

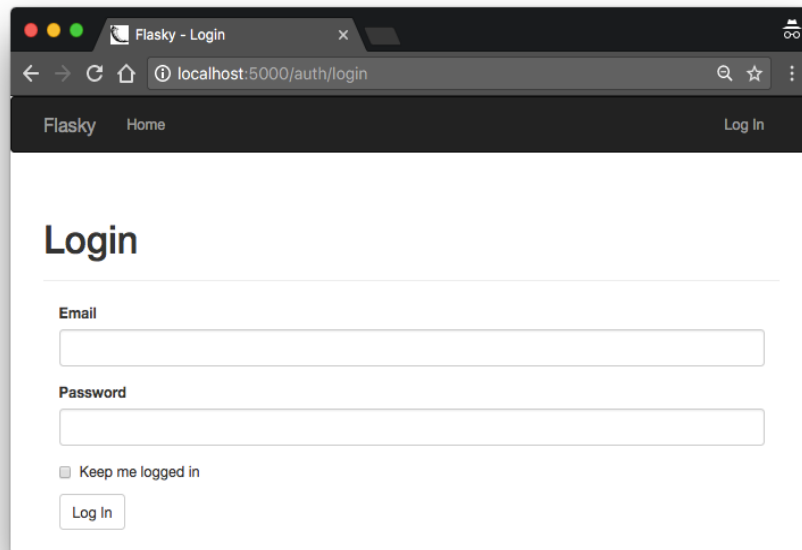
LAB 8:

Example: A Social Blogging Application

Nội dung

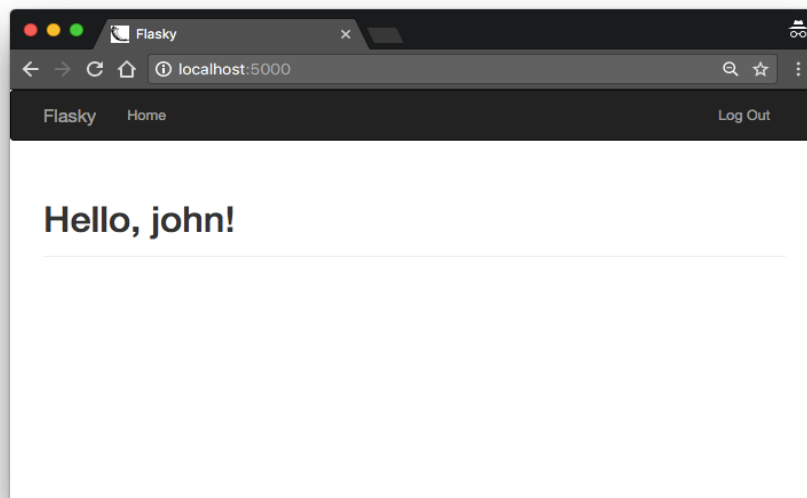
- User Authentication
- Authentication Extensions for Flask
- Hashing Passwords with Werkzeug
- Creating an Authentication Blueprint
- User Authentication with Flask-Login
- Preparing the User Model for Logins
- Protecting Routes
- Adding a Login Form
- Signing Users In
- Signing Users Out
- Testing Logins
- New User Registration
- Adding a User Registration Form
- Registering New Users
- Generating Confirmation Tokens with itsdangerous
- Sending Confirmation Emails
- Database Representation of Roles
- Role Assignment
- Followers on the Profile Page
- Showing Followed Posts on the Home Page
- Database Representation of Comments
- Comment Submission and Display
- Comment Moderation
- Creating an API Blueprint
- Error Handling
- Role Verification
- Profile Information
- User Profile Page
- User-Level Profile Editor
- Administrator-Level Profile Editor
- User Avatars
- Blog Post Submission and Display
- Blog Post on Profile Pages
- Creating Fake Blog Post Data
- Rendering in Pages
- Adding a Pagination Widget
- Rich-Text Posts with Markdown and Flask-PageDown
- Using Flask-PageDown
- Handling Rich Text on the Server
- Permanent Links to Blog Posts
- Blog Post Editor
- Many-to-Many Relationships
- Advanced Many-to-Many Relationships
- User Authentication with Flask-HTTPAuth
- Token-Based Authentication
- Serializing Resources to and from JSON
- Implementing Resource Endpoints
- Pagination of Large Resource Collections

