

19_feb_assignment(flask)

September 23, 2023

0.0.1 Question-1: What is Flask Framework? What are the advantages of Flask Framework?

0.0.2 Ans:-

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by Armin Ronacher who leads an international group of python enthusiasts (POCCO). It is based on WSGI (web server gateway interface) toolkit and jinja2 template engine. Flask is considered as a micro framework.

WSGI: It is an acronym for web server gateway interface which is a standard for python web application development. It is considered as the specification for the universal interface between the web server and web application. Jinja2: Jinja2 is a web template engine which combines a template with a certain data source to render the dynamic web pages.

Advantages of Flask Framework:-

1) It is a lightweight framework that offers hassle-free development. 2) Provide flexibility to the developer to experiment with their modules or architecture. 3) It is suitable for small projects. 4) Offers a built-in development server and fast debugger. 5) Easily scalable for the applications. 6) Support for secure cookies. 7) Uses Ninja2 Template engine. 8) It has integrated support for unit testing. 9) APIs are coherent and neat. 10) Provide strong WSGI support.

0.0.3 Question-2: Create a simple Flask application to display 'Hello World!!'. Attach the screenshot of the output in Jupyter Notebook.

0.0.4 Ans:-

```
[1]: pip install flask
```

```
Requirement already satisfied: flask in /opt/conda/lib/python3.10/site-packages (2.3.3)
```

```
Requirement already satisfied: Jinja2>=3.1.2 in /opt/conda/lib/python3.10/site-packages (from flask) (3.1.2)
```

```
Requirement already satisfied: Werkzeug>=2.3.7 in /opt/conda/lib/python3.10/site-packages (from flask) (2.3.7)
```

```
Requirement already satisfied: click>=8.1.3 in /opt/conda/lib/python3.10/site-packages (from flask) (8.1.3)
```

```
Requirement already satisfied: itsdangerous>=2.1.2 in /opt/conda/lib/python3.10/site-packages (from flask) (2.1.2)
```

```
Requirement already satisfied: blinker>=1.6.2 in /opt/conda/lib/python3.10/site-
```

```
packages (from flask) (1.6.2)
Requirement already satisfied: MarkupSafe>=2.0 in
/opt/conda/lib/python3.10/site-packages (from Jinja2>=3.1.2->flask) (2.1.1)
Note: you may need to restart the kernel to use updated packages.
```

```
[3]: from flask import Flask
```

```
[ ]: app=Flask(__name__)

@app.route('/')
def hello_world():
    return "<h1> Hello World !! </h1>"
if __name__=="__main__":
    app.run(host="0.0.0.0")
```

```
* Serving Flask app '__main__'
* Debug mode: off
```

WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

```
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.18.0.3:5000
```

Press CTRL+C to quit

```
172.18.0.2 - - [22/Sep/2023 11:34:37] "GET / HTTP/1.1" 200 -
172.18.0.2 - - [22/Sep/2023 11:34:38] "GET /favicon.ico HTTP/1.1" 404 -
172.18.0.2 - - [22/Sep/2023 11:35:09] "GET / HTTP/1.1" 200 -
```



0.0.5 Question-3: What is App routing in Flask? Why do we use app routes?

0.0.6 Ans:-

In Flask, the process of defining the URLs that map to specific functions or methods in your application is known as routing. App routing is the technique used to map the specific URL with the associated function intended to perform some task. The Latest Web frameworks use the routing technique to help users remember application URLs. It is helpful to access the desired page directly without navigating from the home page. The URL ('/') is linked with the root URL in every application.

There are two ways through which we can bind the URLs to their associated function:- 1.) Using route() decorator. 2.) Using add_url_rule() function.

The route() decorator in Flask is used to bind an URL to a function.

App routing is used to map the specific URL with the associated function that is intended to perform some task.

0.0.7 Question-4: Create a “/welcome” route to display the welcome message “Welcome to ABC Corporation” and a “/” route to show the following details: Company Name: ABC Corporation Location: India Contact Detail: 999-999-9999

0.0.8 Ans:-

```
[ ]: from flask import Flask

app = Flask(__name__)

@app.route("/Welcome")
def Welcome():
    return "<h1>Welcome to ABC Corporation</h1>"

@app.route("/Welcome/Details")
def Details():
    return "<p> <h1>Company Name : ABC Corporation <br> Location: India <br>↳Contact Detail: 999-999-9999</h1></p>"

if __name__=="__main__":
    app.run(host="0.0.0.0")
```

```
* Serving Flask app '__main__'
* Debug mode: off
```

