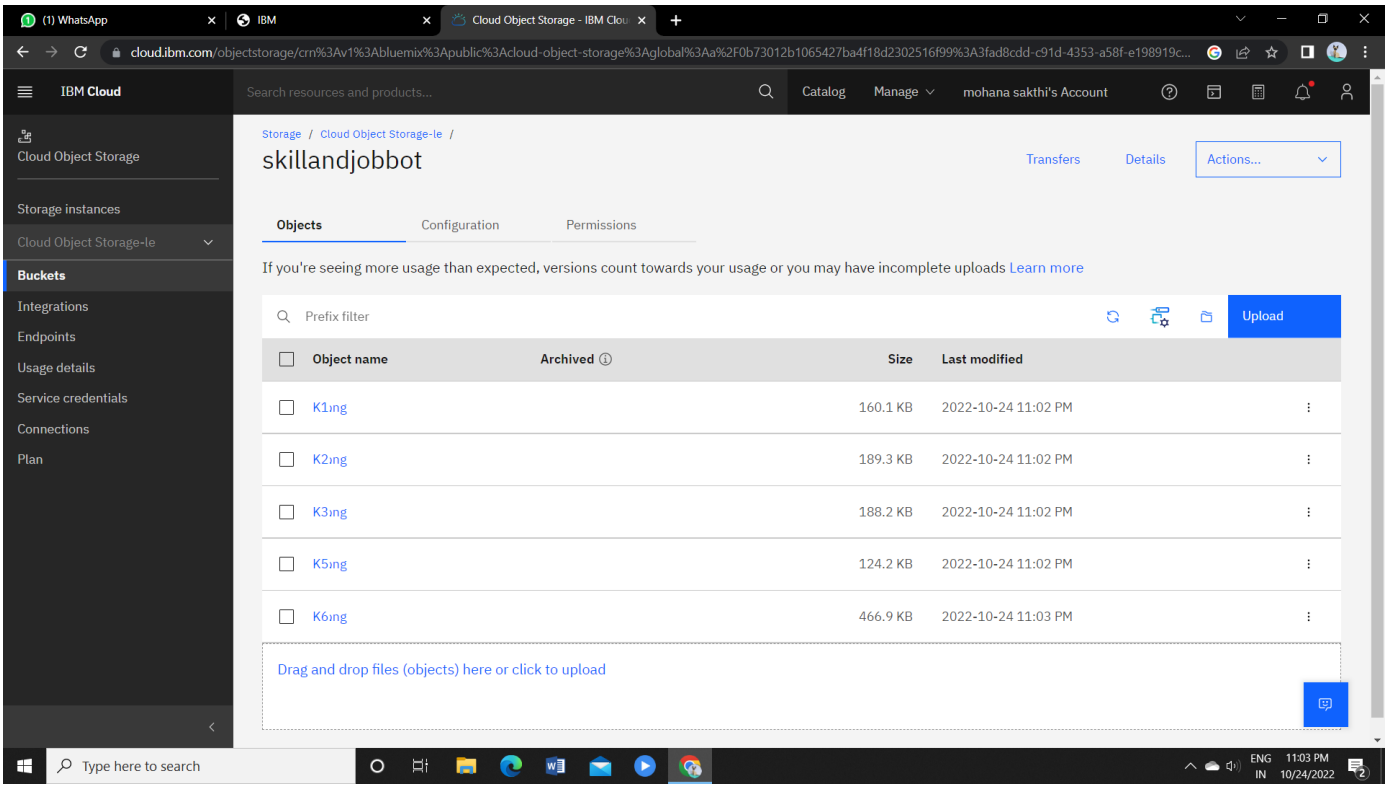
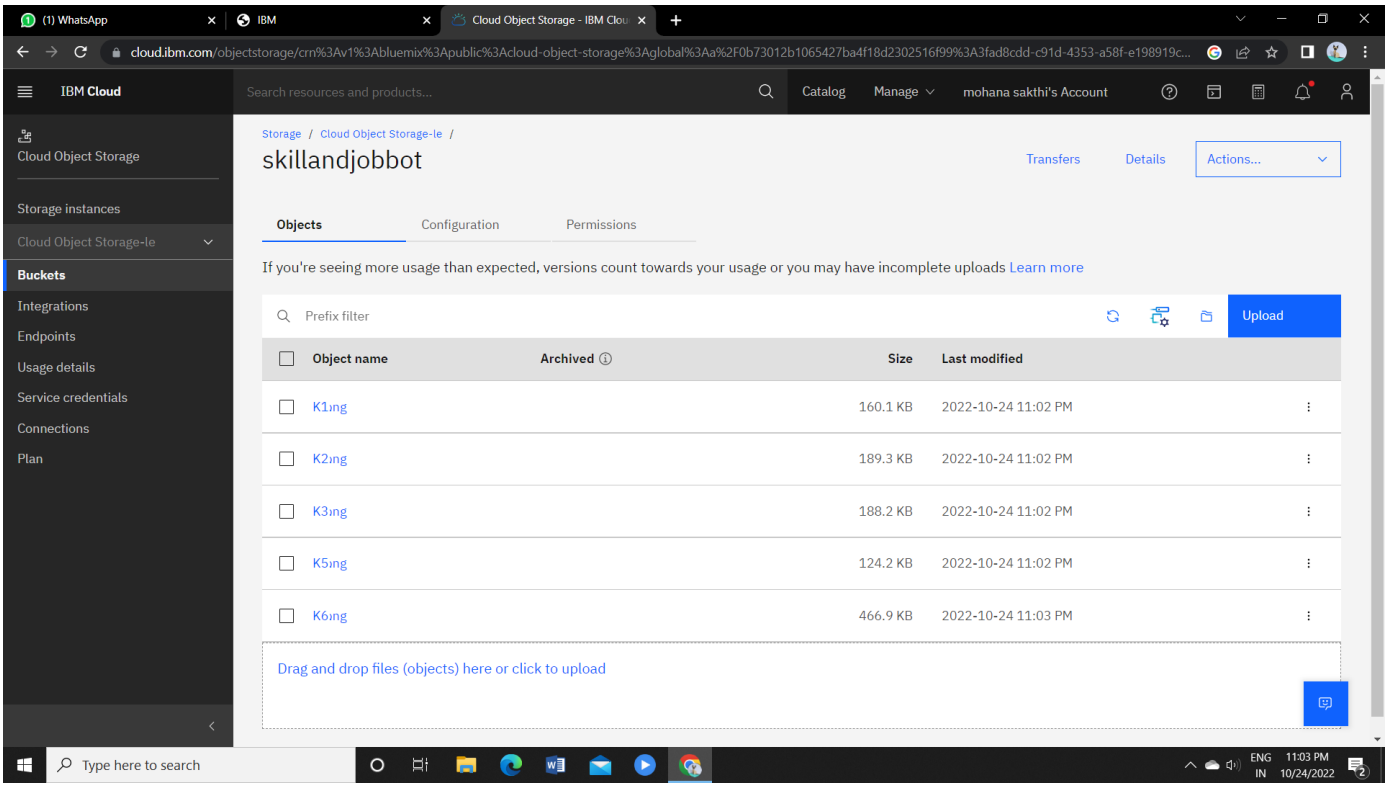
Assignment-3

