

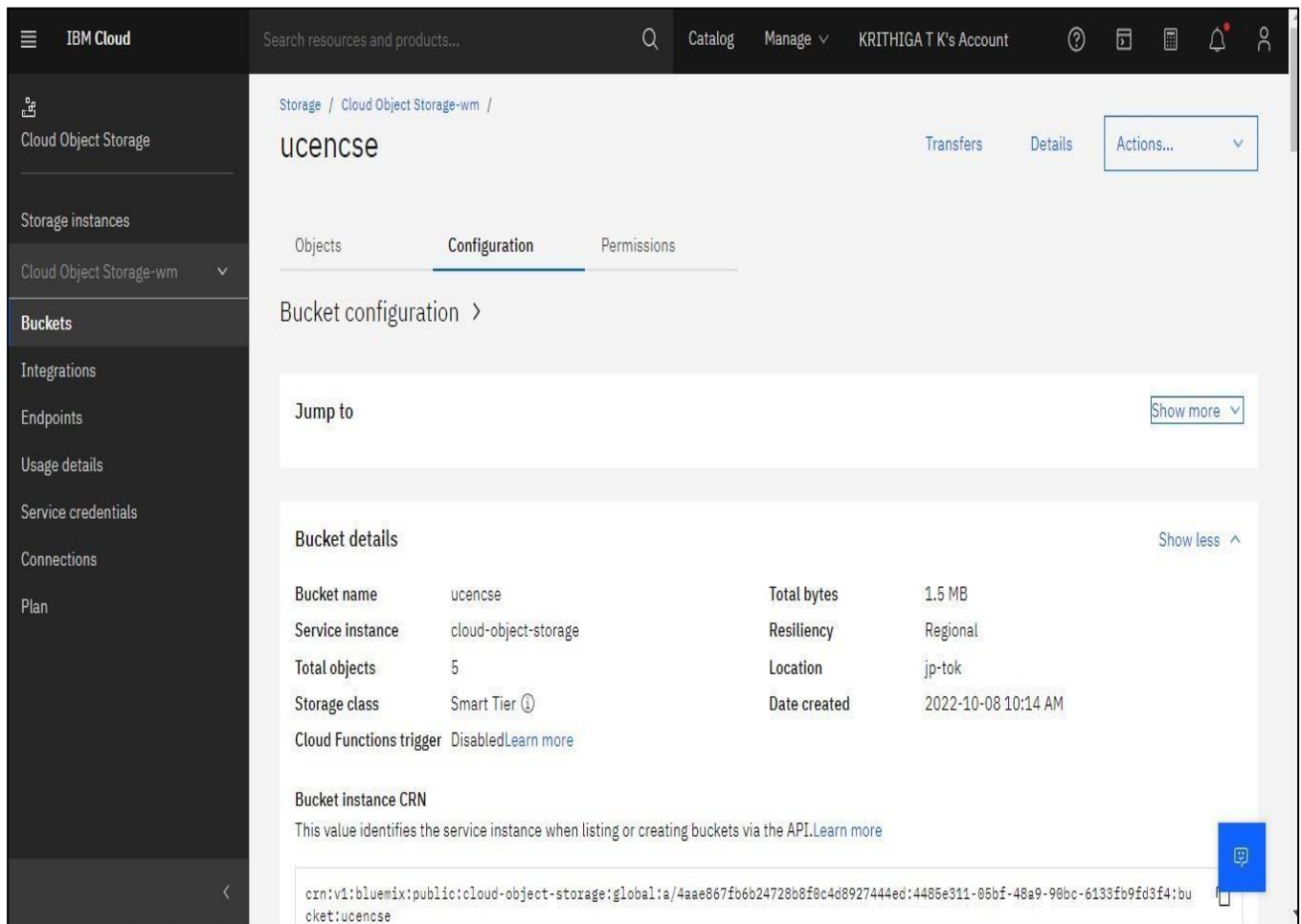
## ASSIGNMENT 3

PROJECT TITLE : SKILL AND JOB RECOMMENDER

TEAM ID : PNT2022TMID11942

### OBJECT STORAGE AND WATSON ASSISTANT IN IBM CLOUD

1. Create a Bucket in IBM object storage.



The screenshot displays the IBM Cloud console interface for managing Cloud Object Storage. The left sidebar shows the navigation menu with 'Buckets' selected. The main content area shows the configuration for a bucket named 'ucencse'. The 'Configuration' tab is active, displaying the 'Bucket configuration' section. Below this, the 'Bucket details' section provides information about the bucket's properties.

