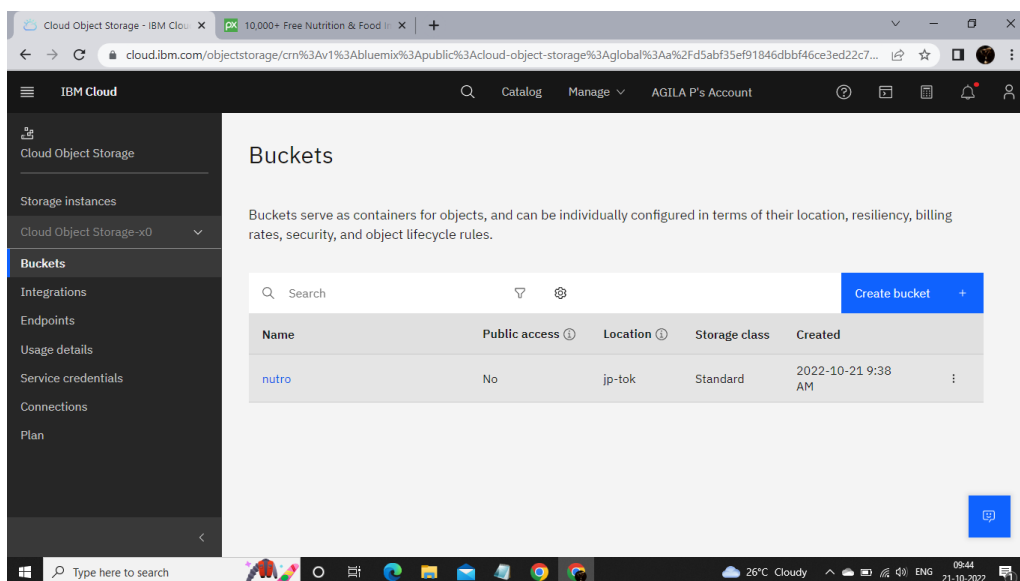
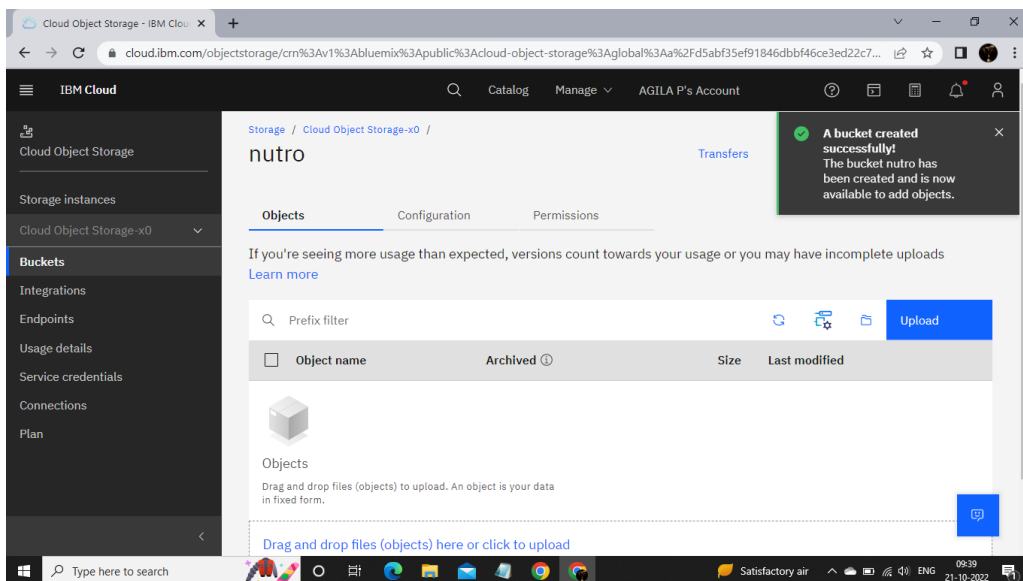