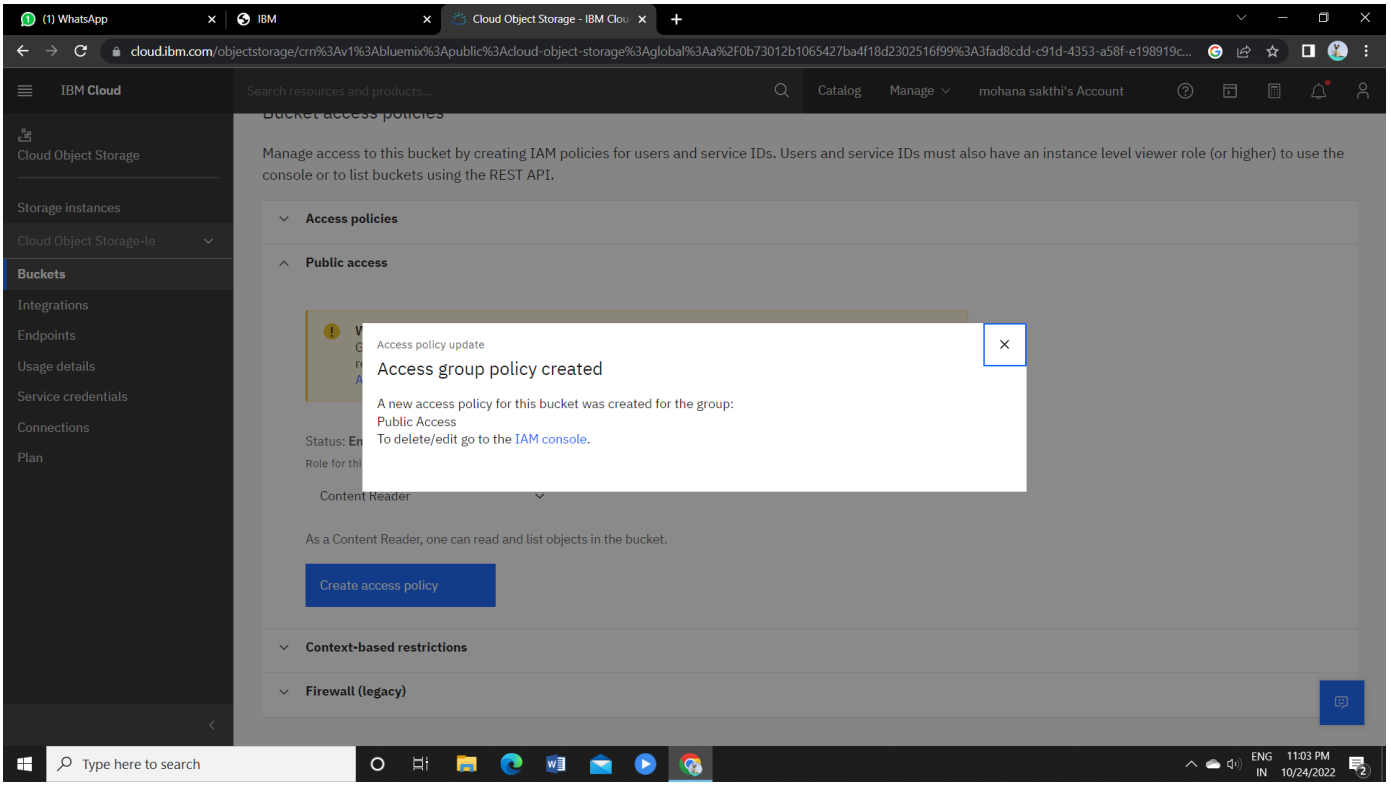
Date	10 October 2022
Name	Mohana sakthi.G.S
Team ID	PNT2022TMID30531
Project Name	Skill and Job Recommender Application

