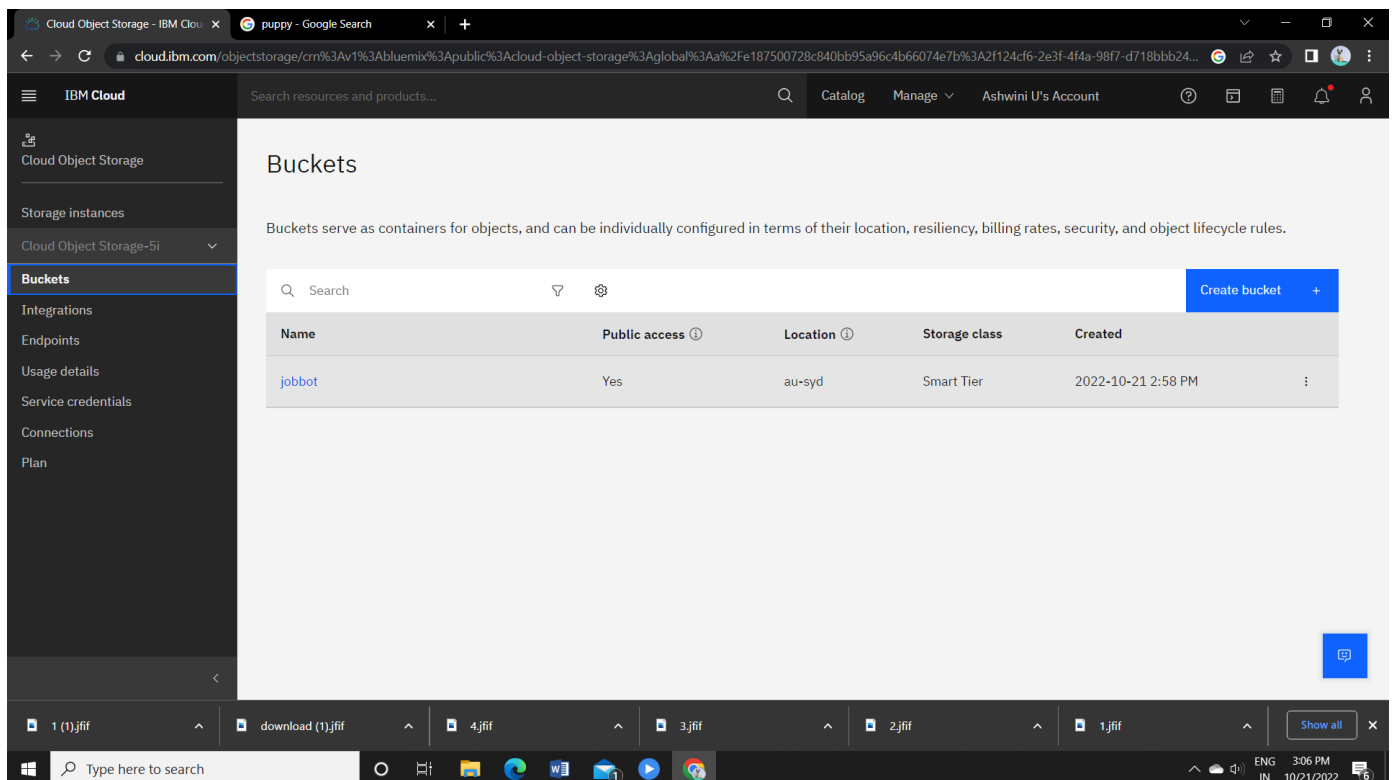


Assignment-3

| | |
|--------------|---------------------------------------|
| Date | 10 October 2022 |
| Team ID | PNT2022TMID30531 |
| Project Name | Skill and Job Recommender Application |

1. CREATE A BUCKET IN IBM OBJECT STORAGE.



**Upload an 5 images to ibm object storage and make it public.
Write html code todisplaying all the 5 images.**

Cloud Object Storage - IBM Cloud

puppy - Google Search

cloud.ibm.com/objectstorage/cm%3Av1%3Abluemix%3Apublic%3Acloud-object-storage%3Aglobal%3AA%2Fe187500728c840bb95a96c4b66074e7b%3A2f124cf6-2e3f-4f4a-98f7-d718bbb24...