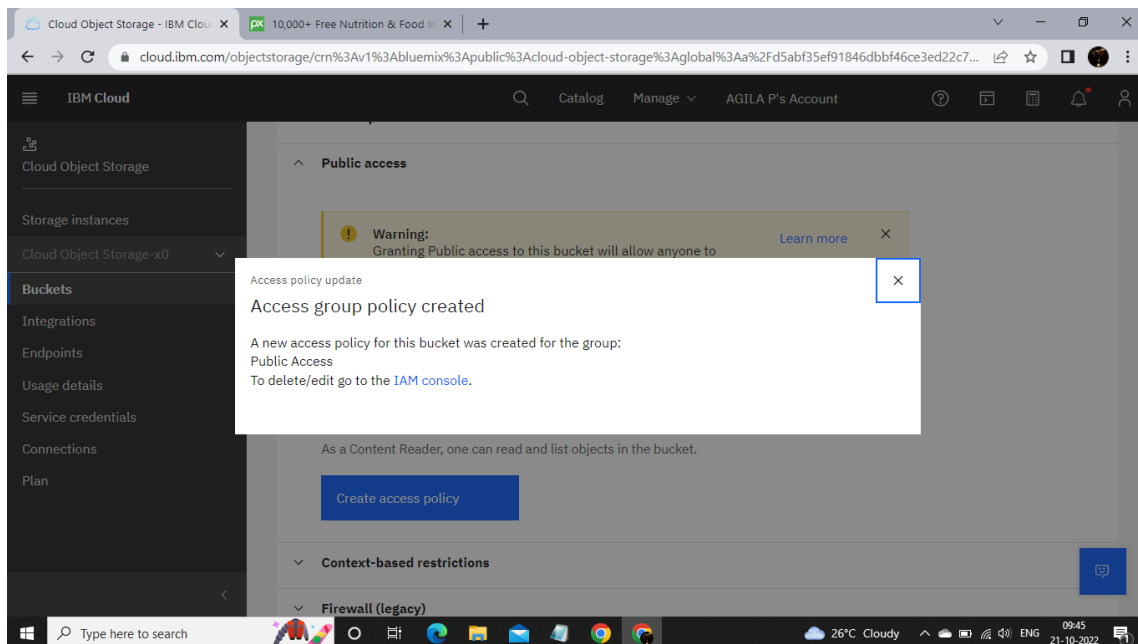
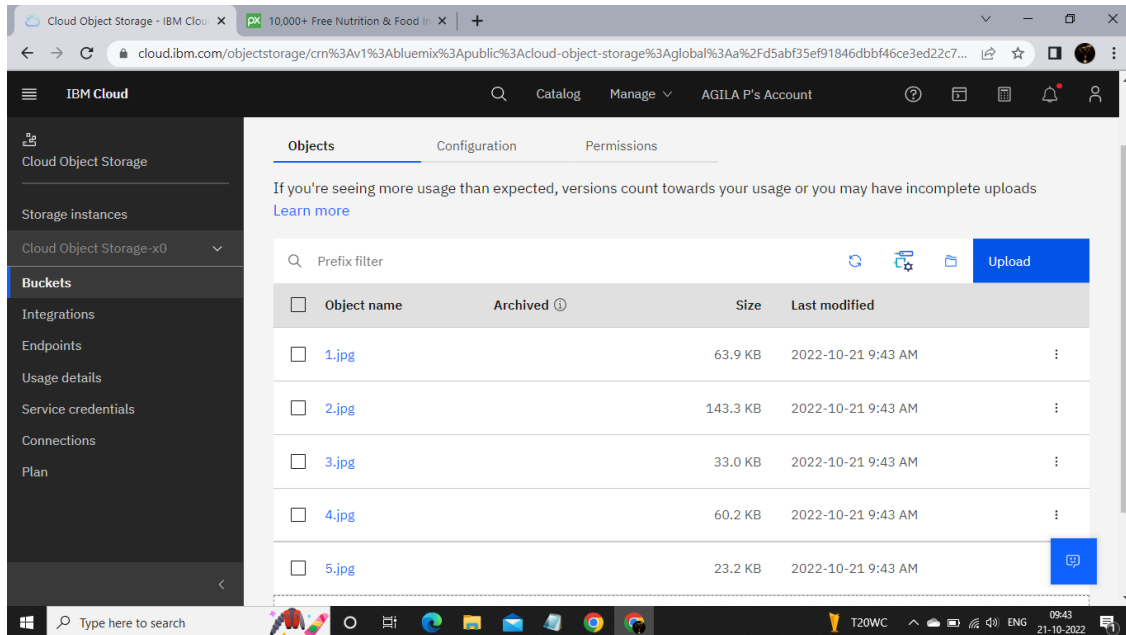
Assignment-3

