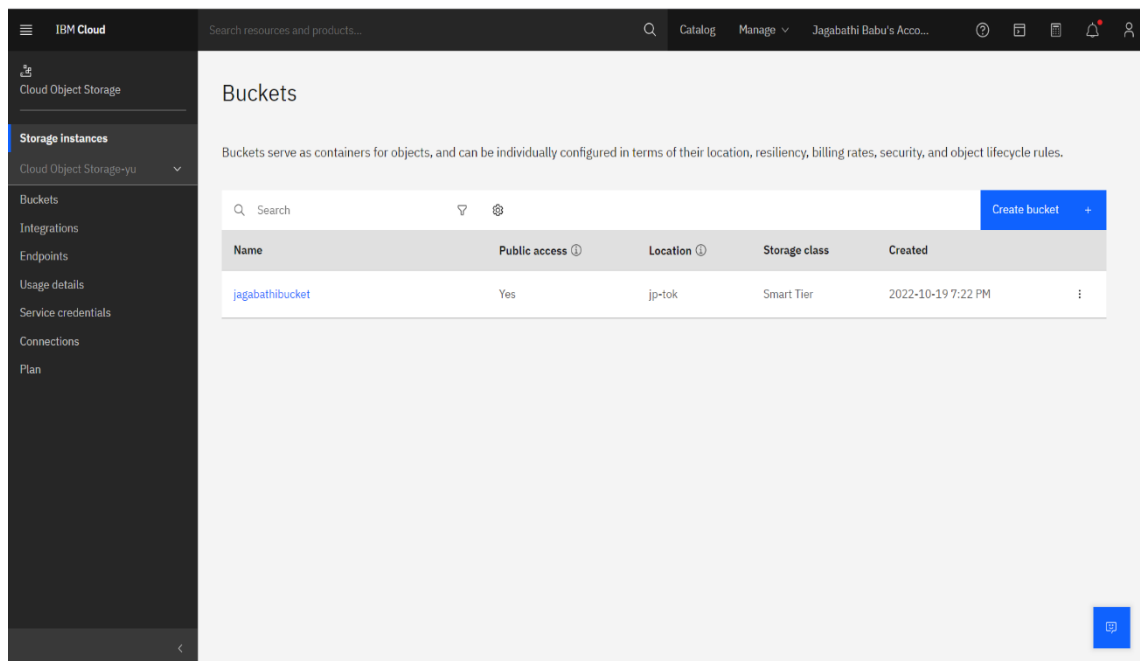


Assignment -3

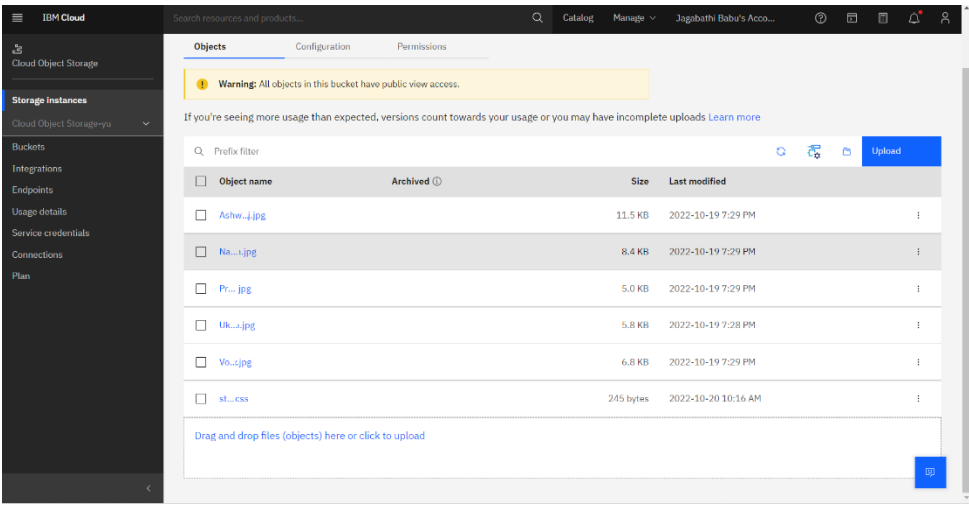
Assignment Date	22 September 2022
Student Name	JAGABATHIBABU V
Student Roll Number	111519205013
Maximum Marks	2 Marks

Question 1:

Create a bucket in IBM Cloud Storage



Question 2:
Upload any 5 images to IBM object storage and make it public. Write HTML code to display all the 5 images.



HTML code for displaying images in cloud:

```
<!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>
</head>
<style>
body{
text-align: center;
background-color: rgb(132, 222, 143);
}
h2{
color: #d831bf;
font-size: 50px;
font-family: cursive;
}
h1{
color: darkblue;
text-align: center;
font-size: 65px;
font-family: Cambria, Cochin, Georgia, Times, 'Times New Roman', serif;
}
img{
height: 230px;
width: 430px;
}
</style>
<body>
<h1>Tamilnadu Volleyball Players</h1>
<h2>Ashwin Raj</h2>

<h2>Naveen</h2>

<h2>Praba</h2>

<h2>Ukkara</h2>

<h2>Volley</h2>

</body>
</html>
```

HTML Output for displaying images from IBM cloud storage:

Tamilnadu Volleyball Players

Ashwin Raj



Naveen



Praba



Ukkara



Volley



Question 3:

Upload a CSS page to the object storage and use the same page in your HTML code. HTML code with CSS file present in IBM cloud storage.

Style.css Code:

```
body{
  background-color: #c66127;
}
h1{
  text-align: center;
  color: #17d324;
  font-family: 'Roboto Slab', serif;
}
p{
  color: #0cd0f3;
  font-family: Georgia, 'Times New Roman', Times, serif;
  font-size:150%;
}
```

The screenshot displays the IBM Cloud console interface. On the left, a sidebar lists navigation options: Cloud Object Storage, Storage instances, Buckets, Integrations, Endpoints, Usage details, Service credentials, Connections, and Plan. The main area shows the 'nutrition-assistant' bucket with a list of objects. The 'style.css' file is selected, and its details are shown in a right-hand pane. The details pane includes tabs for 'Overview' and 'Lifecycle', with 'Overview' currently active. It displays the object's name, last modified date (2022-10-13 12:36 PM), size (245 bytes), and storage class (Smart Tier). Below this, the 'Object Public URL' is provided as `https://nutrition-assistant-application.s3.jp-tok.cloud-object-storage.appdomain.cloud/style.css`. The 'Access with Data Engine' section shows the 'Object Data Engine URL' as `cos://jp-tok/nutrition-assistant-application/style.css`. A message at the bottom states: 'There is no Data Engine instance available. To provision an Data Engine instance, visit Integrations.'

IBM Cloud

Search resources and products...

Cloud Object Storage

Storage instances

Cloud Object Storage-pc

Buckets

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

Storage / Cloud Object Storage-pc

nutrition-assistant

Objects

Warning: All objects in this bucket are not encrypted.

If you're seeing more usage than expected, visit our docs.

Prefix filter

Object name

brooke-lark-jUPOXORNG

joseph-gonzalez-fdlZBV

nadine-primeau-lWVfo

style.css

tangerine-newt-RgT22L

vitalii-pavlyshynets-keR

style.css

Download object

Delete object

Overview

Lifecycle

Object details

Last modified

2022-10-13 12:36 PM

Object size

245 bytes

Storage class

Smart Tier

Tags

0 Tags

Object Public URL

`https://nutrition-assistant-application.s3.jp-tok.cloud-object-storage.appdomain.cloud/style.css`

Access with Data Engine

If this file is of a supported format, you can access the object using an Data Engine instance. To learn more about support file formats, visit our docs.

Object Data Engine URL

`cos://jp-tok/nutrition-assistant-application/style.css`

There is no Data Engine instance available. To provision an Data Engine instance, visit Integrations.

