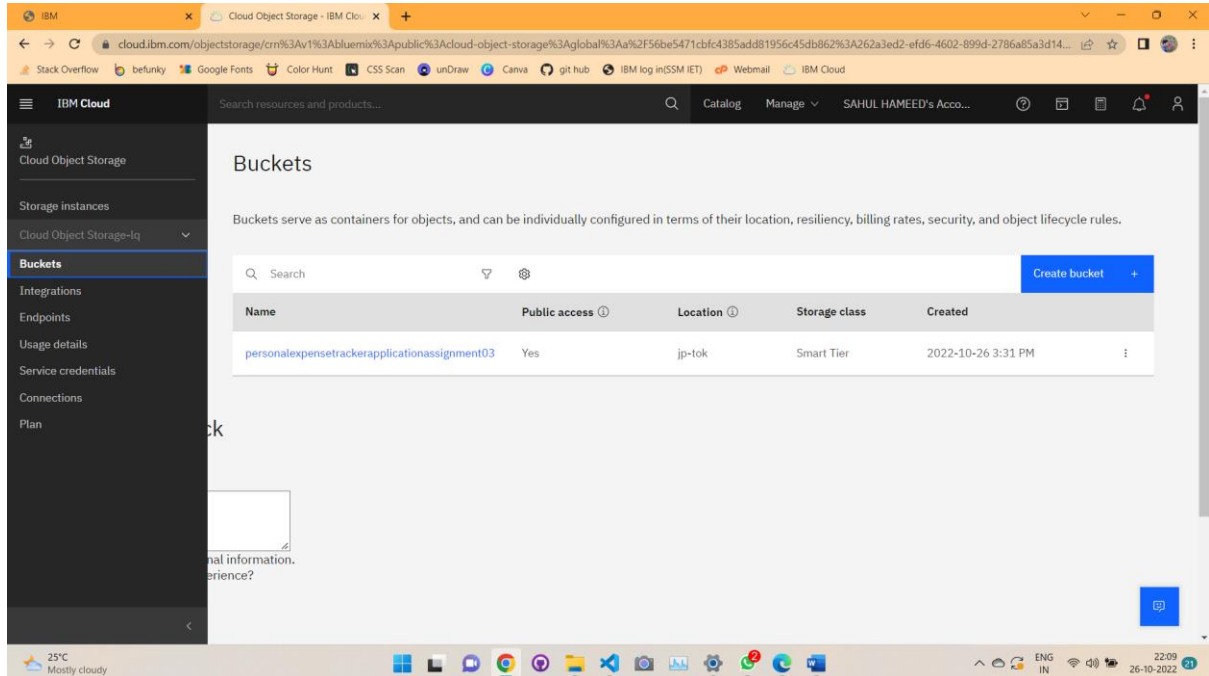
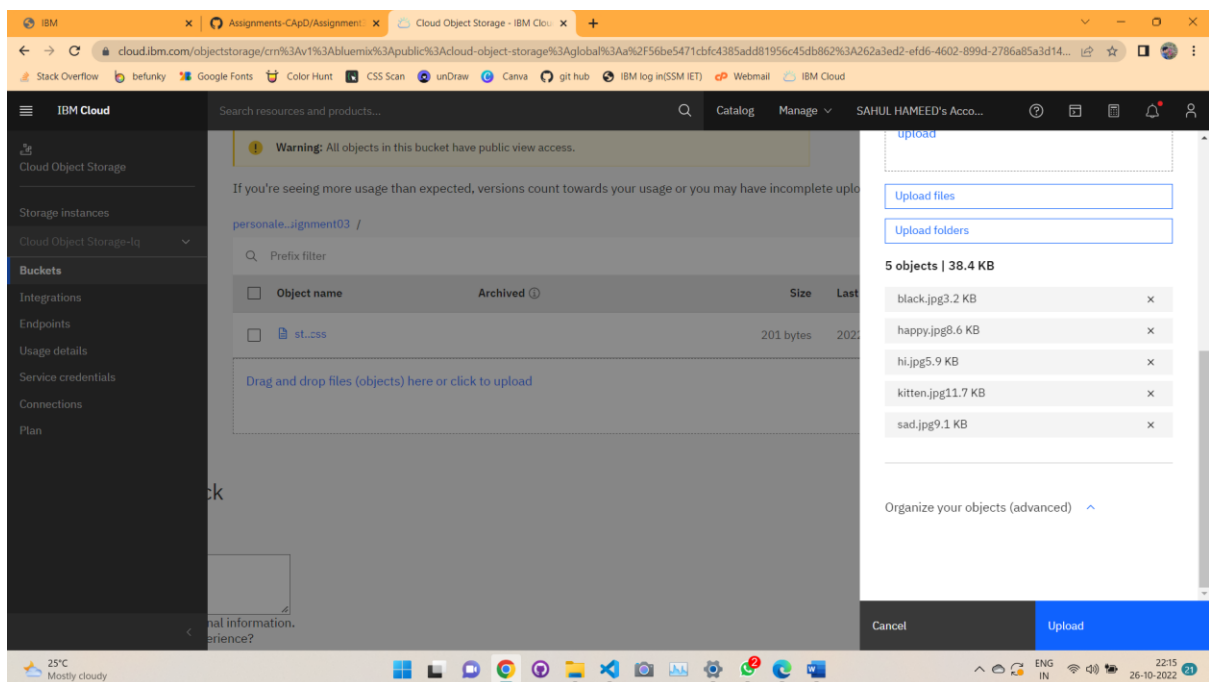


