

Assignment:2

Team id	PNT2022PMID49597
Project Name	Plasma Donor Application
Name	Saravanan S
Roll No	950019104041

1.Create registration page in html with username, email and phone number and by using POST method display it in the next html page.

app.py

```
from flask import Flask, render_template, request, redirect
app = Flask(__name__)
@app.route('/')
def home():
    return 'Welcome! <a href="/login">Register here</a>'

@app.route('/login', methods=['POST', 'GET'])
def login():
    if request.method == 'POST':
        userName = request.form['userName']
        userEmail = request.form['userEmail']
        userPassword = request.form['userPassword']
        return redirect('/')
    return render_template("form.html")

if __name__ == '__main__':
    app.run(debug=True)
```

form.html

```
<form method="POST">

    <div class="form-inline">
        <div class="form-group">
            <input type="text" class="line-input" name="userName"
placeholder="Username">
        </div>
        <div class="form-inline">
            <div class="form-group">
                <input type="email" class="line-input" name="userEmail"
placeholder="Email">
            </div>
            <div class="form-group">
                <input type="password" class="line-input" name="userPassword"
placeholder="Password">
            </div>
        </div>
    </div>
</form>
```

```
        </div>
      </div>
      <div class=" form-group">
        <button type="submit" class="btn btn-light text-primary btn-block"
style="margin : 20px 20px -10px 0px">Register</button>
      </div>

    </form>
```

OUTPUT:



A screenshot of a web browser displaying a login form. The browser's address bar shows the URL "127.0.0.1:5000/login". The form consists of four input fields stacked vertically: the first contains the email "Saravanan10@gmail.com", the second contains masked characters "*****", the third contains the name "Suswin", and the fourth contains the text "hi". Below these fields is a button labeled "Send Message".



2. Develop a flask program which should contain at least 5 packages used from pypi.org

```
"import pandas as pd

import parser
import colorama
from colorama import Fore, Back, Style
import numpy as np
from jinja2 import Template

#panda
data = pd.DataFrame({"x1":["y", "x", "y", "x", "x", "y"], # Construct a pandas
DataFrame
    "x2":range(16, 22),
    "x3":range(1, 7),
    "x4":["a", "b", "c", "d", "e", "f"],
    "x5":range(30, 24, - 1)})
print(data)

#parser
print("Program to demonstrate parser module in Python")
print("\n")
exp = "5 + 8"
print("The given expression for parsing is as follows:")
print(exp)
print("\n")
print("Parsing of given expression results as: ")
st = parser.expr(exp)
print(st)
print("\n")
print("The parsed object is converted to the code object")
code = st.compile()
print(code)
print("\n")
print("The evaluated result of the given expression is as follows:")
res = eval(code)
print(res)

#colorama
colorama.init(autoreset=True)
#Print text using background and font colors
print(Back.RED + Fore.BLUE + "Welcome to LinuxHint")
#Add newline
print()
#Print text using background color
print(Back.GREEN + "I like programming")

#numpy
arr = np.array([1, 2, 3, 4, 5])
print(" NUMPY ")
print(arr)
print(type(arr))
print("\n")
```

```
#jinja2
template = """hostname {{ hostname }}"""
data = {"hostname": "core-sw-waw-01"}
j2_template = Template(template)
print(" JINJA2 ")
print(j2_template.render(data))
print("\n")
```

OUTPUT:

The screenshot shows the Spyder IDE with a Python script in the editor and its output in the console. The script demonstrates various Python modules and Jinja2 templating.

```
1 import pandas as pd
2 import parser
3 import colorama
4 from colorama import Fore, Back, Style
5 import numpy as np
6 from jinja2 import Template
7
8 #panda
9 data = pd.DataFrame({"x1": ["y", "x", "y", "x", "y"], # Construct a pandas DataFrame
10                      "x2": range(16, 22),
11                      "x3": range(1, 7),
12                      "x4": ["a", "b", "c", "d", "e", "f"],
13                      "x5": range(30, 24, -1)})
14 print(data)
15
16 #parser
17 print("Program to demonstrate parser module in Python")
18 print("\n")
19 exp = "5 + 8"
20 print("The given expression for parsing is as follows:")
21 print(exp)
22 print("\n")
23 print("Parsing of given expression results as: ")
24 st = parser.expr(exp)
25 print(st)
26 print("\n")
27 print("The parsed object is converted to the code object")
28 code = st.compile()
29 print(code)
30 print("\n")
31 print("The evaluated result of the given expression is as follows:")
32 res = eval(code)
33 print(res)
34
35 #colorama
36 colorama.init(autoreset=True)
37
38 #Print text using background and font colors
39 print(Back.RED + Fore.BLUE + "Welcome to LinuxHint")
40 #Add newline
41 print()
42 #Print text using background color
43 print(Back.GREEN + "I like programming")
```

The console output shows the execution of the script, including the DataFrame structure, the parsing of the expression "5 + 8", and the rendering of the Jinja2 template.

```
5 y 21 6 + 25
Program to demonstrate parser module in Python

The given expression for parsing is as follows:
5 + 8

Parsing of given expression results as:
<parser.st object at 0x000001B095C37120>

The parsed object is converted to the code object
<code object <module> at 0x000001B095C1AA80, file "<syntax-tree>", line 1>

The evaluated result of the given expression is as follows:
13
Welcome to LinuxHint
I like programming
NUMPY
[1 2 3 4 5]
<class 'numpy.ndarray'>

JINJA2
hostname core-sw-waw-01

In [23]: runfile('C:/flask/pillow.py', wdir='C:/flask')
Hello guest

In [24]:
```

The screenshot shows a Windows Command Prompt window with the execution of a Python script. The output is similar to the one in the Spyder IDE, but with some additional warnings and a different DataFrame structure.

```
C:\flask>python panda.py
C:\flask\panda.py:2: DeprecationWarning: The parser module is deprecated and will be removed in future versions of Python
  import parser
  x1 x2 x3 x4 x5
0 y 16 1 a 30
1 x 17 2 b 29
2 y 18 3 c 28
3 x 19 4 d 27
4 x 20 5 e 26
5 y 21 6 f 25
Program to demonstrate parser module in Python

The given expression for parsing is as follows:
5 + 8

Parsing of given expression results as:
<parser.st object at 0x00000207AB3F1360>

The parsed object is converted to the code object
<code object <module> at 0x00000207ABE1D500, file "<syntax-tree>", line 1>

The evaluated result of the given expression is as follows:
13
Welcome to LinuxHint
I like programming
NUMPY
[1 2 3 4 5]
<class 'numpy.ndarray'>

JINJA2
hostname core-sw-waw-01

C:\flask>
```