

## ASSIGNMENT – 1

Name	M. YOKESH RAJA
Batch	B7-1A3E

### AIM

1. Create registration page in html with username, email and phone number and by using POST method display it in next html page.
2. Develop a flask program which should contain at least 5 packages used from pypi.org

### DIRECTORY STRUCTURE

- static
  - graphs
  - styles
- response.css
- templates
  - form.html
  - response.html
  - graph.html
- app.py

### CODE 1. HTML FILES

#### a) form.html

```
<!DOCTYPE html>

<html>
<body>

    <h1>REGISTRATION FORM</h1>

    <form action="/" method="POST">
        <label>USERNAME</label>
        <br>
        <input type="text" name="username" required>
        <br><br>

        <label>EMAIL</label>
        <br>
```

```

        <input type="email" name="email" required>
        <br><br>

        <label>MOBILE NUMBER</label>
        <br>
        <input type="number" name="mobile_no" required>
        <br><br>

        <button type="submit">SUBMIT</button>
    </form>

</body>
</html>

```

## b) response.html

```

<!DOCTYPE html>

<html>

<link rel="stylesheet" href="{url_for('static', filename='styles/response.css')}">

<body>

    <h1>RESPONSE COPY</h1>

    <table>
        <tr>
            <th>USERNAME</th>
            <td>{{username}}</td>
        </tr>

        <tr>
            <th>EMAIL</th>
            <td>{{email}}</td>
        </tr>

        <tr>
            <th>MOBILE NUMBER</th>
            <td>{{mobile_no}}</td>
        </tr>
    </table>

</body>
</html>

```

### c) graph.html

```
<!DOCTYPE html>

<html>
<body>

    <h2>GRAPH DATA</h2>

    <p>USER COUNT - {{user_count}}</p>

</body>

</html>
```

## 2. CSS FILES

### a) response.css

```
table {    border: 1px
solid black;    text-
align: center;
}

th, td {    border: 1px
solid black;    text-
align: center;
}
```

## 3. app.py

```
from flask import Flask from
flask import render_template
from flask import request
import seaborn as sns import
matplotlib.pyplot as plt import
numpy as np
```

```

import matplotlib
matplotlib.use('Agg')

app = Flask(__name__)

user_count = 0

@app.route("/", methods=["GET", "POST"])
def Form():
    if(request.method=="GET"):
        return render_template("form.html")
    elif(request.method=="POST"):
        username=request.form["username"]
        email=request.form["email"]
        mobile_no=request.form["mobile_no"]

        return
    render_template("response.html",username=username,email=email,mobile_no=mobile_no)

@app.route("/graph", methods=["GET", "POST"])
def Graph():
    global user_count

    if(request.method=="GET"):
        user_count =
        user_count+1

    print(user_count)
    df =
    sns.load_dataset('flights')
    data =
    dict()
    for i in
    range(0,len(df)):
        temp = df.loc[i]
        if(temp.year not in
        data):
            data[temp.year] = 0
            data[temp.year] = data[temp.year] +
            temp.passengers
        year = np.array(list(data.keys()))
        passengers = np.array(list(data.values()))


```

```
plt.clf()
plt.bar(year, passengers, color='green')
plt.xlabel('Year')      plt.ylabel('Passengers')
plt.savefig('static/graphs/img'+str(user_count)+'.png')

return render_template("graph.html", user_count = user_count, filename =
'graphs/img'+str(user_count)+'.png')

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

## OUTPUT 1. REGISTRATION PAGE



A screenshot of a web browser window displaying a registration form. The browser's address bar shows the URL '127.0.0.1:5000'. The form is titled 'REGISTRATION FORM' in a large, bold, black serif font. Below the title, there are three input fields: 'USERNAME' with the value 'steverogers', 'EMAIL' with the value 'user1@gmail.com', and 'MOBILE NUMBER' with the value '7894561230'. Each input field has a small downward arrow on its right side, indicating a dropdown menu. Below these fields is a 'SUBMIT' button.

REGISTRATION FORM

USERNAME  
steverogers

EMAIL  
user1@gmail.com

MOBILE NUMBER  
7894561230

SUBMIT



# RESPONSE COPY

USERNAME	steverogers
EMAIL	user1@gmail.com
MOBILE NUMBER	7894561230

## 2. PACKAGES (NUMPY, MATPLOTLIB, SEABORN)



# GRAPH DATA

USER COUNT - 1