ASSIGNMENT-03

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.

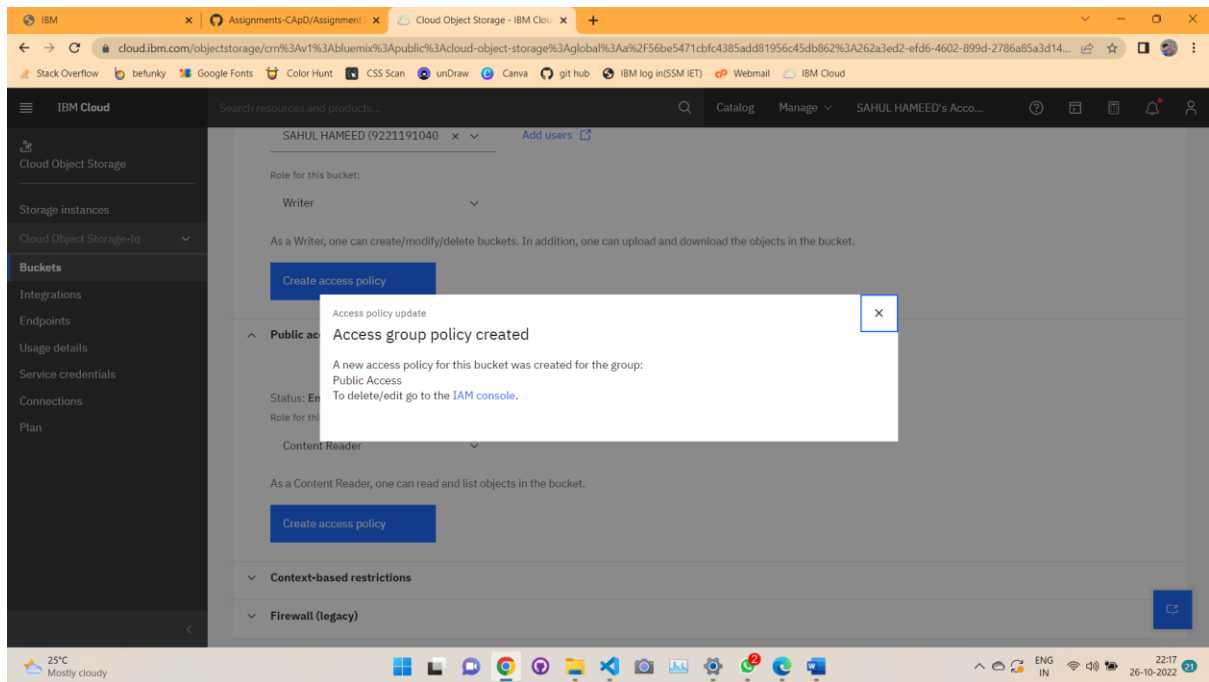


IBM Cloud Object Storage interface showing a bucket named 'personale_ignment03'. A warning message states: "Warning: All objects in this bucket have public view access." A table lists objects in the bucket:

Object name	Archived	Size	Last modified
bl.jpg		3.2 KB	2022-10-26 10:16 PM
ha.jpg		8.6 KB	2022-10-26 10:16 PM
hpg		5.9 KB	2022-10-26 10:16 PM
kl.jpg		11.7 KB	2022-10-26 10:16 PM
s.jpg		9.1 KB	2022-10-26 10:16 PM
st.css		201 bytes	2022-10-26 10:16 PM

A 'Transfers' panel on the right shows an 'Upload success' message for 'black.jpg, happy.jpg, hi.jpg, and 2 others.' The system tray at the bottom shows the date as 26-10-2022 and time as 22:16.

