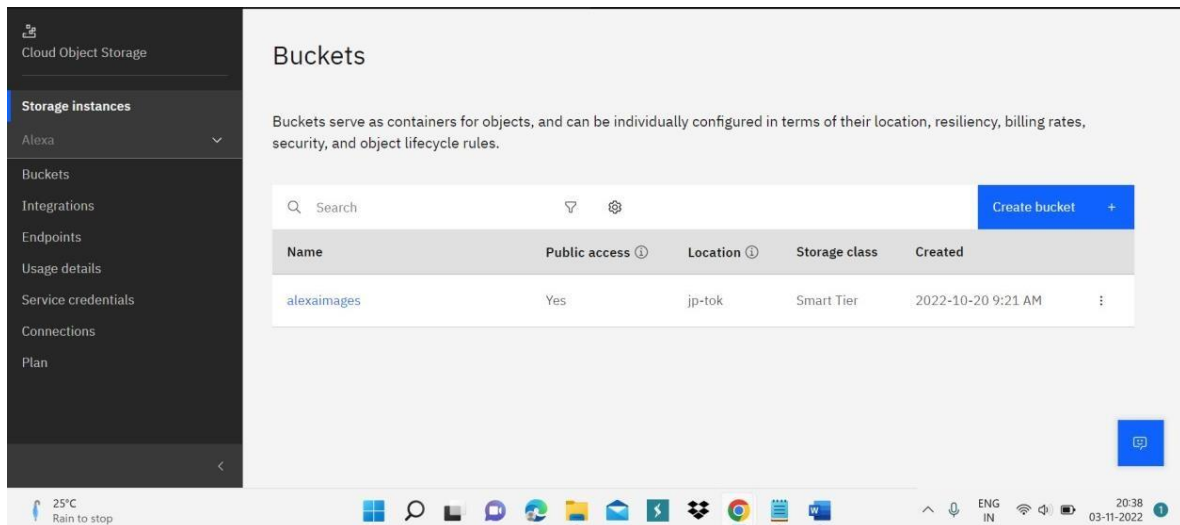


### Assignment -3

|                     |                 |
|---------------------|-----------------|
| Assignment Date     | 06 OCTOBER 2022 |
| Student Name        | DIVYA S         |
| Student Roll Number | 4211191021024   |
| Maximum Marks       | 2 Marks         |

#### 1. Create a Bucket in IBM object storage.



**Buckets**

Buckets serve as containers for objects, and can be individually configured in terms of their location, resiliency, billing rates, security, and object lifecycle rules.

| Name        | Public access | Location | Storage class | Created            |
|-------------|---------------|----------|---------------|--------------------|
| alexaimages | Yes           | jp-tok   | Smart Tier    | 2022-10-20 9:21 AM |

20:38 03-11-2022

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 for a bucket named 'alexaimages'. The left sidebar contains navigation options: Cloud Object Storage, Storage instances (selected), Buckets, Integrations, Endpoints, Usage details, Service credentials, Connections, and Plan. The main area has tabs for Objects, Configuration, and Permissions. A warning message 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 table lists the objects in the bucket:

| Object name | Archived | Size   | Last modified      |
|-------------|----------|--------|--------------------|
| 1..ng       |          | 7.1 KB | 2022-11-03 8:43 AM |
| 2.jpg       |          | 9.8 KB | 2022-11-03 8:43 AM |
| 3.jpg       |          | 7.4 KB | 2022-11-03 8:43 AM |

The bottom of the screen shows a Windows taskbar with the date 03-11-2022 and time 20:39.

This screenshot shows the same IBM Cloud Object Storage console, but now with 7 objects in the 'alexaimages' bucket. The table of objects is as follows:

|            |  |             |                    |
|------------|--|-------------|--------------------|
| 2.jpg      |  | 9.8 KB      | 2022-11-03 8:43 AM |
| 3.jpg      |  | 7.4 KB      | 2022-11-03 8:43 AM |
| 4.jpg      |  | 17.7 KB     | 2022-11-03 8:43 AM |
| 5.jpg      |  | 7.4 KB      | 2022-11-03 8:43 AM |
| buc...html |  | 1.1 KB      | 2022-11-03 8:00 PM |
| st...css   |  | 186 bytes   | 2022-11-03 8:18 PM |
| st...html  |  | 1,018 bytes | 2022-11-03 8:20 PM |

At the bottom of the object list, there is a text box with the instruction: 'Drag and drop files (objects) here or click to upload'. The Windows taskbar at the bottom shows the date 03-11-2022 and time 20:40.

Html code:

```
bucket - Notepad
File Edit View

<!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> Inventory Management System</title>
<style>
body {
background-color: MediumSeaGreen;
display: flex;
align-items: center;
justify-content: center;
flex-direction: column;
}
img {
height: 150px;
border-radius: 40px;
margin: 40px;
}
</style>
</head>
<body>
<h1>Welcome Inventory Management System</h1>
<div>











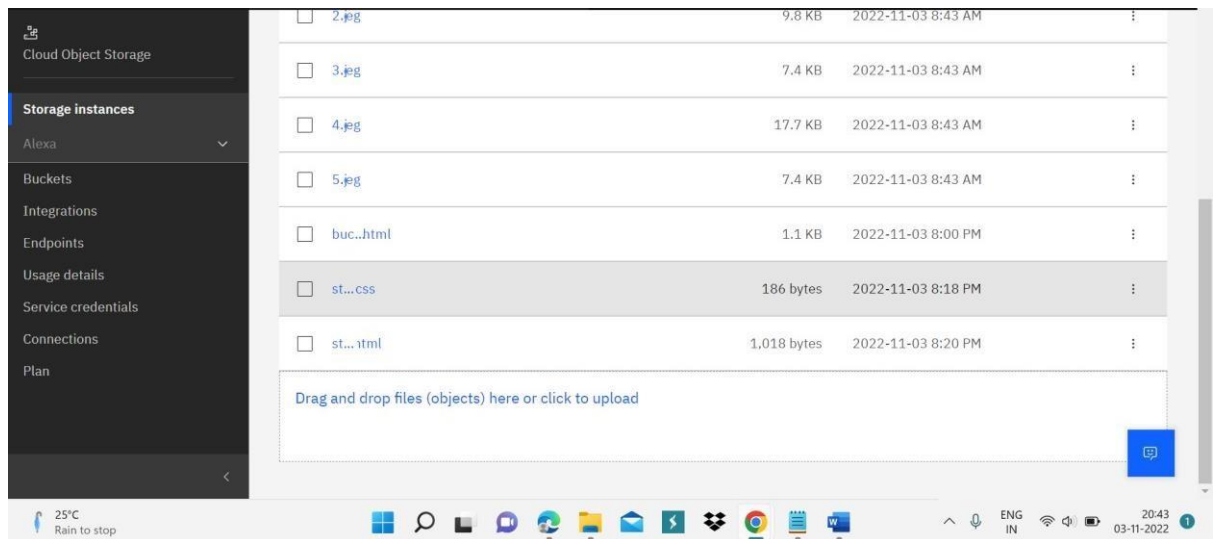
</div>
</body>
</html>

Ln 27, Col 20
60% Windows (CRLF) UTF-8
25°C Rain to stop
```

Output:



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



Html code:

```
style - Notepad
File Edit View

<!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>Inventory Management System </title>
<link rel="stylesheet" href="https://alexaimages.s3.jp-tok.cloud-object-storage.appdomain.cloud/style.css">
</head>
<body>
<h1>Welcome To Inventory Management System</h1>
<div>





</div>
</body>
</html>

Ln 22, Col 1
110% Windows (CRLF) UTF-8
20:56 03-11-2022
```

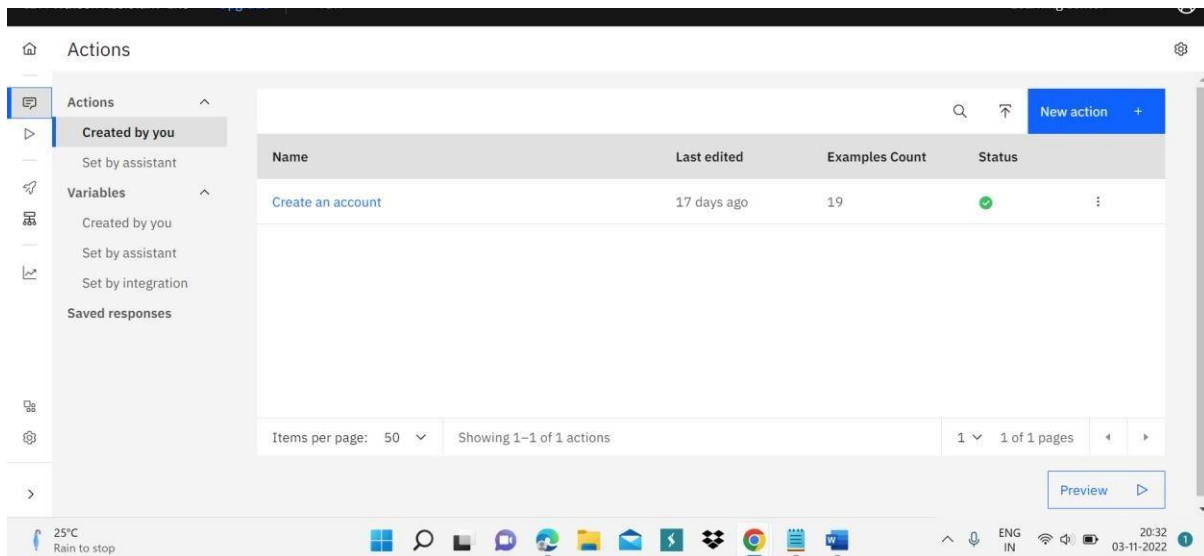
Output:



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 an assignment.

<https://au-syd.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aau-syd%3Aa%2Fc0730717ec23417f814b6cc45740b1bd%3Ac8f677f1-8199-4002-8f72-d193ebbbf75b%3A%3A/assistants/cd533d42-550d-43c2-9152-0f703135f260/actions/actions/custom>

[https://au-syd.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aau-syd%3Aa%2Fc0730717ec23417f814b6cc45740b1bd%3Ac8f677f1-8199-4002-8f72-d193ebbbf75b%3A%3A/assistants/cd533d42-550d-43c2-9152-0f703135f260/actions/actions/custom/edit/action\\_3978](https://au-syd.assistant.watson.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Aconversation%3Aau-syd%3Aa%2Fc0730717ec23417f814b6cc45740b1bd%3Ac8f677f1-8199-4002-8f72-d193ebbbf75b%3A%3A/assistants/cd533d42-550d-43c2-9152-0f703135f260/actions/actions/custom/edit/action_3978)



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

Html page:

```

chat - Notepad
File Edit View

<html>
<head>
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "e891df4f-e963-451d-89ee-c9b5d2458c10", // The ID of this integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "c8f677f1-8199-4002-8f72-d193ebbbf75b", // 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>
</html>

Ln 18, Col 8
100% Windows (CRLF) UTF-8
25°C Rain off and on 21:06 03-11-2022

```

Output:

