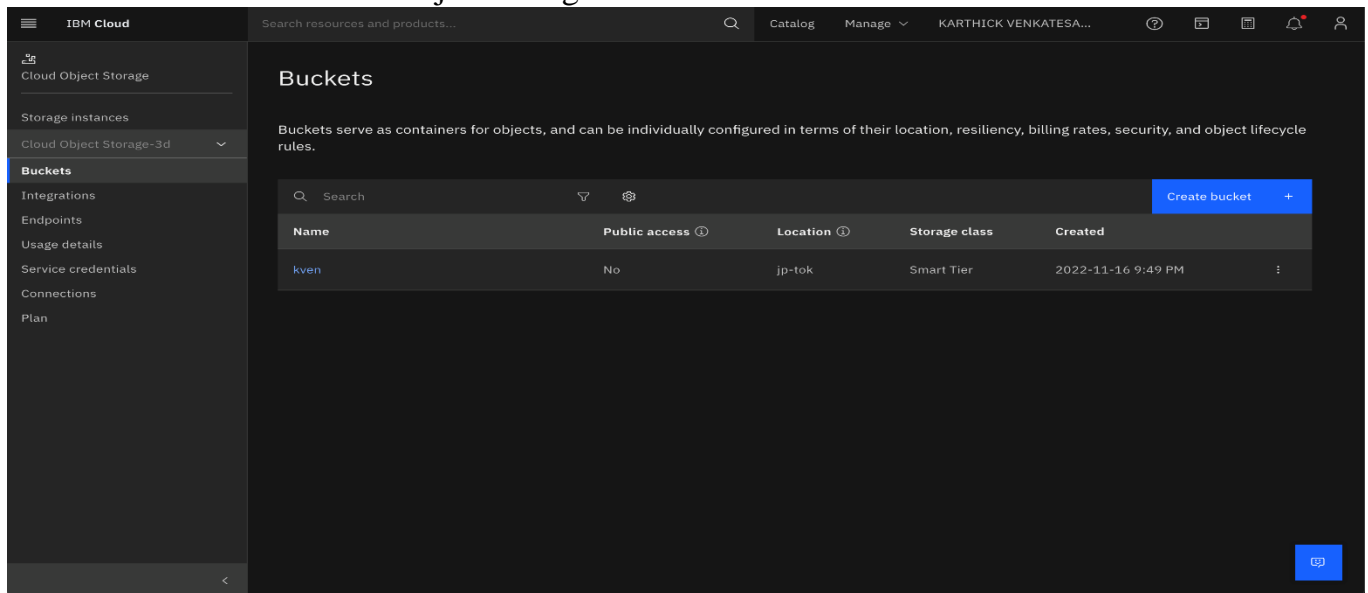


### Assignment-3

Date	10 October 2022
Student Name	Karthick V
Student Roll No	737819ECR073
Max Marks	2

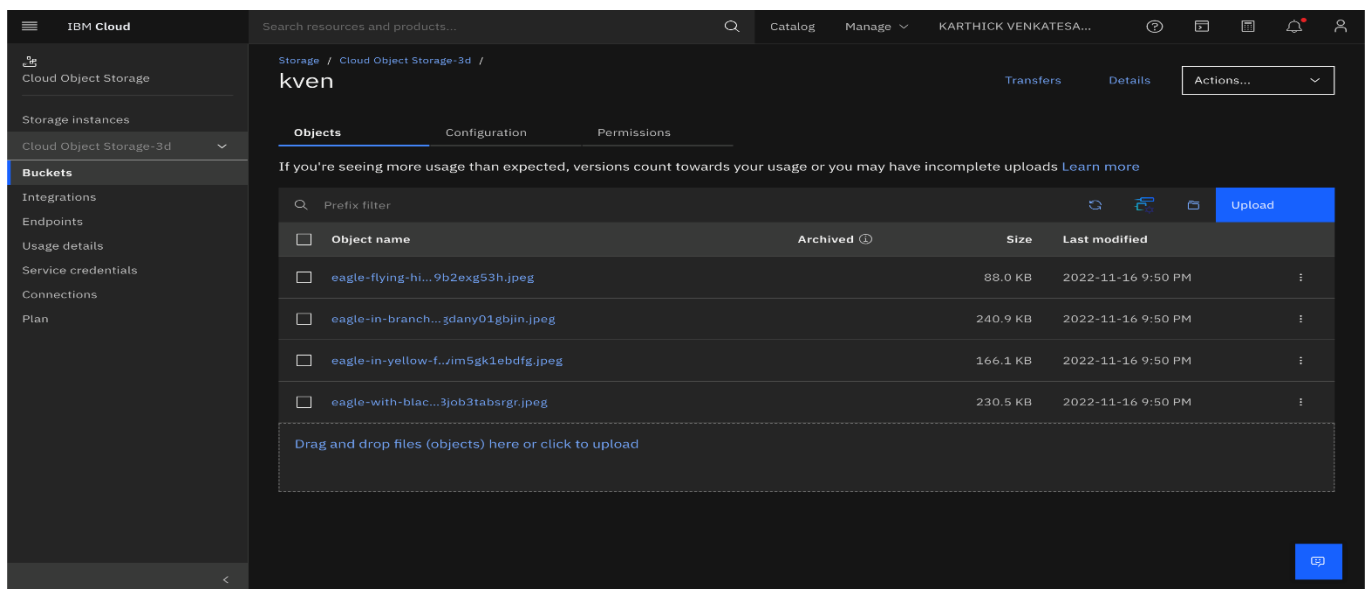
#### 1. Create a Bucket in IBM object storage.



