

Assignment -2

Assignment Date	19 September 2022
Student Name	Deepak Kumar R
Student Roll Number	711119104020
Maximum Marks	2 Marks

Question-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).

Solution:

```
<!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>JOBPORTAL | LOGIN</title>
  <!-- favicon -->
  <!-- <link rel="shortcut icon" href="/assets/img/favicon.ico" type="image/x-icon"> -->
  <!-- <link rel="icon" href="/assets/img/favicon.icos" type="image/x-icon"> -->
  <link rel="icon" type="image/png" sizes="16x16" href="/assets/img/favicon-32x32.png">
  <!-- bootstrap css cdn -->
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z"
crossorigin="anonymous">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.css">
  <!-- css stylesheet -->
  <link rel="stylesheet" href="css/style.css">
  <!-- font styles cdn -->
  <link rel="preconnect" href="https://fonts.gstatic.com">
  <link href="https://fonts.googleapis.com/css2?family=Alegreya&display=swap"
rel="stylesheet">
  <link href="https://fonts.googleapis.com/css2?family=Alegreya:wght@600&display=swap"
rel="stylesheet">
</head>
<body>
  <!-- bootstrap navbar -->
  <div class="logo mt-3 text-center">
    <a class="main-logo-img mt-5" href="#">
      <!-- <a class="navbar-brand" href="index.html">JobPortal</a> -->
    </a>
```

```

</div>
<!-- navbar ends -->
<!-- Login form -->
<div class="login text-center mt-5">
  <h2> Login Form </h2>
  <form action="/login" method="post">
    <div class="msg">{{ msg }}</div>
    <input type="text" name="username" placeholder="Enter Your Username" id="username"
required></br></br>
      <input type="password" name="password" placeholder="Enter Your
Password" id="password" required></br></br>
    </br>
    </br>
    <button type="submit" id="button" class="btn btn-primary"> Login </button>
  </form>
</div>

<div class="note mt-3 text-center"> <!--Register form -->
  <p> Don't have an account yet? Click here to <a href="register">register! </a> </p>
</div>
</body>
</html>

```

Output:

sheep-logo

Login Form

{{ msg }}

Don't have an account yet? Click here to [register!](#)

Question-2:

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

Solution:

```
* {  
    font-family: 'Alegreya', serif !important;  
}
```

```
/* // Small devices (landscape phones, 576px and up) */
```

```
@media (min-width: 576px) {
```

```
}
```

```
/* // Medium devices (tablets, 768px and up) */
```

```
@media (min-width: 768px) {
```

```
}
```

```
/* // Large devices (desktops, 992px and up) */
```

```
@media (min-width: 992px) {
```

```
    .navbar {
```

```
        padding-top:15px;
```

```
        padding-bottom:15px;
```

```
        background-color: white;
```

```
    }
```

```
    .navbar-brand{
```

```
        padding-left: 5px;
```

```
    }
```

```
    .navbar-nav {
```

```
        margin-left: 30px;
```

```
    }
```

```
    .nav-item {
```

```
padding-left:5px;
}
.donate-sponsor{
margin-right:10px;
}
#donate,#sponsor{
margin:5px;
padding: 5px 15px 5px 15px;
}
.homepage-header{
background: url('../img/group.JPG');
background-size: cover;
background-position: center top;
padding: 0;
position: relative;
width: 100%;
overflow: hidden;
display: -webkit-flex;
display: -ms-flexbox;
display: flex;
height: 85vh;
}
.home-for-children {
background-color: #ffeeba;
margin-top:0px;
padding-top:10px;
padding-bottom:30px;
}
.home-for-children h2 {
line-height: 2.5rem !important;
letter-spacing: 3px;
```

```
    font-weight: 600;
}
.home-for-children h5 {
    line-height: 1.8rem;
}
.home-for-children .btn-success {
    padding: 8px 25px;
    font-size: large;
}
/* our focus */
.our-focus .container {
    margin-top: 3rem !important;
}

.our-focus .row {
    margin-top: 1.5rem !important;
}
.our-focus .card {
    border: none !important;
}
.our-focus #focus-first {
    margin-right: 80px;
}
.our-focus #focus-second {
    margin-right: 90px;
}
/* media */
.media .container {
    margin-top: 3rem !important;
}
.media .row {
```

```
        margin-top: 1.5rem !important;
    }
/* footer */
.row-initiative {
    margin-top: 10px;
}
.site-footer {
    margin-top: 30px;
    line-height: 24px;
}
.footer-links {
    padding-left: 0;
    list-style: none;
}
.footer-links li {
    display: block;
}
.footer-links.inline li {
    display: inline-block;
}
.footer-links li a {
    color: black;
}
.footer-logo img {
    width: 100px;
}
.social-icons {
    text-align: right;
    margin-left: 50px;
}
.social-icons li {
```

```

    list-style: none;

    display: inline-block;
}

.social-icons li a {

    border-radius: 50%;

    margin-left: 10px;
}

input {

    padding: 10px 20px;
}

#button {

    border: none;

    padding: 10px 20px;

    border-radius: 10px;

    animation: pulse 3s infinite ease-out;
}
}

/* // X-Large devices (large desktops, 1200px and up) */
@media (min-width: 1200px) {

    .navbar-brand{

        padding-left: 10px;
    }
}

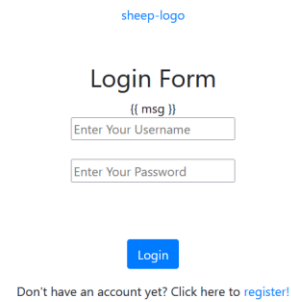
/* // XX-Large devices (larger desktops, 1400px and up) */
@media (min-width: 1400px) {

    .navbar-brand{

        padding-left: 15px;
    }
}