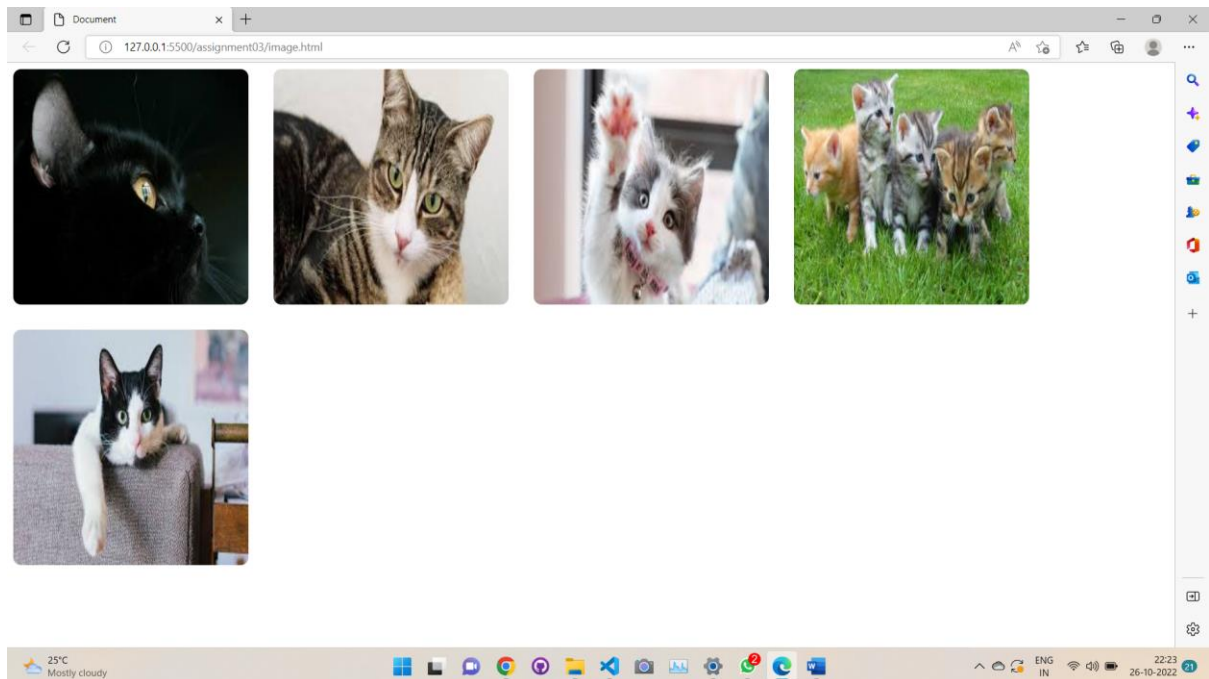
IBM Cloud Object Storage interface showing the 'Public access' settings for the bucket. A dialog box titled 'Enable public access' is displayed, asking: "Granting public access will allow anyone to access this bucket and read all the objects in it. Are you sure you want to enable this functionality?" The dialog has 'Cancel' and 'Enable' buttons. The background shows the 'Public access' section with a 'Create access policy' button and a 'Status: Enabled' indicator. The system tray at the bottom shows the date as 26-10-2022 and time as 22:17.



