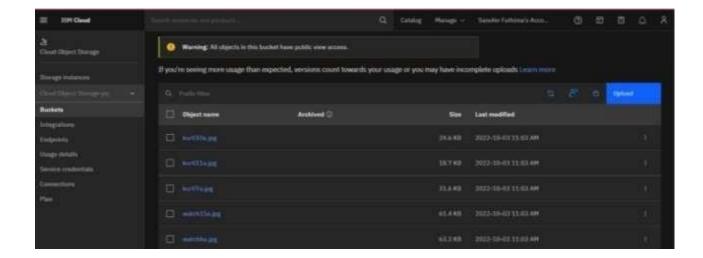
Assignment-3

Date	10 October 2022
Team ID	PNT2022TMID19264
Project Name	Smart Fashion Recommender

- 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.



```
Display.html
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>ASSIGNMENT</title>
k rel="stylesheet" href="{{url_for('redirect_to',link='https://s3.jp-tok.cloud-object-
storage.appdomain.cloud/sanrasheedcss/assign3.css')}}" type="text/css">
</head>
<body>
<h1>DISPLAYING IMAGES FROM CLOUD OBJECT STORAGE</h1>
{% for image in arr: %}
<img src="{{url_for('redirect_to',link=image)}}}">
{% endfor %}
</body>
</html>
App.py
from flask import Flask,render_template,request,redirect,url_for
from connect import *
from objstorage import *
app= Flask( name )
@app.route("/assign3")
def objstorage():
  for i in range(0,len(imagearr)):
    imagearr[i]="https://s3.jp-tok.cloud-object-
storage.appdomain.cloud/sanrasheed29/"+imagearr[i]
    print("objstorage ", imagearr)
    return render_template('assign3.html',arr=imagearr)
@app.route('/redirect_to')
def redirect_to():
  link = request.args.get('link', '/')
```

```
return redirect(link), 301

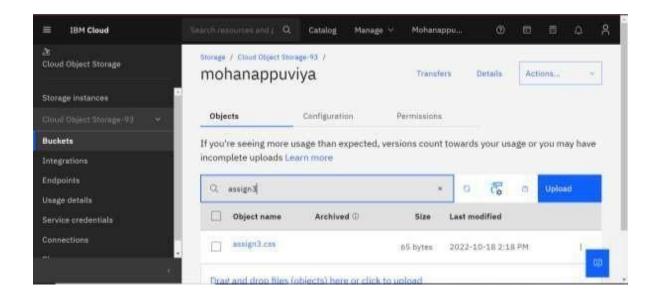
if(_name_=='_main_'):

app.run()
```

Output:

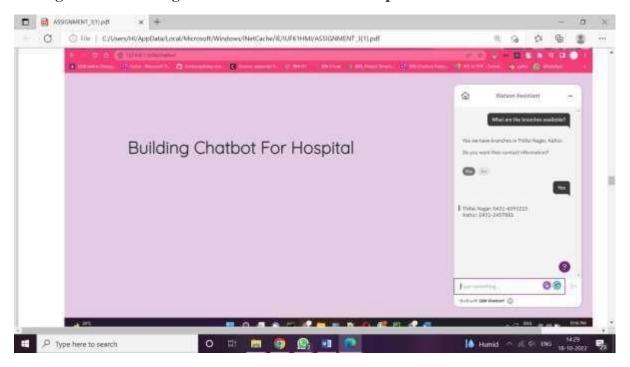


3. Upload a css page to the object storage and use the same page in your HTML code.



```
The Company of the product of the control of the co
```

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



Web URL for Assistant:

```
<script>
```

 $window.watsonAssistantChatOptions = \{\\$

integrationID: "cba0c163-9f9a-42ce-b6e9-979ac6318cc0", // The ID of this integration.

region: "au-syd", // The region your integration is hosted in.

