ASSIGNMENT 3

App.py

```
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"
COS_API_KEY_ID=" "
COS_INSTANCE_CRN=""
# Create resource https://s3.ap.cloud-object-storage.appdomain.cloud
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
```

```
)
        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB
        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('flaskapp123')
    return render_template('index.html', files = files)
@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')
```

```
@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 <a href="/">GO to Home</a>'
   if request.method == 'GET':
      return render_template('upload.html')
if __name__=='__main__':
    app.run(host='0.0.0.0',port=8080,debug=True)
```

delete.html

```
<html>
   <body>
         <a href="/">HOME</a>
         <a href="/uploader">Upload </a>
         <a href="/deletefile">Delete </a>
         <br><hr>
<h1>IBM Object Storage</h1>
      <form action = "/deletefile" method = "POST" >
               <input type = "text" placeholder="Enter bucket name" name = "bucket" />
               <br>
               <br>
               <input type = "text" placeholder="Enter file name" name = "filename" />
```

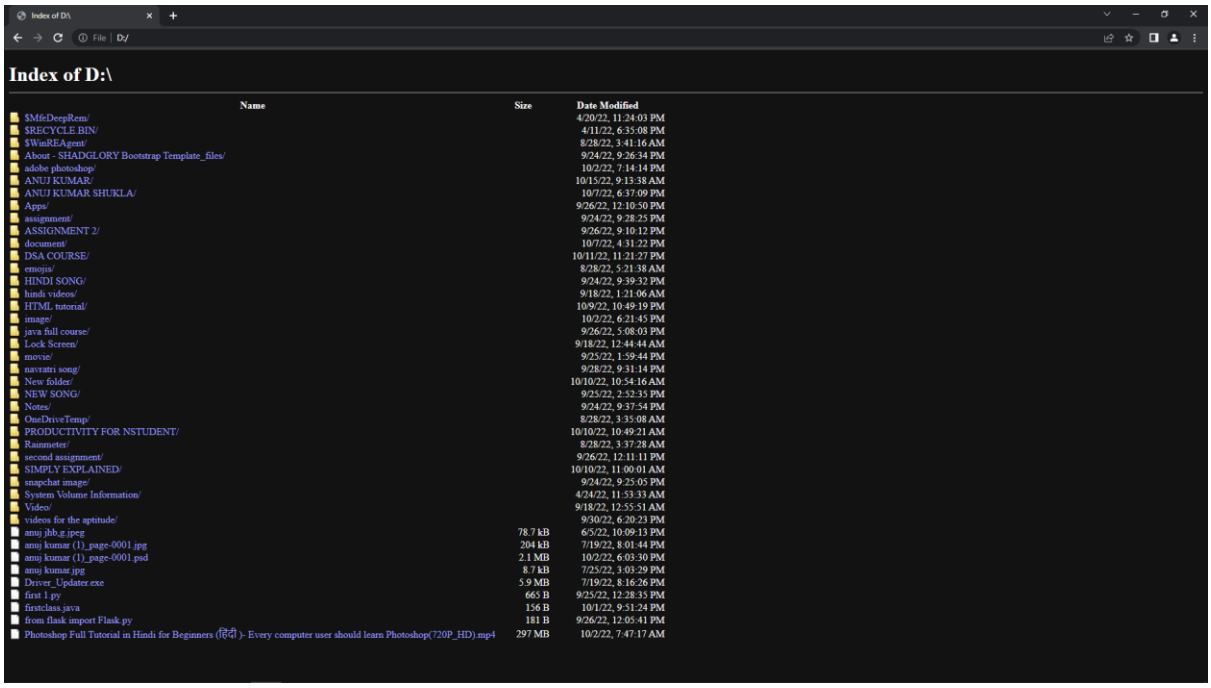
```
<br>
                    <br>
        <input type = "submit"/>
      </form>
   </body>
</html>
```

Index.html

```
<a href="/">HOME</a>
<a href="/uploader">Upload</a>
<a href="/deletefile">Delete</a>
<br><hr>
<h1>IBM Object Storage</h1>
<!doctype html>
<html>
   <body>
      {% for row in files %}
         <div style="border: 1px solid #EFEFEF;margin:10px;">
            <h3>Filename : {{row}} </h3>
            <img src="https://flaskapp123.s3.jp-tok.cloud-object-
storage.appdomain.cloud/{{row}}" width="150px"></td>
         </div>
      {% endfor %}
   </body>
</html>
```

UPLOAD.HTML

```
<html>
  <body>
<a href="/">HOME</a>
<a href="/uploader">Upload </a>
<a href="/deletefile">Delete </a>
<br><hr>
<h1>IBM Upload File</h1>
     <form action = "/uploader" method = "POST"
       enctype = "multipart/form-data">
                    <input type = "text" placeholder="Enter bucket name" name = "bucket" />
                    <br>
                    <br>
                    <input type = "text" placeholder="Enter file name" name = "filename" />
                    <br>
                    <br>
       <input type = "file" name = "file" />
                    <br>
                    <br>
       <input type = "submit"/>
     </form>
  </body>
</html>
```