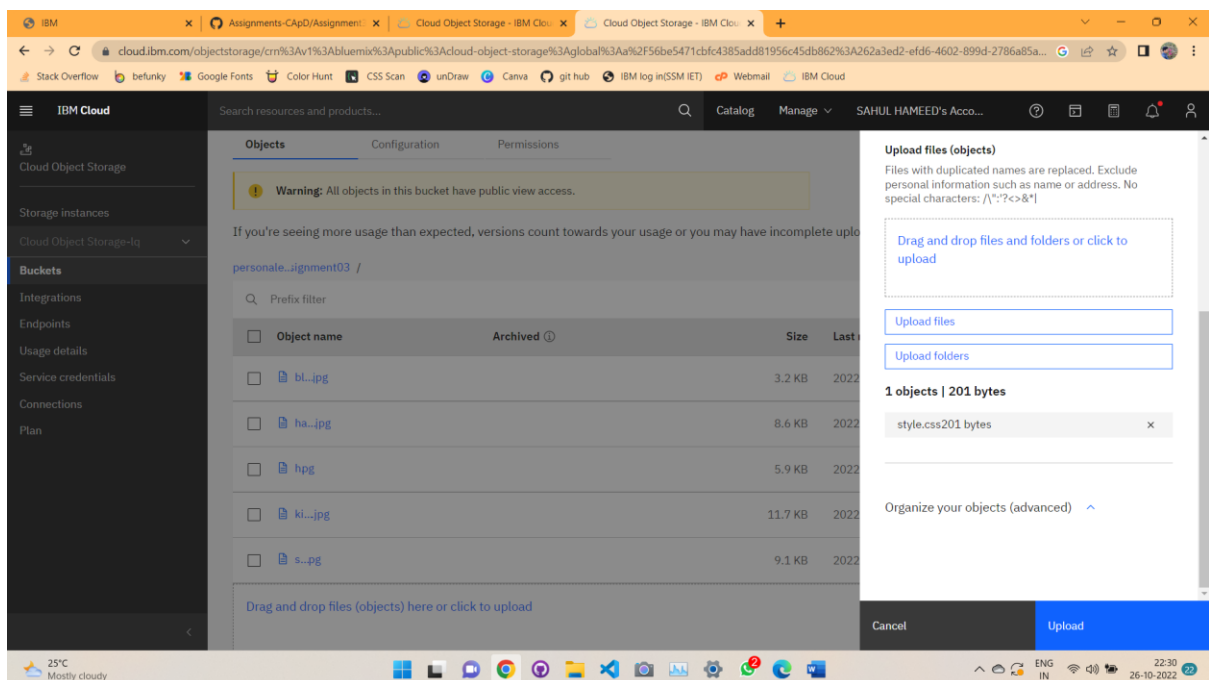
Html code:

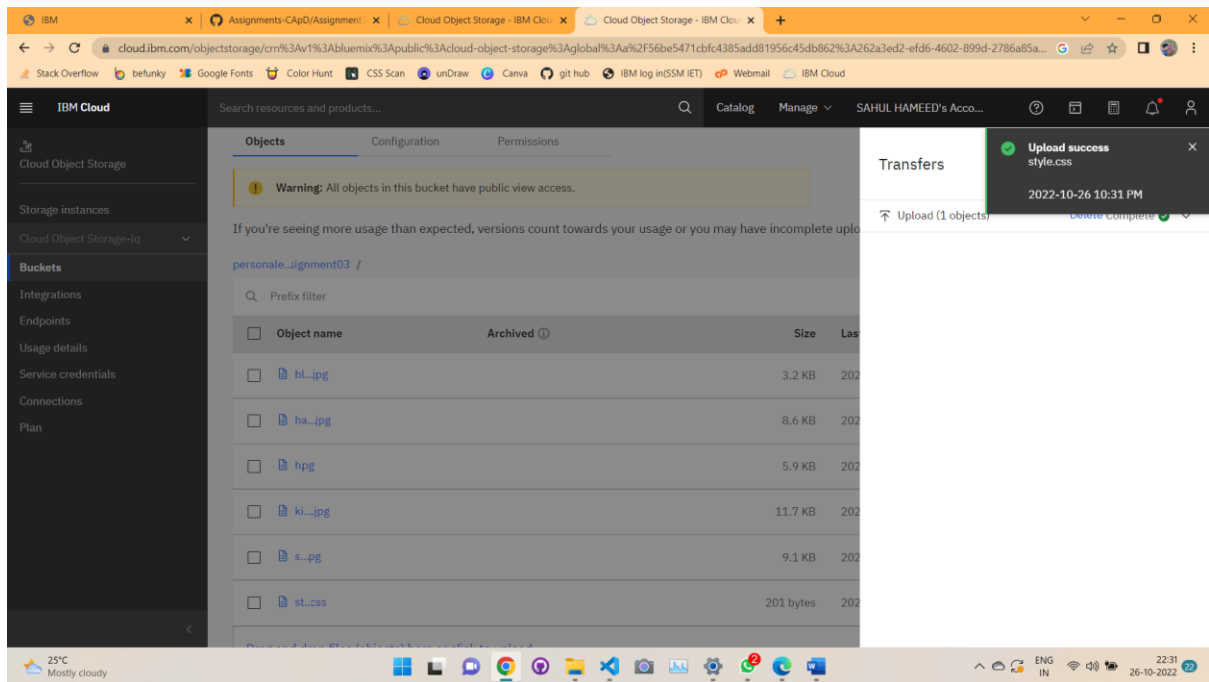
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="https://personalexpensetrackerapplicationassignment03.s3.jp-tok.cloud-object-
storage.appdomain.cloud/style.css" />
    <title>Document</title>
  </head>
  <body>
    <div class="img-container">
      
