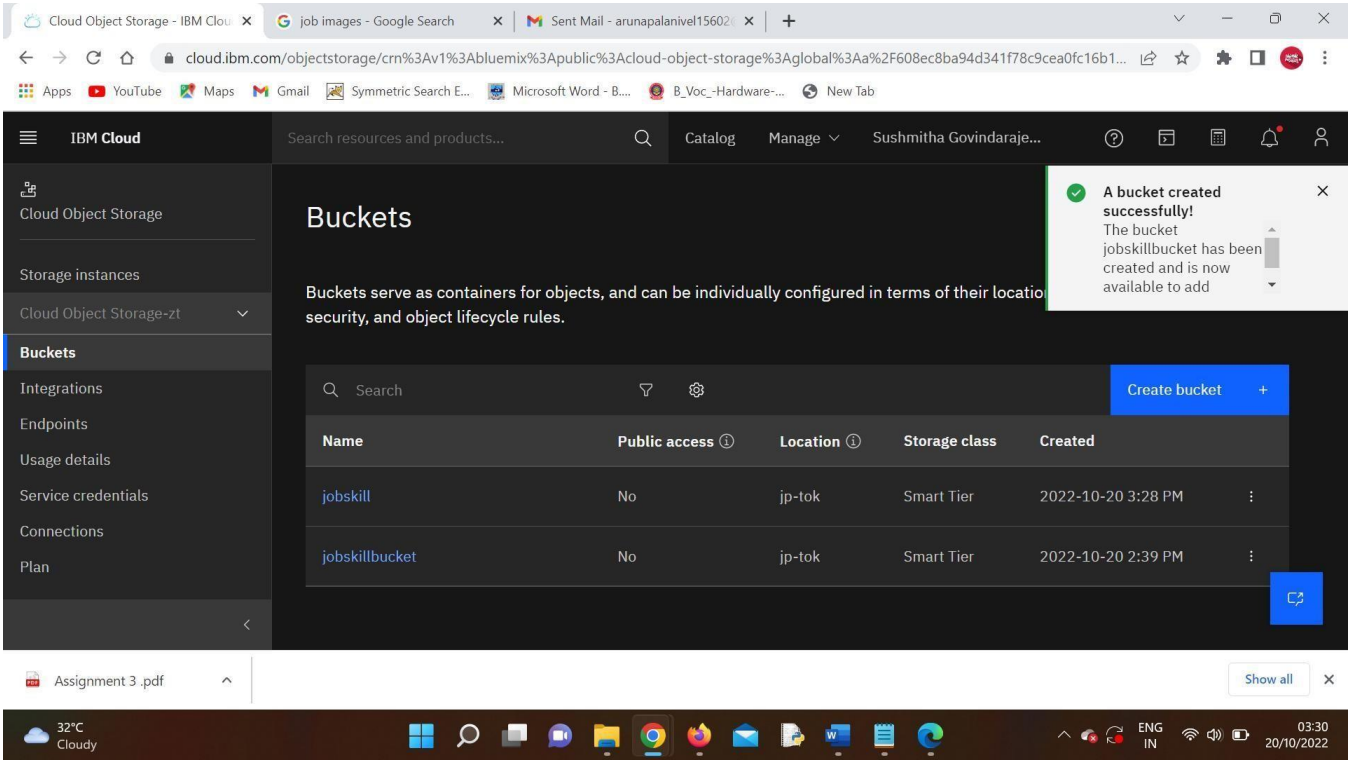
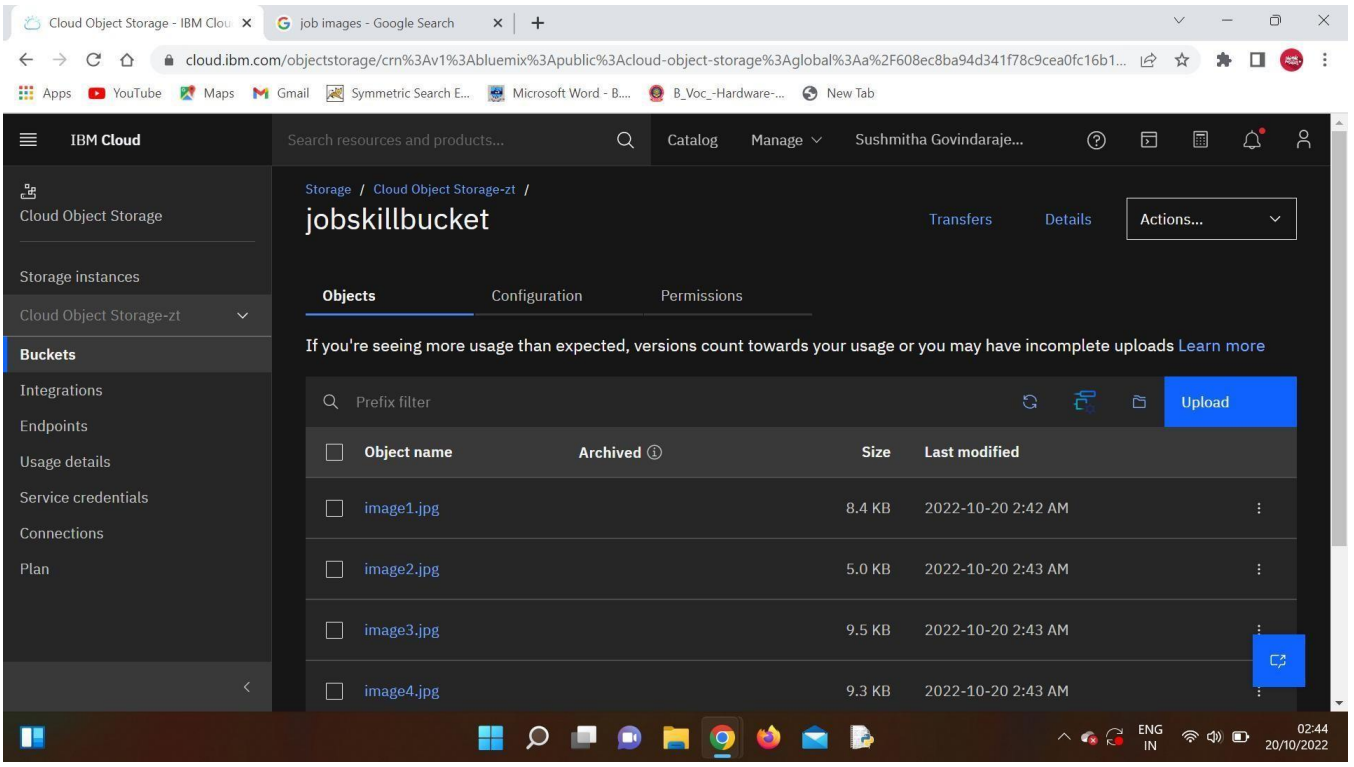
Assignment-3