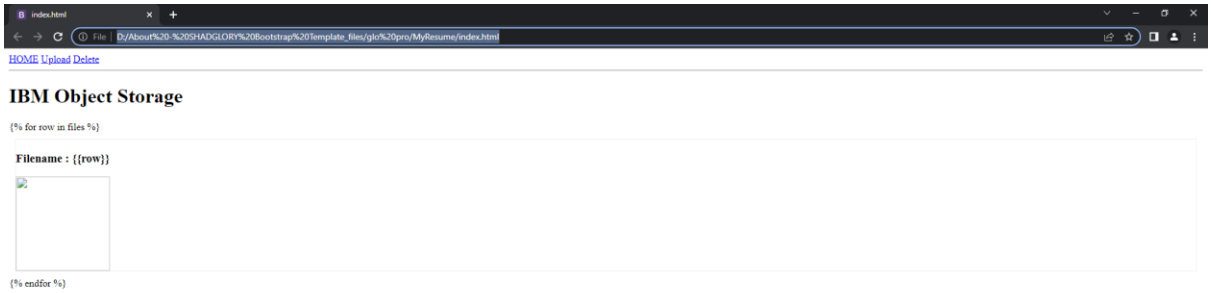


Index of D:\

File | D:/

Index of D:\

Name	Size	Date Modified
$MfeDeepRem/		4/20/22, 11:24:03 PM
$RECYCLE.BIN/		4/11/22, 6:35:08 PM
$WinREAgent/		8/28/22, 3:41:16 AM
About - SHADGLORY Bootstrap Template_files/		9/24/22, 9:26:34 PM
adobe photoshop/		10/2/22, 7:14:14 PM
ANUJ KUMAR/		10/15/22, 9:13:38 AM
ANUJ KUMAR SHUKLA/		10/7/22, 6:37:09 PM
Apps/		9/26/22, 12:10:50 PM
assignment/		9/24/22, 9:28:25 PM
ASSIGNMENT 2/		9/26/22, 9:10:12 PM
document/		10/7/22, 4:31:22 PM
DSA COURSE/		10/11/22, 11:21:27 PM
emojis/		8/28/22, 5:21:38 AM
HINDI SONG/		9/24/22, 9:39:32 PM
hindi videos/		9/18/22, 1:21:06 AM
HTML tutorial/		10/9/22, 10:49:19 PM
image/		10/2/22, 6:21:45 PM
java full course/		9/26/22, 5:08:03 PM
Lock Screen/		9/18/22, 12:44:44 AM
movie/		9/25/22, 1:59:44 PM
navratri song/		9/28/22, 9:31:14 PM
New folder/		10/10/22, 10:54:16 AM
NEW SONG/		9/25/22, 2:52:35 PM
Notes/		9/24/22, 9:37:54 PM
OneDriveTemp/		8/28/22, 3:35:08 AM
PRODUCTIVITY FOR NSTUDENT/		10/10/22, 10:49:21 AM
Rainmeter/		8/28/22, 3:37:28 AM
second assignment/		9/26/22, 12:11:11 PM
SIMPLY EXPLAINED/		10/10/22, 11:00:01 AM
snapchat image/		9/24/22, 9:25:05 PM
System Volume Information/		4/24/22, 11:53:33 AM
Video/		9/18/22, 12:55:51 AM
videos for the aptitude/		9/30/22, 6:20:23 PM
anuj jhb.g.jpeg	78.7 kB	6/5/22, 10:09:13 PM
anuj kumar (1)_page-0001.jpg	204 kB	7/19/22, 8:01:44 PM
anuj kumar (1)_page-0001.psd	2.1 MB	10/2/22, 6:03:30 PM
anuj kumar.jpg	8.7 kB	7/25/22, 3:03:29 PM
Driver_Updater.exe	5.9 MB	7/19/22, 8:16:26 PM
first 1.py	665 B	9/25/22, 12:28:35 PM
firstclass.java	156 B	10/1/22, 9:51:24 PM
from flask import Flask.py	181 B	9/26/22, 12:05:41 PM
Photoshop Full Tutorial in Hindi for Beginners (हिंदी) - Every computer user should learn Photoshop(720P_HD).mp4	297 MB	10/2/22, 7:47:17 AM



index.html

File | D:/About%20-%20SHADGLORY%20Bootstrap%20Template_files/glo%20pro/MyResume/index.html

HOME Upload Delete

IBM Object Storage

{% for row in files %}

Filename : {{row}}

{% endfor %}