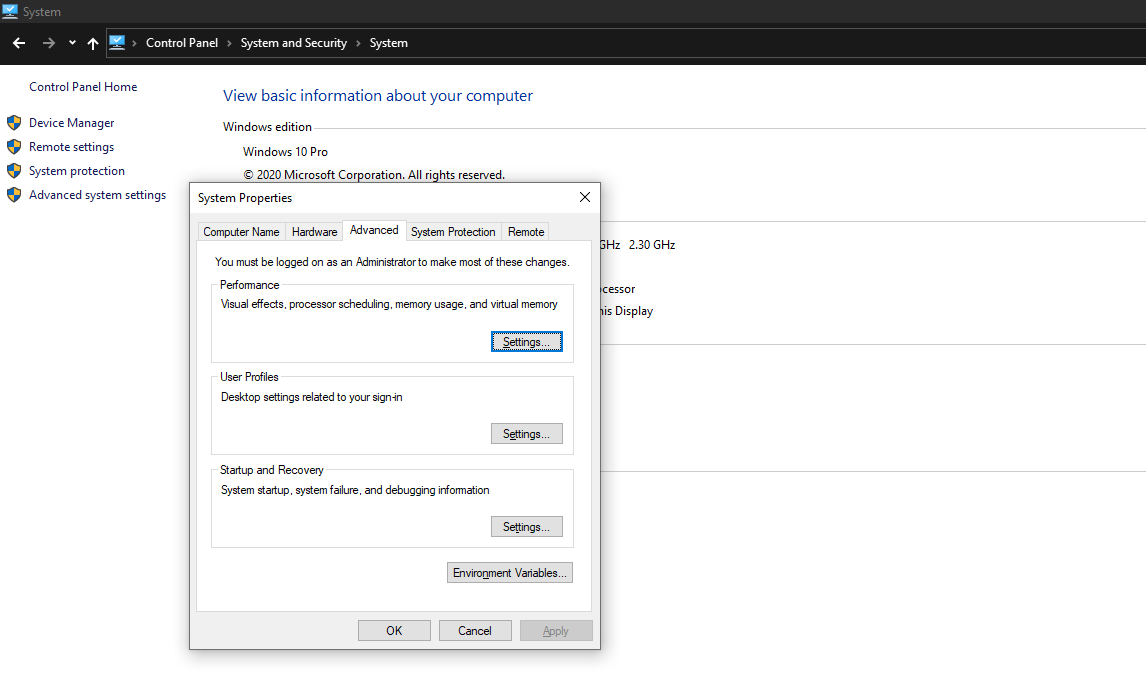
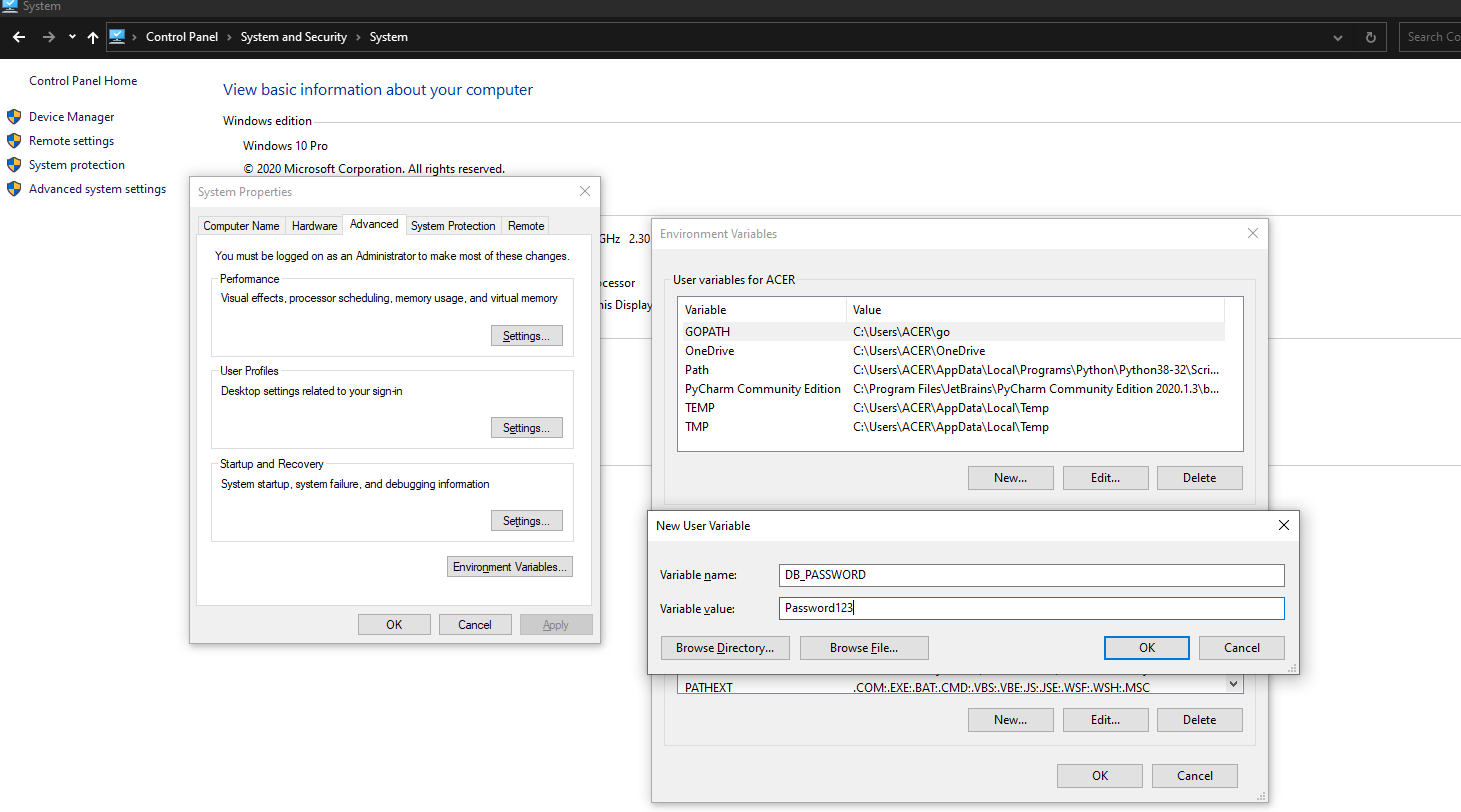
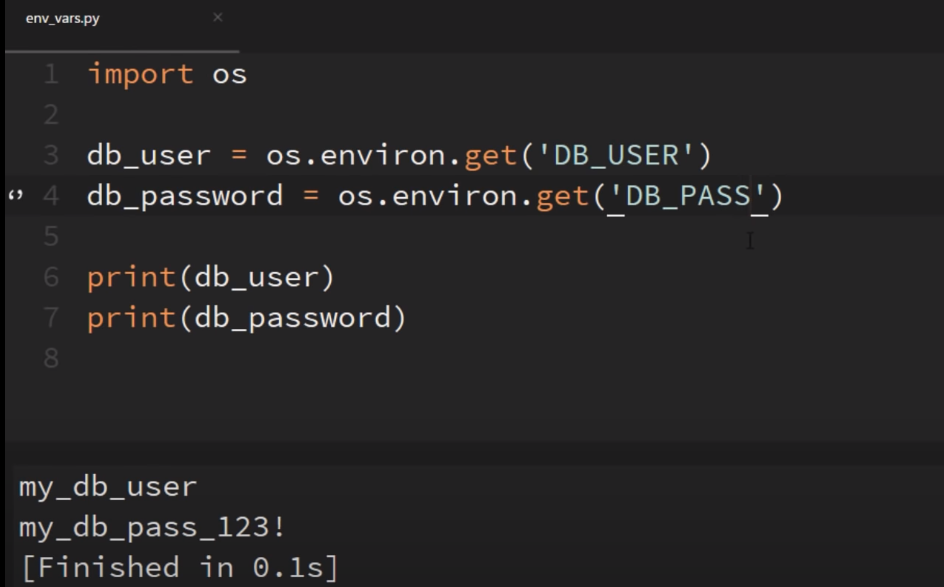
**10/ Chức năng Gửi Email and Reset Password**:

* Cài đặt:
  + \_\_init.py\_\_:
    - 
    -  -> line 19, 20 là biến environment trên máy mình.
  + Routes.py and Models.py: file dùng chức năng gửi mail:
    - Import:  => Line 5, 10.
    - FLOW:
      * Click vào nút “**Quên mật khẩu** ?”
      * => Chuyển qua view “**Reset Password**” gồm có 1 **RequestResetForm** (có **SĐT** hoặc **Email**)
      * => Điền “SĐT” hoặc “Email -> SUBMIT
      * => Gửi email reset password (sử dụng hàm **‘send\_reset\_email‘** (có reset token và đường link )trong User Model để gửi mail)
      * => Truy cập email.
      * => Truy cập vào đường dẫn đến trang “**Đặt lại mật khẩu mới**” có chứa “token”(được tạo ra ở hàm Send\_reset\_email(), có payload chứa user\_id) trong đường dẫn URL.
      * => Điền vào “MẬT KHẨU MỚI” => click nút “Xác nhận“.
      * => Gửi API “/reset-password/<token>” => Nhận được token.
      * => Dùng hàm “verify-reset-token(token)” để verify token => lấy được phần payload => lấy được UserId => Query lấy User ấy => Update lại mật khẩu(nhớ Hash Password).
    - Cách làm: Tự sửa lại theo ý mình, đây chỉ là hướng dẫn kỹ thuật thui
      * Viết hàm “getResetToken()”, **trả về** token tại Model User bằng Serializer:
        + Import: 
      * Viết 1 hàm Static “Verify\_Reset\_Token(token)”, **nhận vào “**token**”, trả về** 1 user hoặc UserId hoặc bất kỳ gì khác:
        + 
      * Viết hàm “Send-reset-mail(user)” để gửi mail cho User :
        +  => line 166: là đường link dẫn đến trang “Đặt lại mật khẩu mới”, có URL chứa token. Sau khi gửi mật khẩu mới lên, verify token lấy từ URL, nếu UserId hợp lệ thì update password, không thì báo lỗi.
        + Import :