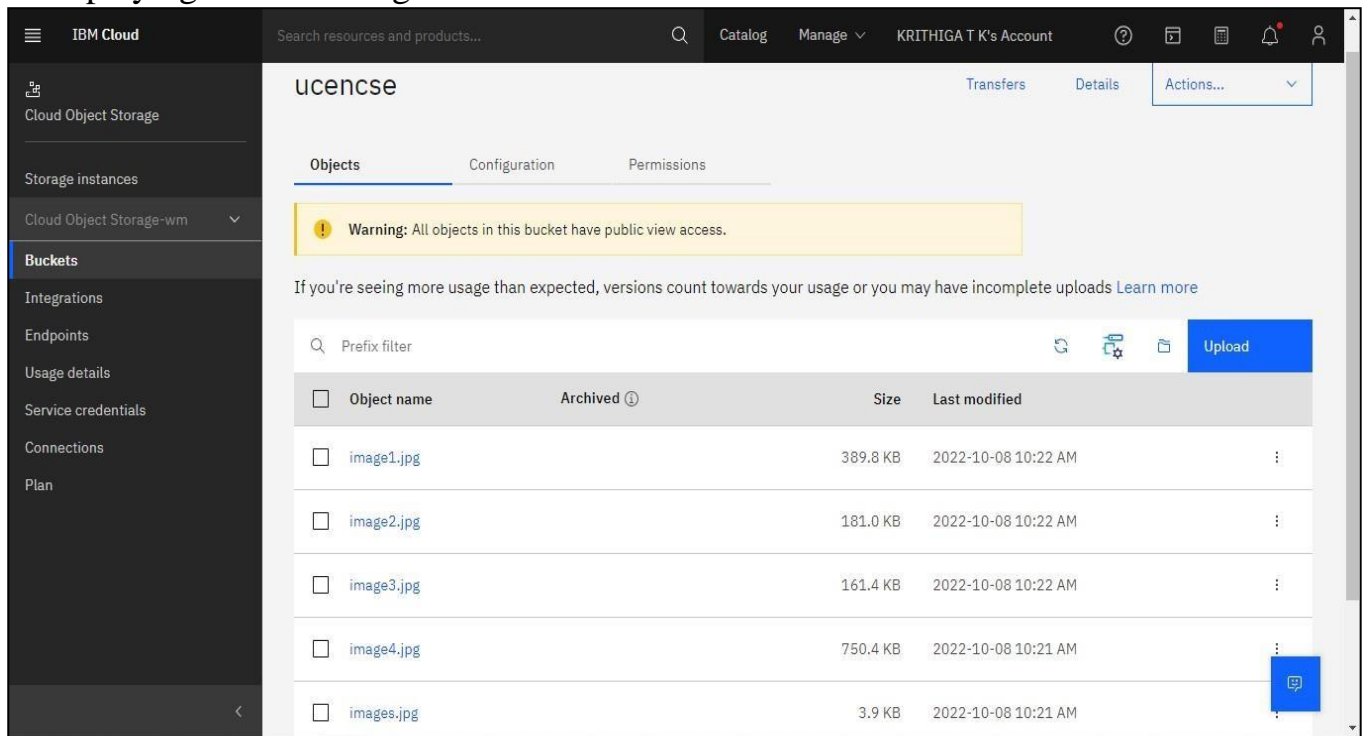
| Bucket details          |                                     |
|-------------------------|-------------------------------------|
| Bucket name             | ucencse                             |
| Service instance        | cloud-object-storage                |
| Total objects           | 5                                   |
| Storage class           | Smart Tier ⓘ                        |
| Cloud Functions trigger | Disabled <a href="#">Learn more</a> |
| Total bytes             | 1.5 MB                              |
| Resiliency              | Regional                            |
| Location                | jp-tok                              |
| Date created            | 2022-10-08 10:14 AM                 |

**Bucket instance CRN**  
This value identifies the service instance when listing or creating buckets via the API. [Learn more](#)

```
crn:v1:bluemix:public:cloud-object-storage:global:a/4aae867fb6b24728b8f0c4d8927444ed:4485e311-85bf-48a9-90bc-6133fb9fd3f4:bucket:ucencse
```

Bucket was created

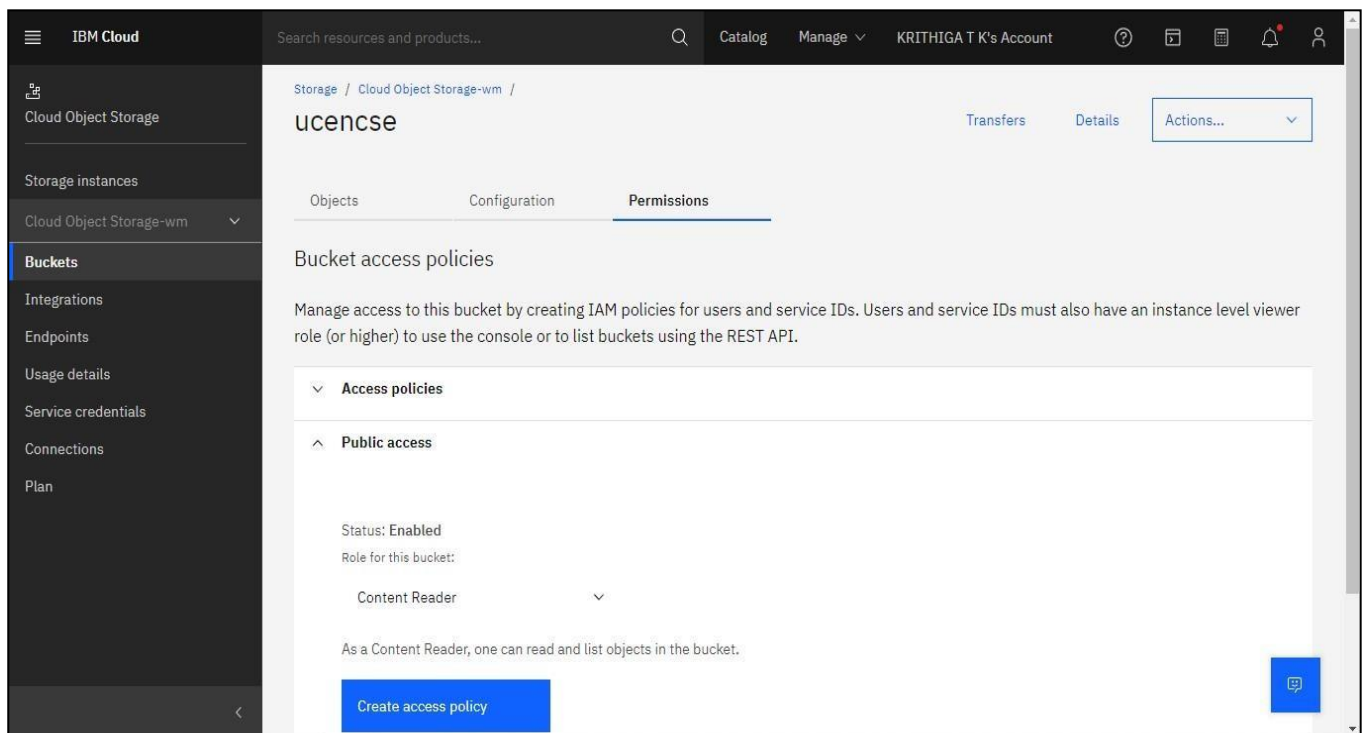
2. Upload 5 images to IBM object storage and make it public. Write html code to displaying all the 5 images.