IBM Cloud

Search resources and products...

CatalogManageAshwini U's Account

Storage / Cloud Object Storage-5i /

jobbot

TransfersDetailsActions...

ObjectsConfigurationPermissions

If you're seeing more usage than expected, versions count towards your usage or you may have incomplete uploads [Learn more](#)

Prefix filter

☐

Object name

Archived ⓘ

Size

Last modified

☐

1.jfif

5.3 KB

2022-10-21 3:03 PM

☐

2.jfif

5.4 KB

2022-10-21 3:04 PM

☐

3.jfif

10.0 KB

2022-10-21 3:04 PM

☐

4.jfif

6.0 KB

2022-10-21 3:04 PM

☐

download (1).jfif

6.6 KB

2022-10-21 3:03 PM

Drag and drop files (objects) here or click to upload

1 (1).jfifdownload (1).jfif4.jfif3.jfif2.jfif1.jfifShow all

Type here to search

ENG 3:07 PM IN 10/21/2022

Cloud Object Storage - IBM Cloud

puppy - Google Search

cloud.ibm.com/objectstorage/cm%3Av1%3Abluemix%3Apublic%3Acloud-object-storage%3Aglobal%3AA%2Fe187500728c840bb95a96c4b66074e7b%3A2f124cf6-2e3f-4f4a-98f7-d718bbb24...

IBM Cloud

Search resources and products...

CatalogManageAshwini U's Account

console or to list buckets using the REST API.

Access policies

Public access

Warning: Creating Public access to this bucket will allow anyone to access the bucket. To learn more, click here.

Access policy update

Access group policy created

A new access policy for this bucket was created for the group: Public Access. To delete/edit go to the IAM console.

Status: Enabled

Role for this policy: Content Reader

As a Content Reader, one can read and list objects in the bucket.