1. CREATE A BUCKET IN IBM OBJECT STORAGE.

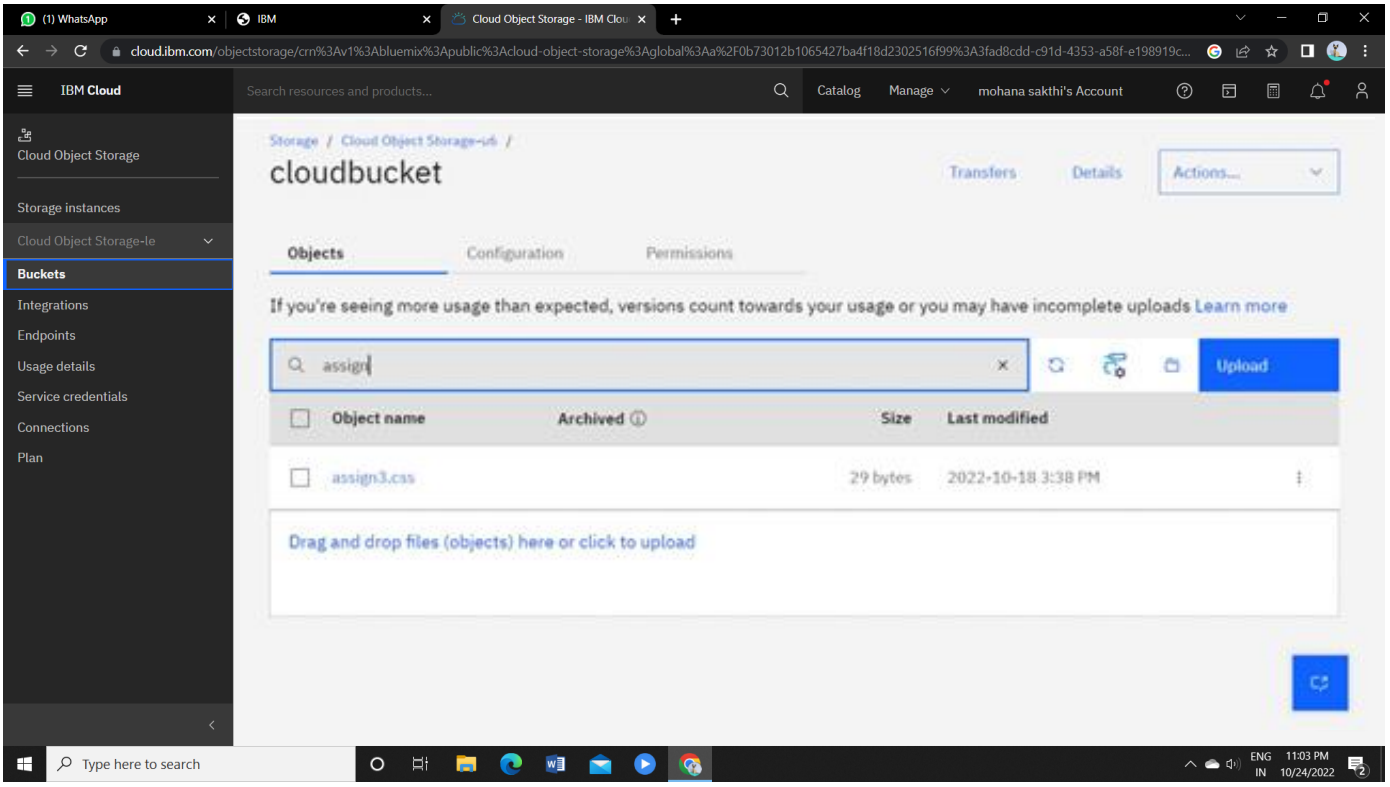


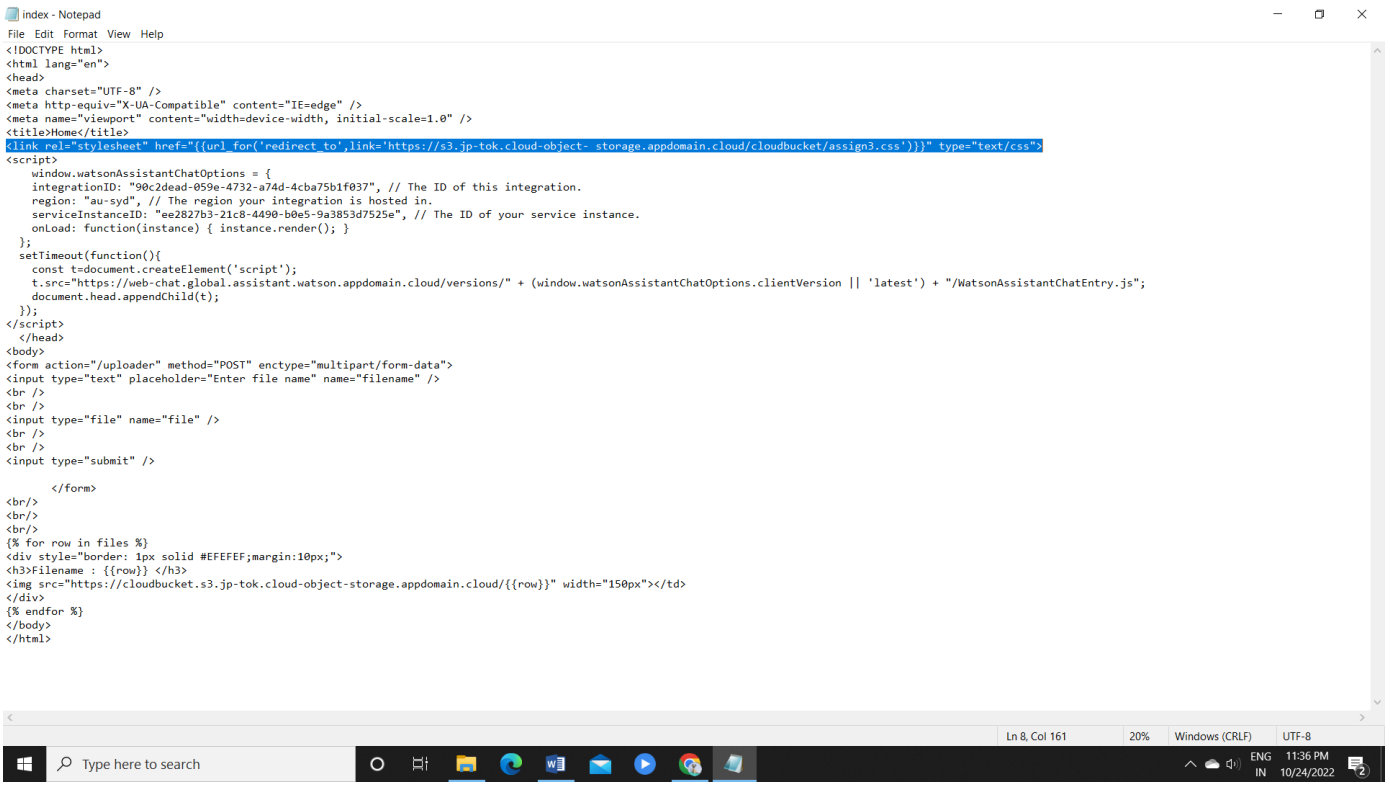
Upload an 5 images to ibm object storage and make it public.
Write html code todisplaying all the 5 images.


IBM Cloud
Search resources and products...
Catalog
Manage
mohana sakthi's Account
Cloud Object Storage
Storage instances
Cloud Object Storage-le
Buckets
Integrations
Endpoints
Usage details
Service credentials
Connections
Plan
Storage / Cloud Object Storage-le /
skillandjobbot
Transfers
Details
Actions...
Objects
Configuration
Permissions
If you're seeing more usage than expected, versions count towards your usage or you may have incomplete uploads Learn more
Prefix filter
Upload
Object name
Archived
Size
Last modified
K1.ng 160.1 KB 2022-10-24 11:02 PM
K2.ng 189.3 KB 2022-10-24 11:02 PM
K3.ng 188.2 KB 2022-10-24 11:02 PM
K5.ng 124.2 KB 2022-10-24 11:02 PM
K6.ng 466.9 KB 2022-10-24 11:03 PM
Drag and drop files (objects) here or click to upload
Type here to search
ENG IN 11:03 PM 10/24/2022