The screenshot shows the IBM Cloud Object Storage console. The left sidebar contains navigation links: IBM Cloud, Cloud Object Storage, Storage instances, Cloud Object Storage-wm, Buckets, Integrations, Endpoints, Usage details, Service credentials, Connections, and Plan. The main panel displays the 'ucencse' bucket. At the top, there are tabs for 'Objects', 'Configuration', and 'Permissions'. A yellow warning banner states: 'Warning: All objects in this bucket have public view access.' Below this, a message says: 'If you're seeing more usage than expected, versions count towards your usage or you may have incomplete uploads. Learn more'. A search bar with 'Prefix filter' and an 'Upload' button are visible. A table lists the objects:

| <input type="checkbox"/> | Object name | Archived ⓘ | Size     | Last modified       |   |
|--------------------------|-------------|------------|----------|---------------------|---|
| <input type="checkbox"/> | image1.jpg  |            | 389.8 KB | 2022-10-08 10:22 AM | ⋮ |
| <input type="checkbox"/> | image2.jpg  |            | 181.0 KB | 2022-10-08 10:22 AM | ⋮ |
| <input type="checkbox"/> | image3.jpg  |            | 161.4 KB | 2022-10-08 10:22 AM | ⋮ |
| <input type="checkbox"/> | image4.jpg  |            | 750.4 KB | 2022-10-08 10:21 AM | ⋮ |
| <input type="checkbox"/> | images.jpg  |            | 3.9 KB   | 2022-10-08 10:21 AM | ⋮ |

5 images was uploaded in the cloud object storage



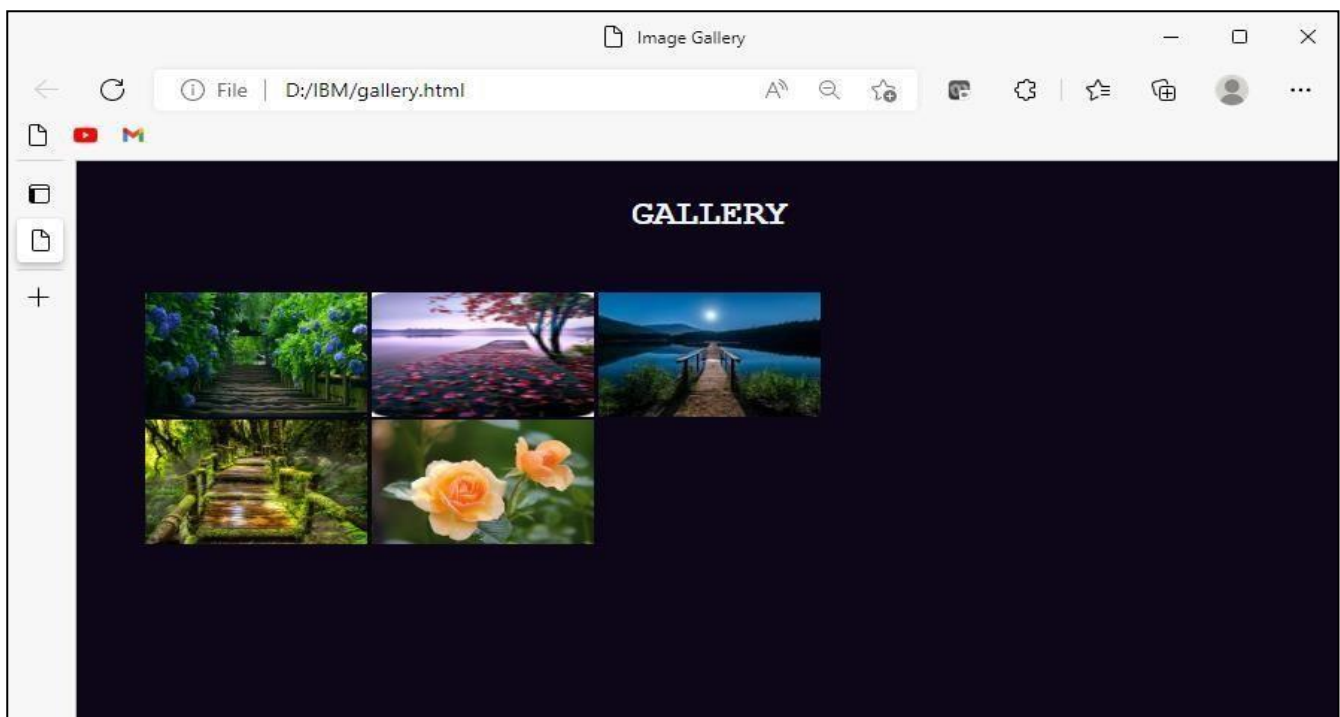
The screenshot shows the IBM Cloud Object Storage console, specifically the 'Permissions' tab for the 'ucencse' bucket. The left sidebar is the same as the previous screenshot. The main panel shows the 'Permissions' tab. A section titled 'Bucket access policies' contains the text: 'Manage access to this 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.' Below this, there is a section for 'Access policies' with a sub-section 'Public access'. The 'Public access' section shows 'Status: Enabled' and 'Role for this bucket: Content Reader'. A note states: 'As a Content Reader, one can read and list objects in the bucket.' A blue button labeled 'Create access policy' is at the bottom.