Date	10 October 2022
Team ID	PNT2022TMID30524
Projectname	Nutrition Assistant Application

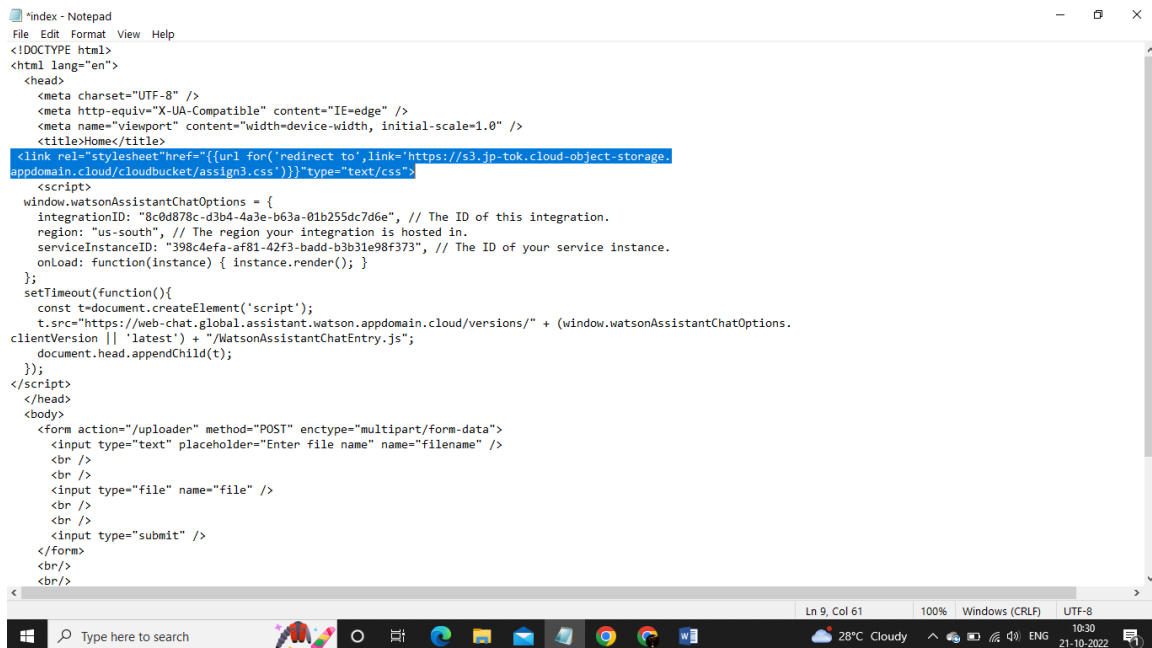
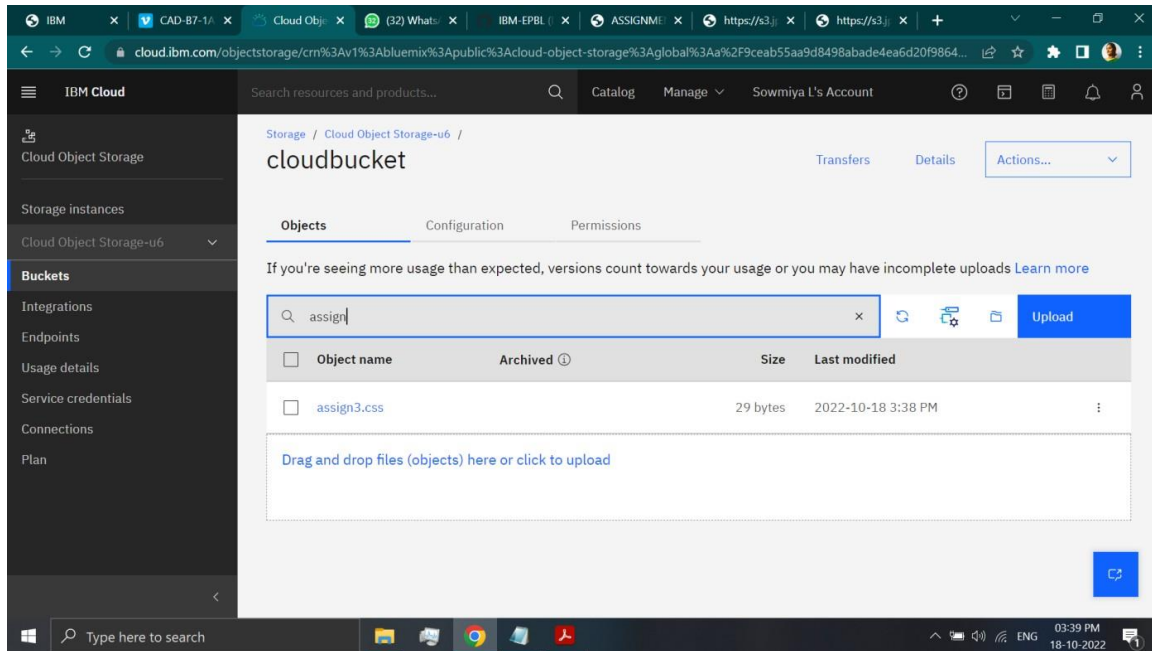
1. CREATE A BUCKET IN IBM OBJECT STORAGE.