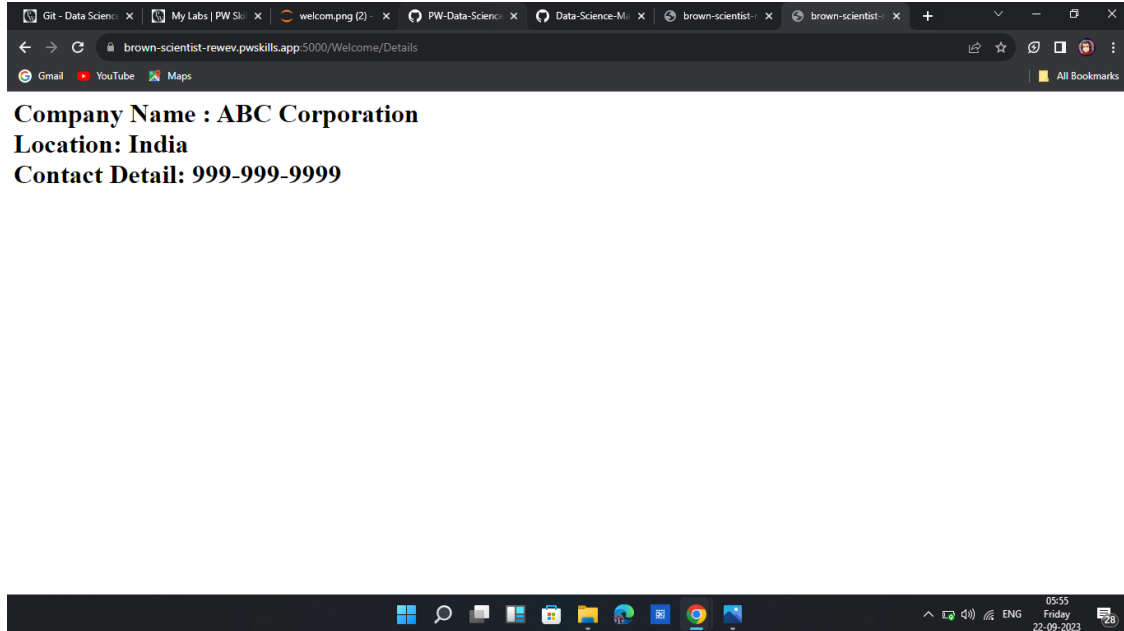
WARNING: This is a development server. Do not use it in a production deployment.

Use a production WSGI server instead.

```
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.18.0.3:5000
```

Press CTRL+C to quit

172.18.0.2 - - [22/Sep/2023 12:51:17] "GET /Welcome/Details HTTP/1.1" 200 -



0.0.9 Question-5: What function is used in Flask for URL Building? Write a Python code to demonstrate the working of the url_for() function.

0.0.10 Ans:-

The `url_for()` function is used to build a URL to the specific function dynamically. The first argument is the name of the specified function, and then we can pass any number of keyword argument corresponding to the variable part of the URL. This function is useful in the sense that we can avoid hard-coding the URLs into the templates by dynamically building them using this function.

```
[ ]: from flask import *

app=Flask(__name__)

@app.route("/admin")
def admin():
    return "admin"

@app.route("/librarian")
def librarian():
    return "librarian"

@app.route("/studentr")
def student():
    return "student"

@app.route("/user/<name>")
```

```
def user(name):

    if name=="admin":
        return redirect(url_for("admin"))

    if name=="librarian":
        return redirect(url_for("librarian"))

    if name=="student":
        return redirect(url_for("student"))

if __name__=="__main__":
    app.run(host="0.0.0.0")
```

```
* Serving Flask app '__main__'
* Debug mode: off
```

WARNING: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.

```
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.18.0.3:5000
```

Press CTRL+C to quit

```
172.18.0.2 - - [22/Sep/2023 13:25:25] "GET /admin HTTP/1.1" 200 -
172.18.0.2 - - [22/Sep/2023 13:26:24] "GET /user HTTP/1.1" 404 -
[2023-09-22 13:26:48,266] ERROR in app: Exception on /user/<name> [GET]
Traceback (most recent call last):
```

```
File "/opt/conda/lib/python3.10/site-packages/flask/app.py", line 2190, in
wsgi_app
```

```
    response = self.full_dispatch_request()
```

```
File "/opt/conda/lib/python3.10/site-packages/flask/app.py", line 1487, in
full_dispatch_request
```

```
    return self.finalize_request(rv)
```

```
File "/opt/conda/lib/python3.10/site-packages/flask/app.py", line 1506, in
finalize_request
```

```
    response = self.make_response(rv)
```

```
File "/opt/conda/lib/python3.10/site-packages/flask/app.py", line 1801, in
make_response
```

```
    raise TypeError(
```

```
TypeError: The view function for 'user' did not return a valid response. The
function either returned None or ended without a return statement.
```

```
172.18.0.2 - - [22/Sep/2023 13:26:48] "GET /user/<name> HTTP/1.1" 500 -
172.18.0.2 - - [22/Sep/2023 13:27:03] "GET /user/admin HTTP/1.1" 302 -
172.18.0.2 - - [22/Sep/2023 13:27:03] "GET /admin HTTP/1.1" 200 -
172.18.0.2 - - [22/Sep/2023 13:27:15] "GET /librarian HTTP/1.1" 200 -
```