Public access is enabled

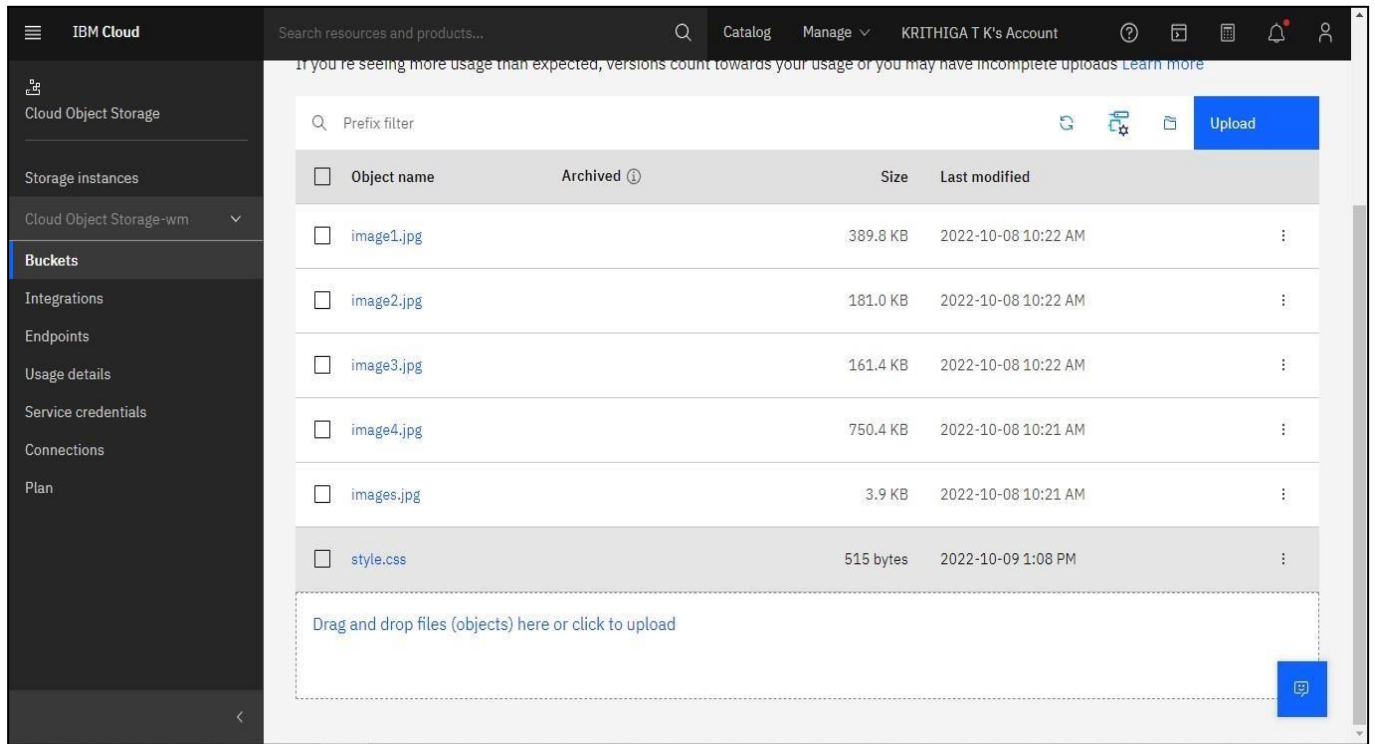
## Gallery.html

```
1  <html>
2  <head>
3    <title>Image Gallery</title>
4    <style type="text/css">
5      body{
6        background-color: rgb(13, 6, 24);
7      }
8      #gallery{
9        margin-top: 10%;
10       margin: 5%;
11       line-height:0;
12       column-count:5;
13       column-gap:5px;
14     }
15     #gallery img{
16       width: 100% !important;
17       height: 40% !important;
18       margin-bottom:5px;
19     }
20     h2{
21       text-align: center;
22       font-family: Courier;
23       font-size: 50px;
24       margin-top: 3%;
25       font-weight: bolder;
26       color: aliceblue;
27     }
28   </style>
29 </head>
30 <body>
31   <h2>GALLERY</h2>
32   <div id="gallery">
33     
34     
35     
36     
37     
38   </div>
39 </body>
40 </html>
```

