1. **Công cụ**

**Flask:**Flask là một web frameworks, nó thuộc loại micro-framework được xây dựng bằng ngôn ngữ lập trình Python. Flask 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.

**Jinja 2:**

Jinja2 là một templates engine, hỗ trợ cho việc show dữ liệu từ app ra html, được kèm theo Flask nên sau khi cài đặt Flask.

Jinja2 được sử dụng trên các page có định dạng: .html , .htm , .xml và .xhtml. Flask đưa vào một số function global và helper trong Jinja2 để có thể sử dụng các mã Python trên các template.

Các biến global sau được mặc định có thể sử dụng trong Jinja2

* **config** Cấu hình hiện tại của app (flask.config)
* **request** Lấy request hiện tại (flask.request). Biến này sẽ vô hiệu nếu template (html) đã được render mà không có request.
* **session** Lấy session hiện tại (flask.session). Biến này sẽ vô hiệu nếu template (html) đã được render mà không có request.
* **g** Lấy các request global mặc định (flask.g). Biến này sẽ vô hiệu nếu template (html) đã được render mà không có request.
* **url\_for()** Convert đường dẫn đến file (css, js, html,…) thành đường dẫn trên host.
* **get\_flashed\_messages()** Passing các giá trị được đặt trong hàm flash() ra html.

1. **Web**

* **Web động:**

Là một tập hợp các dữ liệu số hóa được tổ chức thành cơ sở dữ liệu, các dữ liệu số hóa được gọi ra trình diễn trên các trang web dưới dạng văn bản, âm thanh, hình ảnh.nó có thêm các phần xử lý thông tin và truy xuất dữ liệu còn website tĩnh thì không. Khác với web tĩnh, web động luôn luôn có thông tin mới do các thông tin này được cập nhật bởi phần mềm quản trị web do các công ty thiết kế website cung cấp. Các thông tin mới này được lưu vào cơ sở dữ liệu của website và đưa ra sử dụng dựa theo yêu cầu của người dùng.

* Web tĩnh:

Trang web tĩnh về cơ bản được định nghĩa là một trang web sử dụng hoàn toàn ngôn ngữ chính là HTML (có đuôi html hoặc htm), sau khi tải trang HTML từ máy chủ xuống, trình duyệt sẽ biên dịch mã và hiển thị nội dung trang web, người dùng hầu như sẽ không thể tương tác với trang web được.

* Mô hình Client server:

Trong mô hình client server thì máy khách là các máy tính, các thiết bị điện tử như máy in, máy fax,….các máy khách client gửi yêu cầu đến máy chủ server. máy chủ server tiếp nhận yêu cầu, xử lý các yêu cầu đó và trả về kết quả

**Client** Trong mô hình client/server, người ta còn định nghĩa cụ thể cho một máy client là một máy trạm mà chỉ được sử dụng bởi 1 người dùng với để muốn thể hiện tính độc lập cho nó. Máy client có thể sử dụng các hệ điều hành bình thường như Win9x, DOS, OS/2… Bản thân mỗi một client cũng đã được tích hợp nhiều chức năng trên hệ điều hành mà nó chạy, nhưng khi được nối vào một mạng LAN, WAN theo mô hình client/server thì nó còn có thể sử dụng thêm các chức năng do hệ điều hành mạng (NOS) cung cấp với nhiều dịch vụ khác nhau (cụ thể là các dịch vụ do các server trên mạng này cung cấp), ví dụ như nó có thể yêu cầu lấy dữ liệu từ một server hay gửi dữ liệu lên server đó… Thực tế trong các ứng dụng của mô hình client/server, các chức năng hoạt động chính là sự kết hợp giữa client và server với sự chia sẻ tài nguyên, dữ liệu trên cả hai máy Vai trò của client Trong mô hình client/server, client được coi như là người sử dụng các dịch vụ trên mạng do một hoặc nhiều máy chủ cung cấp và server được coi như là người cung cấp dịch vụ để trả lời các yêu cầu của các clients. Điều quan trọng là phải hiểu được vai trò hoạt động của nó trong một mô hình cụ thể, một máy client trong mô hình này lại có thể là server trong một mô hình khác. Ví dụ cụ thể như một máy trạm làm việc như một client bình thường trong mạng LAN nhưng đồng thời nó có thể đóng vai trò như một máy in chủ (printer server) cung cấp dịch vụ in ấn từ xa cho nhiều người khác (clients) sử dụng. Client được hiểu như là bề nổi của các dịch vụ trên mạng, nếu có thông tin vào hoặc ra thì chúng sẽ được hiển thị trên máy client.

**Server** được định nghĩa như là một máy tính nhiều người sử dụng (multiuser computer). Vì một server phải quản lý nhiều yêu cầu từ các client trên mạng cho nên nó hoạt động sẽ tốt hơn nếu hệ điều hành của nó là đa nhiệm với các tính năng hoạt động độc lập song song với nhau như hệ điều hành UNIX, WINDOWS… Server cung cấp và điều khiển các tiến trình truy cập vào tài nguyên của hệ thống. Các ứng dụng chạy trên server phải được tách rời nhau để một lỗi của ứng dụng này khô ng làm hỏng ứng dụng khác. Tính đa nhiệm đảm bảo một tiến trình không sử dụng toàn bộ tài nguyên hệ thống. Vai trò của server. Như chúng ta đã bàn ở trên, server như là một nhà cung cấp dịch vụ cho các clients yêu cầu tới khi cần, các dịch vụ như cơ sở dữ liệu, in ấn, truyền file, hệ thống… Các ứng dụng server cung cấp các dịch vụ mang tính chức năng để hỗ trợ cho các hoạt động trên các máy clients có hiệu quả hơn. Sự hỗ trợ của các dịch vụ này có thể là toàn bộ hoặc chỉ một phần thông qua IPC. Để đảm bảo tính an toàn trên mạng cho nên server này còn có vai trò như là một nhà quản lý toàn bộ quyền truy cập dữ liệu của các máy clients, nói cách khác đó là vai trò quản trị mạng. Có rất nhiều cách thức hiện nay nhằm quản trị có hiệu quả, một trong những cách đang được sử dụng đó là dùng tên Login và mật khẩu

Python khi làm việc với html: xử lý trên html và trả về kết quả (lấy đi một số thứ và trả lại một số thứ). Tùy từng phần mềm mà sẽ có các giao thức khác nhau

Giao thức: là một cách để truyền đữ liệu, một tập hợp các quy tắc chuẩn dành cho việc biểu diễn dữ liệu, phát tín hiệu.

4 giao thức của HTTP: get, post,put,delete. Đều có thể truyền dữ liệu từ client lên server. Tất cả những gì gõ lên trình duyệt đều là GET, điền thông tin lên web sử dụng giao thức POST (đóng dữ liệu vào hòm r gửi đi)