The screenshot shows the IBM Cloud console interface. On the left, the 'Cloud Object Storage' section is expanded, and 'Buckets' is selected. The main content area displays the 'Buckets' page. At the top, there's a search bar and a 'Create bucket' button. Below, a table lists the existing buckets:

Name	Public access ⓘ	Location ⓘ	Storage class	Created
kven	No	jp-tok	Smart Tier	2022-11-16 9:49 PM

#### 2. Upload an 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 console interface for the 'kven' bucket. The left sidebar shows 'Cloud Object Storage' and 'Buckets' selected. The main content area displays the 'Objects' page. At the top, there's a search bar and an 'Upload' button. Below, a table lists the uploaded objects:

Object name	Archived ⓘ	Size	Last modified
eagle-flying-hi...9b2exg53h.jpeg		88.0 KB	2022-11-16 9:50 PM
eagle-in-branch...3dany01gbjin.jpeg		240.9 KB	2022-11-16 9:50 PM
eagle-in-yellow-f...jim5gk1ebdfg.jpeg		166.1 KB	2022-11-16 9:50 PM
eagle-with-blac...3job3tabsrgr.jpeg		230.5 KB	2022-11-16 9:50 PM

## **Obistorage.html:**

```
<!DOCTYPE html>
<html>
<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>
Bucket
</title>
<link rel="stylesheet" href="./style.css" />

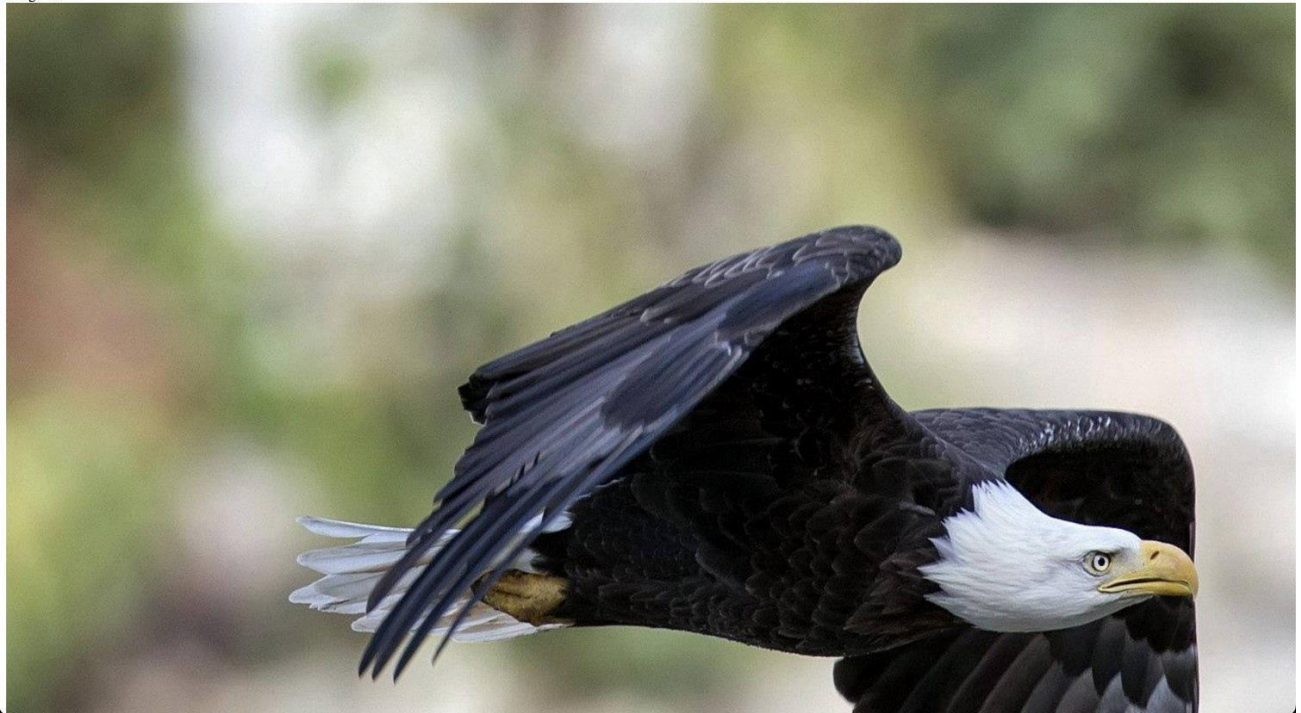
</head>
<body>
<h1>ASSIGNMENT-3</h1>
<table>Image 1:</table>
 <br>
<table>Image 2:</table>
 <br>
<table>Image 3:</table>
 </br>
<table>Image 4:</table>
 <br>
<table>Image 5:</table>
 <br>
</body> <br> </html>
```

## **Output:**

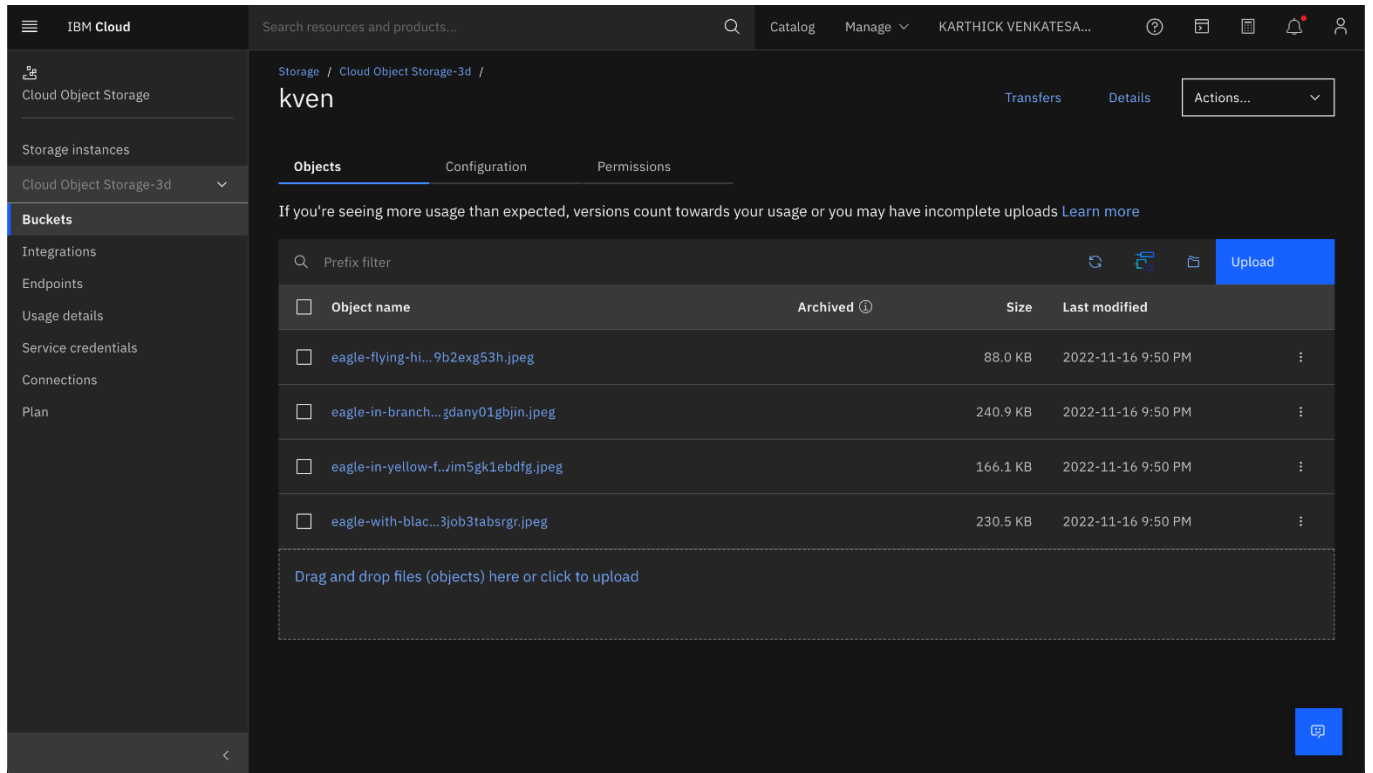
Image 2:



Image 4:



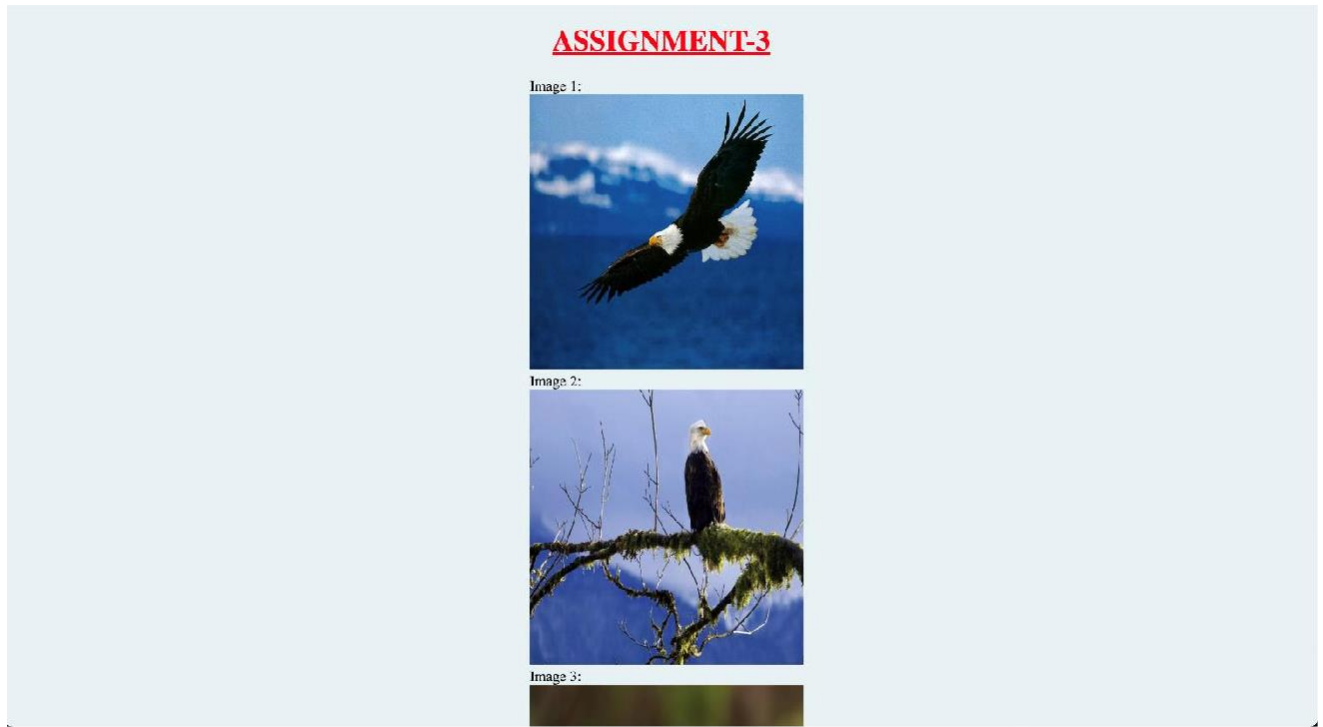
3. Upload a css page to the object storage and use the same page in your HTML code.