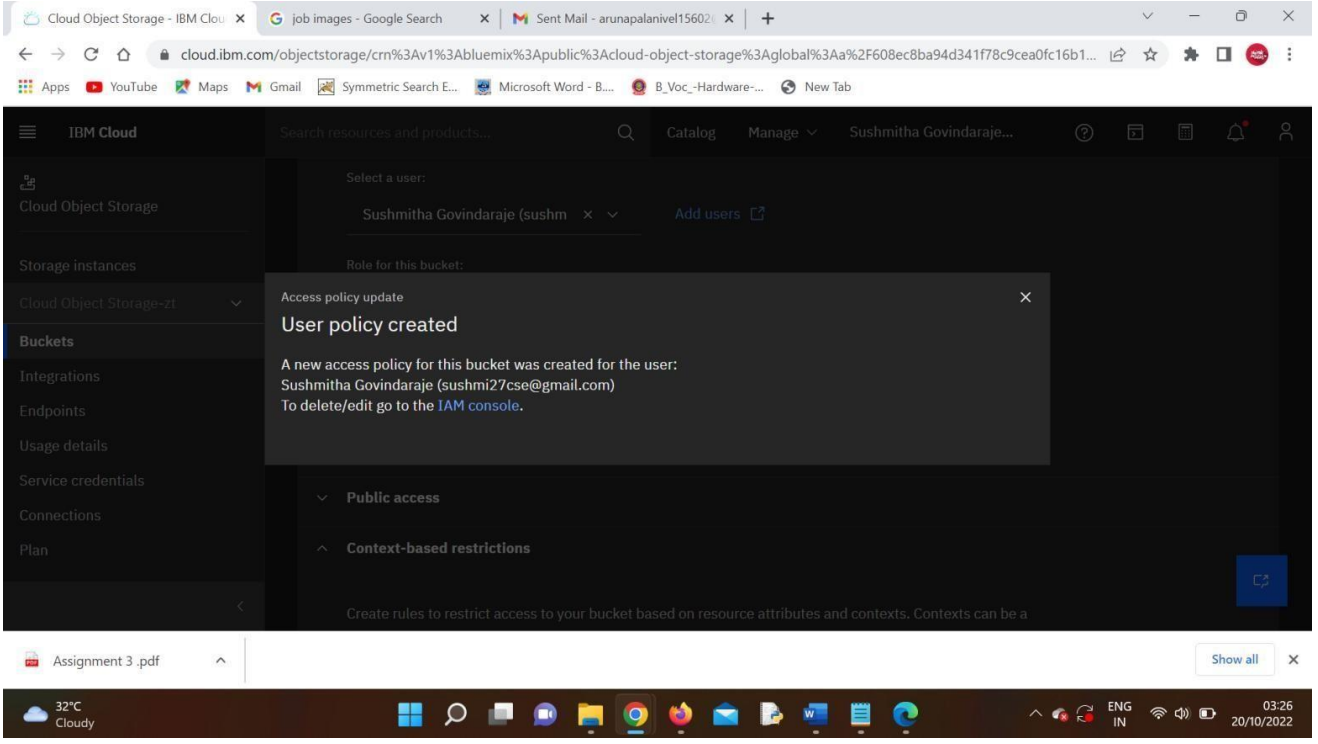
Date	22 October 2022
Team ID	PNT2022TMID29623
Project Name	Skill/Job Recommender Application