```

Output:



sheep-logo

Login Form

{{ msg }}

Don't have an account yet? Click here to [register!](#)

Question-3:

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

Solution:

```
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
import re
```

```
app = Flask(__name__)
```

```
app.secret_key = 'a'
```

```
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-
1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=SSL;SS
LServerCertificate:DigiCertGlobalRootCA;UID=gyj23042;PWD=a5CFymQzrGXGeOnC;", "", "")
```

```
@app.route('/')
```

```
@app.route('/login', methods=['GET', 'POST'])
```

```
def login():
```

```
    global userid
```

```
    msg = "
```

```
if request.method == 'POST' :
```

```
    username = request.form['username']
```

```
    password = request.form['password']
```

```
    sql = "SELECT * FROM users WHERE username =? AND password=?"
```

```
    stmt = ibm_db.prepare(conn, sql)
```



```

ibm_db.bind_param(stmt,1,username)
ibm_db.bind_param(stmt,2,password)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print (account)
if account:
    session['loggedin'] = True
    session['id'] = account['USERNAME']
    userid= account['USERNAME']
    session['username'] = account['USERNAME']
    msg = 'Logged in successfully !'

    msg = 'Logged in successfully !'
    return render_template('dashboard.html', msg = msg)
else:
    msg = 'Incorrect username / password !'
return render_template('login.html', msg = msg)

```

```

@app.route('/register', methods=['GET', 'POST'])
def registet():
    msg = ""
    if request.method == 'POST' :
        username = request.form['username']
        email = request.form['email']
        password = request.form['password']
        sql = "SELECT * FROM users WHERE username =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            msg = 'Account already exists !'
        elif not re.match(r'^@]+@[^@]+\.[^@]+', email):
            msg = 'Invalid email address !'
        elif not re.match(r'[A-Za-z0-9]+', username):
            msg = 'name must contain only characters and numbers !'
        else:
            insert_sql = "INSERT INTO users VALUES (?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, username)
            ibm_db.bind_param(prepare_stmt, 2, email)
            ibm_db.bind_param(prepare_stmt, 3, password)

```

```
        ibm_db.execute(prepare_stmt)
        msg = 'You have successfully registered !'
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg = msg)
```

```
if __name__ == '__main__':
    app.run(host='0.0.0.0')
```