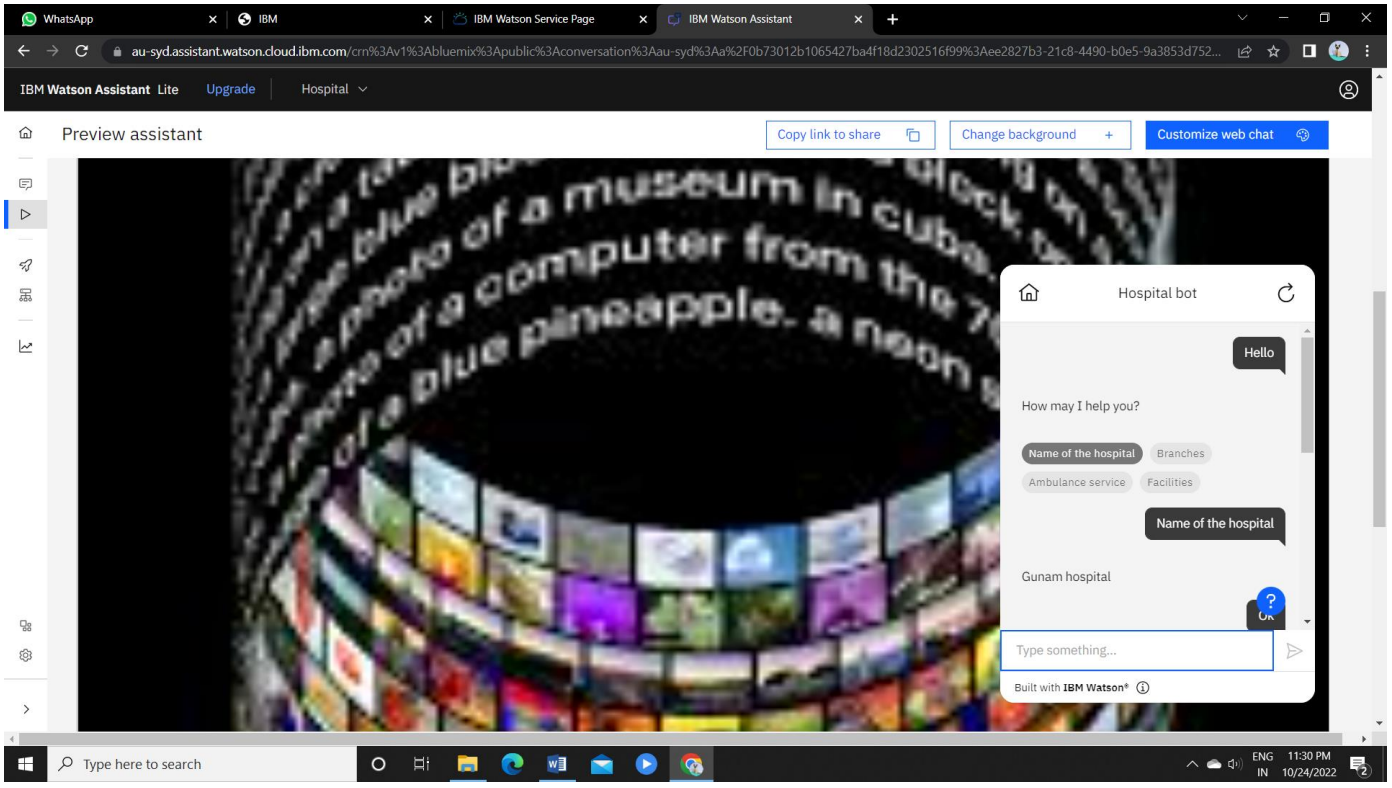

IBM Cloud
Search resources and products...
Catalog
Manage
mohana sakthi's Account
Cloud Object Storage
Storage instances
Cloud Object Storage-le
Buckets
Integrations
Endpoints
Usage details
Service credentials
Connections
Plan
Manage access to this bucket by creating IAM policies for users and service IDs. Users and service IDs must also have an instance level viewer role (or higher) to use the console or to list buckets using the REST API.
Access policies
Public access
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.
Content Reader
As a Content Reader, one can read and list objects in the bucket.
Create access policy
Context-based restrictions
Firewall (legacy)
Type here to search
ENG IN 11:03 PM 10/24/2022

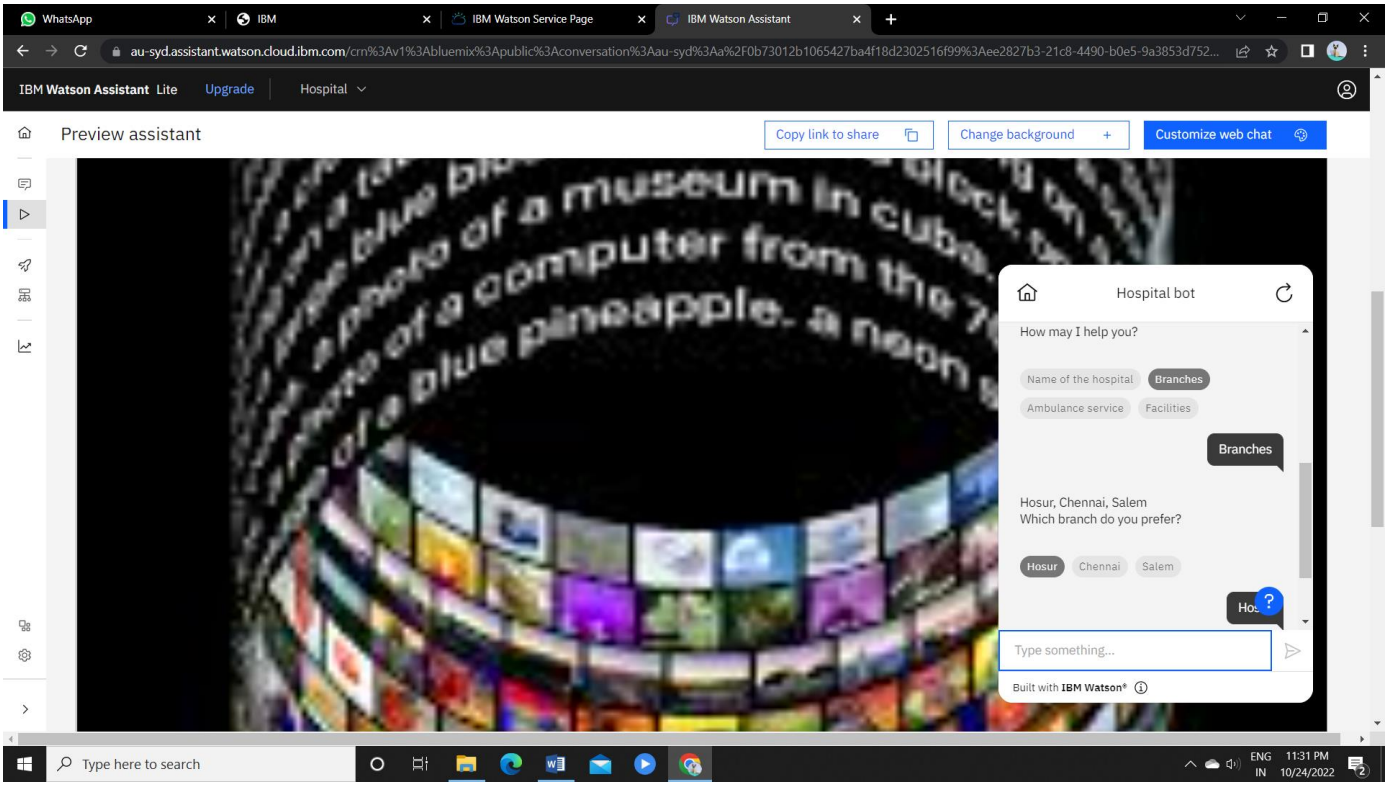
2. Upload a css page to the object storage and use the same page in your HTML code.