Create access policy

Context-based restrictions

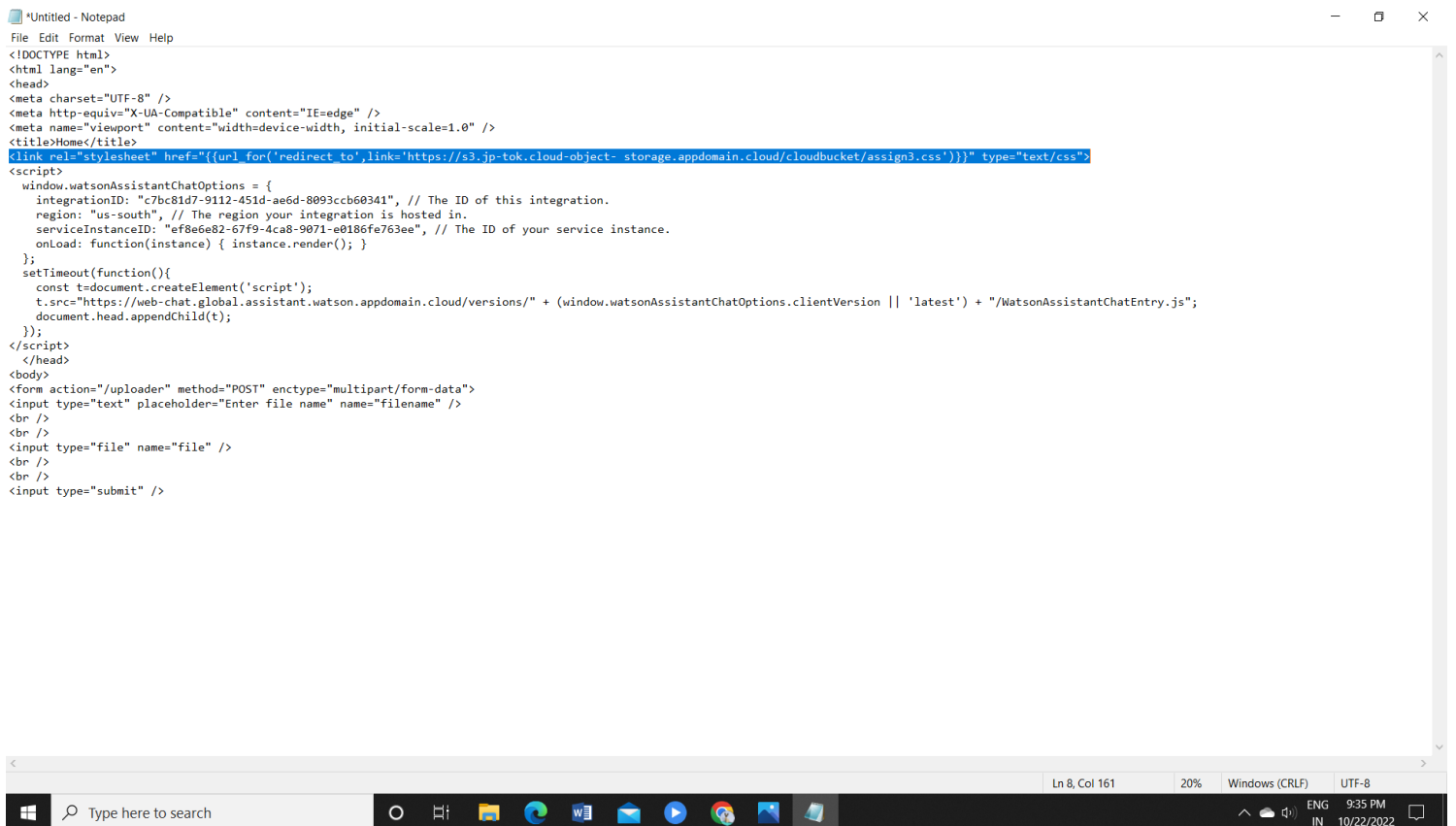