1. CREATE A BUCKET IN IBM OBJECT STORAGE.

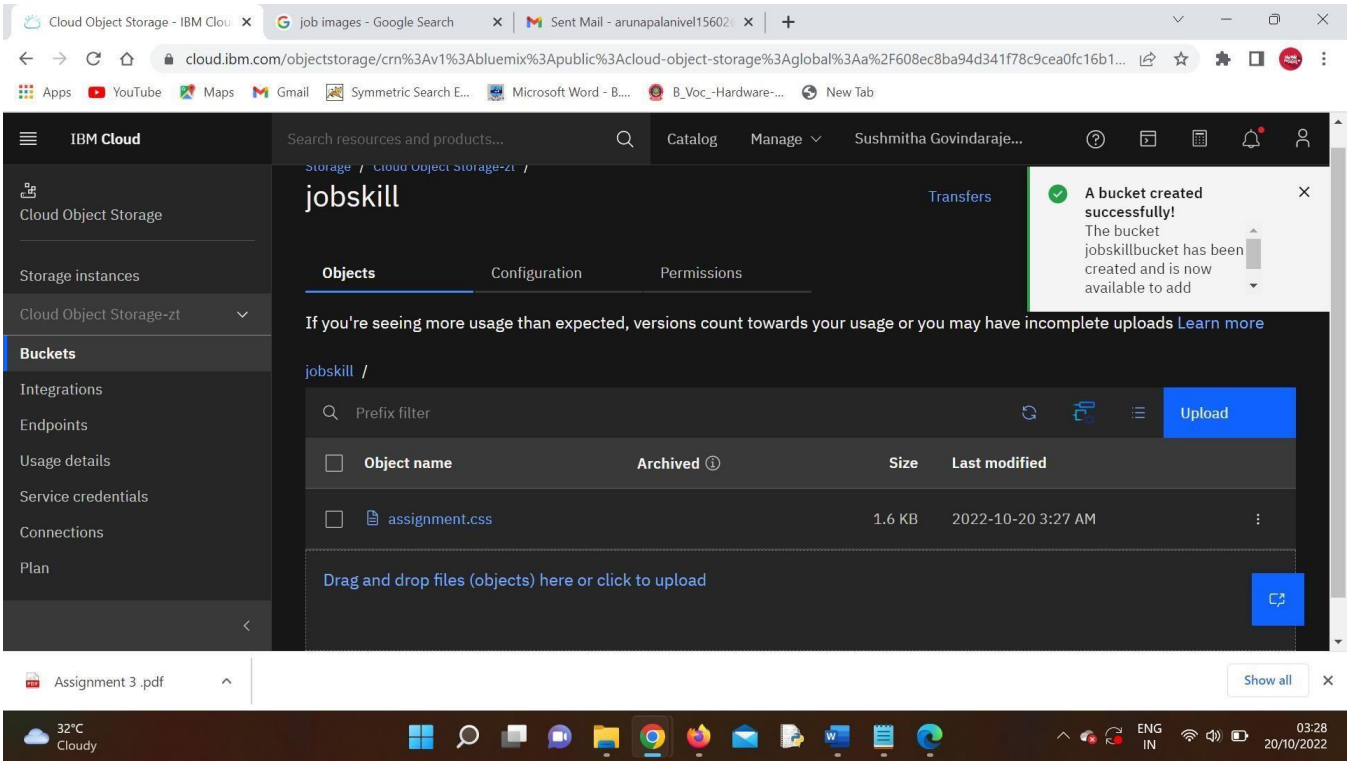


2. Upload an 5 images to ibm object storage and make it public. write html code to displaying all the 5 images.


Cloud Object Storage - IBM Clou
job images - Google Search
cloud.ibm.com/objectstorage/crn%3Av1%3Abluemix%3Apublic%3Acloud-object-storage%3Aglobal%3Aa%2F608ec8ba94d341f78c9cea0fc16b1...
Apps
YouTube
Maps
Gmail
Symmetric Search E...
Microsoft Word - B....
B_Voc_-Hardware-...
New Tab
IBM Cloud
Search resources and products...
Catalog
Manage
Sushmitha Govindaraje...
