Program 9 - Flask

pip3 install -r requirements.txt

pip3 install Flask

To run

python3 application.py

\_\_name\_\_ - name of the program, default built-in variable

7 - takes route and methods (decorator)

/ - default route

/usnvalidator - other route (pseudo path )

Flask methods

- render\_template - renders a HTML template

- url\_for - find the url associated with a method

{{ }} - to include python code in HTML ( uses templating engine - Jinja )

url\_for(<method\_name>)

self loop with index.html

GET - render the form

POST - process the form

client validation - validate form - returns boolean ( true if form details valid, else false )

Program 6

Closures - anonymous functions ( no name )

\*\*\*\*\*\*\*\*\*

'''

(v) What is the meaning of the code - @app.route("/serversidepython", methods=['GET', ‘POST']) ? Demonstrate a simple

change made to this code and explain why it used ?

Answer: Here 'serversidepython' is just a place-holder. Flask will look for the html files in the 'templates' folder of the 'root' or

the folder from where the python application is run. The IP Address and the port number is visible on the console when the

python aplication is invoked. If this placeholder is used, then the URL entered in the browser to run this program

must have this placeholder. Else a HTTP-404 'Not Found' error is displayed.

Correct Useage in browser: http://127.0.0.1:5000/serversidepython

(vi) How is the client-side html and the Python Code at server-side connected? Explain the flow of the execution.

Answer: Execution Flow

Run in console the command python3 serversidepython.py

The URL is displayed - IP Address and Port Number where Flask with this python code, is listening to HTTP Requests

Open the broswer and type this URL. Note: Check 'approutes' in python to see if URL Placeholder is used

Correct Useage in browser: http://127.0.0.1:5000/serversidepython for this program

Since it is a GET request, the serversidevalidation.html is loaded

The HTML's Submit button has a FORM ACTION tag that directs it to the Python function (here 'myindex') inside the python application run.

This function checks if it is a GET request, and renders the HTML page (serversidevalidation.html)

Aftet the user enters the value, the Submit button makes a POST request

This request comes back to the ServerSidePython.py in the POST part of the 'myindex' function

Null validations are checked and the appropriate error / success messages are diaplayed

'''