      
      
      
      
    </div>
  </body> </html>
```

Output:



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





Css:

```
.img-container {
  display: flex;
  flex-direction: row;
  gap: 2rem;
  flex-wrap: wrap;
}

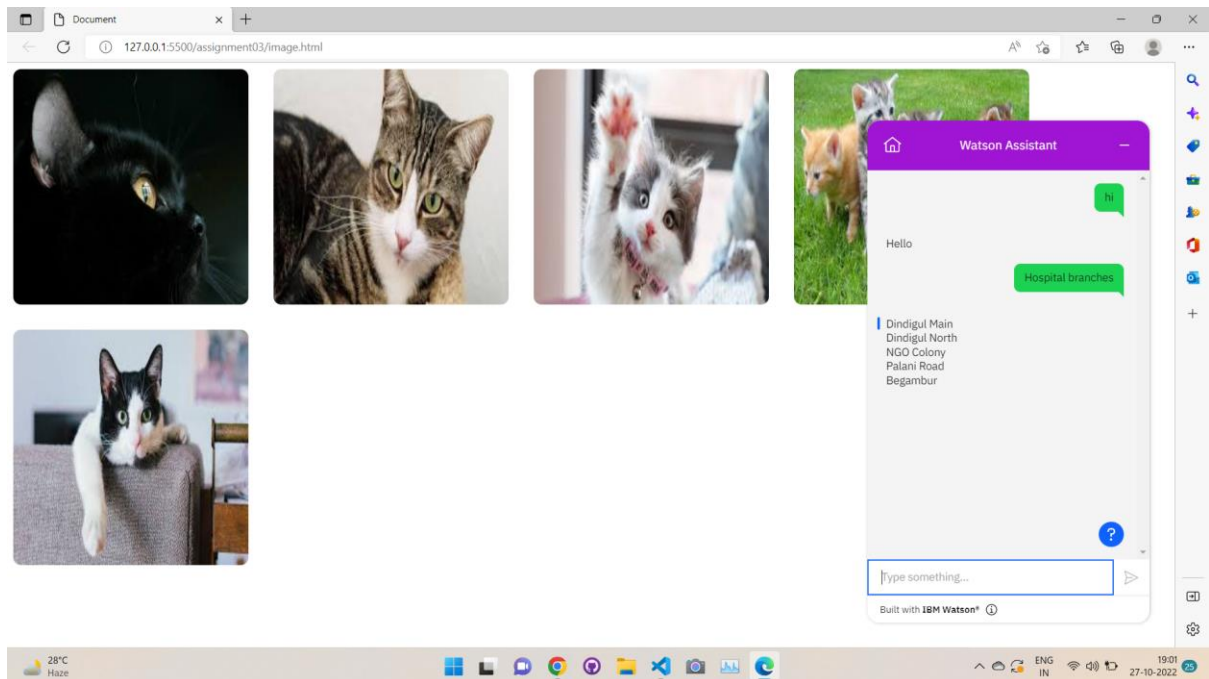
.img-container img {
  width: 300px;
  height: 300px;
  border-radius: 10px;
}
```

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

<https://web->

chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fau-syd.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-27413bf2-e2c4-4422-bb6d-

30a6b2f75a3c%3A%3Acb343695-711a-41aa-9ef0-d01b567b4f36&integrationID=bc3bbd30-2788-42fc-810f-6ee967ad6490®ion=au-syd&serviceInstanceID=27413bf2-e2c4-4422-bb6d-30a6b2f75a3c



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 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" />

    <link rel="stylesheet" href="https://skilandjobassignment3.s3.jp-tok.cloud-object-
storage.appdomain.cloud/style.css" />

    <title>Document</title>

  </head>

  <body>

    <div class="img-container">
```

```

</div>

<script>

window.watsonAssistantChatOptions = {

    integrationID: "bc3bbd30-2788-42fc-810f-6ee967ad6490", // The ID of this integration.

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

    serviceInstanceID: "27413bf2-e2c4-4422-bb6d-30a6b2f75a3c", // 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>

```

Output:

