Assignment -1

Assignment Date	19 September 2022
Student Name	Rithika S
Student Roll Number	737819ECR142
Maximum Marks	2 Marks

Question-1:

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

app.py

```
from flask import Flask, request, render_template


app = Flask(__name__)

app.config['DEBUG'] = True


@app.route('/login', methods=("POST", "GET"))

def login():

   name = request.form['name']

   phone = request.form['phone']

   email = request.form['email']

   return f'Name: {name}<br/>Register mobile num: {phone}<br/>Email: {email}'


@app.route('/')

def home():

   return render_template('index.html')


if __name__ == '__main__':

   app.run()
```

templates/index.html

```
<!DOCTYPE html>

<html lang="en">



<head>

   <meta charset="UTF-8">

   <meta http-equiv="X-UA-Compatible" content="IE=edge">

   <meta name="viewport" content="width=device-width, initial-scale=1.0">

   <title>Registration Page</title>

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

</head>



<body>

   <div class="login-page">

       <div class="form">

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

               <input name="name" type="text" placeholder="name" />

               <input name="email" type="email" placeholder="email" />

               <input name="phone" type="tel" placeholder="phone" />

               <button>create</button>

           </form>

       </div>

   </div>

</body>



</html>
```

static/css/index.css

```
@import url(https://fonts.googleapis.com/css?family=Roboto:300);


.login-page {
 width: 360px;
 padding: 8% 0 0;
 margin: auto;
}
.form {
 position: relative;
 z-index: 1;
 background: #FFFFFF;
 max-width: 360px;
 margin: 0 auto 100px;
 padding: 45px;
 text-align: center;
 box-shadow: 0 0 20px 0 rgba(0, 0, 0, 0.2), 0 5px 5px 0 rgba(0, 0, 0, 0.24);
}
input {
 font-family: "Roboto", sans-serif;
 outline: 0;
 background: #f2f2f2;
 width: 100%;
 border: 0;
 margin: 0 0 15px;
 padding: 15px;
 box-sizing: border-box;
```

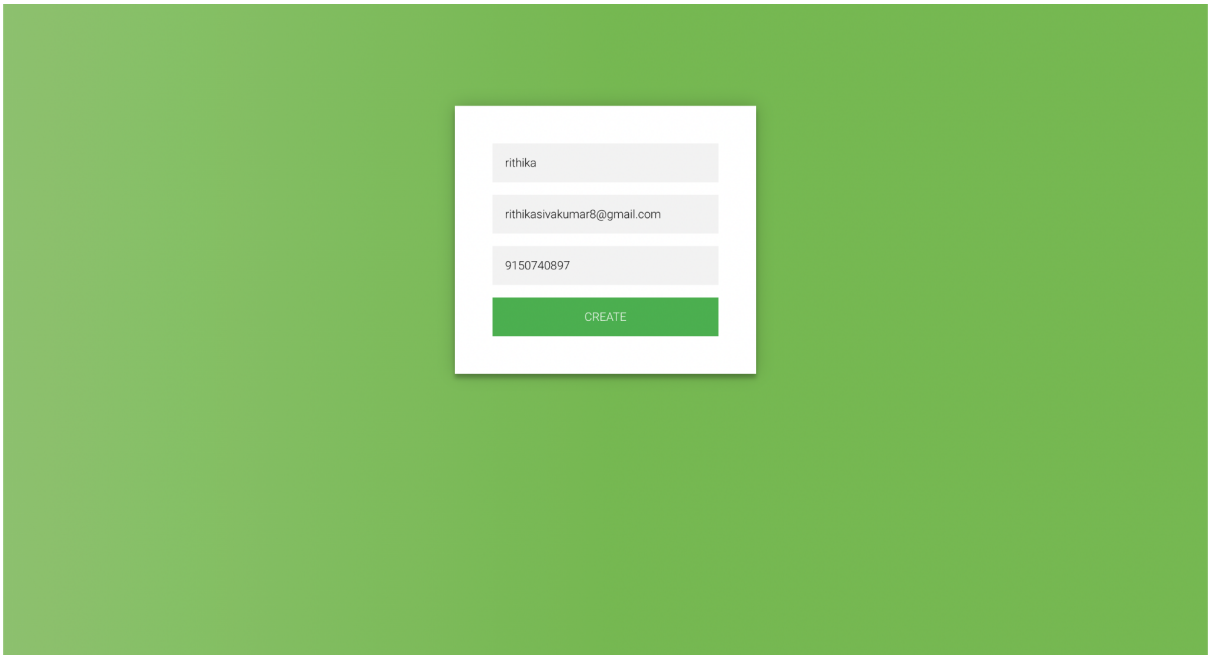
```
 font-size: 14px;
}
button {
 font-family: "Roboto", sans-serif;
 text-transform: uppercase;
 outline: 0;
 background: #4CAF50;
 width: 100%;
 border: 0;
 padding: 15px;
 color: #FFFFFF;
 font-size: 14px;
 -webkit-transition: all 0.3 ease;
 transition: all 0.3 ease;
 cursor: pointer;
}
.form button:hover,.form button:active,.form button:focus {
 background: #43A047;
}

body {
 background: #76b852;
 background: rgb(141,194,111);
 background: linear-gradient(90deg, rgba(141,194,111,1) 0%, rgba(118,184,82,1)
50%);
 font-family: "Roboto", sans-serif;
 -webkit-font-smoothing: antialiased;
```

```
 -moz-osx-font-smoothing: grayscale;

}
```

Output



Name: rithika
Register mobile num: 9150740897
Email: rithikasivakumar8@gmail.com

Question-2:

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

app.py

```
from flask import Flask

from datetime import datetime
import pytz
from color_convert import color
import numpy as np
import pandas as pd

app = Flask(__name__)
app.config['DEBUG'] = True

@app.route("/", methods=["GET"])
def register():
    # DateTime & pytz
    print('----------------------------')
    time = datetime.now(tz=pytz.timezone('Asia/Kolkata'))
    print(time)

    # color_convert
    print('----------------------------')
    rgbValue = color.hex_to_rgb("#fff000")
    print(rgbValue)

    # Numpy
    print('----------------------------')
    a = [(1, 2), [3, 4, (5)], (6, 7, 8)]
    b = np.array(a, dtype=object)
    print(b)

    # Pandas
    print('----------------------------')
    dates = pd.date_range("20130101", periods=6)
    print(dates)
```

```
    return 'Packages used<br/>Pandas Numpy DateTime PYTZ Color_Convert'

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

Output:

Terminal

```
---------------------------
2022-10-17 22:16:26.839397+05:30
---------------------------
rgb(255,240,0)
---------------------------
[(1, 2) list([3, 4, 5]) (6, 7, 8)]
---------------------------
DatetimeIndex(['2013-01-01', '2013-01-02', '2013-01-03', '2013-01-04',
               '2013-01-05', '2013-01-06'],
              dtype='datetime64[ns]', freq='D')
127.0.0.1 - - [17/Oct/2022 22:16:26] "GET / HTTP/1.1" 200 -
```

Browser