### Style.css:

```
h1{
  text-align: center;
  text-decoration: underline;
  color: rgb(255, 0, 0);
}
body{
  background-color: rgb(231, 240, 243);
  display: block;
  margin-left: auto;
  margin-right: auto;
  width: 20%;
}
```

## **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 a assignment.

WebURL: <https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Fau-syd.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-e8f7a1f4-2ab8-4dcb-a24b-1ee1b218d5be%3A%3Ab2672b65-99ae-4dcd-83fe-c9c46ff1a56a&integrationID=95cb7cba-d12c-4426-bc57-1c8a8fd4df2d&region=au-syd&serviceInstanceID=e8f7a1f4-2ab8-4dcb-a24b-1ee1b218d5be>

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

The screenshot shows the IBM Watson Assistant configuration interface. The top bar includes 'IBM Watson Assistant Lite', 'Upgrade', 'bot', and 'Learning center'. The main title is 'Book a meeting'. The interface is divided into two main sections: a list of steps on the left and a detailed view of the selected step on the right.

**Steps:**

- Step 1: 'To set up a meeting, I'll need a few details. First, what's your first name?' (Free text input)
- Step 2: 'Thanks! And what's your last name?' (Free text input)
- Step 3: 'Where would you like me to email the calendar invite?' (Regex input, highlighted)
- Step 4: 'How long would you like to meet for?' (15 minutes, 1 hour, +1 buttons)
- Step 5: 'What day works best?' (Date input)

**Step 3 details:**

- Step 3 is taken without conditions
- Assistant says: 'Where would you like me to email the calendar invite?'
- Assistant recognizes pattern in user's text
- And then: Continue to next step

Buttons at the bottom include 'New step +', 'Preview', and 'Edit response'.

## Sample website

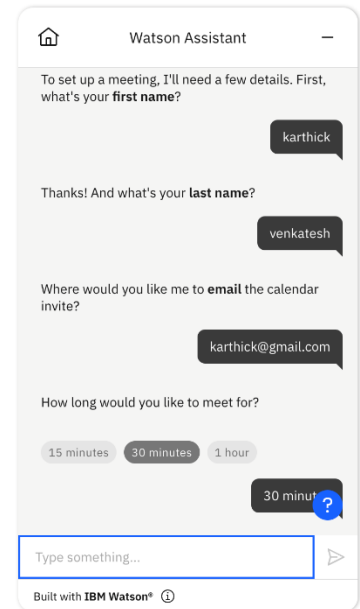
The screenshot shows a sample website layout with a chatbot overlay. The website has a header 'Sample website' and a main content area with several placeholder boxes. The chatbot overlay is titled 'Watson Assistant' and displays a conversation flow:

- Assistant: 'To set up a meeting, I'll need a few details. First, what's your first name?'
- User: 'karthick'
- Assistant: 'Thanks! And what's your last name?'
- User: 'venkatesh'
- Assistant: 'Where would you like me to email the calendar invite?'
- User: 'karthick@gmail.com'
- Assistant: 'How long would you like to meet for?'
- User: '15 minutes'

The chatbot interface includes a 'Type something...' input field and a 'Built with IBM Watson' footer.

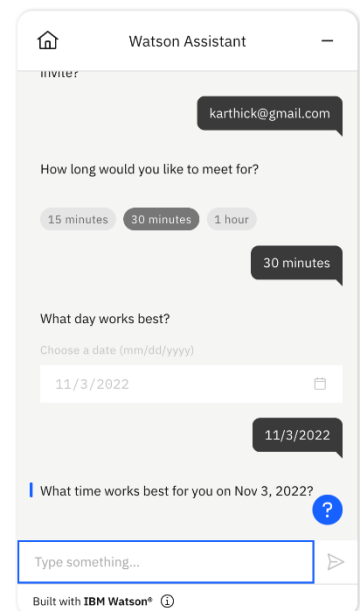
### ASSIGNMENT-3

#### Watson assistant service



### ASSIGNMENT-3

#### Watson assistant service





### Html code:

```
<!DOCTYPE html>
<html>
<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>Bucket</title>
</head>
<body>
<h1
style="
text-align: center;
color: rgb(255, 0, 0);
text-decoration: underline;
"
>
ASSIGNMENT-3
</h1>
<h2 style="text-align: center">Watson assistant service</h2>
<script>
window.watsonAssistantChatOptions = {
integrationID: "5f10cdab-264e-40df-a968-ae17f3c2545c", // The ID of this integration.
region: "au-syd", // The region your integration is hosted in.
serviceInstanceID: "e8f7a1f4-2ab8-4dcb-a24b-1ee1b218d5be", // 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>
<br />
</html>
```