Cloud Object Storage
Storage instances
Cloud Object Storage-zt
Buckets
Integrations
Endpoints
Usage details
Service credentials
Connections
Plan
Storage / Cloud Object Storage-zt /
jobskillbucket
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
image1.jpg
8.4 KB
2022-10-20 2:42 AM
image2.jpg
5.0 KB
2022-10-20 2:43 AM
image3.jpg
9.5 KB
2022-10-20 2:43 AM
image4.jpg
9.3 KB
2022-10-20 2:43 AM
ENG
IN
02:44
20/10/2022


Cloud Object Storage - IBM Clou
job images - Google Search
Sent Mail - arunapalanivel15602
cloud.ibm.com/objectstorage/crn%3Av1%3Abluemix%3Apublic%3Acloud-object-storage%3Aglobal%3Aa%2F608ec8ba94d341f78c9cea0fc16b1...
Apps
YouTube
Maps
Gmail
Symmetric Search E...
Microsoft Word - B....
B_Voc_-Hardware-...
New Tab
IBM Cloud
Catalog
Manage
Sushmitha Govindaraje...
Cloud Object Storage
Select a user:
Sushmitha Govindaraje (sushm
Add users
Storage instances
Role for this bucket:
Cloud Object Storage-zt
Buckets
Integrations
Endpoints
Usage details
Service credentials
Connections
Plan
Access policy update
User policy created
A new access policy for this bucket was created for the user:
Sushmitha Govindaraje (sushmi27cse@gmail.com)
To delete/edit go to the IAM console.
Public access
Context-based restrictions
Create rules to restrict access to your bucket based on resource attributes and contexts. Contexts can be a
Assignment 3 .pdf
Show all
32°C
Cloudy
ENG
IN
03:26
20/10/2022

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



index.html - Notepad

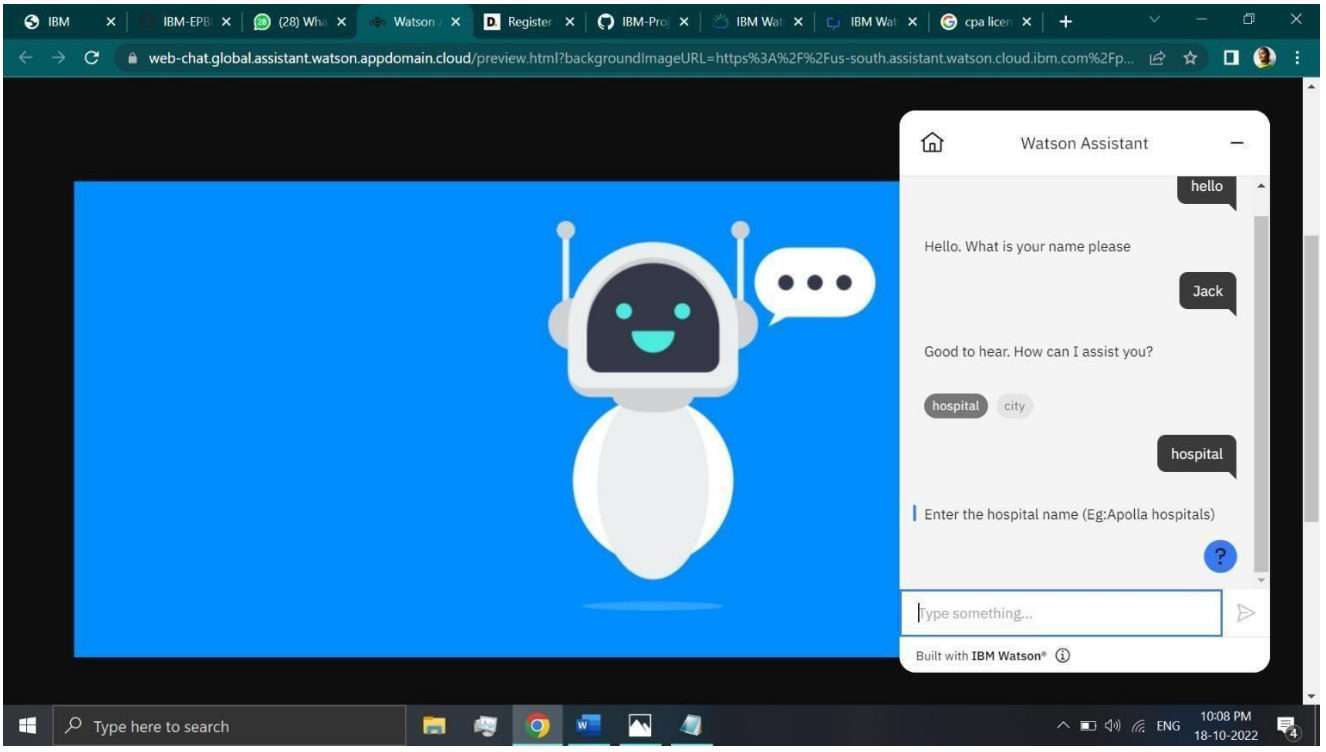
File Edit Format View Help

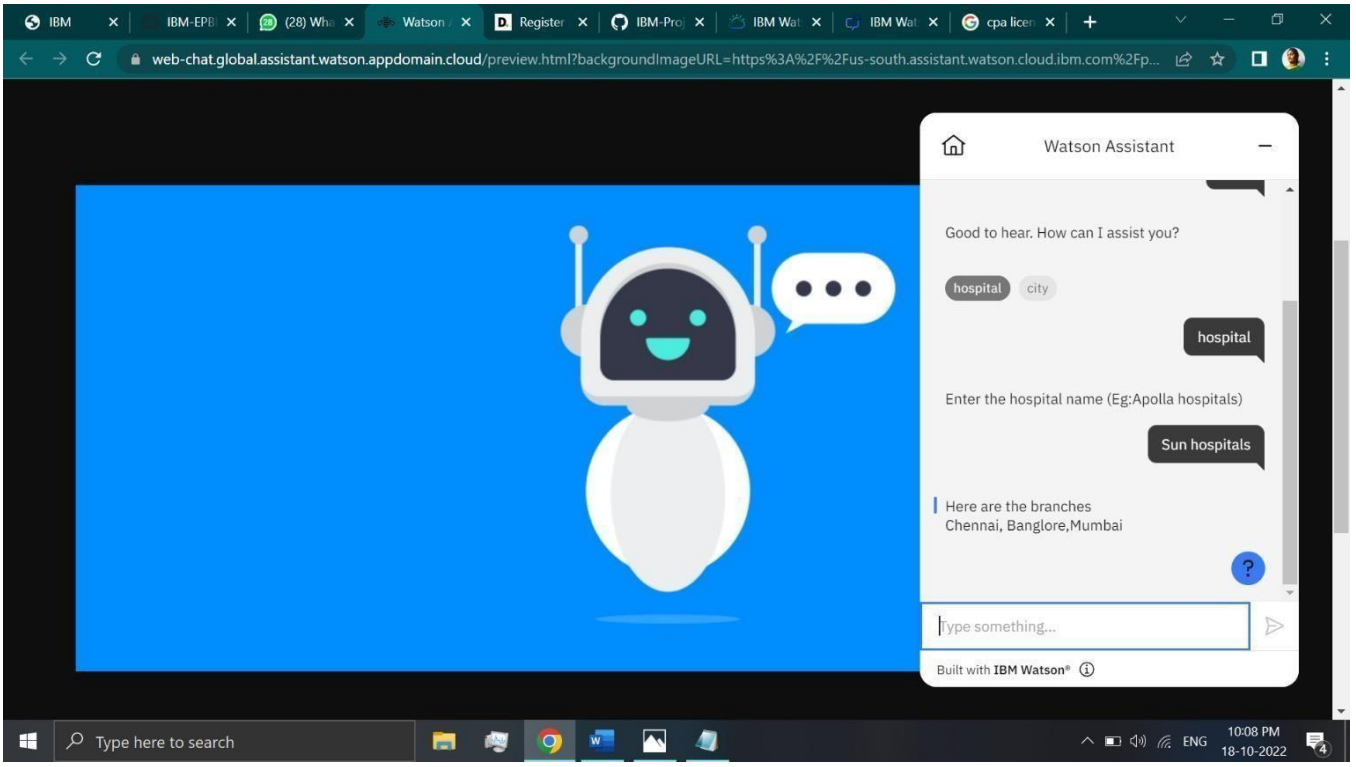
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Home</title>
<link rel="stylesheet" href="{{url_for('redirect_to',link='https://s3.jp-tok.cloud-object-storage.appdomain.cloud/cloudbucket/assign3.css')}}"
type="text/css">
    <script>
  window.watsonAssistantChatOptions = {
    integrationID: "14b83b8f-3dfd-405f-9520-b550092892aa", // The ID of this integration.
    region: "us-south", // The region your integration is hosted in.
    serviceInstanceID: "6e95bee9-8d0b-49f6-8a2f-4125fb3a7945", // 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">
```

Ln 8, Col 160 100% Windows (CRLF) UTF-8

Type here to search ENG 09:24 PM

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





Web URL for Assistant:

https://webchat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fus-south.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-304fa071-91cc-44bc-bfd7-98e7f124ef92%3A%3A0a1b8f67-b7d3-452a-b1bd-b9c995a4a9bb&integrationID=955f7fec-2d74-4165-bb9a-3befbfd74cd9&region=us-south&serviceInstanceID=304fa071-91cc-44bc-bfd7-98e7f124ef92

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

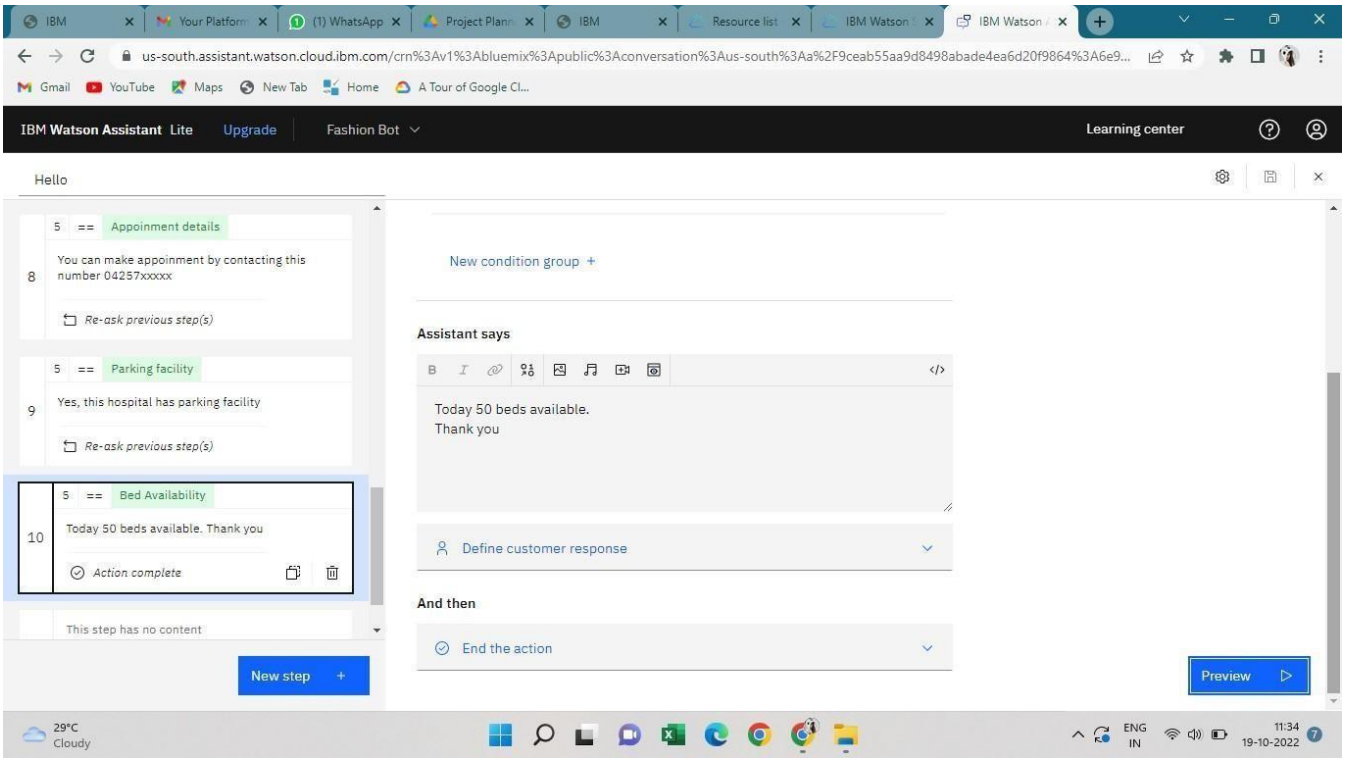
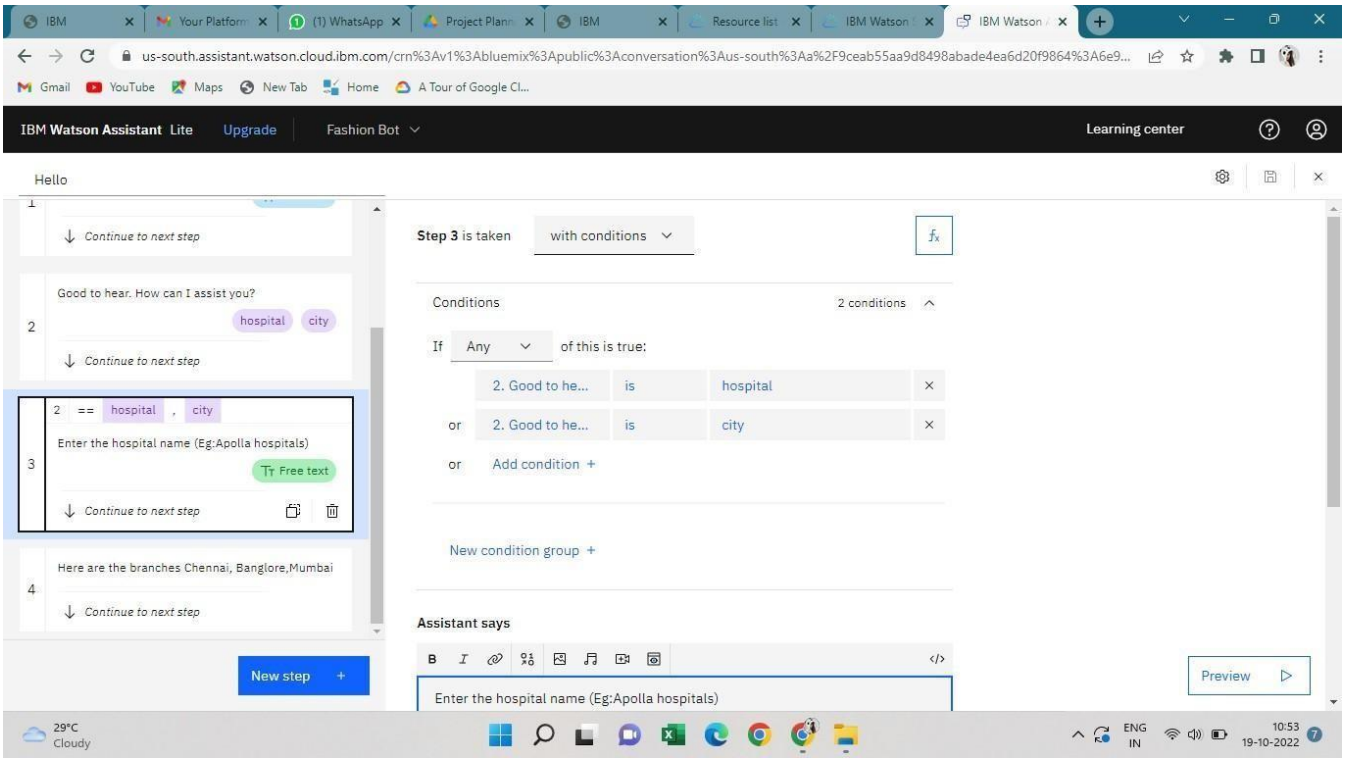
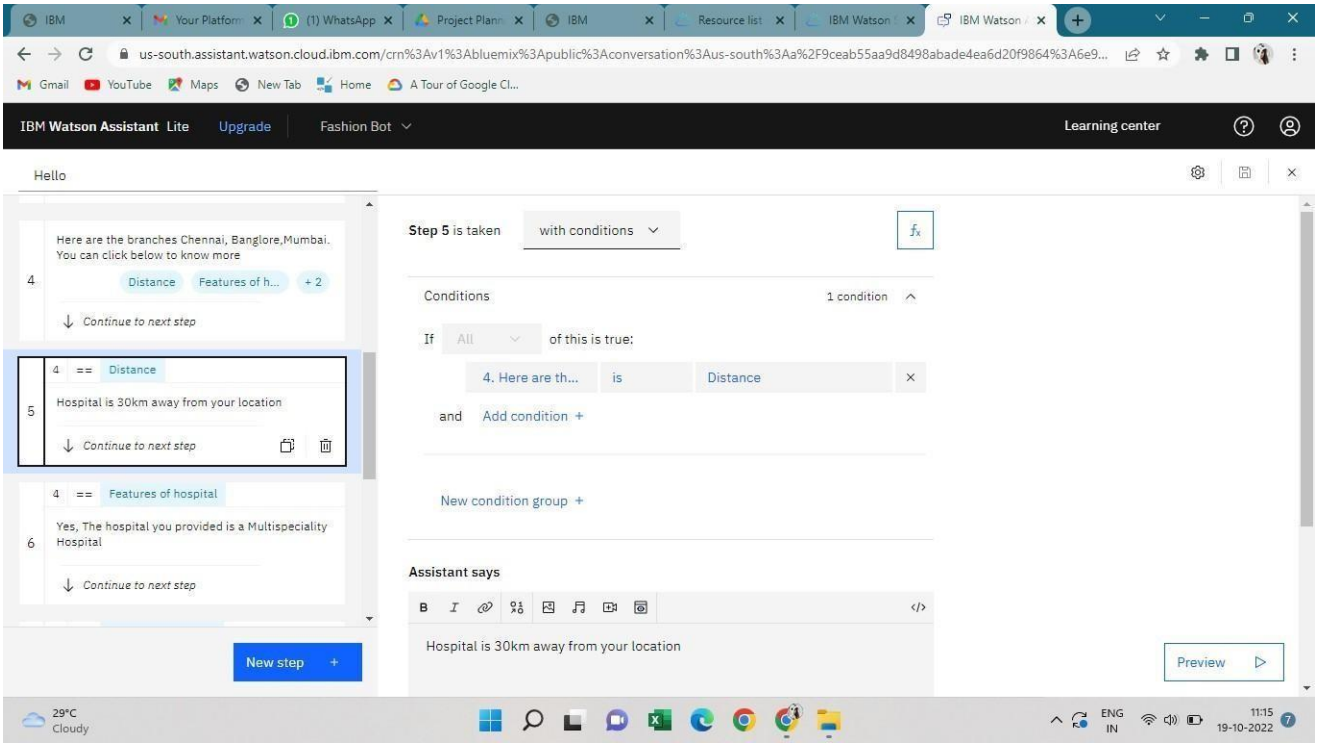
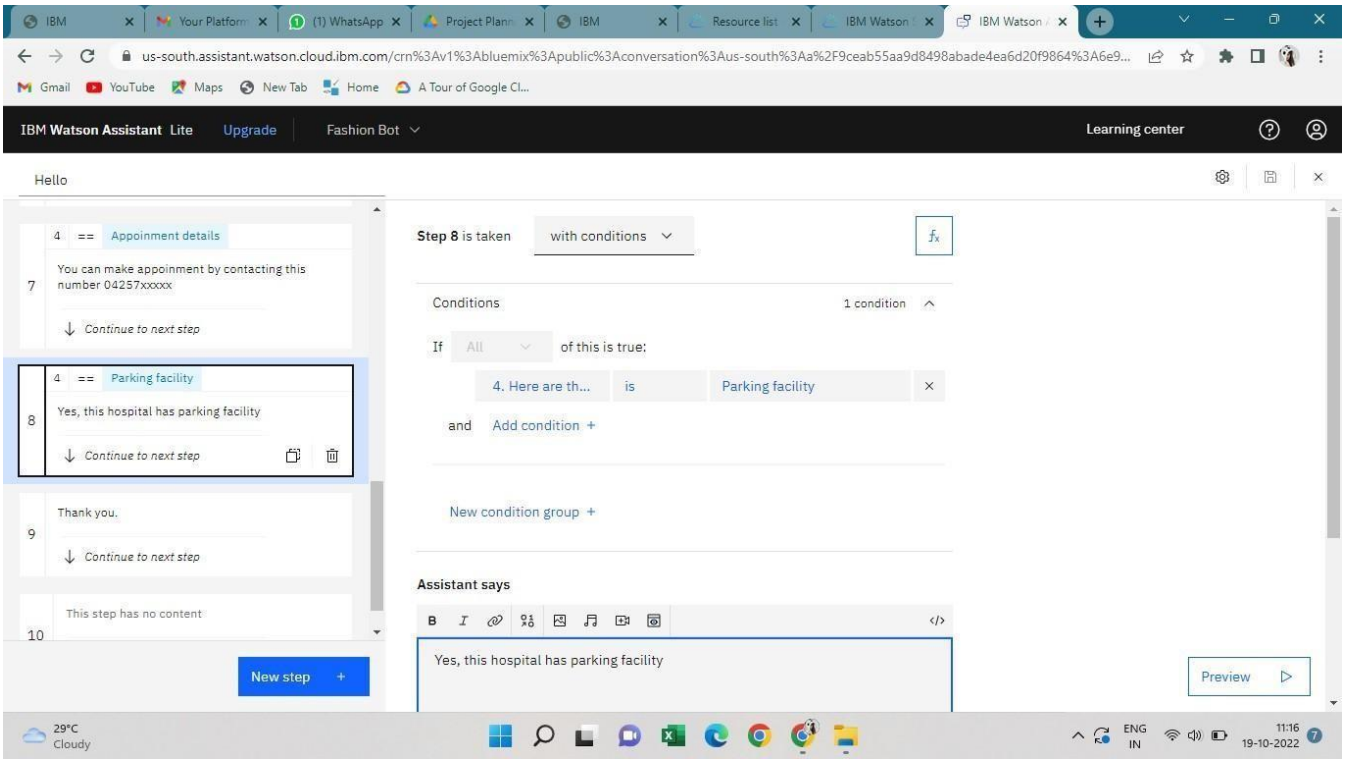