Data Engine instance

No available instance

Open in Data Engine

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 href="https://fonts.googleapis.com/css2?family=Roboto+Slab:wght@200&display=swap"
rel="stylesheet">
<link rel="stylesheet" href="https://jagabathibucket.s3.jp-tok.cloud-object-
storage.appdomain.cloud/style.css">
<title>VOLLEYBALL</title>
</head>
<body>
    <h1>What is Volleyball?</h1>
<p style="text-align: justify;">Volleyball, game played by two teams, usually of six players on a
side, in which the players use their hands to bat a ball back and forth over a high net, trying to
make the ball touch the court within the opponents' playing area before it can be returned. To
prevent this a player on the opposing team bats the ball up and toward a teammate before it touches
the court surface—that teammate may then volley it back across the net or bat it to a third
teammate who volleys it across the net. A team is allowed only three touches of the ball before it
must be returned over the net.</p>
</body>
</html>
```

HTML Code Output:

What is Volleyball?

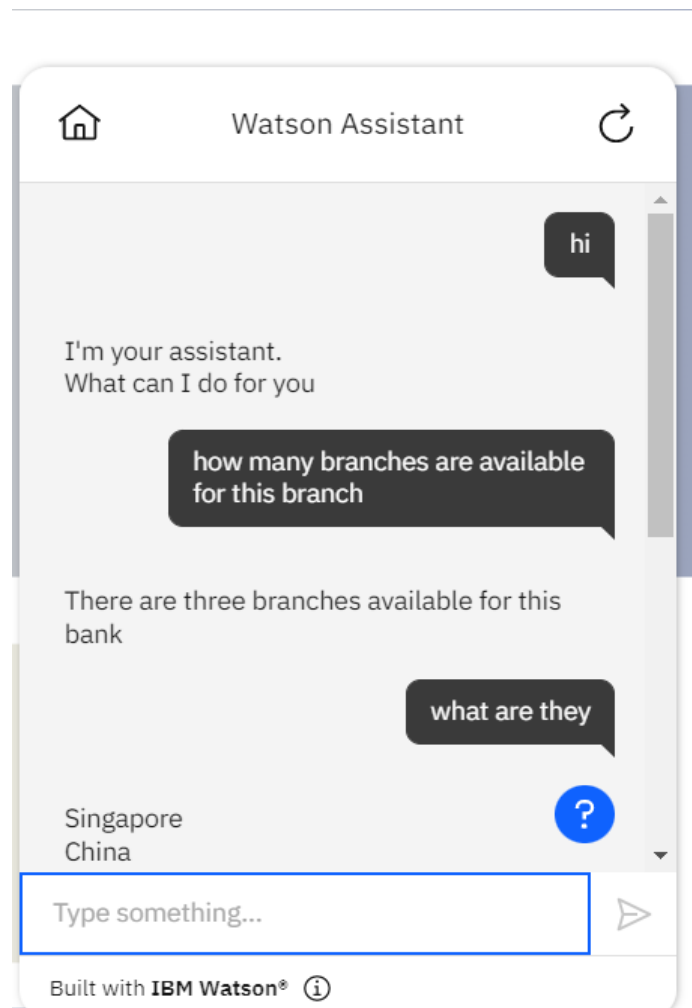