1. Viết chương trình python để tạo 1 form/cửa sổ HTML với giao diện Log in như sau:



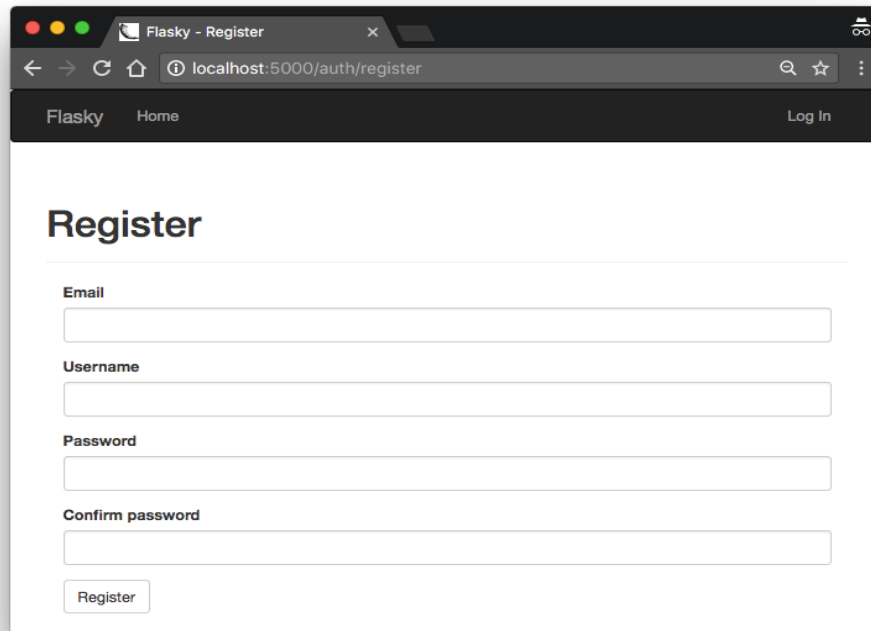
The screenshot shows a web browser window titled 'Flasky - Login'. The address bar displays 'localhost:5000/auth/login'. The page features a dark navigation bar with 'Flasky' and 'Home' links on the left, and a 'Log In' button on the right. The main content area is white and contains a 'Login' heading. Below the heading are two input fields labeled 'Email' and 'Password'. Under the 'Password' field is a checkbox labeled 'Keep me logged in' and a 'Log In' button.

2. Viết chương trình python giúp người dùng có thể Log in và sau khi Log in thành công sẽ trở thành giao diện như hình dưới đây:



The screenshot shows a web browser window titled 'Flasky'. The address bar displays 'localhost:5000'. The page features a dark navigation bar with 'Flasky' and 'Home' links on the left, and a 'Log Out' button on the right. The main content area is white and displays the text 'Hello, john!'.