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

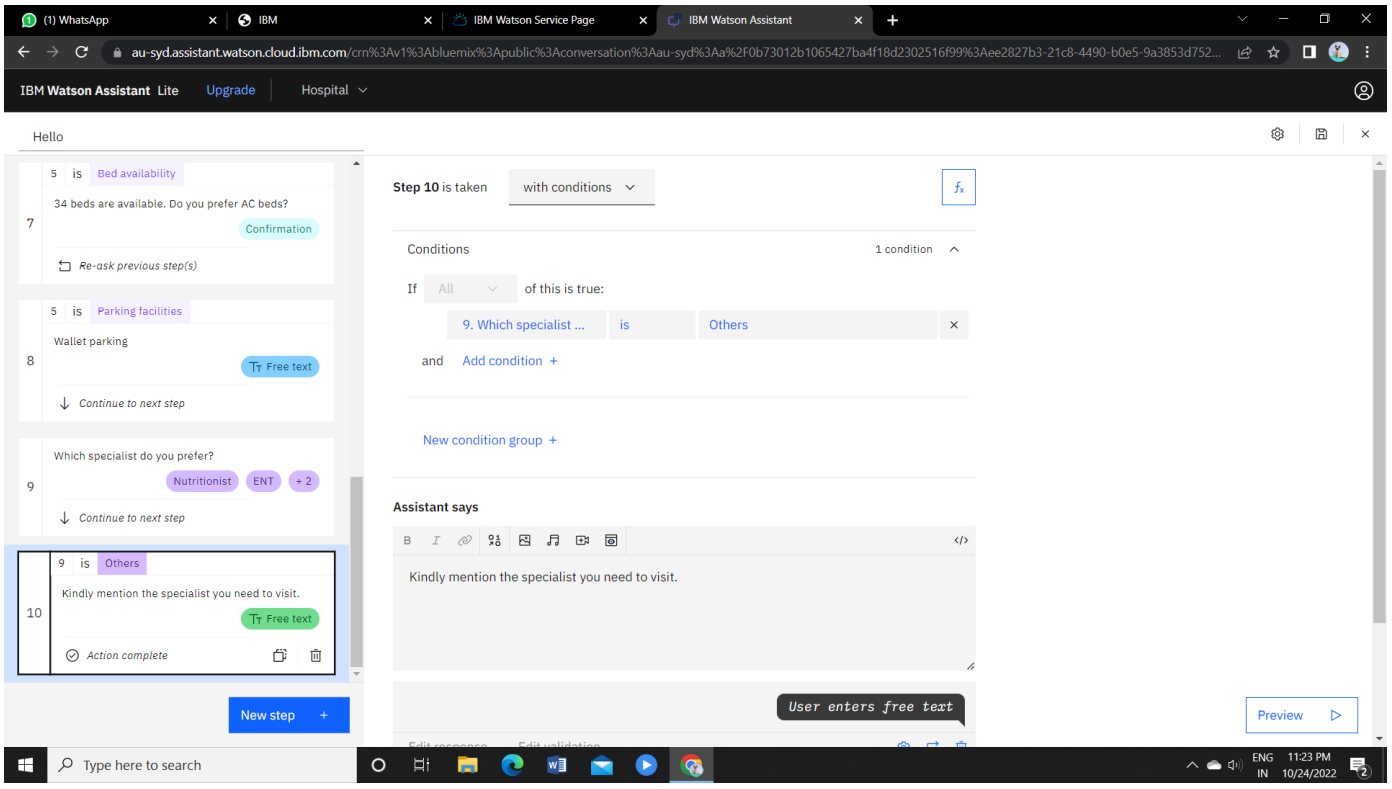




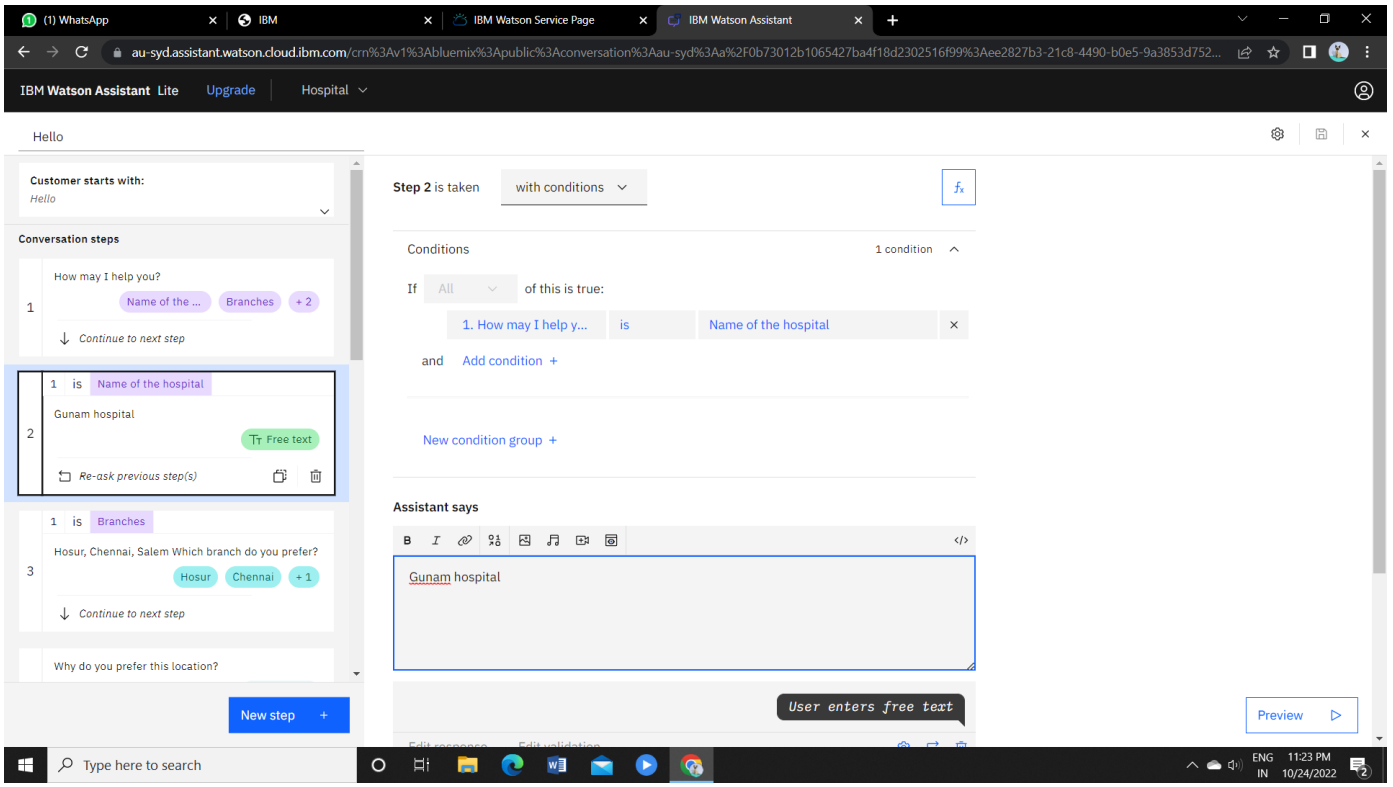
Web URL for Assistant:

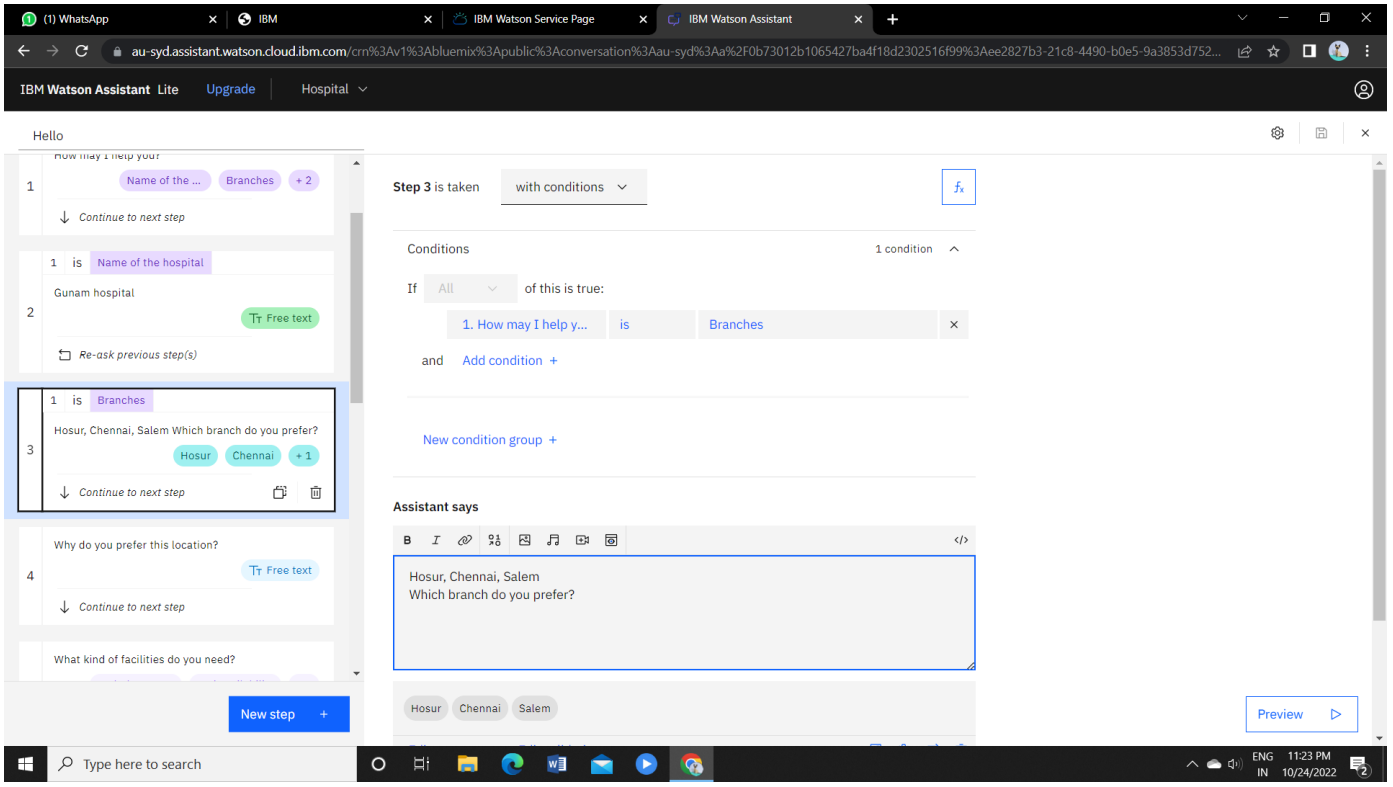
https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fau-syd.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-ee2827b3-21c8-4490-b0e5-9a3853d7525e%3A%3Aa631a952-bd36-4487-81f2-246f2a791a5b&integrationID=90c2dead-059e-4732-a74d-4cba75b1f037&region=au-syd&serviceInstanceID=ee2827b3-21c8-4490-b0e5-9a3853d7525e

