

ASSIGNMENT - 3

Date	11 Nov 2022
Team ID	PNT2022TMID37814
Project Name	News Tracker Application
Maximum Marks	2 Marks

1. OBJECT STORAGE AND WASTON IN IBM CLOUD

1. Create a Bucket in IBM object storage

The screenshot displays the IBM Cloud Object Storage console. The left sidebar shows the navigation menu with 'Buckets' selected. The main content area is titled 'Bucket configuration' and includes a 'Jump to' section with links to 'Bucket details', 'Endpoints', 'Versioning', 'Key management', 'Activity Tracker', 'Monitoring', 'Quota enforcement', 'Lifecycle policy', and 'Cross-Origin Resource Sharing (CORS) rules'. Below this, the 'Bucket details' section shows the following information:

Bucket name	gallerysite	Total bytes	529.6 KB
Service instance	cloud-object-storage	Resiliency	Single Site
Total objects	7	Location	ams03
Storage class	Smart Tier ⓘ	Date created	2022-11-03 7:58 AM

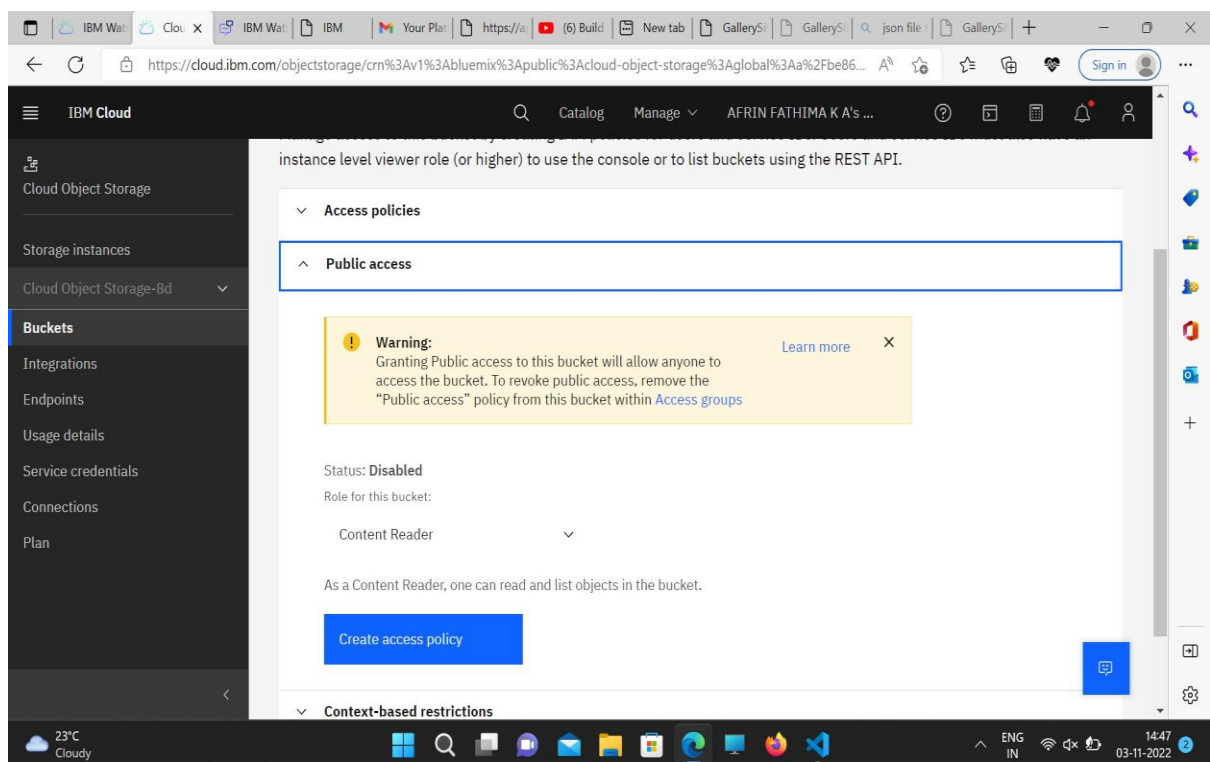
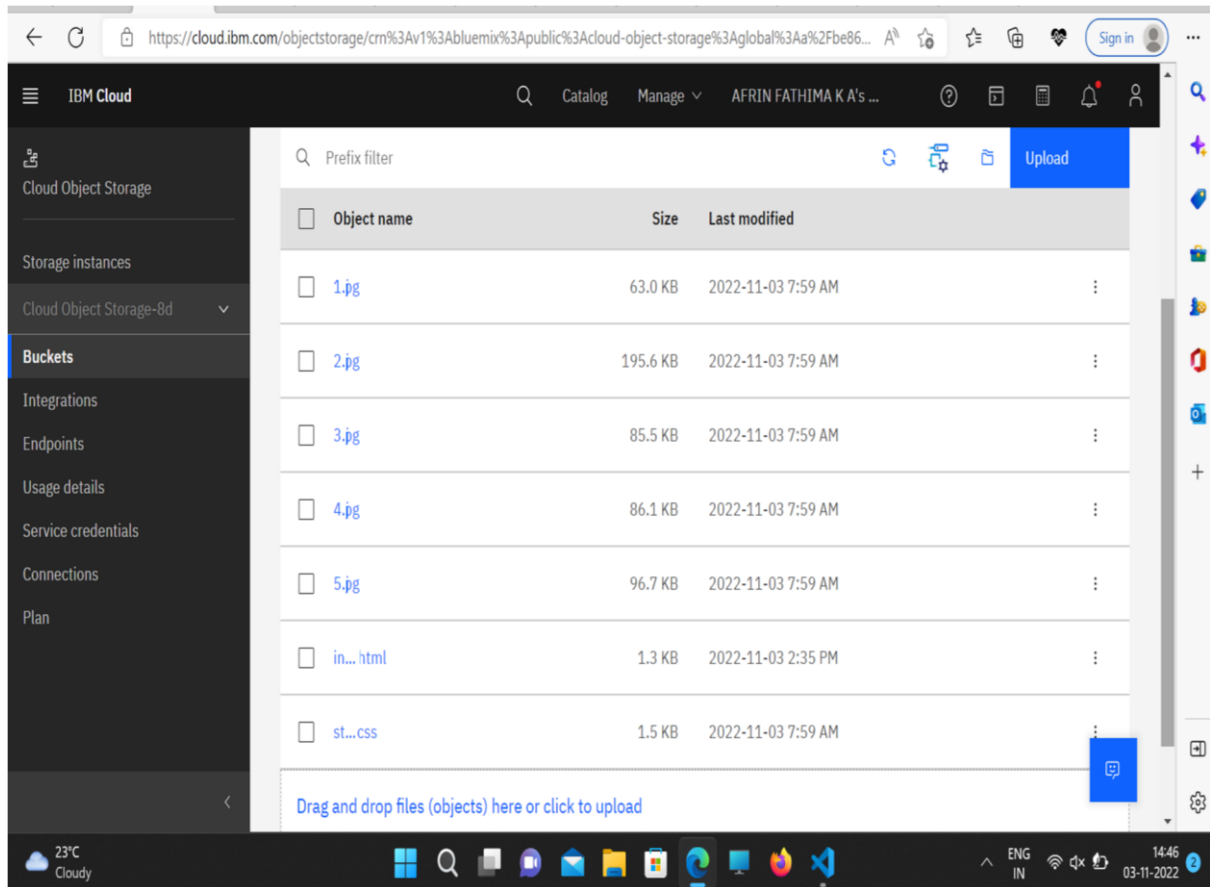
Cloud Functions trigger: Disabled [Learn more](#)