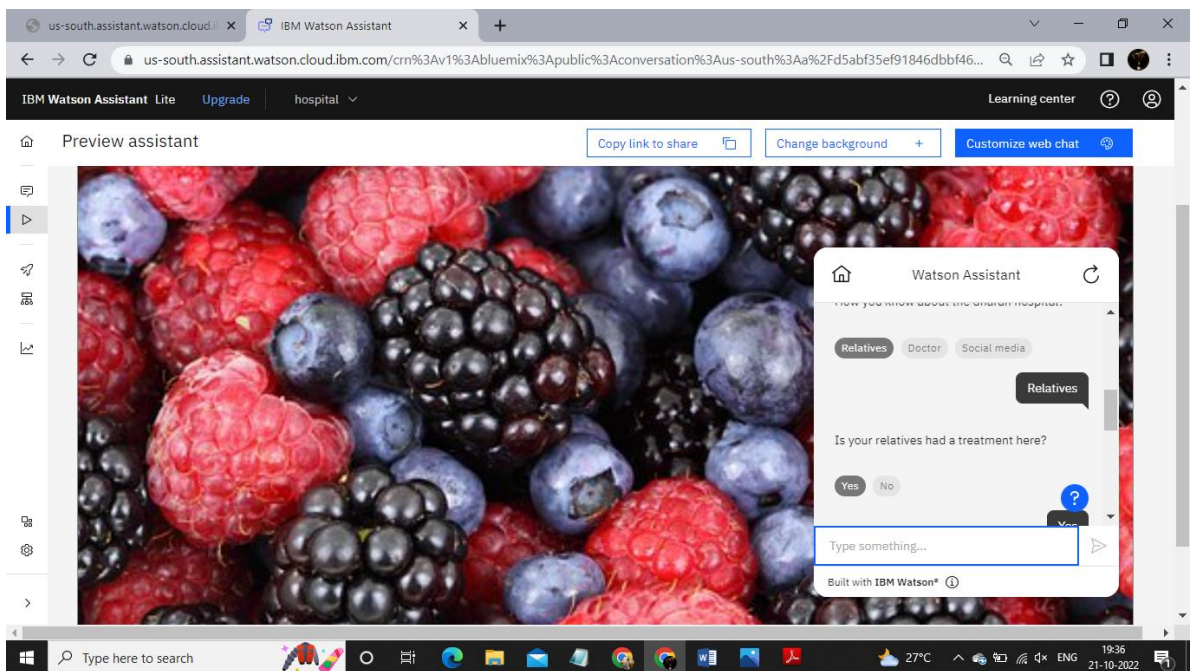
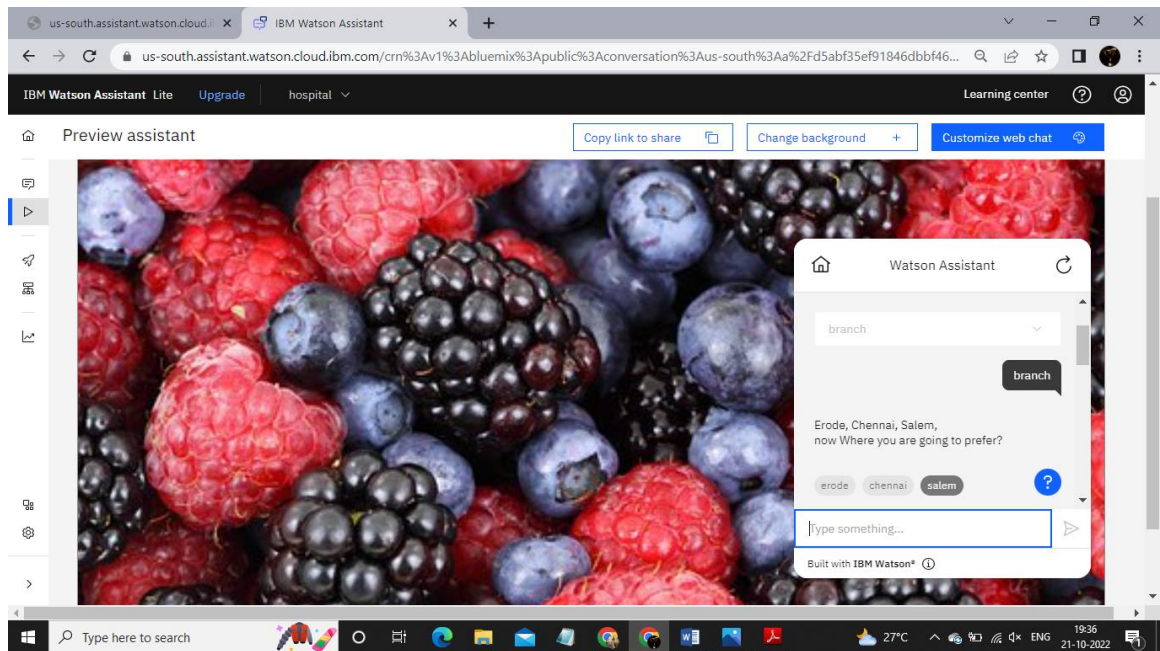
2. Upload an 5 images to ibm object storage and make it public. write html code to displaying all the 5 images.



