

# Assignment-2

-Name: Sangamitra S

-Reg No: 711119104052

1. Create form of type input text, email, password, radio button text Area, drop down and navigate to success page and display files of form in table (CSS, HTML).

## PROGRAM:

```
<html>
<head>
<title> Input Form
</title>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="stylesheet" href="style.css">

<body>
<div class="login-box">
<h2>Login</h2>
<form>
<table align="center" style="padding-top:50px;">
<tr>
<td>
<label>Name<span style="color:red;"> * </span></label></td>
<td><input type="text" name="" required=""></td>
</tr>

<tr>
<td><label>Qualification</label></td>
<td><select name="qualify" id="qualify" style="width:180px;" >
<option value="">--select qualification--</option>
<option value="tenth">10th</option>
<option value="twelvth">12th</option>
<option value="diploma">Diploma</option>
<option value="bachelor">Bachelor's Degree</option>
<option value="master">Master's Degree</option>
</select></td>
</tr>

<tr>
```

```

<td> <label>Age</label></td>
<td> <input type="radio" name="age10" required="" ><span style="color:#fff">10-
20</span>&nbsp;<input type="radio" name="age20" required="" ><span
style="color:#fff">20-30</span>&nbsp;<input type="radio" name="age30"
required=""><span style="color:#fff">30-40</span></td>
</tr>

<tr>
<td><label>Email<span style="color:red;"> * </span></label></td>
<td> <input type="email" name="email" required=""></td>
</tr>

<tr>
<td></td>
<td><button type="submit" name="submit" class="btn"
value="submit">submit</button></td>
</tr>

</table>
</form>
</div>
</body>
</html>

```

Output:

**Login**

Name \*

Qualification

Age ☐ 10-20 ☒ 20-30 ☐ 30-40

Email \*

2. For CSS create external style sheet for above task (separate CSS file and link that in html).

```
tr,td {
    padding:10px;
    border-radius:3px;
}
body {

    font-family: sans-serif;

}

.login-box {
    position: absolute;
    top: 50%;
```

```

left: 50%;
width: 500px;
padding: 40px;
color: #000;
transform: translate(-50%, -50%);
background: rgba(0,0,0,.5);
box-sizing: border-box;
box-shadow: 0 15px 25px rgba(0,0,0,.6);
border-radius: 10px;
}
.login-box h2 {
margin: 0 0 30px;
color: #fff;
text-align: center;
text-shadow: 1px 2px #03e9f4;
}
.login-box label {
margin: 0 0 30px;
color: #fff;
text-align: center;
}

.login-box input {
color: #fff;
padding: 5px;
}

.login-box select {
padding: 5px;
}

.btn {
width: 50%;
background: transparent;
border: 2px solid #03e9f4;
color: #fff;
margin: 18px 0 9px 0;
padding: 5px 5px 5px 5px;
border-radius: 5px;
box-shadow: 2px 3px #03e9f4;
}
}

