

<b>Assignment No.</b>	1
<b>Name</b>	Kirunraj.C
<b>Registration No.</b>	SSNCE195001053
<b>Team No.</b>	PNT2022TMID53036

### **Question-1:**

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

### **index.html**

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <title>Registration</title>
    <link rel="stylesheet"
href="{{ url_for('static',filename='styles/index.css') }}">
  </head>
  <body>
    <center>
      <br><br>
      <h1>Registration</h1><br>
      <form action="{{ url_for('result') }}" method="post">
        <table>
          <tr>
            <td>
              <label>Name</label>
            </td>
            <td>
              : <input type="text" class="name-input
name mb-3" id="name" name="name">
            </td>
          </tr>
          <tr>
            <td>
              <label>Email</label>
            </td>
            <td>
              : <input type="email" class="name-
input name mb-3" id="email" name="email">
            </td>
          </tr>
        </table>
      </form>
    </center>
  </body>
</html>
```

```

        </tr>
        <tr>
            <td>
                <label>Mobile</label>
            </td>
            <td>
                : <input type="number" class="name-
input name mb-3" id="mobile" name="mobile">
            </td>
        </tr>
    </table>
    <br><br>
    <input class="btn btn-outline-primary" type="submit"
value="Submit">
    </form>
    </center>
</body>
</html>

```

### **result.html**

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <title>Profile</title>
        <link rel="stylesheet"
href="{{ url_for('static',filename='styles/index.css') }}">
    </head>
    <body>
        <center>
            <br><br>
            <h1>Welcome !!!</h1>
            <br>
            <form action="{{ url_for('result') }}" method="post">
                <table>
                    <tr>
                        <h3>Name : {{ name }}<br></h3>
                    </tr>
                    <tr>
                        <h3>Email : {{ email }}<br></h3>
                    </tr>
                    <tr>

```

```

                <h3>Mobile : {{ mobile }}<br></h3>
            </tr>
        </table>
        <br><br>
    </form>
</center>
</body>
</html>

```

### **app.py**

```

from flask import *;
import os

app = Flask(__name__)

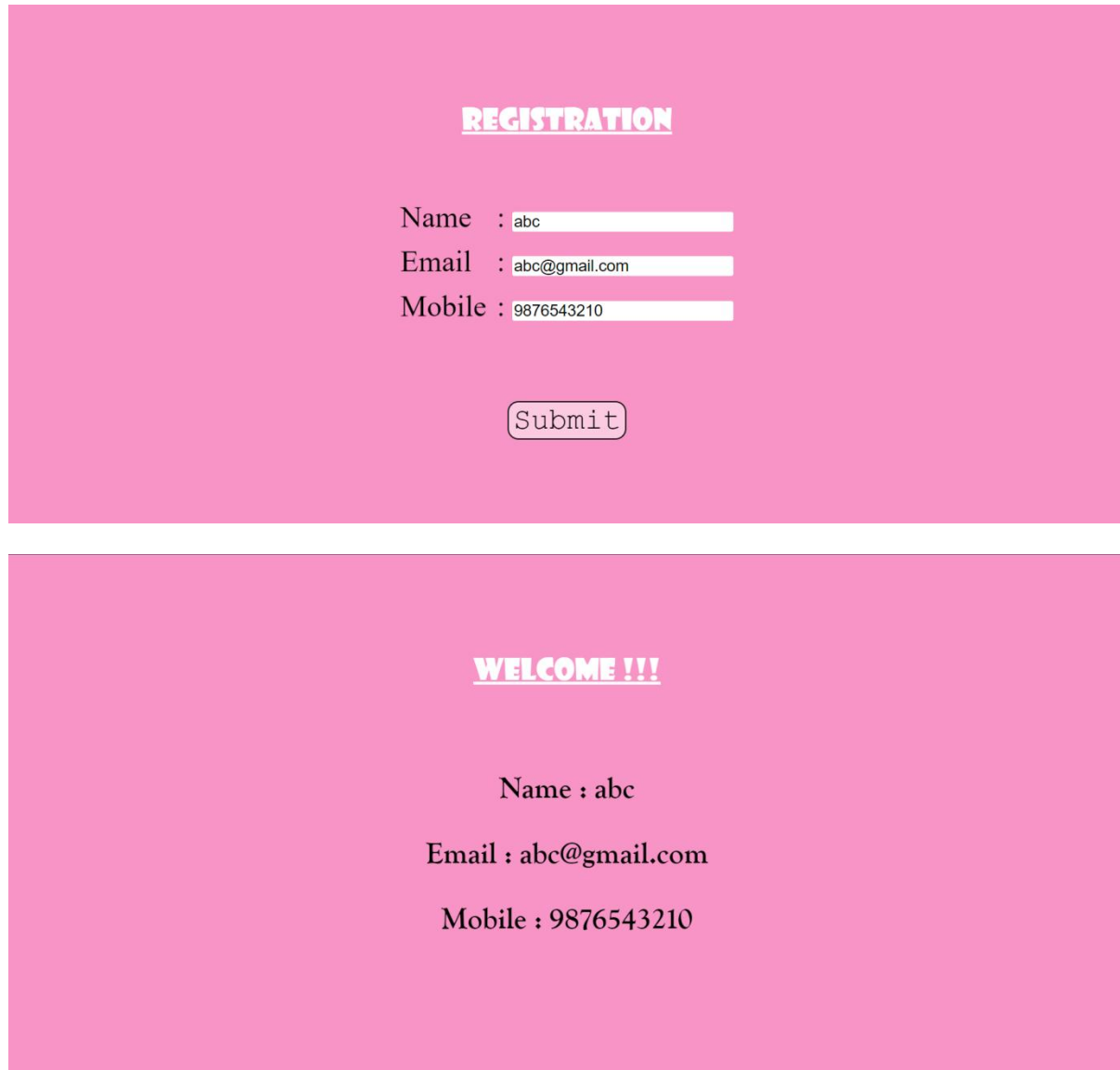
@app.route('/', methods=['GET', 'POST'])
def home():
    if request.method == 'POST':
        name = request.form["name"]
        email= request.form["email"]
        mobile = request.form["mobile"]
        return redirect(url_for('result', name=name, email=email,
mobile=mobile))
    return render_template('index.html')

@app.route("/result", methods=['GET', 'POST'])
def result():
    name = request.form.get('name')
    email= request.form.get('email')
    mobile = request.form.get('mobile')
    return render_template('result.html', name=name, email=email,
mobile=mobile)

if __name__ == "__main__":
    app.run(debug=True, port=2807)