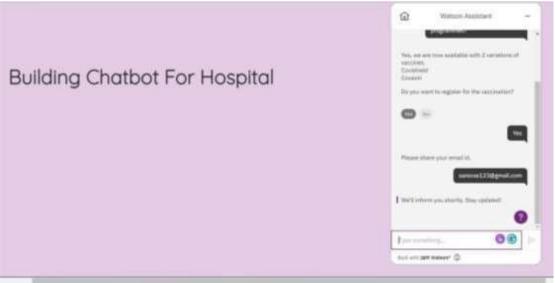
serviceInstanceID: "245858a6-01a2-41fc-8707-d603be17b8f0", // The ID of your service instance.

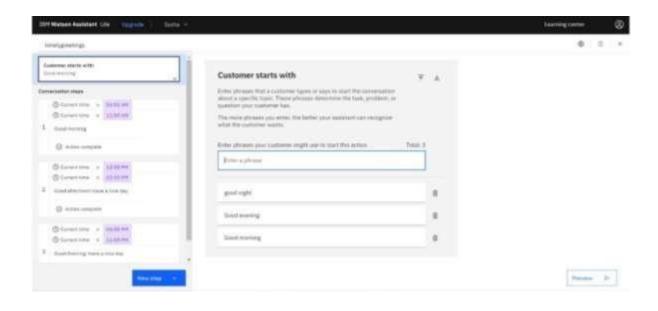
onLoad: function(instance) { instance.render(); }

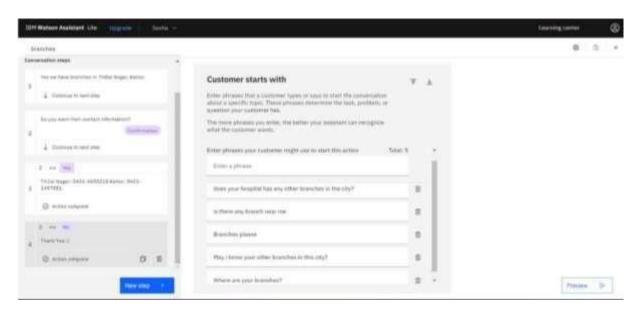
```
};
setTimeout(function(){
const t=document.createElement('script');
t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
document.head.appendChild(t);
});</script>
```

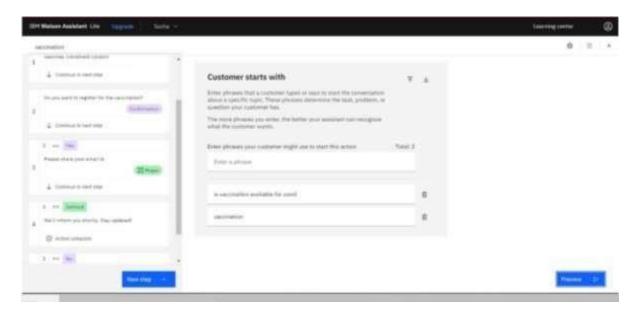
5. Create Watson assistant service with 10 steps and use 3 conditions in it. Load that script in HTML page.











Html code:

```
<!DOCTYPE html>
<html><head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Chatbot </title>
k rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npv4HI+N"
crossorigin="anonymous">
k rel="stylesheet" href="{{url_for('static',filename='styles/login.css')}}" type="text/css">
k rel="preconnect" href="https://fonts.googleapis.com">
k rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<script defer src="https://use.fontawesome.com/releases/v5.13.1/js/all.js"></script>
link
href="https://fonts.googleapis.com/css2?family=Alex+Brush&family=Quicksand:wght@300
&display=swap" rel="stylesheet"></head>
<body style="background:rgba(128,0,128,0.2">
<h1 style="font-weight: bolder; margin: 12%;
font-size: xxx-large;">Building Chatbot For Hospital</h1> <script>
window.watsonAssistantChatOptions = {
integrationID: "cba0c163-9f9a-42ce-b6e9-979ac6318cc0", // The ID of this integration.
region: "au-syd", // The region your integration is hosted in.
serviceInstanceID: "245858a6-01a2-41fc-8707-d603be17b8f0", // 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>
</body></html>
```