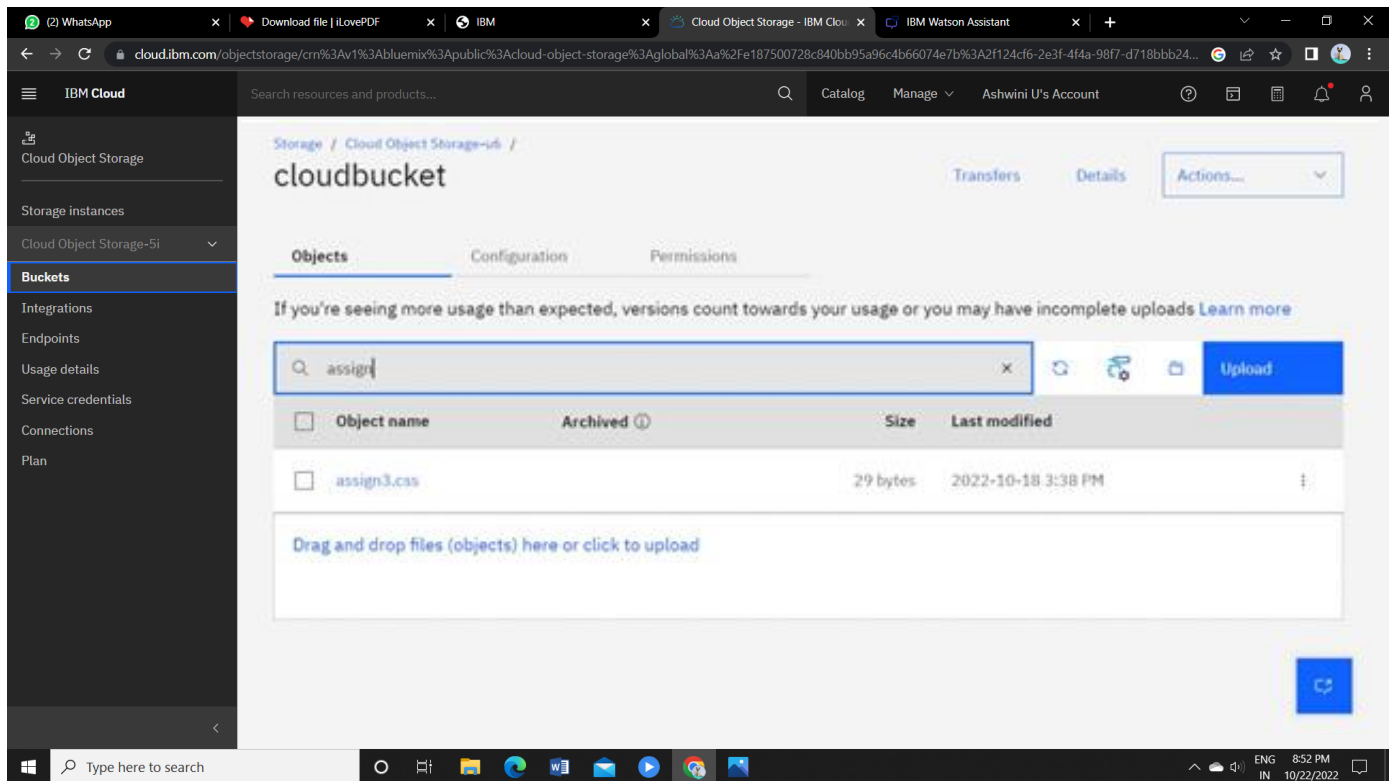
Firewall (legacy)