```

**OUTPUT:**



The image displays two screenshots of a web application interface. The top screenshot shows a registration form titled "REGISTRATION" in bold, underlined, red text. Below the title, there are three input fields: "Name" with the value "abc", "Email" with the value "abc@gmail.com", and "Mobile" with the value "9876543210". A "Submit" button is located below the input fields. The bottom screenshot shows a welcome message titled "WELCOME !!!" in bold, underlined, red text. Below the title, the user's details are displayed: "Name : abc", "Email : abc@gmail.com", and "Mobile : 9876543210". Both screenshots have a light blue background.

**REGISTRATION**

Name : abc

Email : abc@gmail.com

Mobile : 9876543210

Submit

**WELCOME !!!**

Name : abc

Email : abc@gmail.com

Mobile : 9876543210

**Question-2:**

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



## #1. NUMPY

```
import numpy as np

arr = np.array([[ -1,  2,  0,  4],
                [ 4, -0.5, 6,  0],
                [ 2.6, 0,  7,  8],
                [ 3, -7,  4,  2.0]])

print("Initial Array: ")
print(arr)
```

```
Initial Array:
[[-1.  2.  0.  4. ]
 [ 4. -0.5  6.  0. ]
 [ 2.6  0.  7.  8. ]
 [ 3. -7.  4.  2. ]]
```

## #2. PANDAS

```
import pandas as pd

s1 = pd.Series([1, 3, 4, 5, 6, 2, 9])

s2 = pd.Series([1.1, 3.5, 4.7, 5.8, 2.9, 9.3])

s3 = pd.Series(['a', 'b', 'c', 'd', 'e'])

Data = {'first':s1, 'second':s2, 'third':s3}

df = pd.read_csv('/content/sample_data/ds_salaries.csv')
print(df)
```

```
   Unnamed: 0  work_year  experience_level  employment_type  \
0            0        2020                MI              FT
1            1        2020                SE              FT
2            2        2020                SE              FT
3            3        2020                MI              FT
4            4        2020                SE              FT
..          ...        ...                ...              ...
602          602        2022                SE              FT
603          603        2022                SE              FT
604          604        2022                SE              FT
605          605        2022                SE              FT
606          606        2022                MI              FT

   job_title  salary  salary_currency  salary_in_usd
\
0      Data Scientist    70000          EUR        79833
1  Machine Learning Scientist  260000          USD       260000
2      Big Data Engineer   85000          GBP       109024
```

3	Product Data Analyst	20000	USD	20000
4	Machine Learning Engineer	150000	USD	150000
..	...	...	...	...
602	Data Engineer	154000	USD	154000
603	Data Engineer	126000	USD	126000
604	Data Analyst	129000	USD	129000
605	Data Analyst	150000	USD	150000
606	AI Scientist	200000	USD	200000

	employee_residence	remote_ratio	company_location	company_size
0	DE	0	DE	L
1	JP	0	JP	S
2	GB	50	GB	M
3	HN	0	HN	S
4	US	50	US	L
..	...	...	...	...
602	US	100	US	M
603	US	100	US	M
604	US	0	US	M
605	US	100	US	M
606	IN	100	US	L

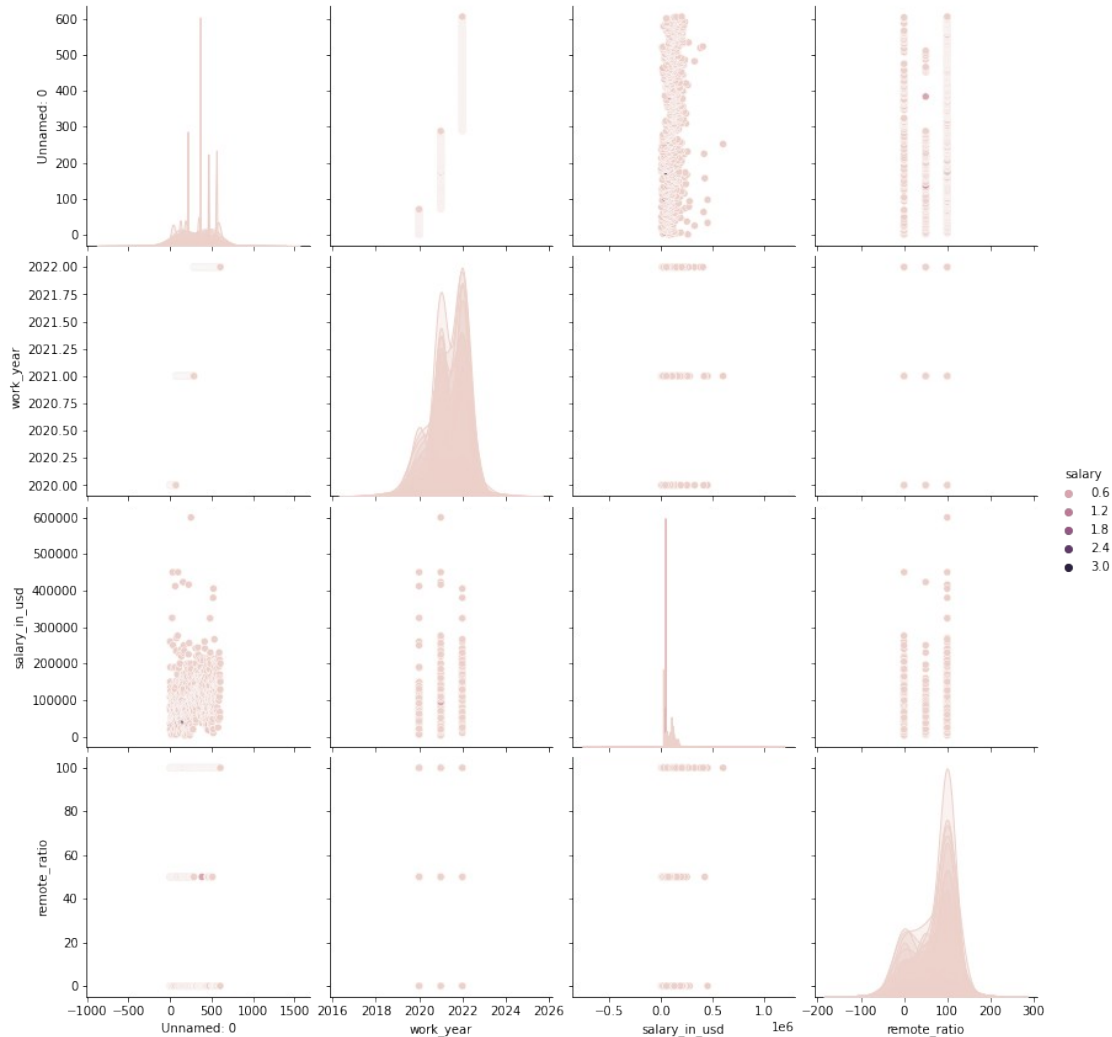
[607 rows x 12 columns]

### #3. SEABORN

```
import seaborn as sns
```

```
sns.pairplot(df,hue="salary",height=3)
```

```
<seaborn.axisgrid.PairGrid at 0x7f32024d5650>
```



#### #4. TENSORFLOW

```
import tensorflow as tf
```

```
mnist = tf.keras.datasets.mnist
(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10)
])
```

#### #5. PYTZ

```
from pytz import timezone
from datetime import datetime
```

```
format = "%Y-%m-%d %H:%M:%S %Z%z"
```



```
now_utc = datetime.now(timezone('UTC'))  
print(now_utc.strftime(format))
```

```
now_asia = now_utc.astimezone(timezone('Asia/Kolkata'))  
print(now_asia.strftime(format))
```

```
2022-10-07 16:36:09 UTC+0000  
2022-10-07 22:06:09 IST+0530
```