

Assignment -1
Python
Programming

Assignment Date	19 September 2022
Student Name	Akshay Ajit
Student Roll Number	2019115013
Maximum Marks	2 Marks

Question
1:

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

Solution 1:

register.html

```
<!DOCTYPE HTML>

<html>

    <head>

        <title> User Registration Form </title>

    </head>

    <body>

        <H2> User Registration Form </H2>

        <form action="{{ url_for('register')}}" method="POST">

            <div class="msg">{{ msg }}</div>

            <input id="username" name="username" type="text" placeholder="Enter Your Username" class="textbox"/></br></br>

            <input id="email" name="email" type="text" placeholder="Enter Your Email ID" class="textbox"/></br></br>

            <input id="mobile" name="mobile" type="number" placeholder="Enter Your Mobile No." class="textbox"/></br></br>

            <input type="submit" class="btn" value="SUBMIT"></br>

        </FORM>

    </BODY>

</HTML>
```

dashboard.html

```
<!DOCTYPE HTML>

<html>

    <head>

        <title>User application</title>

    </head>

    <body>

        <div class="msg1">Username : {{ msg1 }}</div>

        <div class="msg2">Email : {{ msg2 }}</div>

        <div class="msg3">Phone no: {{ msg3 }}</div>

    </BODY>

</HTML>
```

app.py

```
from flask import Flask, render_template, request, redirect, url_for, session
```

```
app = Flask(__name__)
```

```
@app.route('/', methods = ['GET',
'POST'])def register():
```

```
    msg = "
```

```
if request.method == 'POST' and 'username' in request.form and 'email' in request.form and 'mobile' in request.form:
```

```
    username =
```

```
    request.form['username'] email =
```

```
    request.form['email']
```

```
    mobile =
```

```
    request.form['mobile'] msg =
```

```
    'Registered Successfully!'
```

```
    return render_template('dashboard.html', msg1 = username, msg2 = email, msg3 =
```

```
    mobile) return render_template('register.html', msg = msg)
```

```
if __name__ == '__main__':
```

```
    app.run()
```

OUTPUT:



A screenshot of a web browser window. The address bar shows a local IP address: 127.0.0.1:5000. The page title is "User Registration Form". Below the title, there are three input fields. The first field contains the text "Akshay Ajit". The second field contains the text "akshay@gmail.com". The third field contains the text "9090978787". Below these fields is a button labeled "SUBMIT".

Username : Akshay Ajit
Email : akshay@gmail.com
Phone no: 9090978787

Question 2:

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

```
from flask import Flask, render_template, request, redirect, url_for, session

app = Flask(__name__)
@app.route('/', methods = ['GET', 'POST'])
def register():
    msg = ''
    if request.method == 'POST' and 'username' in request.form and 'email' in request.form and 'mobile' in request.form:
        username = request.form['username']
        email = request.form['email']
        mobile = request.form['mobile']
        msg = 'Registered Successfully!'
        return render_template('dashboard.html', msg1 = username, msg2 = email, msg3 = mobile)
        return render_template('register.html', msg = msg)

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

Output:

The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project structure with 'OPEN EDITORS' containing 'app.py' and 'ASSIGNMENT1' containing 'flask-env' and 'app.py'. The main editor area displays the code for 'app.py', which is a Flask application with a single route for a registration form. The code includes imports for Flask, render_template, request, redirect, url_for, and session. It defines a 'register()' function that checks for POST requests and validates 'username', 'email', and 'mobile' fields. If valid, it renders a 'dashboard.html' template with the user details. The application is run using 'app.run()'.

```
1 from flask import Flask, render_template, request, redirect, url_for, session
2
3 app = Flask(__name__)
4 @app.route('/', methods=['GET', 'POST'])
5 def register():
6     msg = ''
7     if request.method == 'POST' and 'username' in request.form and 'email' in request.form and 'mobile' in request.form:
8         username = request.form['username']
9         email = request.form['email']
10        mobile = request.form['mobile']
11        msg = 'Registered Successfully!'
12        return render_template('dashboard.html', msg1 = username, msg2 = email, msg3 = mobile)
13    return render_template('register.html', msg = msg)
14
15 if __name__ == '__main__':
16     app.run()
```

The bottom panel shows the 'TERMINAL' output, which details the pip installation process for the application's dependencies. It lists packages like Werkzeug, itsdangerous, click, Jinja2, and flask, along with their versions and sizes. A warning message indicates that the current pip version (22.0.4) is older than the available version (22.3). The terminal also shows the command used to run the application: '(flask-env) (base) → Assignment1 python app.py'. The application is running on http://127.0.0.1:5000.

Using cached click-8.1.3-py3-none-any.whl (96 kB)
Collecting Werkzeug>=2.2.2
Using cached Werkzeug-2.2.2-py3-none-any.whl (232 kB)
Collecting itsdangerous>=2.0
Using cached itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting importlib-metadata>=3.6.0
Using cached importlib_metadata-5.0.0-py3-none-any.whl (21 kB)
Collecting zipp>=0.5
Downloading zipp-3.10.0-py3-none-any.whl (6.2 kB)
Collecting MarkupSafe>=2.0
Using cached MarkupSafe-2.1.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (25 kB)
Installing collected packages: zipp, MarkupSafe, itsdangerous, click, Werkzeug, Jinja2, importlib-metadata, flask
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 flask-2.2.2 importlib-metadata-5.0.0 itsdangerous-2.1.2 zipp-3.10.0
WARNING: You are using pip version 22.0.4; however, version 22.3 is available.
You should consider upgrading via the '/run/media/akshayajit/Large drive/academics/7thsem/IBMPProject/Assignments/Assignment1/flask-env/bin/python -m pip install --upgrade pip' command.
○ (flask-env) (base) → Assignment1 python app.py
* Serving Flask app 'app'
* 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 http://127.0.0.1:5000
Press CTRL-C to quit