1 (1).jfifdownload (1).jfif4.jfif3.jfif2.jfif1.jfifShow all

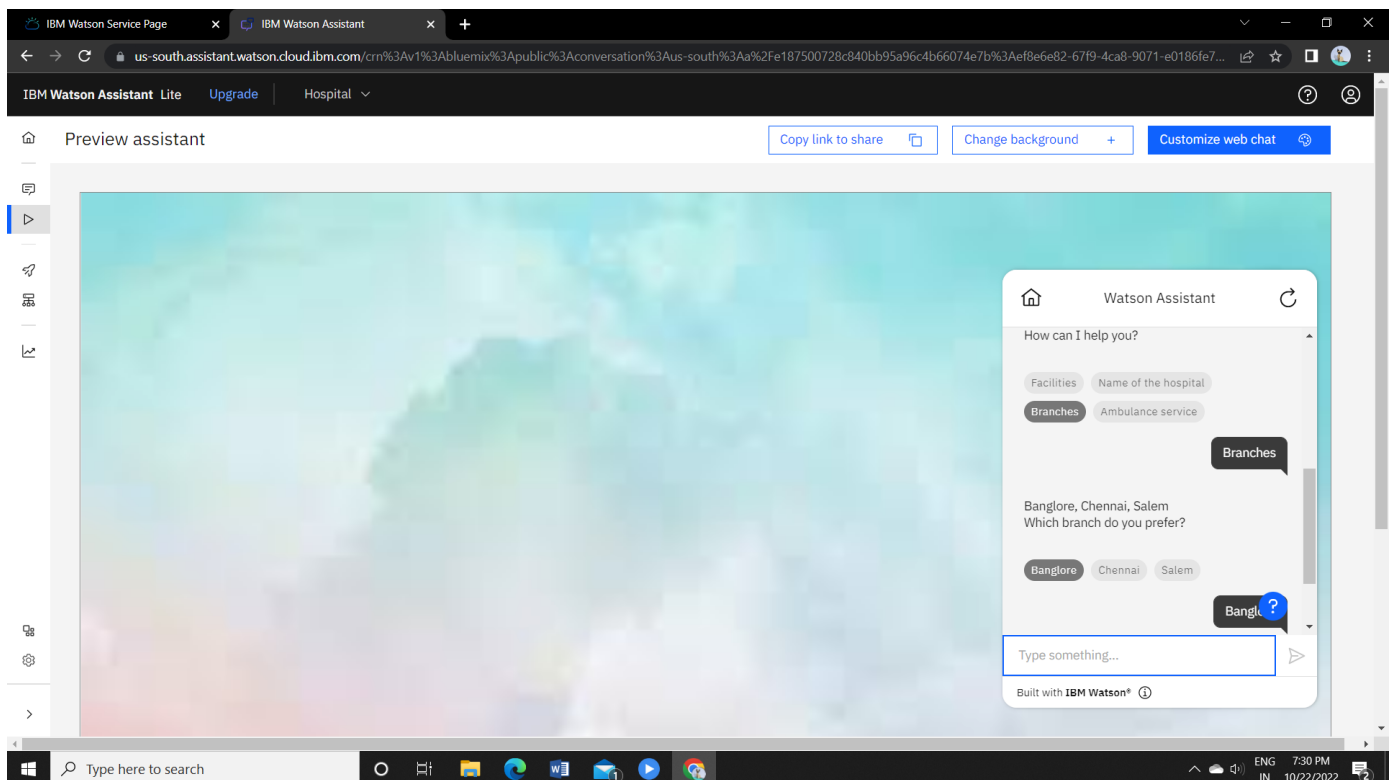
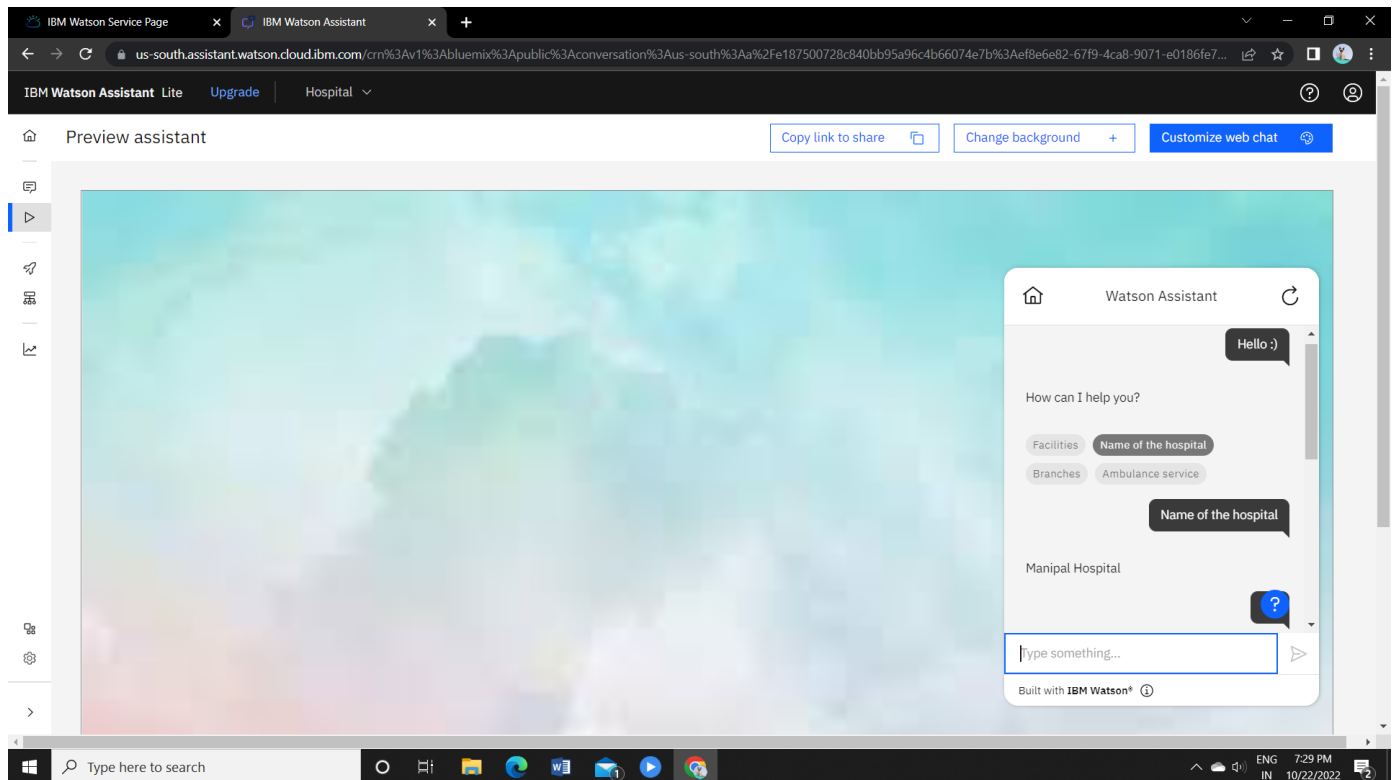
Type here to search

