**Assignment -1**

Dashboard

|  |  |
| --- | --- |
| Assignment Date | 10 October 2022 |
| Student Name | Yogendrakumar S |
| Student Roll Number | 2116190801201 |
| Maximum Marks | 2 Marks |

**Index page:**

<!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">

    <link rel="stylesheet" href="{{ url\_for('static', filename='style.css')}}">

    <title>Home</title>

</head>

<body>

    <h2 class="home">Home Page</h2>

    <div class="space">

    <a class="text" href="/about"> About </a>

    <a class="text" href="/signin"> Log In </a>

    <a class="text" href="/signup"> Sign Up </a>

    </div>

</body>

</html>

**Sign in page:**

<!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">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"

    integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">

    <link rel="stylesheet" href="{{ url\_for('static', filename='style.css')}}">

    <title>Sign In</title>

</head>

<body>

    <div class="space">

      <a class="text" href="/"> Home </a>

      <a class="text" href="/about"> About </a>

      <a class="text" href="/signup"> Sign Up </a>

    </div>

    <h2 class="home">Log In</h2>

 <div class="signin">

    <form>

      <div class="form-group">

          <label for="exampleInputEmail1">Email</label>

          <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter email">

        </div> <br>

        <div class="form-group">

            <label for="exampleInputEmail1">password</label>

            <input type="password" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter password">

          </div>

          <br>

        <button type="submit" class="btn btn-primary">Submit</button>

    </form>

</div>

</body>

</html>

**Sign up Page:**

<!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">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"

    integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">

    <link rel="stylesheet" href="{{ url\_for('static', filename='style.css')}}">

    <title>Sign Up</title>

</head>

<body>

    <div class="space">

    <a class="text" href="/"> Home </a>

    <a class="text" href="/about"> About </a>

    <a class="text" href="/signin"> Log In </a>

    </div>

    <h2 class="home">Sign Up</h2>

    <div class="signup">

    <form>

        <div class="form-group">

            <label for="Name">Name</label>

            <input class="form-control" type="text" name="Name" placeholder="Enter Name">

          </div><br>

      <div class="form-group">

          <label for="exampleInputEmail1">Email</label>

          <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter email">

        </div><br>

        <div class="form-group">

            <label for="exampleInputEmail1">password</label>

            <input type="password" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter password">

          </div><br>

          <div class="form-group">

            <label for="exampleInputEmail1">Re - password</label>

            <input type="password" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Enter password">

          </div><br>

        <button type="submit" class="btn btn-primary">Submit</button>

    </form>

    </div>

</body>

</html>

**Style code :**

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;500;600;700&display=swap');

\*{

font-family: 'Poppins', sans-serif;

top: 0;

margin: 0;

}

body{

background-image:url("https://cdn.discordapp.com/attachments/999321523577966613/1023800785215176865/img.png") ;

}

.home{

font-size: 25px;

text-align: center;

margin-top: 10px;

color: black;

}

.text{

text-decoration: none;

font-size: 20px;

color: rgb(128, 42, 207);

margin-left: 10px;

margin-right: 10px;

margin-top: 10px;

}

p{

margin-left: 25px;

margin-top: 10px;

color:rgb(122, 2, 42);

font-size: 20px;

}

.signin, .signup{

height: 50vh;

width: 50vw;

margin-top: 10px;

margin-left: 25px;

}

.space{

margin-top:15px;

text-align: right;

margin-right: 10px;

**App python :**

from flask import Flask

from flask import render\_template

app = Flask(\_name\_)

@app.route('/')

def main():

return render\_template('index.html')

@app.route('/about')

def about():

return render\_template('about.html')

@app.route('/signin')

def signin():

return render\_template('signin.html')

@app.route('/signup')

def signup():

return render\_template('signup.html')

if \_name\_ == "\_main\_":

app.run()

**Source file:**

from flask import Flask,redirect,url\_for,render\_template,request  
import ibm\_boto3  
from ibm\_botocore.client import Config, ClientError  
  
