

## Assignment -1

### Flask Programs- with HTML

Assignment Date	19 September 2022
Student Name	N. Ramya
Student Roll Number	2019115076
Maximum Marks	2 Marks

Q1. Create registration page in html with username, email and phone number and by using POST method display it in next html page.

#### Solution:

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

app = Flask(__name__)

@app.route("/")
def home():
    return render_template('form.html')

@app.route("/Register", methods=['GET', 'POST'])
def login():
    if request.method=="POST":
        return render_template('result.html', name=request.form["name"],
                               email=request.form["email"],phno=request.form["phno"])
        return redirect(url_for('home'))

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

#### Register.html

```
<!doctype html>
<html>
  <head>
    <title>Registration</title>
  </head>
  <body>
    <form action="/Register" method="POST">
      <table>
        <tr>
          <td><label for="name">Username: </label></td>
          <td><input type="text" name="name" id="name" /></td>
```

```

        </tr>
        <tr>
            <td><label for="email">Email: </label></td>
            <td><input type="email" name="email" id="email" /></td>
        </tr>
        <tr>
            <td><label for="phno">Phone Number: </label></td>
            <td><input type="number" name="phno" id="phno" /></td>
        </tr>

    </table>
    <div>
        <input type="submit" value="Submit" />
    </div>
</form>
</body>
</html>

```

### Result.html

```

<!doctype html>
<html>
    <head>
        <title>Login</title>
    </head>
    <body>
        <table>
            <tr>
                <th>
                    Username
                </th>
                <td>
                    {{name}}
                </td>
            </tr>
            <tr>
                <th>
                    Email
                </th>
                <td>
                    {{email}}
                </td>
            </tr>
            <tr>
                <th>
                    Phone Number
                </th>
                <td>

```

```
        {{phno}}
    </td>
</tr>
</body>
</html>
```

```
(base) C:\Users\DELL>cd documents
(base) C:\Users\DELL\Documents>cd 7th sem
(base) C:\Users\DELL\Documents\7th sem>cd IBM
(base) C:\Users\DELL\Documents\7th sem\IBM>python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with watchdog (windowsapi)
* Debugger is active!
* Debugger PIN: 134-189-173
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

← → ↻ ⓘ localhost:5000

---

Username:

Email:

Phone Number:

Displaying results in next html page:

← → ↻ ⓘ localhost:5000/Register

---

**Username** Ramya  
**Email** ramyanmsr@gmail.com  
**Phone Number** 123456789

Q2. Develop a flask program which should contain at least 5 packages used from pypi.org.

**Solution:**

from flask import Flask, request, redirect, url\_for, render\_template

import cowsay

```

import art
import io
import numpy as np
import base64
import random
from randimage import get_random_image
from PIL import Image

app = Flask(__name__)

@app.route("/")
def home():
    img = Image.fromarray((get_random_image((128, 128)) * 255).astype(np.uint8))
    stream = io.BytesIO()
    img.save(stream, format="PNG")
    imgstream = base64.b64encode(stream.getvalue()).decode()

    return render_template('index.html',
        cowsay=cowsay.get_output_string(random.choice(cowsay.char_names), 'Hello World'),
        art=art.text2art("ART", "rnd-xlarge"), img=imgstream)

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

```

```

<!doctype html>
<html>
  <head>
    <title>Login</title>
  </head>
  <body>
    <table>
      <tr>
        <th>
          Cowsay
        </th>
        <td>
          <pre>{{ cowsay }}</pre>
        </td>
      </tr>
    </table>
  </body>
</html>

```

← → ↻ ⓘ localhost:5000

[illegible]