## Output



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



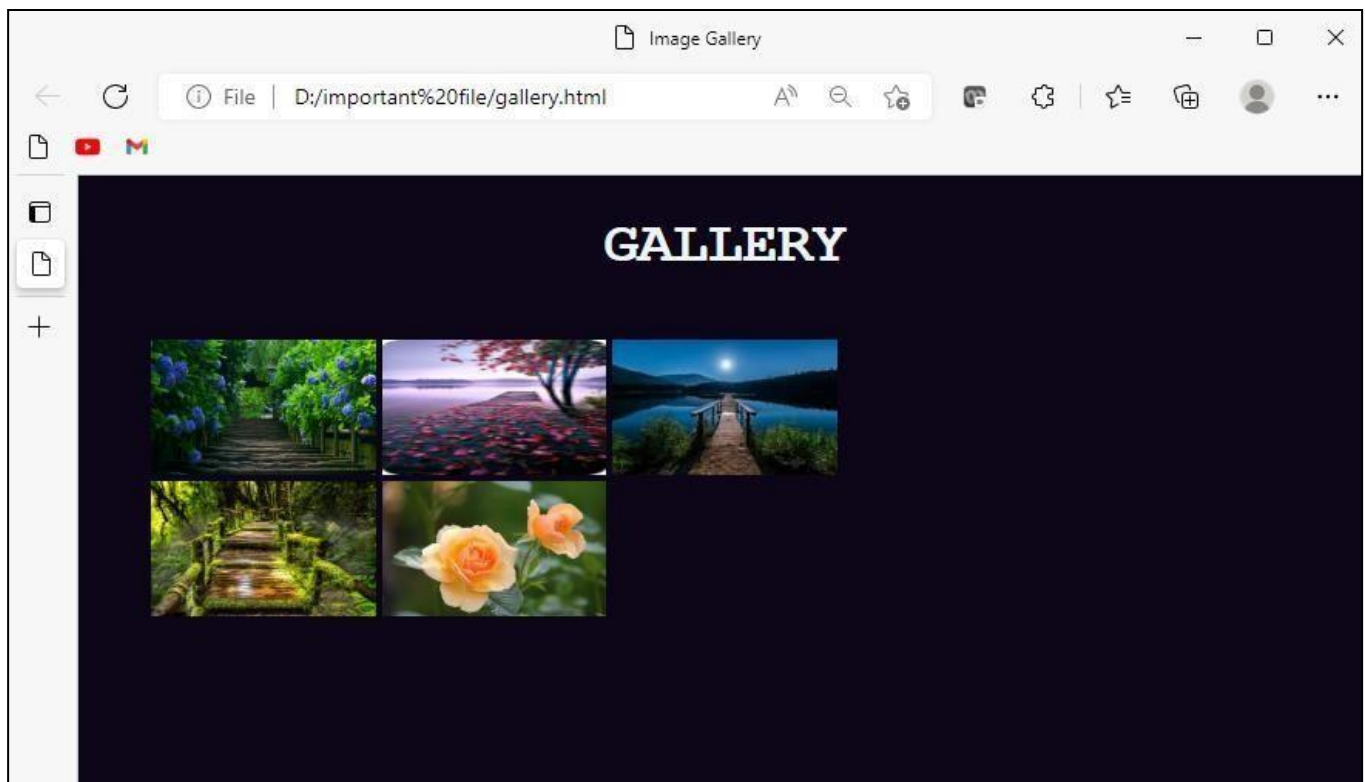
### Style.css



## Gallery.html

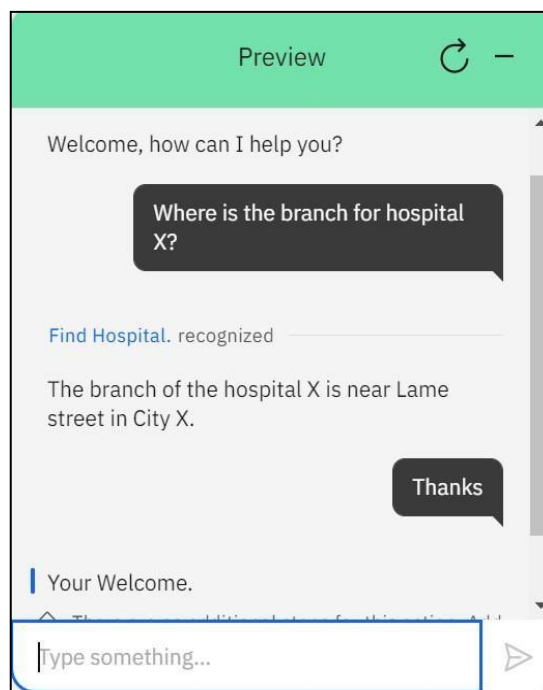
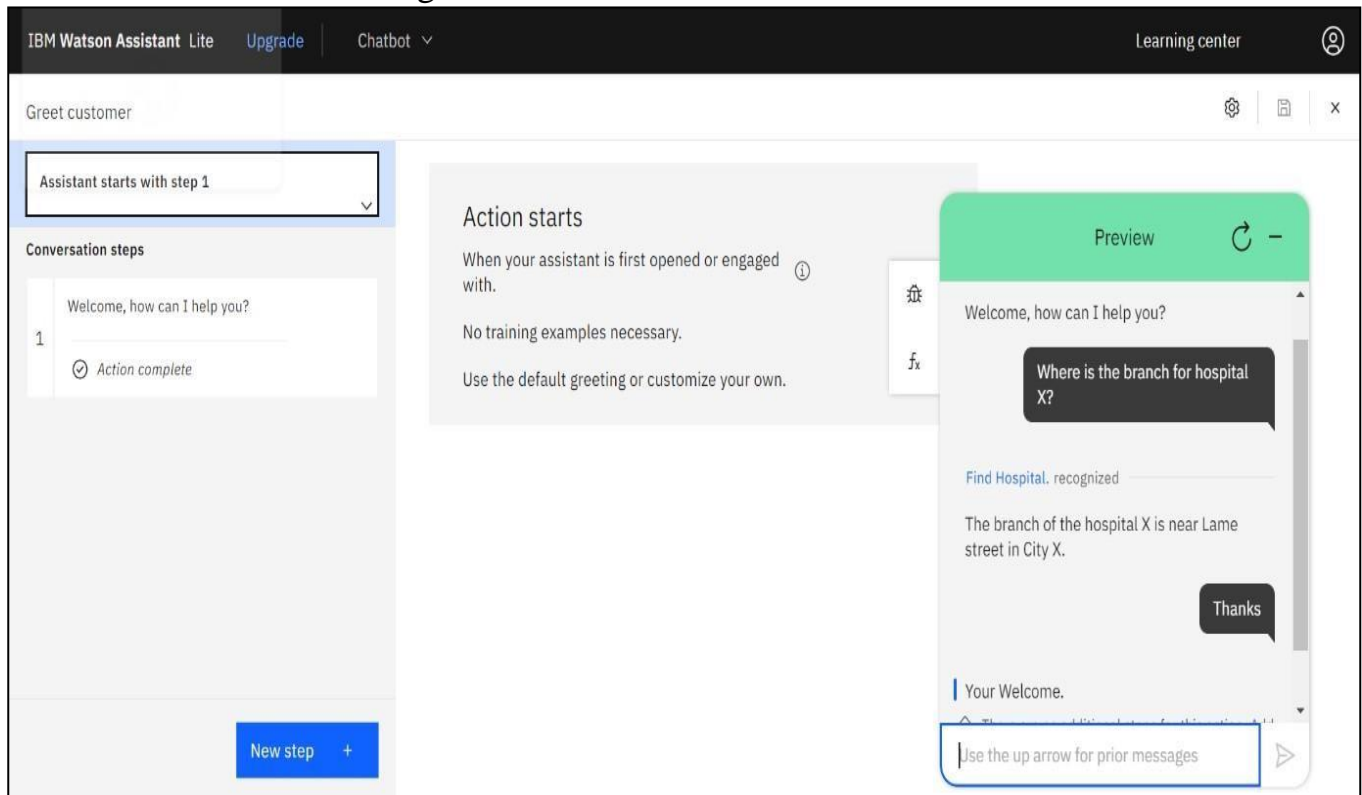
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Image Gallery</title>
5     <link rel="stylesheet" href="https://ucencse.s3.jp-tok.cloud-object-storage.appdomain.cloud/style.css">
6   </head>
7   <body>
8     <h2>GALLERY</h2>
9     <div id="gallery">
10      
11      
12      
13      
14      
15    </div>
16  </body>
17 </html>
```

## Output





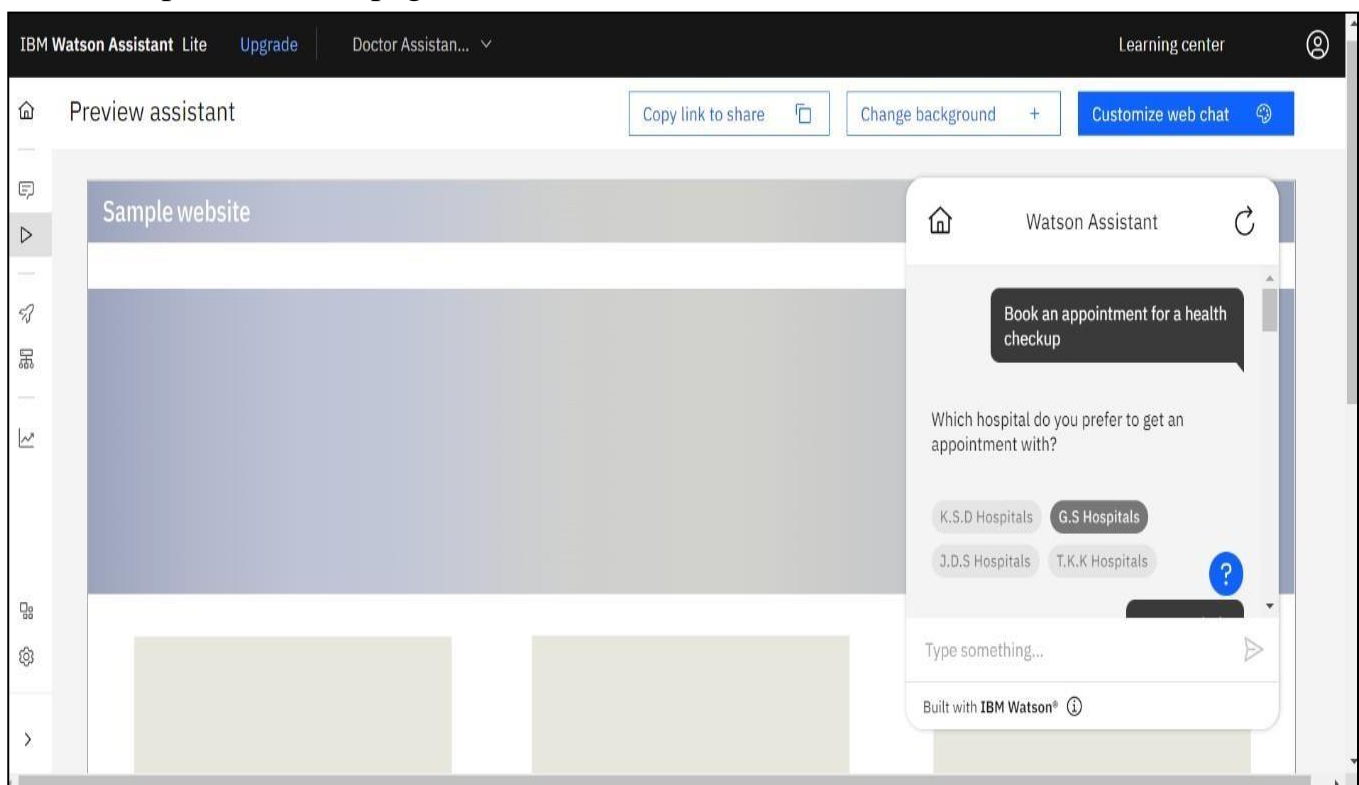
4. Design a chat bot using IBM Watson assistant for a hospital. Example the user comes with a query to know the branches of that hospital in your city. Submit the web URL of that chat bot as an assignment.

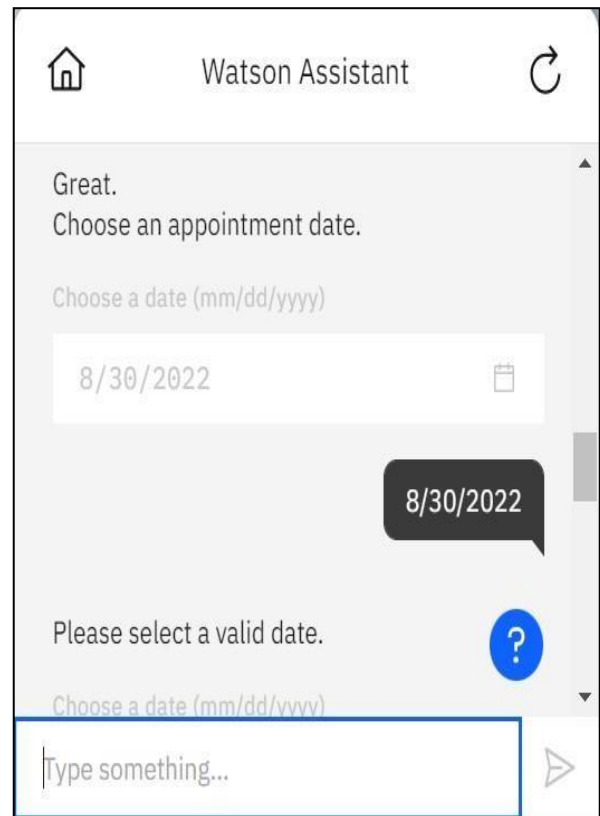
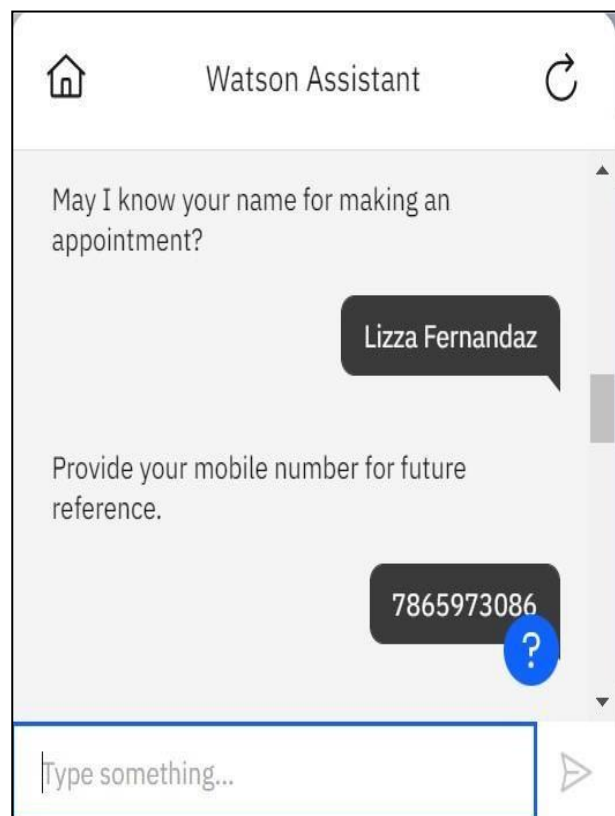
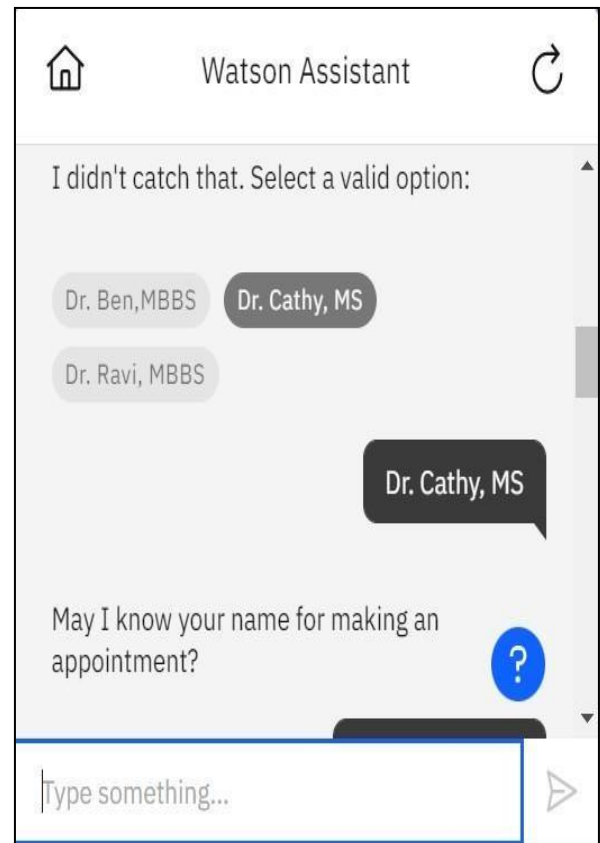
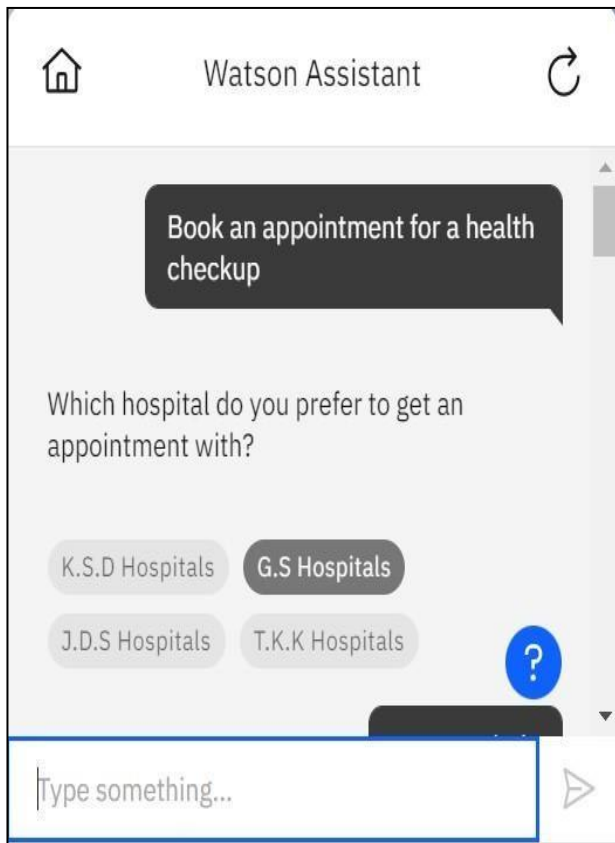


## Chatbot Access URL

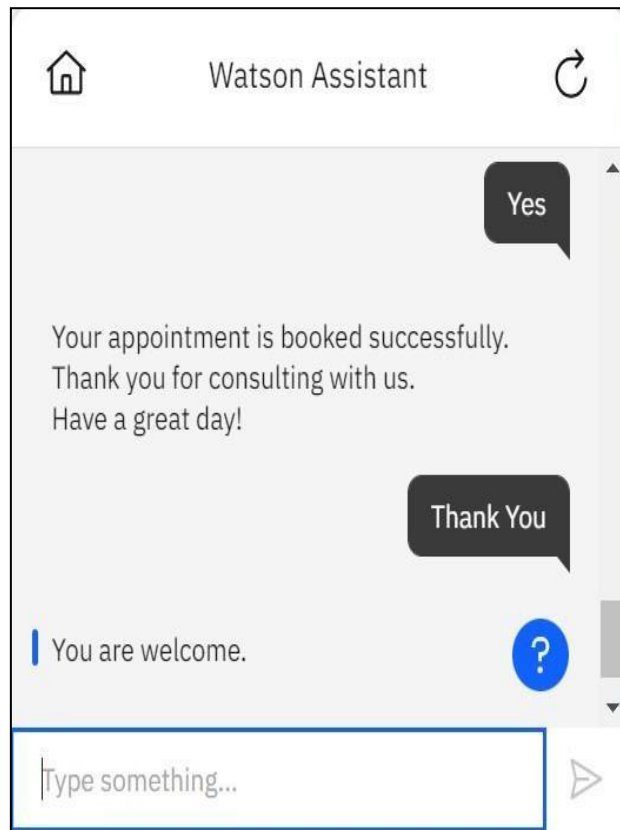
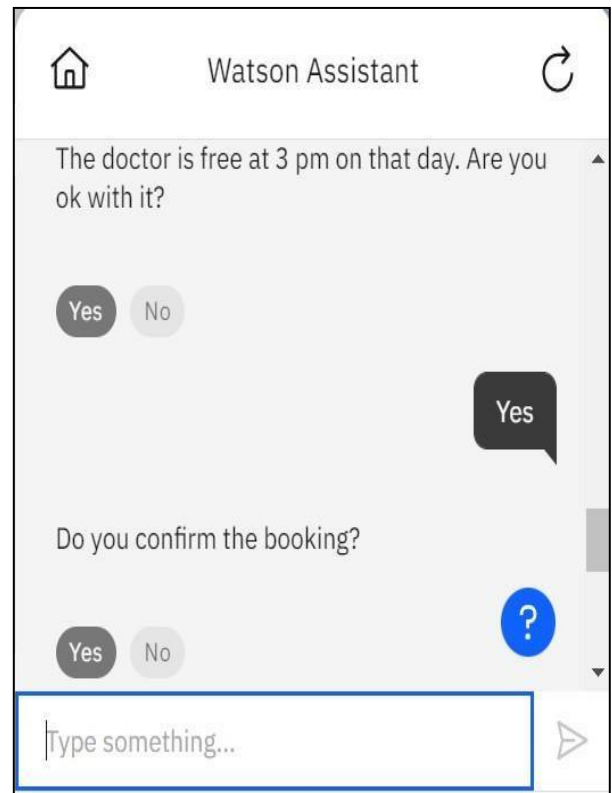
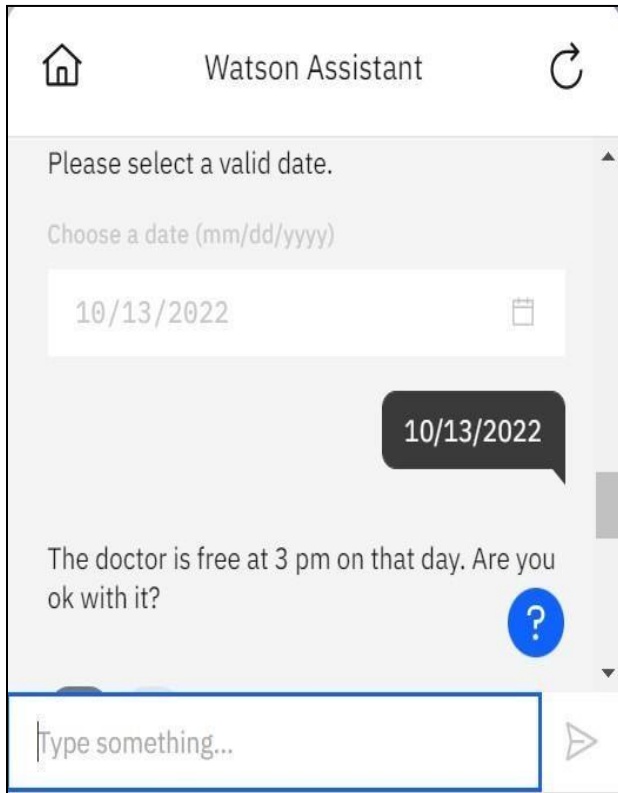
<https://webchat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fjp-tok.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupxc1c094d9-c048-4877-b09e-202b2a623fda%3A%3A28be94d0-4a06-4bb7-a0af358479d049ca&integrationID=640b8a9b-167a-4301-8cf9-f956c3444e5f&region=jptok&serviceInstanceID=c1c094d9-c048-4877-b09e-202b2a623fda>

5. Create a Watson assistant service with ten steps and use three conditions in it. Load that script in the html page.









## Watsonassistantchatbot.html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Home Page</title>
5 </head>
6 <body>
7 <script>
8     window.watsonAssistantChatOptions = {
9         integrationID: "0edf8511-f378-498b-9540-7bcf7e985396", // The ID of this integration.
10        region: "jp-tok", // The region your integration is hosted in.
11        serviceInstanceID: "c1c094d9-c048-4877-b09e-202b2a623fda", // The ID of your service instance.
12        onLoad: function(instance) { instance.render(); }
13    };
14    setTimeout(function(){
15        const t=document.createElement('script');
16        t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
17        document.head.appendChild(t);
18    });
19 </script>
20 </body>
21 </html>
```

## Chatbot access URL

<https://webchat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageUrl=https%3A%2F%2Fjp-tok.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupxc1c094d9-c048-4877-b09e-202b2a623fda%3A%3A970fcc61-3fe8-4aa3-962b7c9d376ceb12&integrationID=0edf8511-f378-498b-9540-7bcf7e985396&region=jptok&serviceInstanceID=c1c094d9-c048-4877-b09e-202b2a623fda>