```

3. Create sample program for Flask HTTP methods (list or map and Perform operations of PUT, GET, DELETE and POST).

Program:

```
from flask import Flask, render_template, request, flash, redirect, url_for
import sqlite3 as sql

app = Flask(__name__)
app.secret_key='admin123'

con=sql.connect("db_web.db")
con.execute("CREATE TABLE IF NOT EXISTS users(pid INTEGER PRIMARY KEY ,UNAME
TEXT,CONTACT TEXT)")
con.close()
app=Flask(__name__)

@app.route("/")
@app.route("/index")
def index():
    con=sql.connect("db_web.db")
    con.row_factory=sql.Row
    cur=con.cursor()
    cur.execute("select * from users")
    data=cur.fetchall()
    return render_template("index.html",datas=data)

@app.route("/add_user",methods=['POST','GET'])
def add_user():
    if request.method=='POST':
        uname=request.form['uname']
        contact=request.form['contact']
        con=sql.connect("db_web.db")
        cur=con.cursor()
        cur.execute("insert into users(UNAME,CONTACT) values
(?,?)",(uname,contact))
        con.commit()
        # flash('User Added','success')
        return redirect(url_for("index"))
    return render_template("add_user.html")

@app.route("/edit_user/<string:uid>",methods=['POST','GET'])
def edit_user(uid):
    if request.method=='POST':
        uname=request.form['uname']
        contact=request.form['contact']
        con=sql.connect("db_web.db")
        cur=con.cursor()
        cur.execute("update users set UNAME=?,CONTACT=? where
pid=?",(uname,contact,uid))
```

```

        con.commit()
        # flash('User Updated','success')
        return redirect(url_for("index"))
con=sql.connect("db_web.db")
con.row_factory=sql.Row
cur=con.cursor()
cur.execute("select * from users where pid=?", (uid,))
data=cur.fetchone()
return render_template("edit_user.html",datas=data)

@app.route("/delete_user/<string:uid>",methods=['GET'])
def delete_user(uid):
    con=sql.connect("db_web.db")
    cur=con.cursor()
    cur.execute("delete from users where pid=?", (uid,))
    con.commit()
    # flash('User Deleted','warning')
    return redirect(url_for("index"))

if __name__=='__main__':
    app.secret_key='admin123'
    app.config['SESSION_TYPE'] = 'filesystem'
    app.config['SECRET_KEY'] = 'admin123'
    app.run(debug=True)

```

Flask HTTP method :

1. Login page

## Add user

User Name

Contact

Submit

2. Data stored that can be edit, delete and add another id.

---

Flask CRUD Application				
				<a href="#">+Add User</a>
SNo	Name	Contact	Edit	Delete
1	sangamitra	+91 9876512345	<a href="#">Edit</a>	<a href="#">Delete</a>

3. Can Edit the Use ID

## Edit user

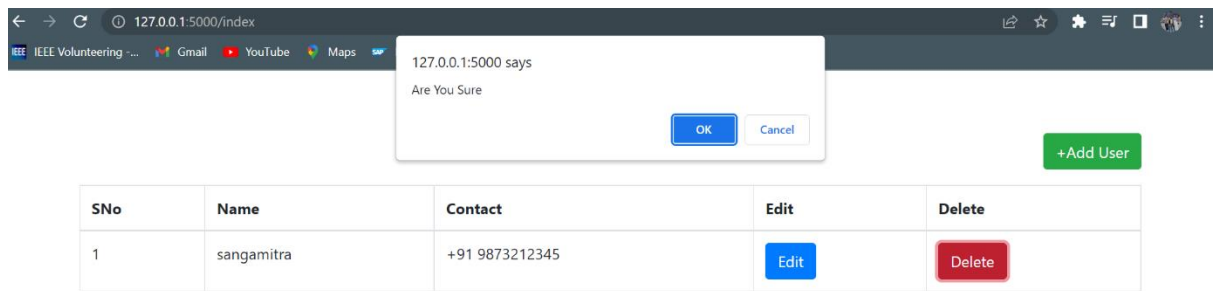
User Name

Contact

4. Can see the changes

Flask CRUD Application				
				<a href="#">+Add User</a>
SNo	Name	Contact	Edit	Delete
1	sangamitra	+91 9873212345	<a href="#">Edit</a>	<a href="#">Delete</a>

## 5. If needed can delete



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/index". A confirmation dialog box is open, asking "Are You Sure" with "OK" and "Cancel" buttons. Below the dialog, there is a table with user information and a "+Add User" button.

SNo	Name	Contact	Edit	Delete
1	sangamitra	+91 9873212345	<a href="#">Edit</a>	<a href="#">Delete</a>

[+Add User](#)

## 6. It will be deleted from DB

### Flask CRUD Application

					<a href="#">+Add User</a>
SNo	Name	Contact	Edit	Delete	