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/be06bbdf138947359c1b3b44bcaa15ac:5533e6da-e155-44a9-90

Bucket was created

2. Upload 5 image to IBM object storage and make it public .write html code to Displaying all the 5 images.



Public access is enable

Index.html

```
<!DOCTYPE html>

<html>

<head>

  <title>GallerySite</title>

  <link rel="stylesheet" type="text/css" href="style.css">

</head> <script>  window.watsonAssistantChatOptions = {  integrationID: "bbbe859b-e53d-4b07-
b175-ee06bf004253", // The ID of this integration.  region: "us-south", // The region your
integration is hosted in.  serviceInstanceID: "07268260-5b95-4281-bb5e-eb1a1db0159a", // 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>

  <div class="container">

    <div class="heading">

      <h3>Photo <span>Gallery</span></h3>

    </div>

    <div class="box">

      <div class="dream">



      </div>

    </div>

  </div>

</body>

</html>
```

```
<div class="dream">

</div>

<div class="dream">

</div>

</div>

<div class="btn">

  <a href="#">More</a>

</div>

</div>

</body>

</html>
```

STYLE.CSS

```
*{ margin: 0; padding: 0; box-
sizing: border-box; font-family:
'Poppins', sans-serif; }
```

```
.container{

  display: flex; flex-