COS\_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud" # Current list avaiable at https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints  
COS\_API\_KEY\_ID = "xVWVt8tjo5lL5eY-jii6RAZu4IAfMrRQY2uaD0\_-9LZj"   
COS\_INSTANCE\_CRN = "crn:v1:bluemix:public:iam-identity::a/37641ef4661f44a7bfab12af5b4ba6fe::serviceid:ServiceId-dd1803b4-6727-46a8-b09c-1e921e8211c6"  
  
# Create resource  
cos = ibm\_boto3.resource("s3",  
 ibm\_api\_key\_id=COS\_API\_KEY\_ID,  
 ibm\_service\_instance\_id=COS\_INSTANCE\_CRN,  
 config=Config(signature\_version="oauth"),  
 endpoint\_url=COS\_ENDPOINT  
)  
  
app=Flask(\_name\_)  
  
def get\_item(bucket\_name, item\_name):  
 print("Retrieving item from bucket: {0}, key: {1}".format(bucket\_name, item\_name))  
 try:  
 file = cos.Object(bucket\_name, item\_name).get()  
 print("File Contents: {0}".format(file["Body"].read()))  
 except ClientError as be:  
 print("CLIENT ERROR: {0}\n".format(be))  
 except Exception as e:  
 print("Unable to retrieve file contents: {0}".format(e))  
  
  
def get\_bucket\_contents(bucket\_name):  
 print("Retrieving bucket contents from: {0}".format(bucket\_name))  
 try:  
 files = cos.Bucket(bucket\_name).objects.all()  
 files\_names = []  
 for file in files:  
 files\_names.append(file.key)  
 print("Item: {0} ({1} bytes).".format(file.key, file.size))  
 return files\_names  
 except ClientError as be:  
 print("CLIENT ERROR: {0}\n".format(be))  
 except Exception as e:  
 print("Unable to retrieve bucket contents: {0}".format(e))  
  
def delete\_item(bucket\_name, object\_name):  
 try:  
 cos.delete\_object(Bucket=bucket\_name, Key=object\_name)  
 print("Item: {0} deleted!\n".format(object\_name))  
 except ClientError as be:  
 print("CLIENT ERROR: {0}\n".format(be))  
 except Exception as e:  
 print("Unable to delete object: {0}".format(e))  
  
def multi\_part\_upload(bucket\_name, item\_name, file\_path):  
 try:  
 print("Starting file transfer for {0} to bucket: {1}\n".format(item\_name, bucket\_name))  
 # set 5 MB chunks  
 part\_size = 1024 \* 1024 \* 5  
  
 # set threadhold to 15 MB  
 file\_threshold = 1024 \* 1024 \* 15  
  
 # set the transfer threshold and chunk size  
 transfer\_config = ibm\_boto3.s3.transfer.TransferConfig(  
 multipart\_threshold=file\_threshold,  
 multipart\_chunksize=part\_size  
 )  
  
  
 with open(file\_path, "rb") as file\_data:  
 cos.Object(bucket\_name, item\_name).upload\_fileobj(  
 Fileobj=file\_data,  
 Config=transfer\_config  
 )  
  
 print("Transfer for {0} Complete!\n".format(item\_name))  
 except ClientError as be:  
 print("CLIENT ERROR: {0}\n".format(be))  
 except Exception as e:  
 print("Unable to complete multi-part upload: {0}".format(e))  
  
  
@app.route('/')  
def index():  
 files = get\_bucket\_contents('jeeva-bucket1')  
   
 return render\_template('index.html', files = files)  
  
@app.route('/uploader', methods = ['GET', 'POST'])  
def upload():  
 if request.method == 'POST':  
 bucket=request.form['bucket']  
 name\_file=request.form['filename']  
 f = request.files['file']  
 multi\_part\_upload(bucket,name\_file,f.filename)  
 return 'file uploaded successfully'  
   
 if request.method == 'GET':  
 return render\_template('upload.html')  
   
@app.route('/deletefile', methods = ['GET', 'POST'])  
def deletefile():  
 if request.method == 'POST':  
 bucket=request.form['bucket']  
 name\_file=request.form['filename']  
   
 delete\_item(bucket,name\_file)  
 return 'file deleted successfully'  
   
 if request.method == 'GET':  
 return render\_template('delete.html')  
  
if \_name=='main\_':  
 app.run(debug=True)