Volleyball, game played by two teams, usually of six players on a side, in which the players use their hands to bat a ball back and forth over a high net, trying to make the ball touch the court within the opponents' playing area before it can be returned. To prevent this a player on the opposing team bats the ball up and toward a teammate before it touches the court surface—that teammate may then volley it back across the net or bat it to a third teammate who volleys it across the net. A team is allowed only three touches of the ball before it must be returned over the net.

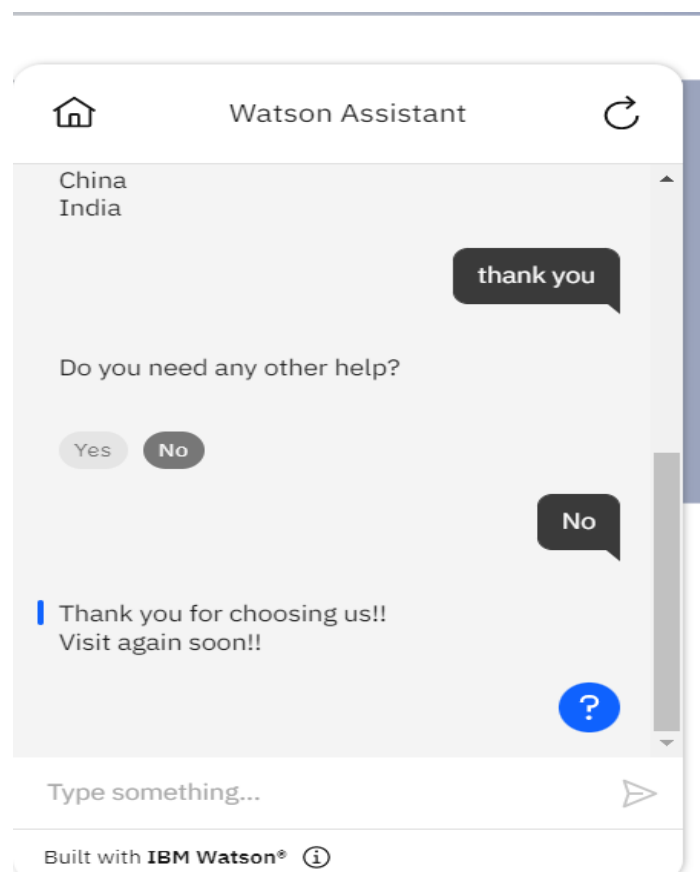
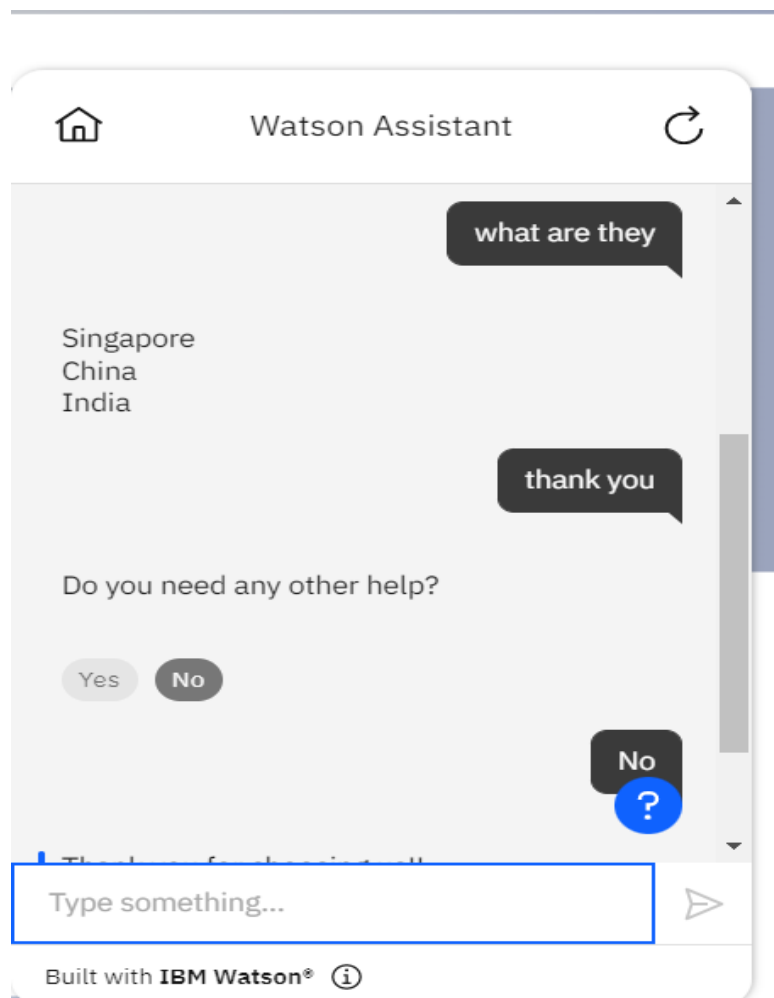
Question 4:

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

URL For Chat Bot:

<https://webchat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageUrl=https%3A%2F%2Fjp-tok.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-0d0c6968-cd6e-4649-a6d9-c4d01f30fb1d%3A%3Ae7376884-9af7-41f8-9f0e-6e931de10b1a&integrationID=ea93da8a-3f06-40e5-b9e3-566adc68e80b®ion=jp-tok&serviceInstanceID=0d0c6968-cd6e-4649-a6d9-c4d01f30fb1d>





Question 5:

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

HTML CODE:

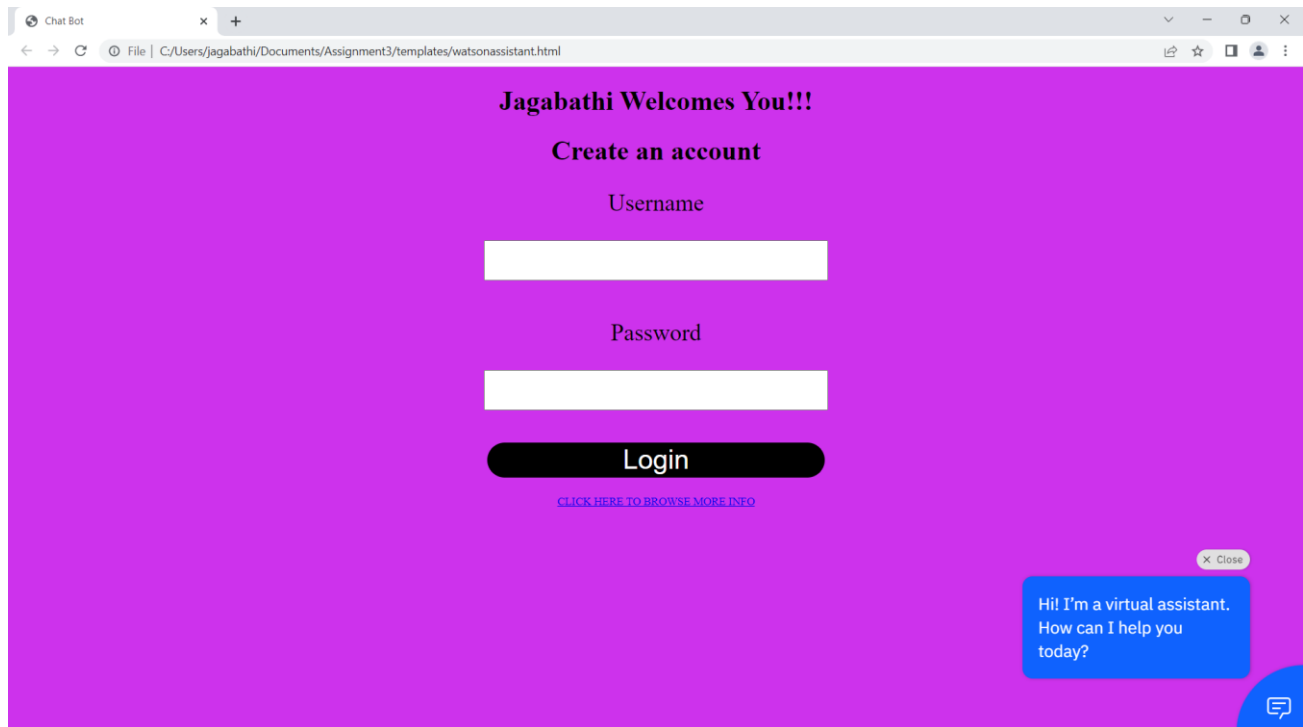