ENG 3:06 PM IN 10/21/2022

2. Upload a css page to the object storage and use the same page in your HTML code.



3. Design a chatbot using IBM Watson assistant for hospital.



Web URL for Assistant:

<https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fus-south.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-ef8e6e82-67f9-4ca8-9071-e0186fe763ee%3A%3Ace8f7f9b-be52-47ec-a9c0-9fc243082847&integrationID=c7bc81d7-9112-451d-ae6d-8093ccb60341®ion=us-south&serviceInstanceID=ef8e6e82-67f9-4ca8-9071-e0186fe763ee>

4. Create Watson assistant service with 10 steps and use 3 conditions in it. Load that script in HTML page.

The screenshot displays the IBM Watson Assistant console interface. On the left, a workflow editor shows 10 steps:

- Step 7: Confirmation button. Action: Re-ask previous step(s).
- Step 8: Free text input. Action: Continue to next step.
- Step 9: Selection of specialist (ENT, Cardiologist, +2). Action: Continue to next step.
- Step 10: Free text input. Action: End the action.

The main area shows the 'Assistant says' section with the text: 'Kindly mention the specialist you need to contact.' Below this, the 'And then' section shows the action 'End the action'.

On the right, a 'Preview' window shows a chat interface with the following content:

- Branches: Bangalore, Chennai, Salem.
- Question: Which branch do you prefer?
- Buttons: Bangalore, Chennai, Salem.
- Selected: Bangalore.
- Message: Hello :) recognized. There are no additional steps for this action. Add a new step or end the action.
- Warning: Save your action. Save to see your changes appear below.

The bottom of the screenshot shows the Windows taskbar with various application icons and the system clock indicating 4:29 PM on 10/21/2022.

Included 3 conditions in steps:

This screenshot shows the IBM Watson Assistant configuration interface. On the left, a list of steps is visible, with Step 3 highlighted. Step 3 is titled 'Branches' and contains the text 'Banglore, Chennai, Salem Which branch do you prefer?'. Below the text are buttons for 'Banglore' and 'Chennai', and a '+1' button. The 'Continue to next step' button is also visible. In the center, the 'Step 3 is taken' section shows a condition: '1. How can I help yo... is Branches'. The 'Assistant says' section contains the text 'Banglore, Chennai, Salem Which branch do you prefer?'. On the right, a 'Preview' window shows a chat conversation with the assistant asking for a branch preference and the user responding with 'Banglore'.

This screenshot shows the IBM Watson Assistant configuration interface, focusing on Step 7. Step 7 is titled 'Bed availability' and contains the text 'We have 67 beds available. Are you in need of AC beds?'. Below the text is a 'Confirmation' button. The 'Continue to next step' button is also visible. In the center, the 'Step 7 is taken' section shows a condition: '5. What kind of facili... is Bed availability'. The 'Assistant says' section contains the text 'We have 67 beds available. Are you in need of AC beds?'. On the right, a 'Preview' window shows a chat conversation with the assistant asking about the need for AC beds and the user responding with 'Banglore'.