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-398c4efa-af81-42f3-badd-b3b31e98f373%3A%3A6299a82c-671e-4181-a618-b8c6251f725a&integrationID=8c0d878c-d3b4-4a3e-b63a-01b255dc7d6e®ion=us-south&serviceInstanceID=398c4efa-af81-42f3-badd-b3b31e98f373>

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

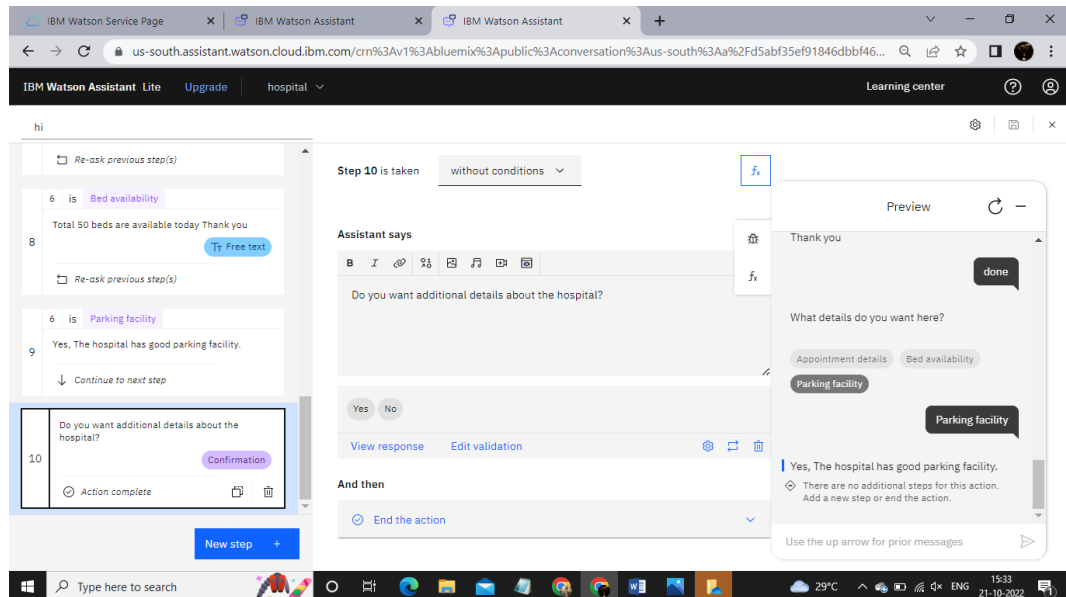


Figure 1. 10 steps of conversation

Included 3 conditions in steps:

This screenshot shows the IBM Watson Assistant configuration interface. On the left, the 'Conversation steps' panel lists three steps: Step 1 asks for details, Step 2 identifies the hospital as 'Dharan hospital', and Step 3 asks for a branch. The main area shows 'Step 10 is taken without conditions'. The 'Assistant says' section contains the text 'Do you want additional details about the hospital?' with 'Yes' and 'No' buttons. A 'Preview' window on the right shows a chat simulation where the user asks for details, and the assistant responds with 'Thank you' and 'What details do you want here?' followed by buttons for 'Appointment details', 'Bed availability', and 'Parking facility'. The 'Parking facility' button is highlighted.

This screenshot shows the IBM Watson Assistant configuration interface for Step 4. The 'Conversation steps' panel on the left shows Step 3 asking for a branch, with 'salem' and 'chennai' as options. The main area shows 'Step 4 is taken with conditions'. The 'Conditions' section is configured with 'If All of this is true:' followed by '3. Erode, Che...' and 'salem'. The 'Assistant says' section contains the text 'How you know about the dharan hospital?'. A 'Preview' window on the right shows a chat simulation where the user asks for details, and the assistant responds with 'Thank you' and 'What details do you want here?' followed by buttons for 'Appointment details', 'Bed availability', and 'Parking facility'. The 'Parking facility' button is highlighted.

IBM Watson Assistant Lite Upgrade hospital Learning center

hi

3 is salem

4 How you know about the dharan hospital?

4 is Relatives

5 Is your relatives had a treatment here?

6 What details do you want here?

6 is Appointment details

Step 5 is taken with conditions

Conditions

If All of this is true:

4. How you k... is Relatives

and Add condition +

New condition group +

Assistant says

Is your relatives had a treatment here?

Preview

Thank you

What details do you want here?

Appointment details Bed availability

Parking facility

Parking facility

Yes, The hospital has good parking facility.

There are no additional steps for this action. Add a new step or end the action.

Type something...

IBM Watson Assistant Lite Upgrade hospital Learning center

hi

6 is Bed availability

8 Total 50 beds are available today Thank you

6 is Parking facility

9 Yes, The hospital has good parking facility.

10 Do you want additional details about the hospital?

Confirmation

Action complete

Step 6 is taken with conditions

Conditions

If All of this is true:

6. What detai... is Parking facility

and Add condition +

New condition group +

Assistant says

Yes, The hospital has good parking facility.

Preview

Thank you

What details do you want here?

Appointment details Bed availability

Parking facility

Parking facility

Yes, The hospital has good parking facility.

There are no additional steps for this action. Add a new step or end the action.

Type something...

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: "8c0d878c-d3b4-4a3e-b63a-01b255dc7d6e", // The ID of this integration.
        region: "us-south", // The region your integration is hosted in.
        serviceInstanceID: "398c4efa-af81-42f3-badd-b3b31e98f373",
        // 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>

    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>
```

```
<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
```