Figure 1. 10 steps of conversation

Included 3 conditions in steps:


IBM Watson Assistant Lite
Upgrade
Fashion Bot
Learning center
Hello
Continue to next step
Good to hear. How can I assist you?
hospital
city
2 == hospital , city
Enter the hospital name (Eg:Apolla hospitals)
Free text
Here are the branches Chennai, Banglore,Mumbai
Step 3 is taken
with conditions
Conditions
2 conditions
If Any of this is true:
2. Good to he... is hospital
or 2. Good to he... is city
or Add condition +
New condition group +
Assistant says
Enter the hospital name (Eg:Apolla hospitals)
New step
Preview


IBM Watson Assistant Lite
Upgrade
Fashion Bot
Learning center
Hello
Here are the branches Chennai, Banglore,Mumbai. You can click below to know more
Distance
Features of h...
+ 2
4 == Distance
Hospital is 30km away from your location
4 == Features of hospital
Yes, The hospital you provided is a Multispeciality Hospital
Step 5 is taken
with conditions
Conditions
1 condition
If All of this is true:
4. Here are th... is Distance
and Add condition +
New condition group +
Assistant says
Hospital is 30km away from your location
New step
Preview



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-objectstorage.appdomain.cloud/cloudbucket/assign3.css')}}" type="text/css">

<script>

  window.watsonAssistantChatOptions = {      integrationID: "14b83b8f-3dfd-405f-9520-b550092892aa", // The ID of this integration.      region: "us-south", // The region your
```

```
integration is hosted in.     serviceInstanceID: "6e95bee9-8d0b-49f6-8a2f-
4125fb3a7945", // 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__)


@a
pp.
rou
te('
```

```
/')
def
ind
ex()
:
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
```