IBM Watson Assistant interface showing a workflow for a hospital assistant.

Workflow Steps:

- Step 5: **Bed availability**. Message: "We have 67 beds available. Are you in need of AC beds?". Action: **Confirmation**. Option: **Re-ask previous step(s)**.
- Step 8: **Parking facility**. Message: "Wallet parking system is available.". Action: **Free text**. Option: **Continue to next step**.
- Step 9: **Which specialist you are in need of?**. Options: **ENT**, **Cardiologist**, **+ 2**. Option: **Continue to next step**.
- Step 9: **Others**. Message: "Kindly mention the specialist you need to contact.".
- Step 10: **New step**.

Conditions:

- Step 8 is taken with conditions.
- Condition: If **All** of this is true:
 - 5. What kind of faci...** is **Parking facility**.
- and **Add condition +**.
- New condition group +**.

Assistant says:

Wallet parking system is available.

User enters free text

Preview:

- Facilities: **Name of the hospital**, **Branches**, **Ambulance service**.
- Branches: **Bangalore**, **Chennai**, **Salem**.
- Message: "Bangalore, Chennai, Salem Which branch do you prefer?".
- Options: **Bangalore**, **Chennai**, **Salem**.
- Message: "Hello :) recognized".
- Message: "There are no additional steps for this action. Add a new step or end the action."
- Message: "Use the up arrow for prior messages"

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>Home</title>

    <link rel="stylesheet" href="{ {url_for('redirect_to',link='https://s3.jp-tok.cloud-object-storage.appdomain.cloud/cloudbucket/assign3.css')} }" type="text/css">

    <script>

      window.watsonAssistantChatOptions = {

        integrationID: "c7bc81d7-9112-451d-ae6d-8093ccb60341", // The ID of this integration.

        region: "us-south", // The region your integration is hosted in.

        serviceInstanceID: "ef8e6e82-67f9-4ca8-9071-e0186fe763ee", // The ID of your service instance.

        onLoad: function(instance) { instance.render(); }

      };

      setTimeout(function(){

        const t=document.createElement('script');

        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
        (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";

        document.head.appendChild(t);

      });

    </script>

  </head>

  <body>

    <form action="/uploader" method="POST" enctype="multipart/form-data">

      <input type="text" placeholder="Enter file name" name="filename" />

      <br />

      <br />

      <input type="file" name="file" />

      <br />

      <br />

      <input type="submit" />

    </form>

  </body>

</html>
```



```

</form>

<br/>

<br/>

<br/>

{% for row in files %}

    <div style="border: 1px solid #EFEFEF;margin:10px;">

        <h3>Filename : {{row}} </h3>

        </td>

    </div>

{% endfor %}

</body>

</html>

```

App.py

```

import io

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=""


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__)

@app.route('/')
def index():
    try:
        files = cos.Bucket('cloudbucket').objects.all()
        files_names = []
        for file in files:
            files_names.append(file.key)
            print(file)
            print("Item: {0} ({1} bytes)".format(file.key, file.size))
        return render_template('index.html',files=files_names)

    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
        return render_template('index.html')
    except Exception as e:
        print("Unable to retrieve bucket contents: {0}".format(e))
        return render_template('index.html')

@app.route('/uploader',methods=['POST'])
def upload():
    name_file=request.form['filename']
    f = request.files['file']
    try:
        part_size = 1024 * 1024 * 5

        file_threshold = 1024 * 1024 * 15

        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,

```

```
        multipart_chunksize=part_size
    )

    content = f.read()
    cos.Object('cloudbucket', name_file).upload_fileobj(
        Fileobj=io.BytesIO(content),
        Config=transfer_config
    )
    return redirect(url_for('index'))
```

```
except ClientError as be:
    print("CLIENT ERROR: {0}\n".format(be))
    return redirect(url_for('index'))
```

```
except Exception as e:
    print("Unable to complete multi-part upload: {0}".format(e))
    return redirect(url_for('index'))
```

```
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080, debug=True)
```