3. Viết chương trình python một form đăng ký để giúp người dùng đăng ký/tạo một tài khoản mới:



Flasky - Register

localhost:5000/auth/register

Flasky Home Log In

Register

Email

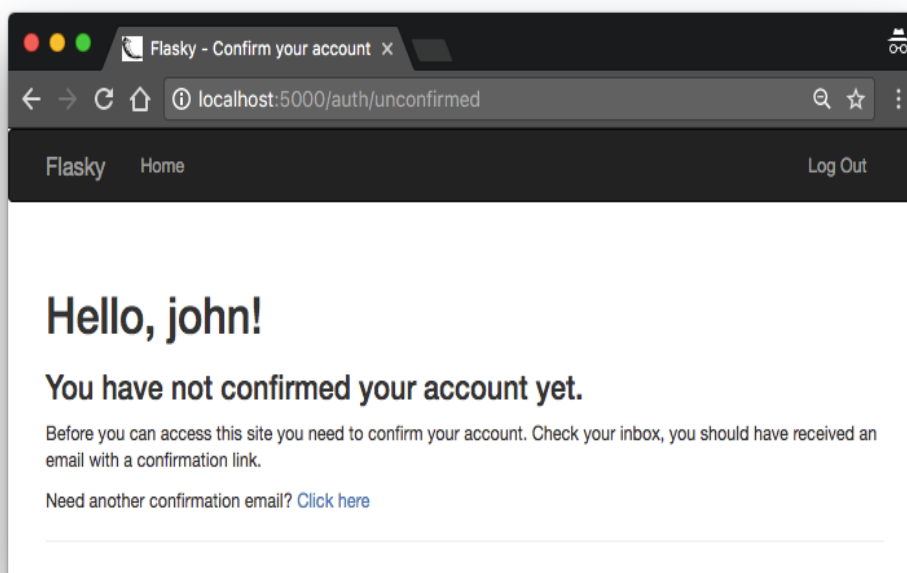
Username

Password

Confirm password

Register

4. Viết chương trình python để sau khi tạo tài khoản xong sẽ gửi email về cho người dùng là đã được chấp thuận hay chưa như hình dưới:



Flasky - Confirm your account

localhost:5000/auth/unconfirmed

Flasky Home Log Out

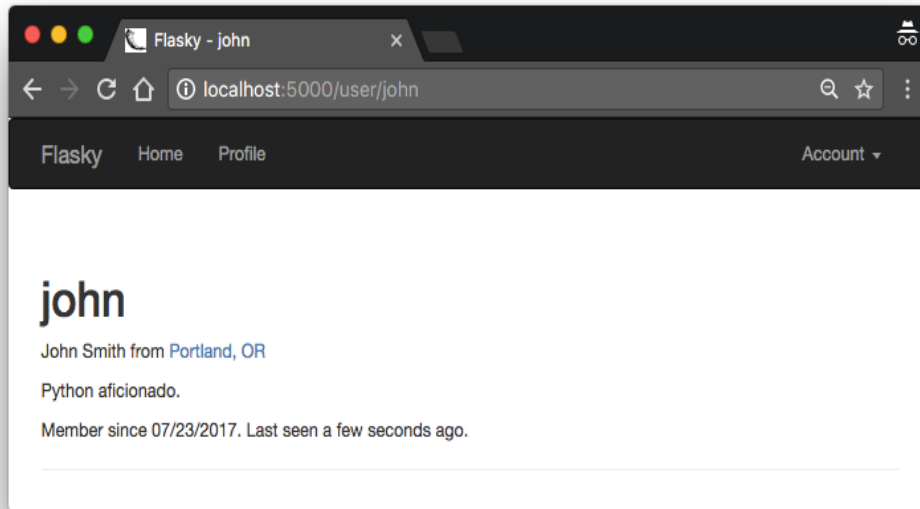
Hello, john!

You have not confirmed your account yet.

Before you can access this site you need to confirm your account. Check your inbox, you should have received an email with a confirmation link.

Need another confirmation email? [Click here](#)