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Chat Bot</title>
<style>
h1{
text-align: center;
color: black;
}
body{
background-color: rgb(205, 50, 236);
}
input[type="text"],input[type="password"]{
height: 42px;
width: 400px;
font-family: cursive;
}
input[type="submit"]{
height: 42px;
width: 400px;
color:white;
background-color: black;
font-size: xx-large ;
border-radius: 25px;
margin-top: 20px;
}
p{
color: black;
font-size: 28px;
}
form{
text-align: center;
}
small{
color: rgb(255, 238, 0);
}
img{
fill: auto;
}
</style>
</head>
<body>
<h1>Jagabathi Welcomes You!!!</h1>
<h1>Create an account</h1>
```

```
<form>
<p>Username </p>
<input type="text" >
<br><br>
<p>Password </p>
<input type="password">
<br><br>
<input type="submit" value="Login" >
<br><br>
<small><a href="https://www.youtube.com/watch?v=E40UEoa19As">CLICK HERE TO BROWSE MORE
INFO</a></small>
</form>
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "9c7de7a4-e052-4756-9172-c985b650bd75", // The ID of this integration.
    region: "jp-tok", // The region your integration is hosted in.
    serviceInstanceID: "0d0c6968-cd6e-4649-a6d9-c4d01f30fb1d", // 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>
```