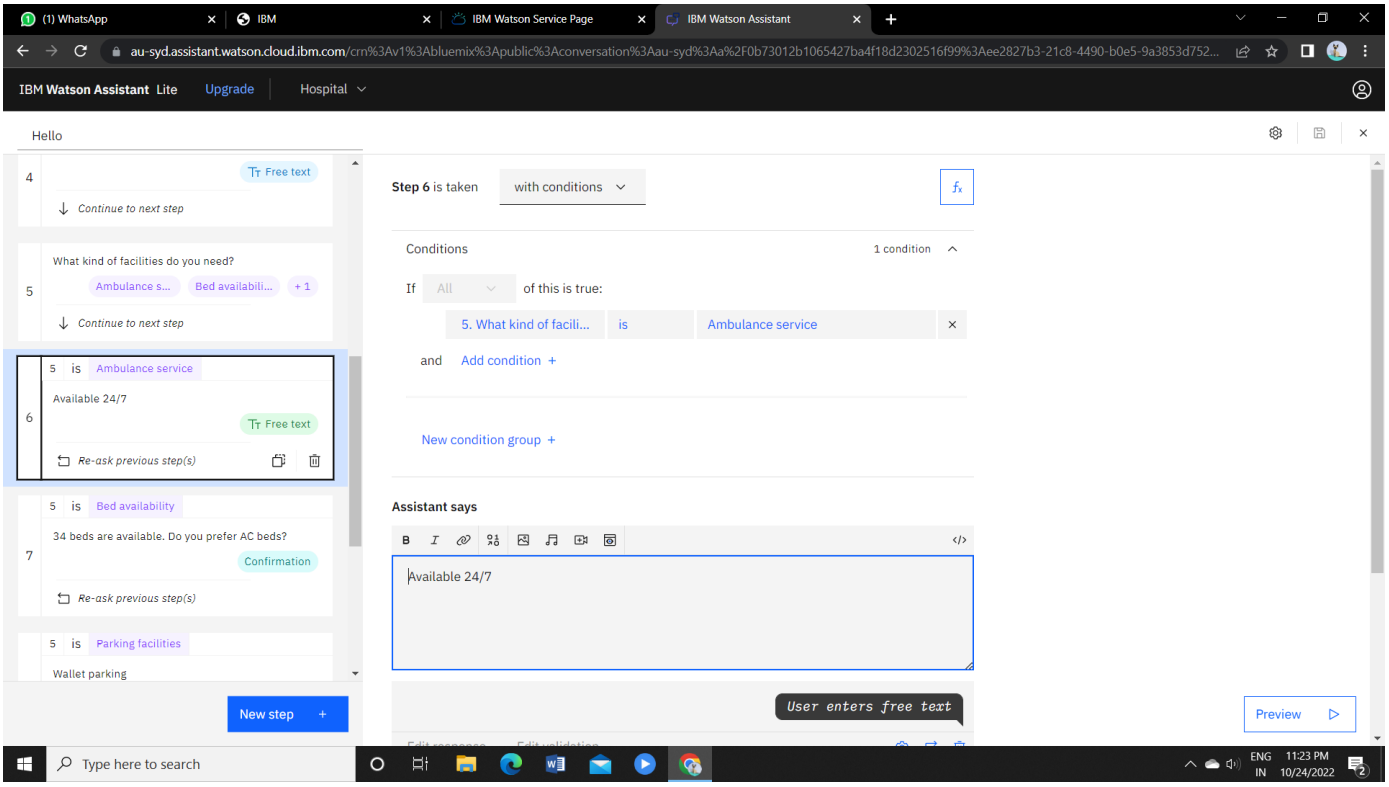
4. Create Watson assistant service with 10 steps and use 3 conditions in it. Load thatscript in HTML page.



Included 3 conditions in steps:






(1) WhatsApp
IBM
IBM Watson Service Page
IBM Watson Assistant
au-syd.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aau-syd%3Aa%2F0b73012b1065427ba4f18d2302516f99%3Aee2827b3-21c8-4490-b0e5-9a3853d752...
IBM Watson Assistant Lite
Upgrade
Hospital
Hello
4
Free text
Continue to next step
What kind of facilities do you need?
5
Ambulance s...
Bed availabili...
+ 1
Continue to next step
5 is Ambulance service
Available 24/7
6
Free text
Re-ask previous step(s)
5 is Bed availability
34 beds are available. Do you prefer AC beds?
7
Confirmation
Re-ask previous step(s)
5 is Parking facilities
Wallet parking
New step
Step 6 is taken
with conditions
Conditions
1 condition
If All of this is true:
5. What kind of facili...
is
Ambulance service
and
Add condition +
New condition group +
Assistant says
Available 24/7
User enters free text
Preview
Type here to search
ENG
IN
11:23 PM
10/24/2022

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: "90c2dead-059e-4732-a74d-4cba75b1f037", // The ID of this integration.
  region: "au-syd", // The region your integration is hosted in.
  serviceInstanceID: "ee2827b3-21c8-4490-b0e5-9a3853d7525e", // 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>
    <br/>
    <br/>
    <br/>
    {% for row in files %}
        <div style="border: 1px solid #EFEFEF;margin:10px;">
          <h3>Filename : {{row}} </h3>
          <img src="https://cloudbucket.s3.jp-tok.cloud-object-storage.appdomain.cloud/{{row}}"
width="150px"></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
```