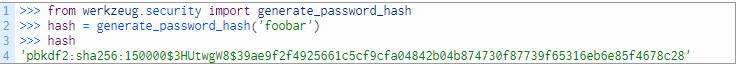




**10/ Biến Environtment:**

* 
* 
* 

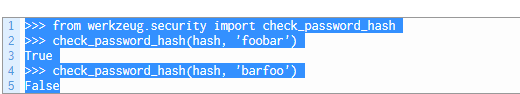
**Mã băm password:**



tạo hash cho mật mã bằng thư viện Werkzeug.

**Xác thực mật khẩu:**

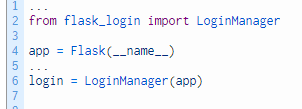
Để xác thực mật mã (verification), chúng ta sử dụng hàm thứ hai từ thư viện Werkzeug:



**Cài đặt Flask-login:**



Flask-Login phải được khởi tạo ngay sau thực thể ứng dụng trong file app/\_\_init\_\_.py như sau:

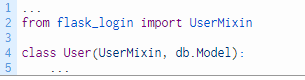


* + is\_authenticated: một thuộc tính sẽ được gán là True nếu user có tên và mật mã hợp lệ, False nếu một trong hai không đúng.
  + is\_active: một thuộc tính được gán là True nếu tài khoản user trong chế độ hoạt động (active) và False nếu ngược lại.
  + is\_anonymous: một thuộc tính được gán là False cho những user bình thường, và True cho những user ẩn danh (anonymous)
  + get\_id(): một phương thức để trả về định danh người dùng (id) dưới dạng chuỗi

**Hàm UserMixin:**

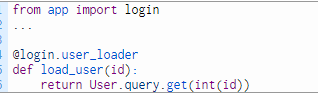
Lớp này có sẵn các khai báo và mã thực thi có thể sử dụng được với phần lớn mô hình dữ liệu User. Sau đây là cách thêm lớp mixin vào mô hình dữ liệu:

**app/models.py:** lớp mixin cho user trong Flask-Login



**Hàm tải thông tin User:**

Flask-Login không trực tiếp làm việc với cơ sở dữ liệu. Vì vậy, thư viện này sẽ cần có một hàm hỗ trợ để tải thông tin user. Hàm này sẽ tìm kiếm và tải các thông tin về user từ cơ sở dữ liệu dựa trên Id của user đó. Thêm mã cho hàm này vào module app/models.py như sau:



đăng ký hàm này với Flask-Login qua decorator @login.user\_loader. Tham số id sẽ được truyền vào hàm này dưới dạng chuỗi (String), vì vậy nếu cơ sở dữ liệu định dạng Id kiểu số nguyên (integer), cần chuyển đổi Id thành số nguyên