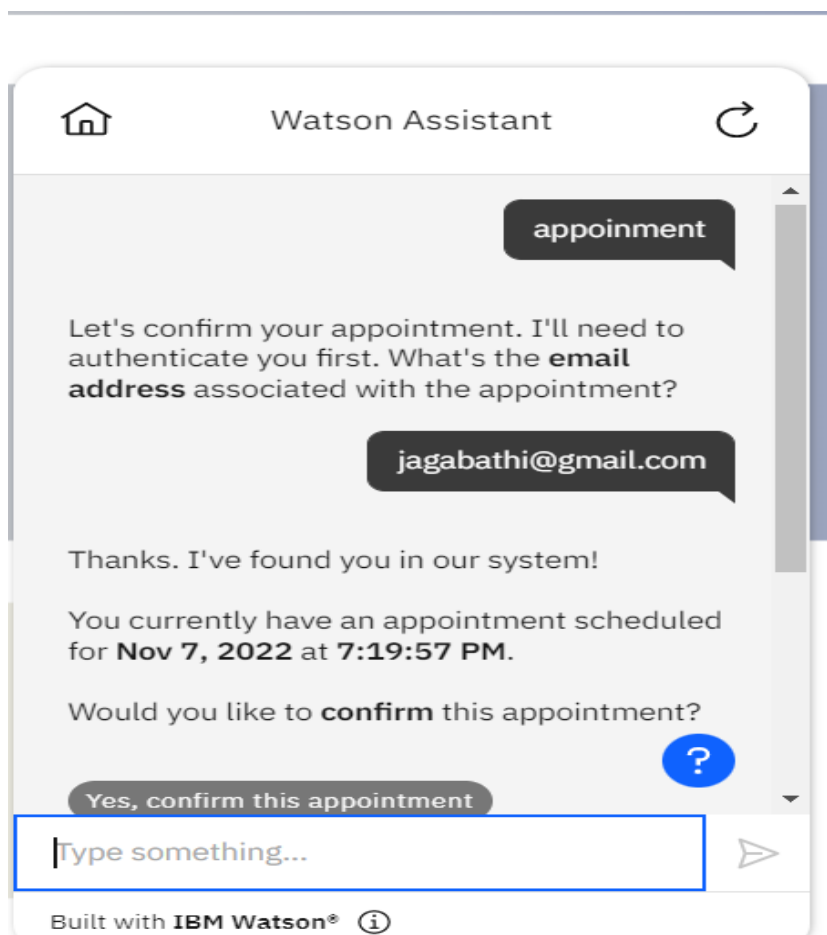
App.py Code:

```
from flask import Flask, render_template
app = Flask(__name__)
@app.route("/")
def home():
    return render_template("watsonassistant.html")
@app.route("/salvador")
def salvador():
    return "Hello, Salvador"
if __name__ == "__main__":
    app.run(debug=True)
```

HTML CODE OUTPUT:



ASSISTANT OUTPUT:





Watson Assistant



Thanks. I've found you in our system!

You currently have an appointment scheduled for **Nov 7, 2022 at 7:19:57 PM.**

Would you like to **confirm** this appointment?

Yes, confirm this appointment

No, I'd like to make a modification

Yes, confirm this appointment

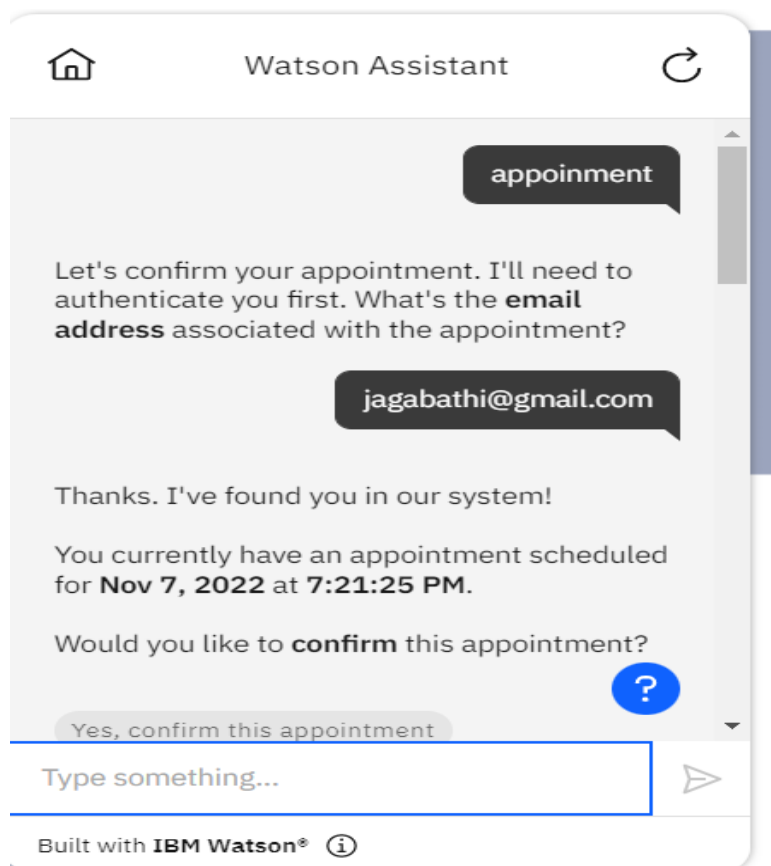
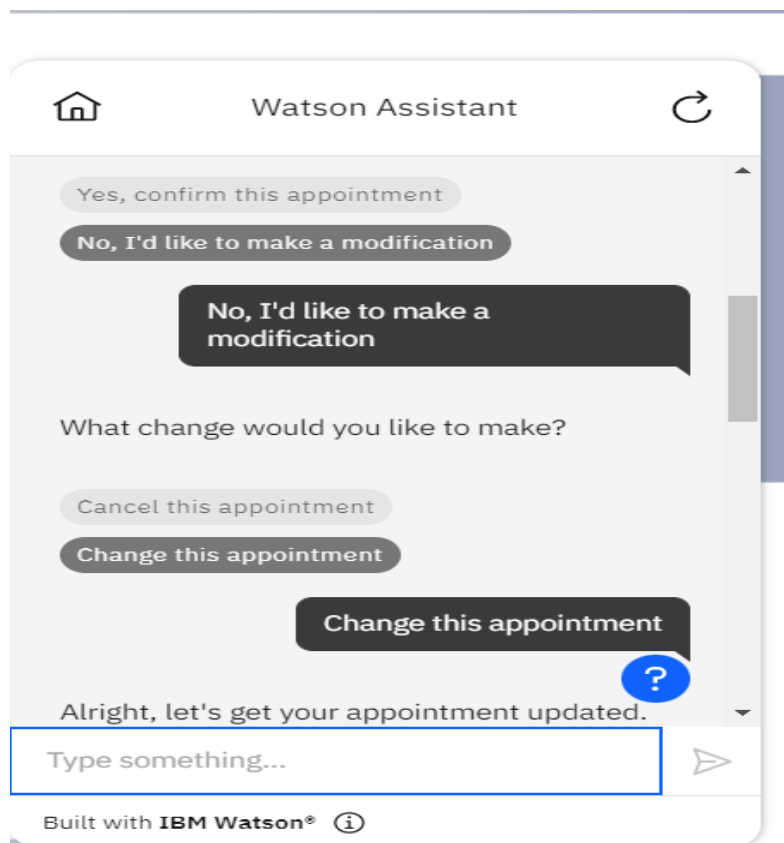
Great! You're all set. We'll see you at your upcoming appointment!



Type something...



Built with **IBM Watson®** ⓘ





Watson Assistant



Alright, let's get your appointment updated.

How long would you like to meet for?

15 minutes

30 minutes

1 hour

30 minutes

What **day** works best for you?

Choose a date (mm/dd/yyyy)

11/12/2022



11/12/20



Type something...



Built with **IBM Watson®** ⓘ



Watson Assistant



What **time** works best for you on Nov 12, 2022?

10 am

Great. Just to confirm, you'd like to book an appointment on Nov 12, 2022 at 10:00:00 AM for 30 minutes.

Is that correct?

Yes that's right

Not quite

Yes that's right



You're all set. I've sent the calendar invite to

Type something...



Built with IBM Watson® ⓘ



Watson Assistant



Great. Just to confirm, you'd like to book an appointment on Nov 12, 2022 at 10:00:00 AM for 30 minutes.

Is that correct?

Yes that's right

Not quite

Yes that's right

You're all set. I've sent the calendar invite to your email at jagabathi@gmail.com

Enjoy the rest of your day!



Type something...



Built with IBM Watson® ⓘ