direction: column; justify-
content: center; align-
items: center; text-align:
center; margin: 40px 20px
0 20px; }
```

```
.container .heading{

width: 50%; padding-
bottom: 50px; }
```

```
.container .heading h3{
```

```
font-size: 3em; font-weight:
bolder; padding-bottom: 10px;
border-bottom: 3px solid #222; }
```

```
.container .heading h3 span{ font-
weight: 100; }
```

```
.container .box{
display: flex; flex-direction:
row; justify-content: space-
between; }
```

```
.container .box .dream{
display: flex; flex-direction:
column; width: 32.5%; }
```

```
.container .box .dream img{
width: 100%; padding-bottom:
15px; border-radius: 5px; }
```

```
.container .btn{
margin: 40px 0 70px 0;
background: #222;
padding: 15px 40px ;
border-radius: 5px; }
```

```
.container .btn a{
color: #fff; font-
size: 1.2em; text-
decoration: none;
font-weight: bolder;
letter-spacing: 3px; }
@media only screen
```

```

and (max-width:
769px){

    .container .box{ flex-
direction: column; }

.container .box .dream{
width: 100%; }

/* responsive */
}

@media only screen and (max-width: 643px) {

.container .heading {
width: 100%;

}

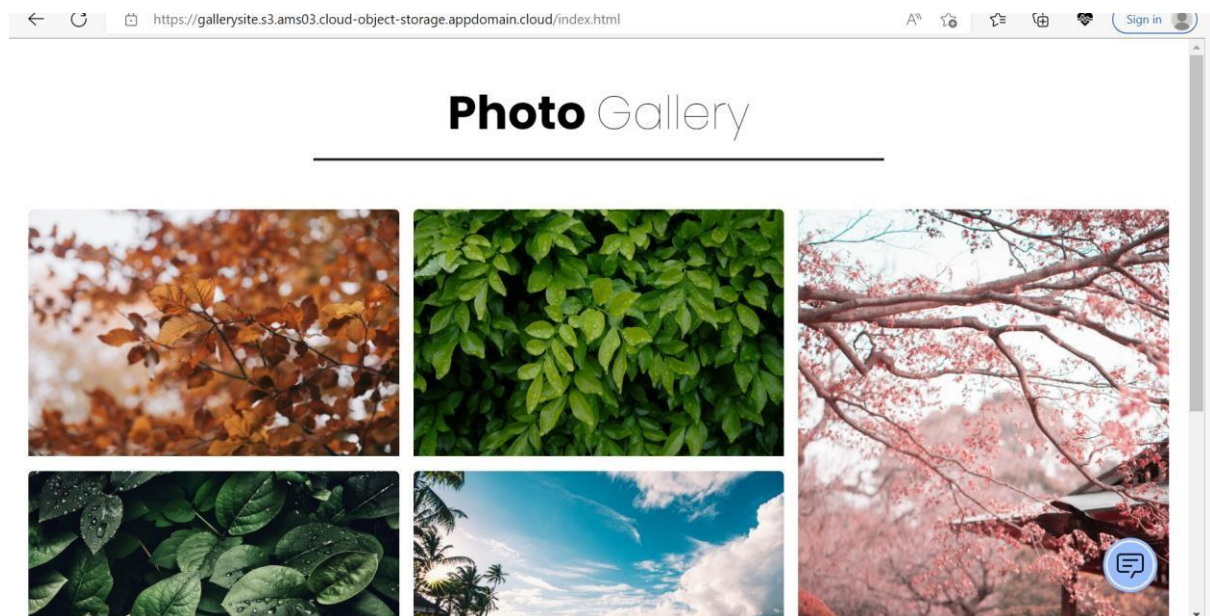
.container .heading h3 { font-
size: 1em;

}

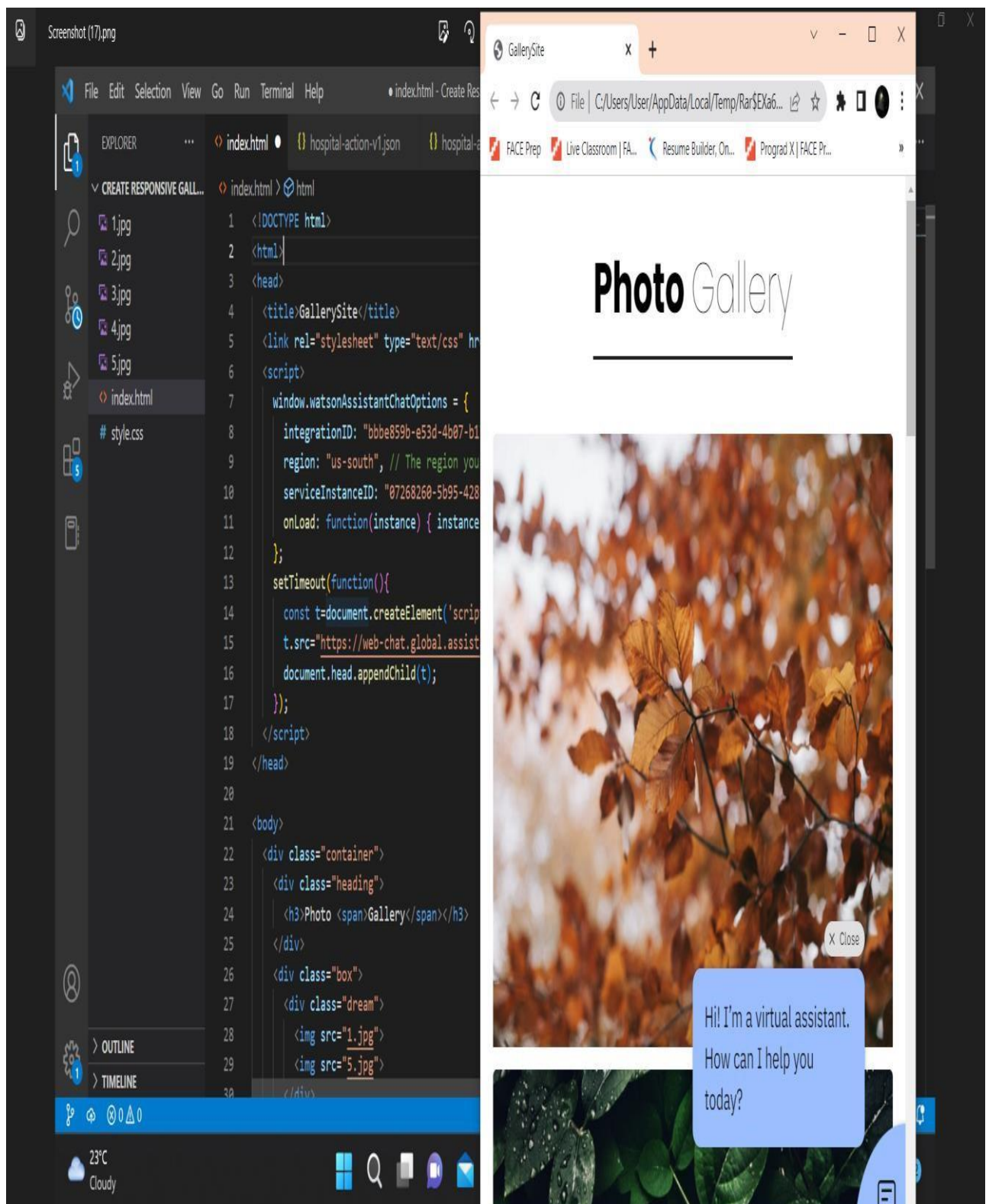
}

```

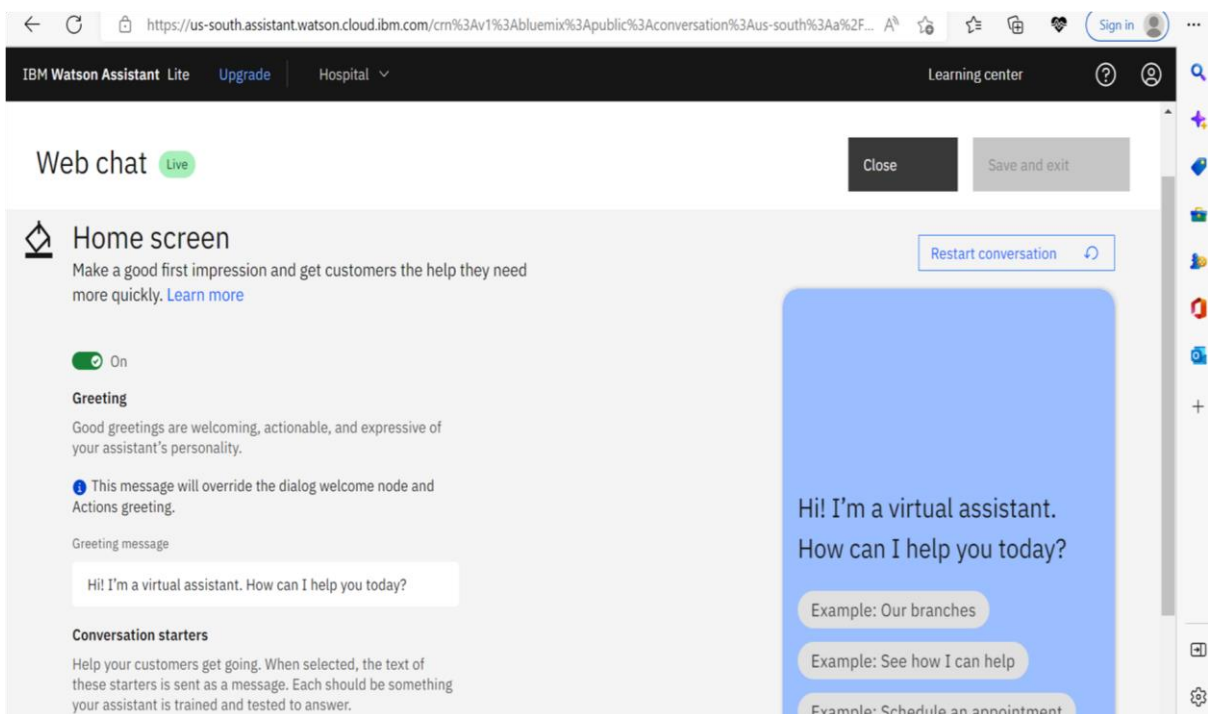
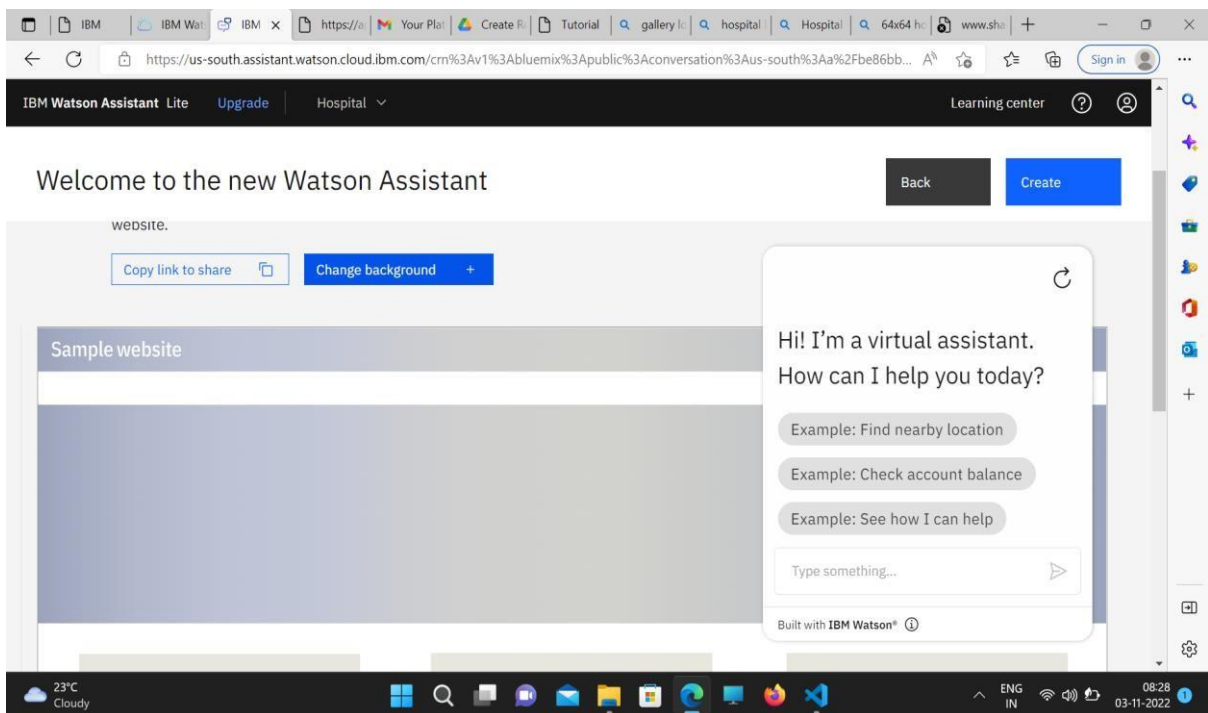
OUTPUT:



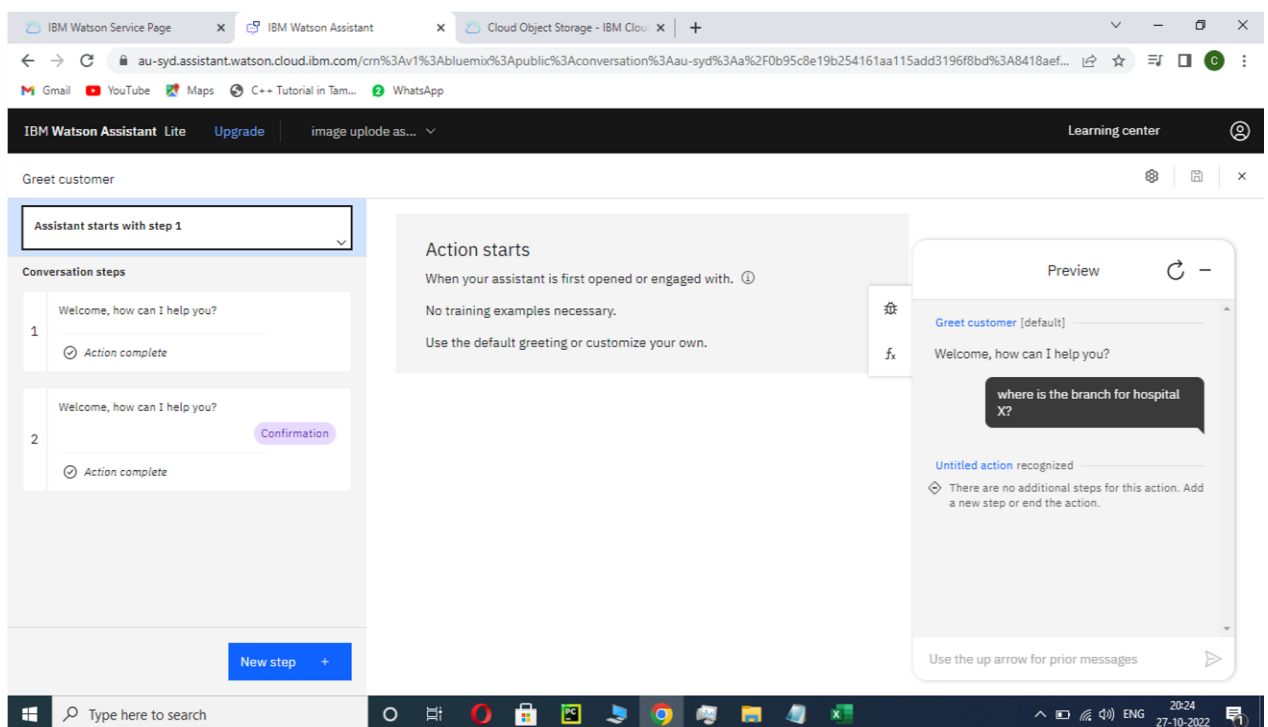
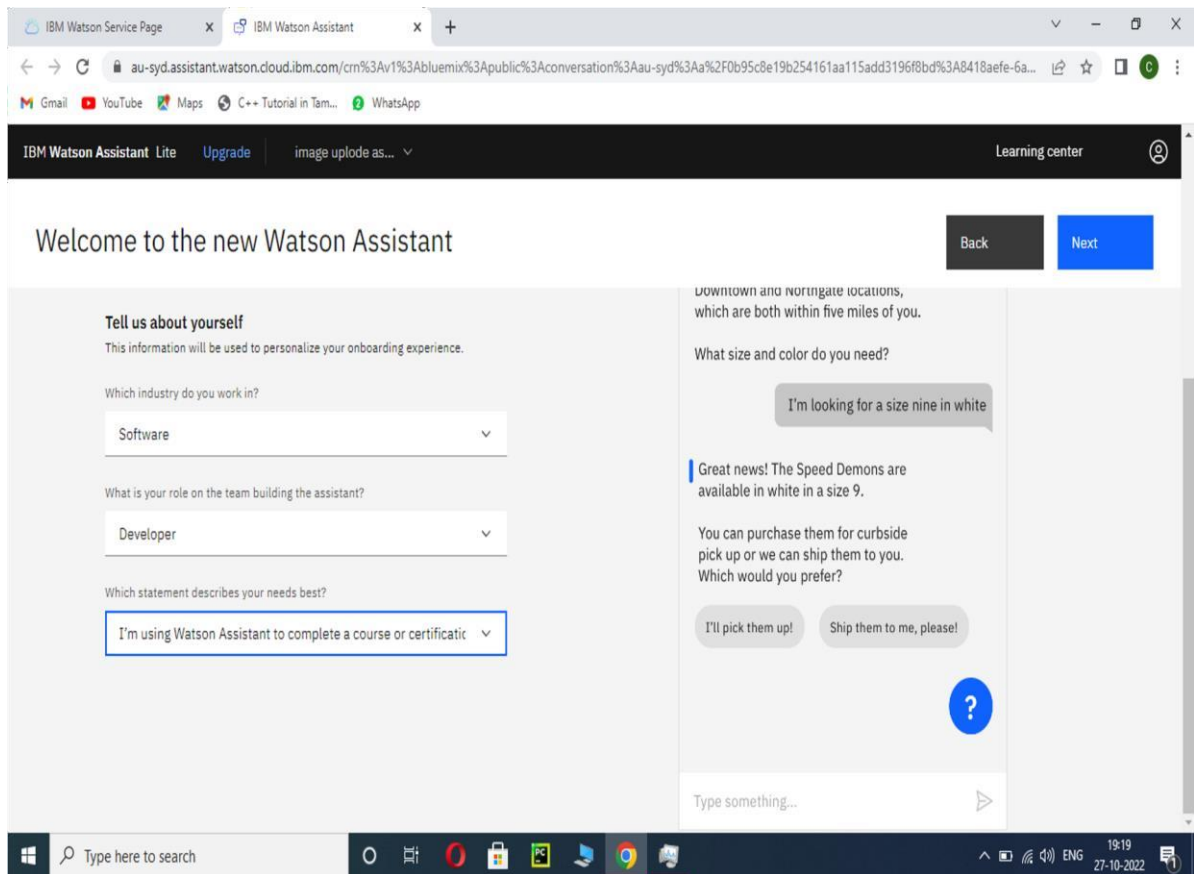
3. Upload css page to the storage and use the same in your html

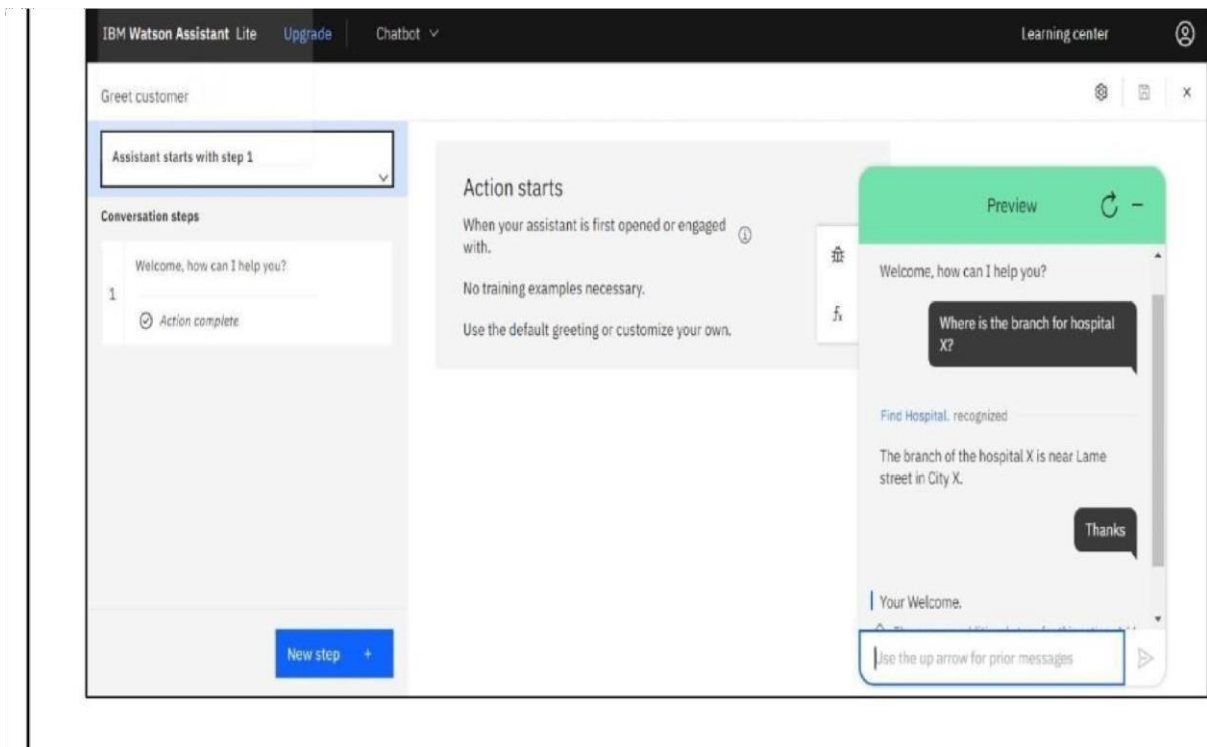
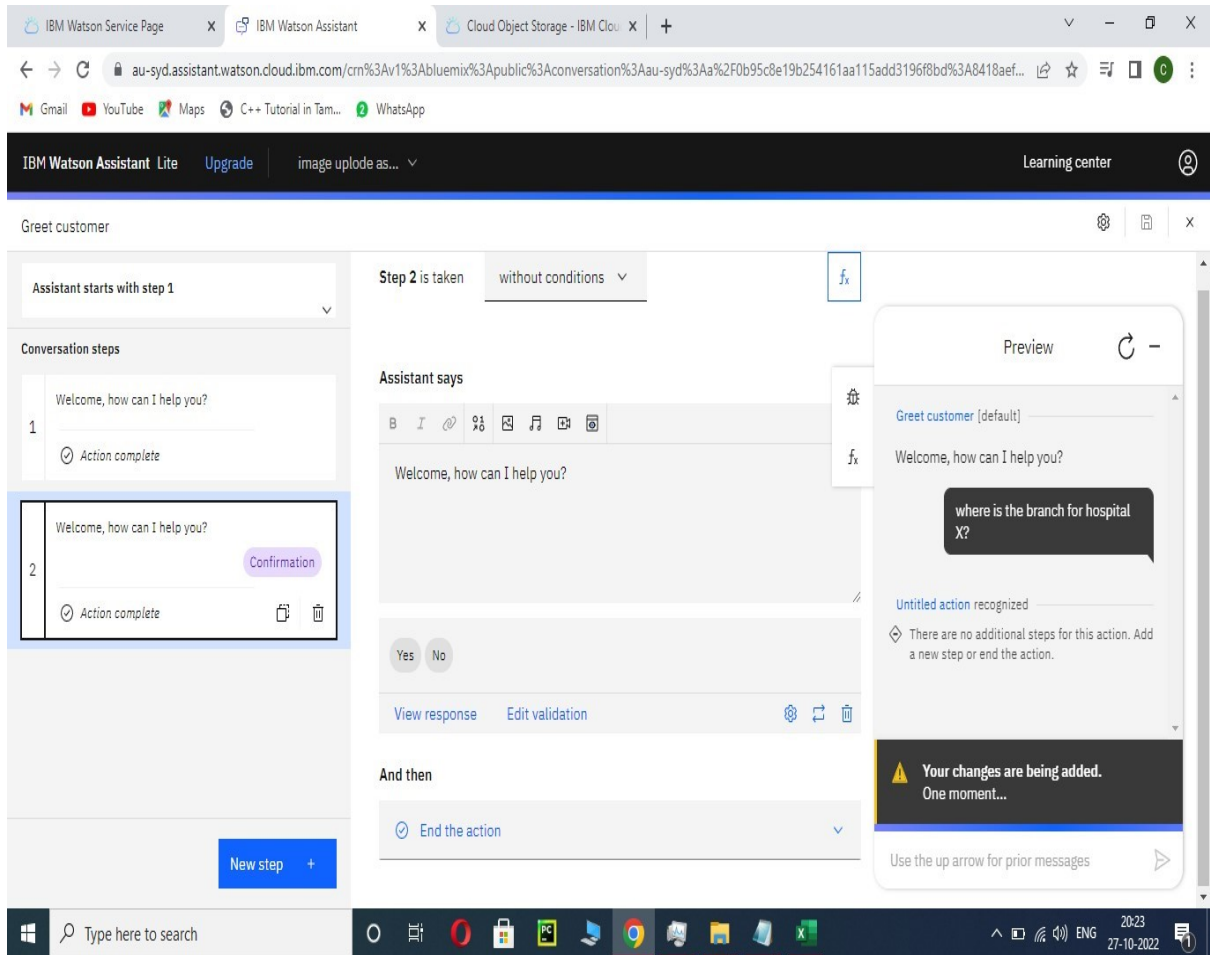


4.Design a chat bot using Watson assistant for a hospital .Example the user come with a query to know the branches of that hospital in you city. Submit the web URL of that chat bot as an assignment.



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





Chatbox Access URL

<https://gallerysite.s3.ams03.cloud-object-storage.appdomain.cloud/index.html>

Watson assistant chatbox.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>Document</title>

  <script>

    window.watsonAssistantChatOptions = {

      integrationID: "bbbe859b-e53d-4b07-b175-ee06bf004253", // The ID of this integration.

      region: "us-south", // The region your integration is hosted in.

      serviceInstanceID: "07268260-5b95-4281-bb5e-eb1a1db0159a", // 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>

</body>

</html>
```