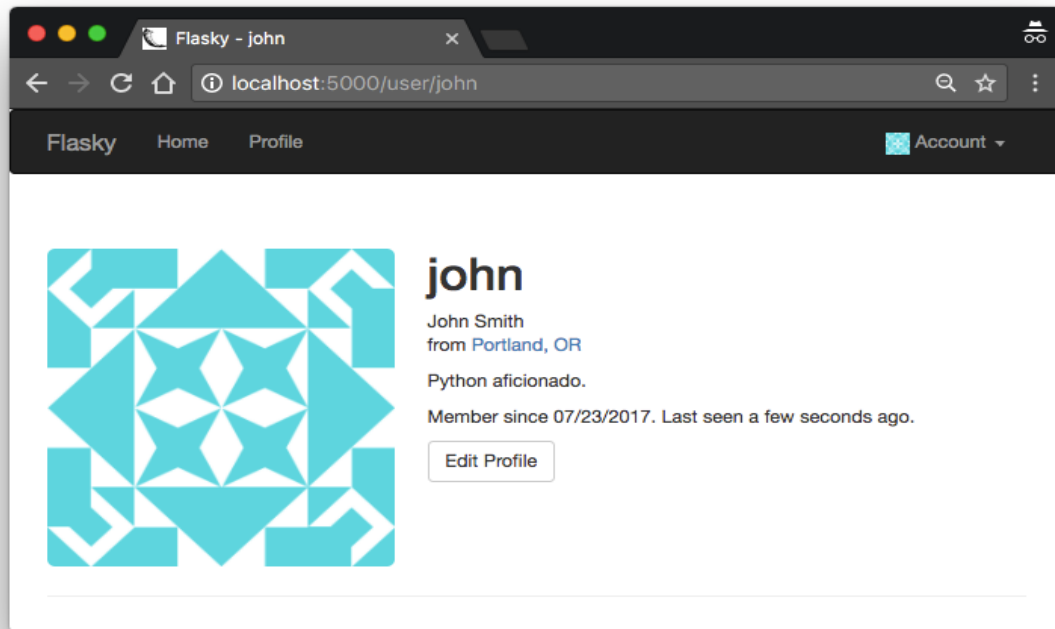
5. Viết chương trình python để tạo trang gồm thông tin cá nhân của người dùng đó:



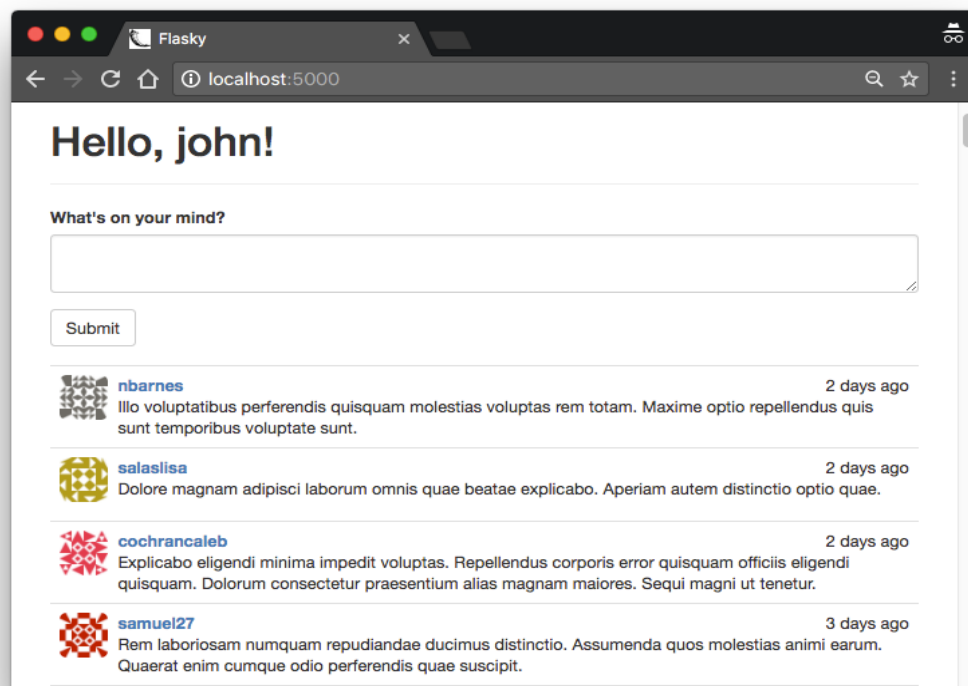
6. Viết chương trình python để tạo form giúp người dùng có thể chỉnh sửa trang thông tin cá nhân của họ:

A screenshot of a web browser window titled 'Flasky - Edit Profile'. The address bar shows 'localhost:5000/edit-profile'. The page has a dark navigation bar with 'Flasky', 'Home', 'Profile', and 'Account' (with a dropdown arrow). The main content area is titled 'Edit Your Profile'. It contains three form fields: 'Real name' with the value 'John Smith', 'Location' with the value 'Portland, OR', and 'About me' with the value 'Python aficionado.'. Below these fields is a 'Submit' button.

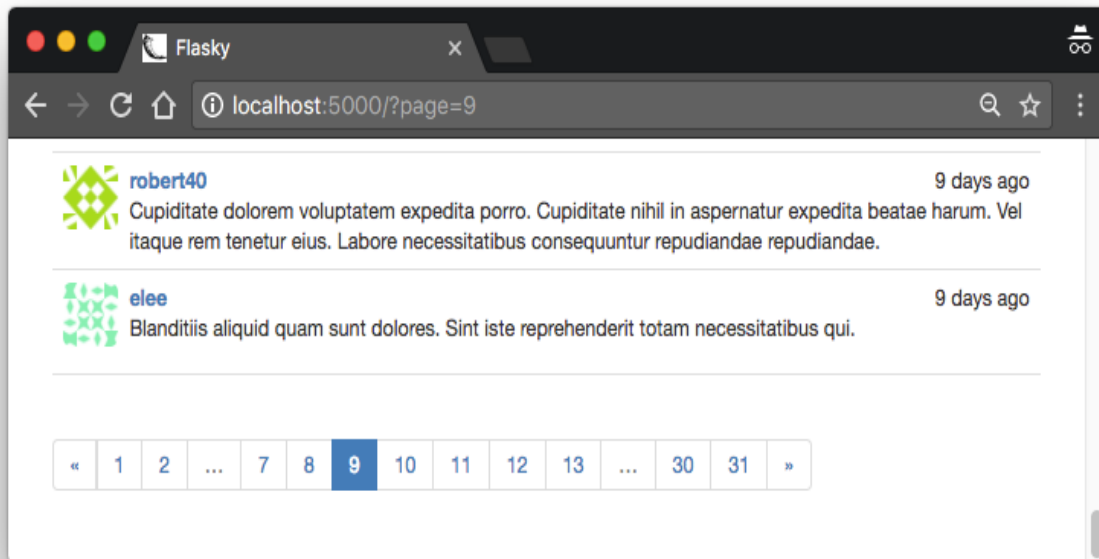
7. Viết chương trình python để tạo một trang hiển thị thông tin cá nhân của người dùng có Avatar của người dùng đó như hình sau:



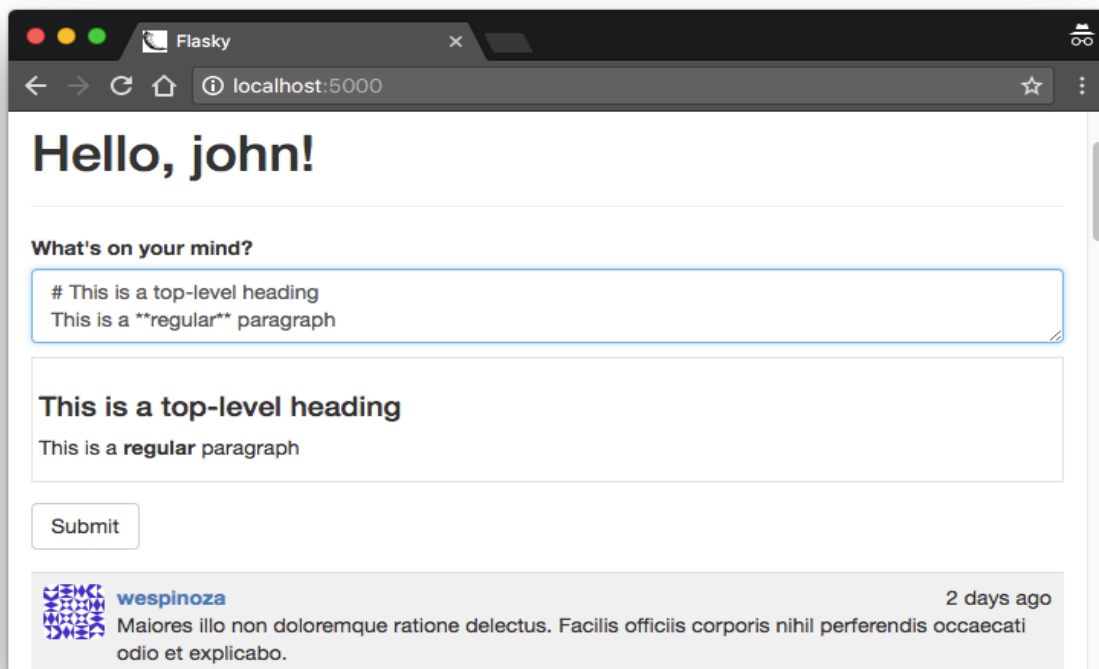
8. Viết chương trình python để tạo trang chủ với biểu mẫu giúp người dùng đăng blog và danh sách bài đăng trên blog như sau:



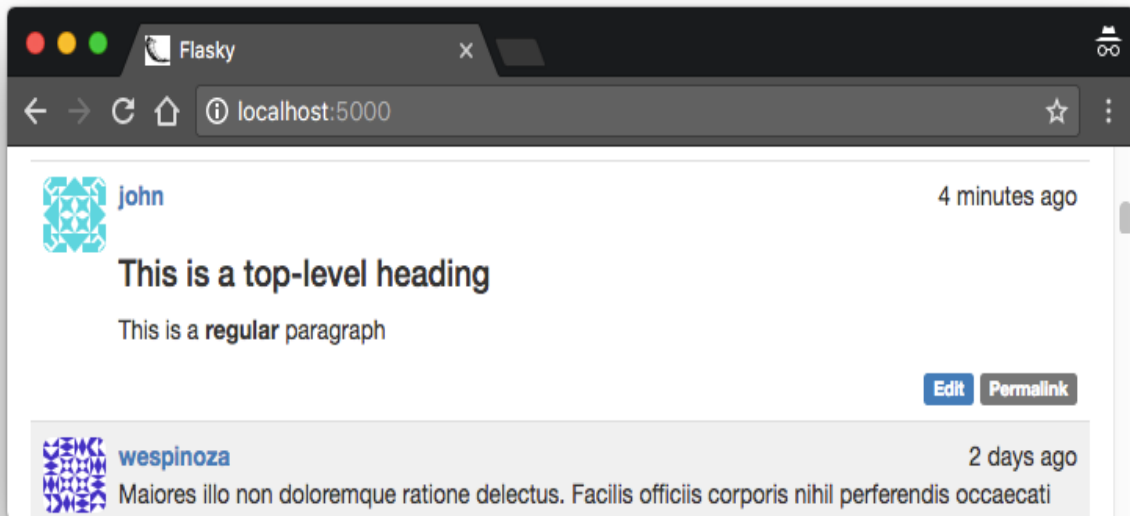
9. Viết chương trình python để trang Blog có thể phân ra các bài đăng như hình dưới:



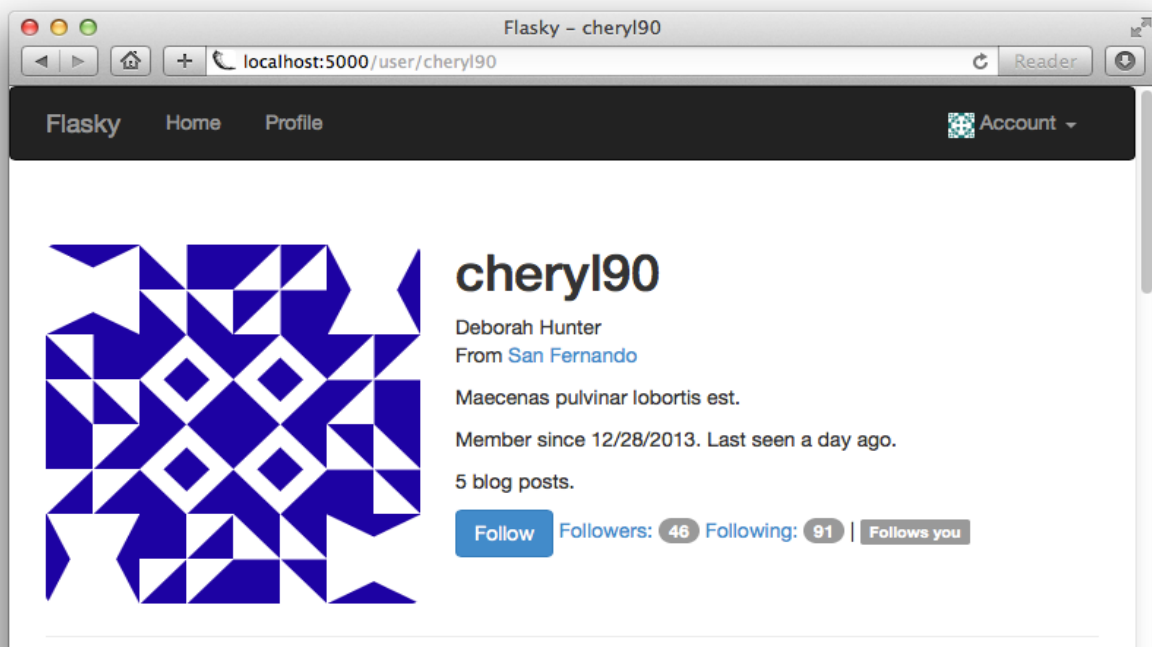
10. Viết chương trình python để người dùng có thể đăng bài đăng blog đa văn bản như hình dưới:



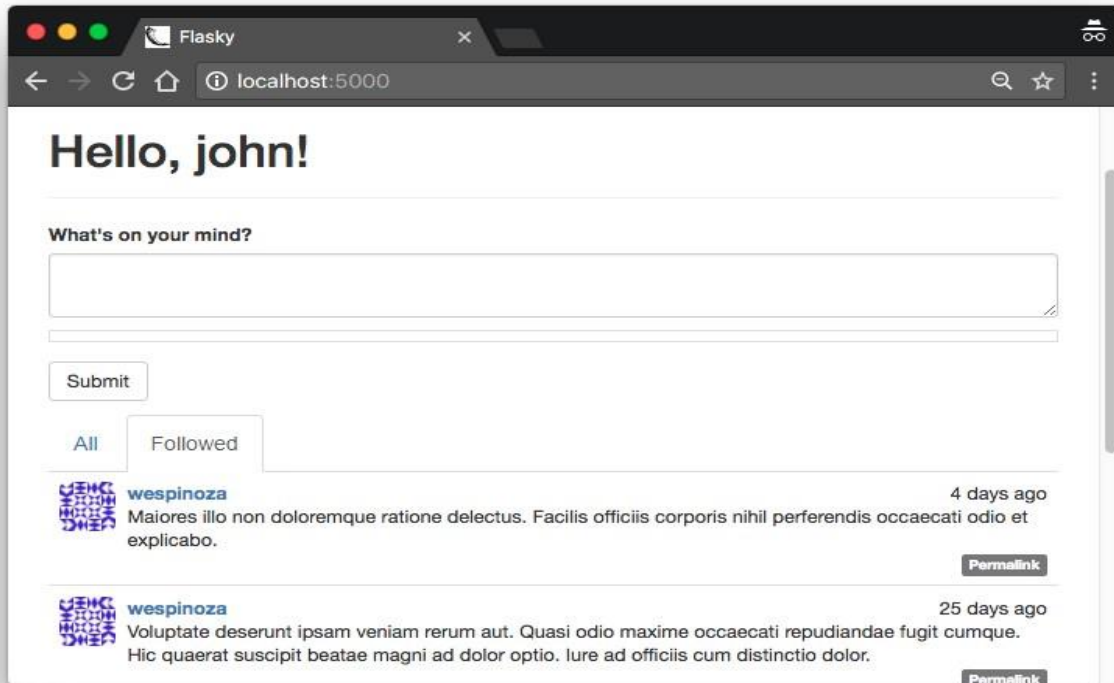
11. Viết chương trình python giúp người dùng có thể chỉnh sửa lại bài đăng đã đăng trên blog:



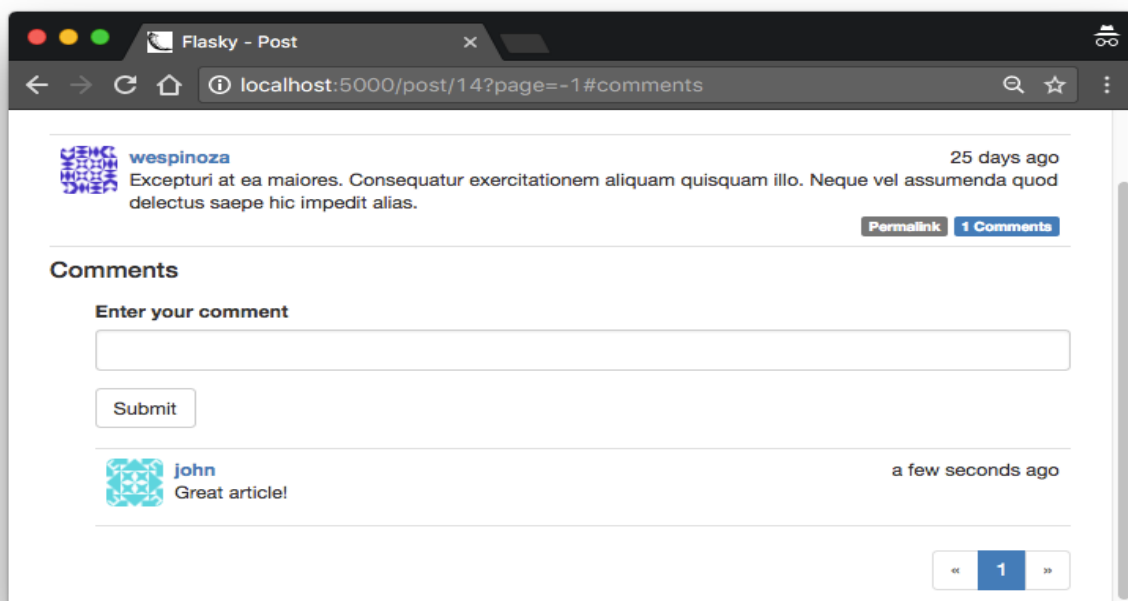
12. Viết chương trình python hiển thị nút follow giúp người dùng có thể follow và unfollow bất kỳ trang cá nhân hay trang nào khác, như hình dưới:



13. viết chương trình python hiển thị danh sách tất cả những trang cá nhân hoặc trang khác mà người dùng đã follow họ:



14. viết chương trình python hiển thị khung comment giúp người dùng có thể post comment của mình hoặc của những người bạn khác lên:



15. Viết chương trình Python giúp người dùng có thể điều chỉnh, kiểm duyệt các comment mà họ hoặc người bạn khác đã đăng lên:

