Assignment 3
Creation of IBM object storage and Watson chat bot

Date	07 October 2022
Student Name	Harsini S.R.
Student Roll Number	111519205011
Maximum Marks	2 Marks

Question:

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

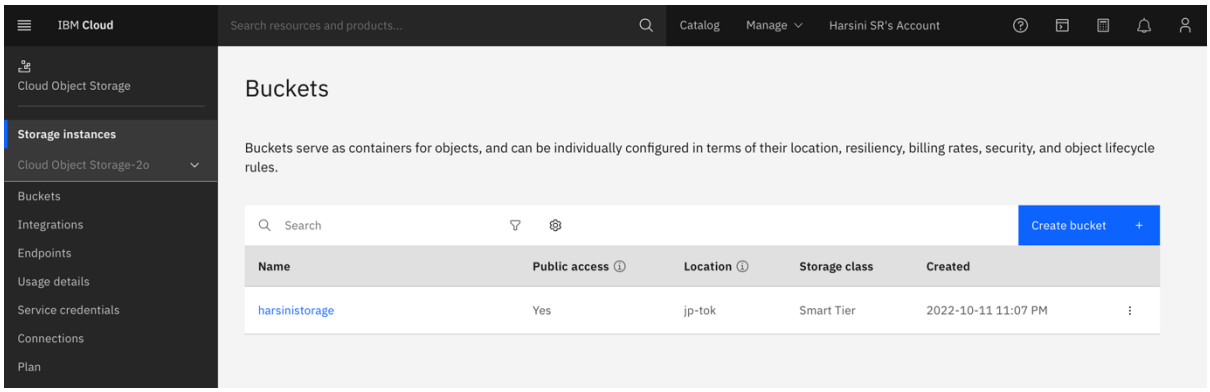
3.Upload a css page to the object storage and use the same page in your HTML code.

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 assignment.

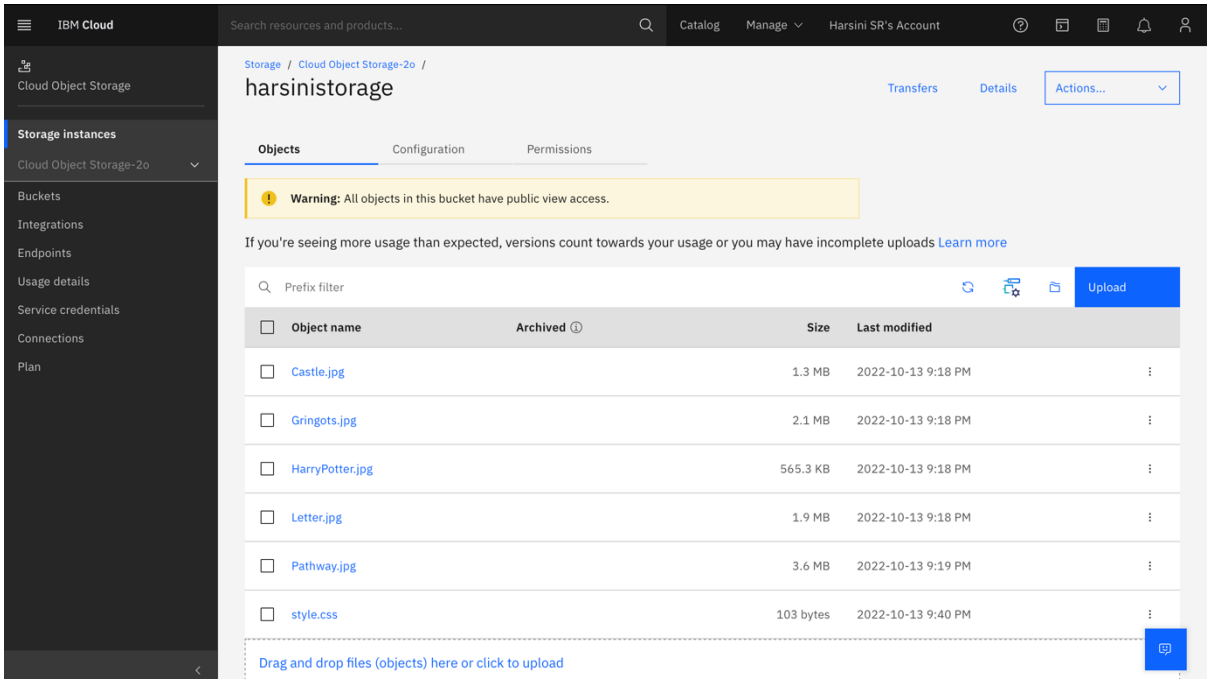
5.Create Watson assistant service with 10 steps and use 3 conditions in it. Load that script in HTML page.

Solution:

Creating bucket in IBM object storage



Upload 5 images and display using html code. The images are made public. Upload a css page to the object storage and use the same page in your HTML code.



Using html displaying:

Index.html

```
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Harry Potter World</title>
  </head>
  <body>
    <h1>Harry Potter World<h1>
    <h2>Castle </h2>
    <img style="width:400px ;height:200px;" src="https://harsinistorage.s3.jp-
tok.cloud-object-storage.appdomain.cloud/Castle.jpg" alt="Castle">
    <h2>Gringots</h2>
    <img style="width:400px ;height:200px;" src="https://harsinistorage.s3.jp-
tok.cloud-object-storage.appdomain.cloud/Gringots.jpg" alt="Gringots">
    <h2>HarryPotter</h2>
    <img style="width:400px ;height:200px;" src="https://harsinistorage.s3.jp-
tok.cloud-object-storage.appdomain.cloud/HarryPotter.jpg" alt="HarryPotter">
    <h2>Letter</h2>
    <img style="width:400px ;height:200px;" src="https://harsinistorage.s3.jp-
tok.cloud-object-storage.appdomain.cloud/Letter.jpg" alt="Letter">
```

```
    <h2>Pathway</h2>
    <img style="width:400px ;height:200px;" src="https://harsinistorage.s3.jp-tok.cloud-object-storage.appdomain.cloud/Pathway.jpg" alt="Pathway">
  </body>
</html>
```

Output:

Harry Potter World

Castle

Gringots

HarryPotter

Letter

Pathway

Using flask

App.py

```python
from flask import Flask,url_for,render_template,request
import ibm_boto3
from ibm_botocore.client import Config, ClientError

COS_ENDPOINT="https://s3.jp-tok.cloud-object-storage.appdomain.cloud"
COS_API_KEY_ID="qC4sWRXAYbEU5nrpzHWvcce5wjlWd0p-s7A5-9RRnQQk"
COS_INSTANCE_CRN="crn:v1:bluemix:public:cloud-object-
storage:global:a/ef4be31ad3b24eb8bed57ab1f9b98873:ea44501d-196a-4fe3-98c6-
0019ae352f16::"
# Create resource https://s3.ap.cloud-object-storage.appdomain.cloud
cos = ibm_boto3.resource("s3",
    ibm_api_key_id=COS_API_KEY_ID,
    ibm_service_instance_id=COS_INSTANCE_CRN,
    config=Config(signature_version="oauth"),
    endpoint_url=COS_ENDPOINT
)

app=Flask(__name__)



def get_item(bucket_name, item_name):
    print("Retrieving item from bucket: {0}, key: {1}".format(bucket_name,
item_name))
    try:
        file = cos.Object(bucket_name, item_name).get()

        print("File Contents: {0}".format(file["Body"].read()))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve file contents: {0}".format(e))


def get_bucket_contents(bucket_name):
    print("Retrieving bucket contents from: {0}".format(bucket_name))
    try:
        files = cos.Bucket(bucket_name).objects.all()
        files_names = []
        for file in files:
            files_names.append(file.key)
            print("Item: {0} ({1} bytes).".format(file.key, file.size))
        return files_names
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to retrieve bucket contents: {0}".format(e))


def delete_item(bucket_name, object_name):
    try:
        cos.delete_object(Bucket=bucket_name, Key=object_name)
```

```
        print("Item: {0} deleted!\n".format(object_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to delete object: {0}".format(e))



def multi_part_upload(bucket_name, item_name, file_path):
    try:
        print("Starting file transfer for {0} to bucket: {1}\n".format(item_name,
bucket_name))
        # set 5 MB chunks
        part_size = 1024 * 1024 * 5

        # set threadhold to 15 MB
        file_threshold = 1024 * 1024 * 15

        # set the transfer threshold and chunk size
        transfer_config = ibm_boto3.s3.transfer.TransferConfig(
            multipart_threshold=file_threshold,
            multipart_chunksize=part_size
        )

        # the upload_fileobj method will automatically execute a multi-part upload
        # in 5 MB chunks for all files over 15 MB
        with open(file_path, "rb") as file_data:
            cos.Object(bucket_name, item_name).upload_fileobj(
                Fileobj=file_data,
                Config=transfer_config
            )

        print("Transfer for {0} Complete!\n".format(item_name))
    except ClientError as be:
        print("CLIENT ERROR: {0}\n".format(be))
    except Exception as e:
        print("Unable to complete multi-part upload: {0}".format(e))


@app.route('/')
def index():
    files = get_bucket_contents('harsinistorage')
    return render_template('index.html', files = files)

@app.route('/deletefile', methods = ["GET", "POST"])  # type: ignore
def deletefile():
   if request.method == 'POST':
       bucket=request.form['bucket']
       name_file=request.form['filename']
       print(bucket,name_file)
       delete_item(bucket,name_file)
```

```
        return 'file deleted successfully'

    if request.method == 'GET':
        return render_template('delete.html')


@app.route('/uploader', methods = ['GET', 'POST'])  # type: ignore
def upload():
    if request.method == 'POST':
        bucket=request.form['bucket']
        name_file=request.form['filename']
        f = request.files['file']
        print(f.filename)
        multi_part_upload(bucket,name_file,f.filename)
        return 'file uploaded successfully <a href="/">GO to Home</a>'


    if request.method == 'GET':
        return render_template('upload.html')

if __name__=='__main__':
     app.run(host='0.0.0.0',port=8080,debug=True)
```

index.html

```
<a href="/">HOME</a>
<a href="/uploader">Upload</a>
<a href="/deletefile">Delete</a>
<br><hr>
<h1>IBM Object Storage</h1>


<!doctype html>
<html>
    <head>
        <link href="https://harsinistorage.s3.jp-tok.cloud-object-
storage.appdomain.cloud/style.css" rel="stylesheet">

    </head>
   <body>
      {% for row in files %}
         <div style="border: 1px solid #EFEFEF;margin:10px;">
            <h3>Filename : {{row}} </h3>
            <img src="https://harsinistorage.s3.jp-tok.cloud-object-
storage.appdomain.cloud/{{row}}" width="150px"></td>
         </div>
      {% endfor %}
   </body>
</html>
```

IBM Object Storage

Filename : Castle.jpg

Filename : Gringots.jpg

Filename : HarryPotter.jpg

Filename : Letter.jpg

Filename : Pathway.jpg

Style.css

style.css

Download object | Delete object

Overview | Lifecycle

Object details

Last modified	Object size	Storage class
2022-10-13 9:40 PM	103 bytes	Smart Tier

Object Public URL

```
https://harsinistorage.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

Object Data Engine URL

```
cos://jp-tok/harsinistorage/style.css
```

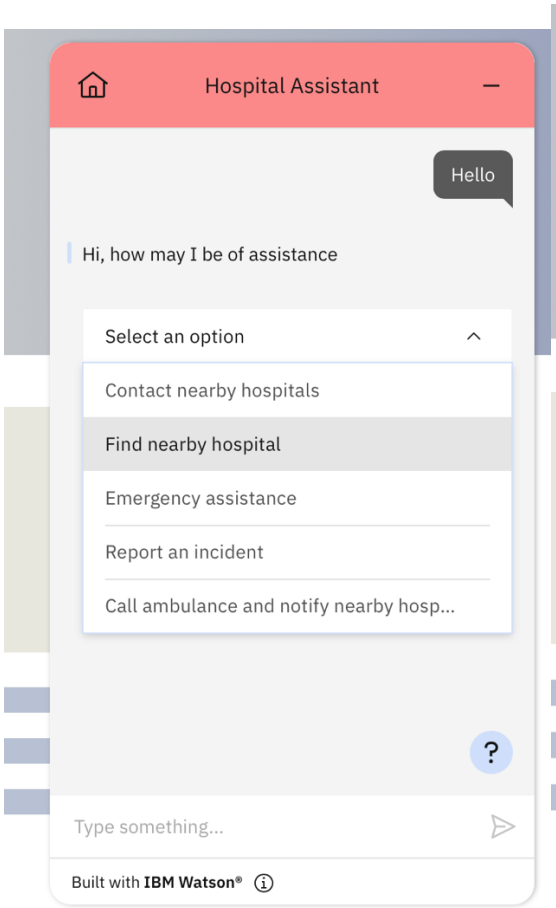
There is no Data Engine instance available. To provision an Data Engine instance, visit Integrations.

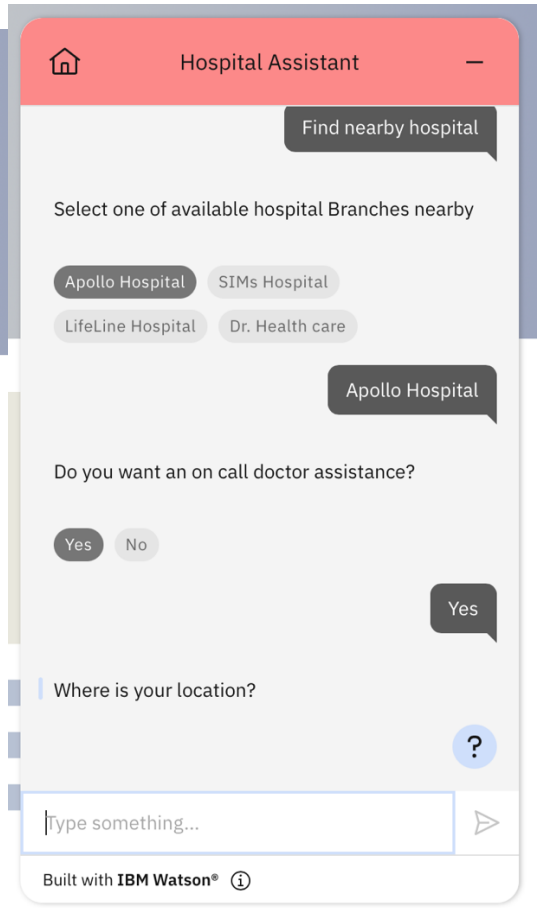
Chat bot using IBM Watson assistant for hospital

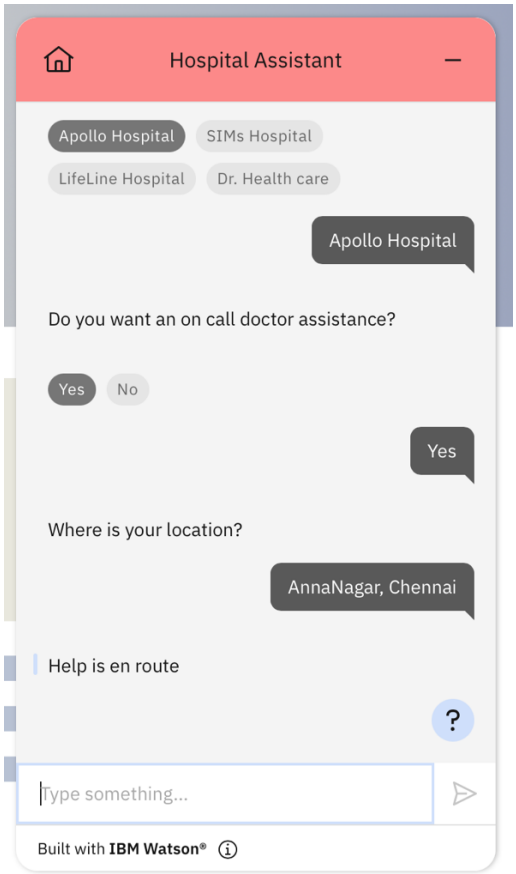
Chat bot web URL: Hospital assistance

https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageURL=https%3A%2F%2Feu-gb.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-eeb0f9b2-ba26-4bce-ad4d-08b36df6bcf6%3A%3A917e7281-cbc0-432b-8f80-db337f562cc3&integrationID=5113f1ba-d90a-46ea-a91a-0f11d3533c5f&region=eu-gb&serviceInstanceID=eeb0f9b2-ba26-4bce-ad4d-08b36df6bcf6

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


Hospital Assistant
Hello
Hi, how may I be of assistance
Select an option
Contact nearby hospitals
Find nearby hospital
Emergency assistance
Report an incident
Call ambulance and notify nearby hosp...
Type something...
Built with IBM Watson®


Hospital Assistant
Find nearby hospital
Select one of available hospital Branches nearby
Apollo Hospital
SIMs Hospital
LifeLine Hospital
Dr. Health care
Apollo Hospital
Do you want an on call doctor assistance?
Yes
No
Yes
Where is your location?
Type something...
Built with IBM Watson®


Hospital Assistant
Apollo Hospital
SIMs Hospital
LifeLine Hospital
Dr. Health care
Apollo Hospital
Do you want an on call doctor assistance?
Yes
No
Yes
Where is your location?
AnnaNagar, Chennai
Help is en route
Type something...
Built with IBM Watson®

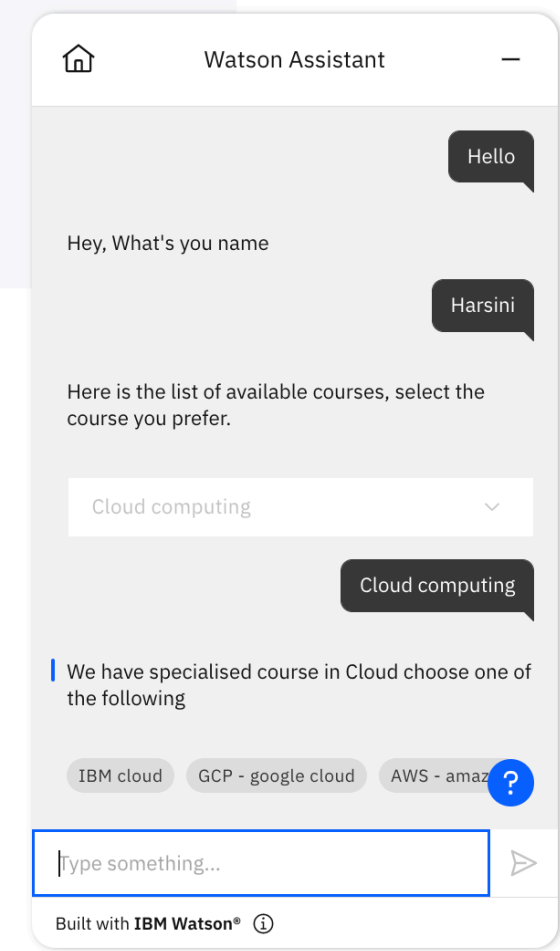
App.py

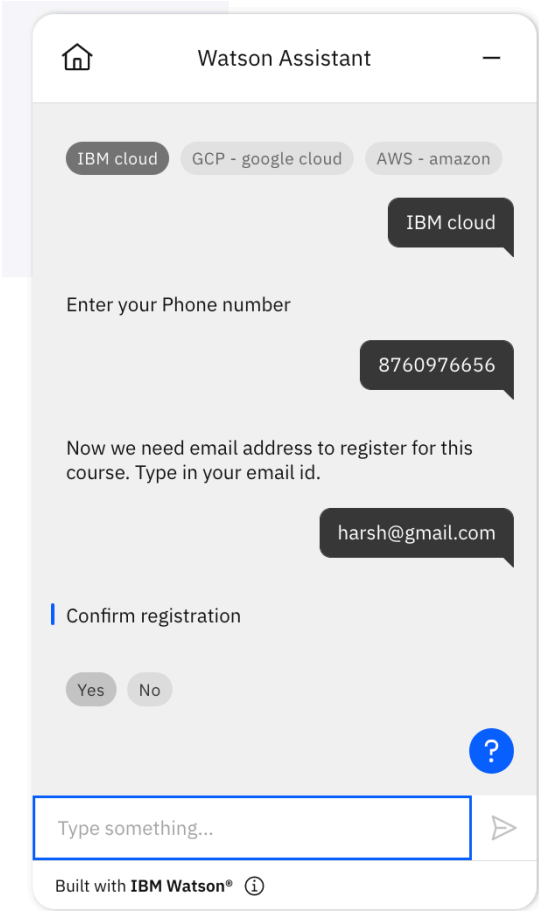
```python
from flask import Flask, render_template, url_for

app = Flask(__name__)

@app.route("/")
def index():
    return render_template('index.html')
```

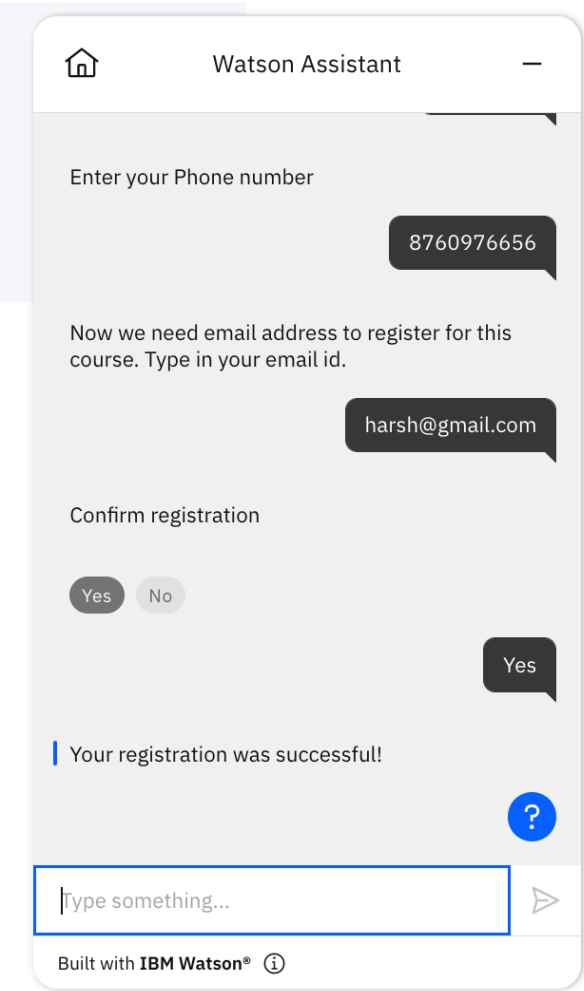
base.html

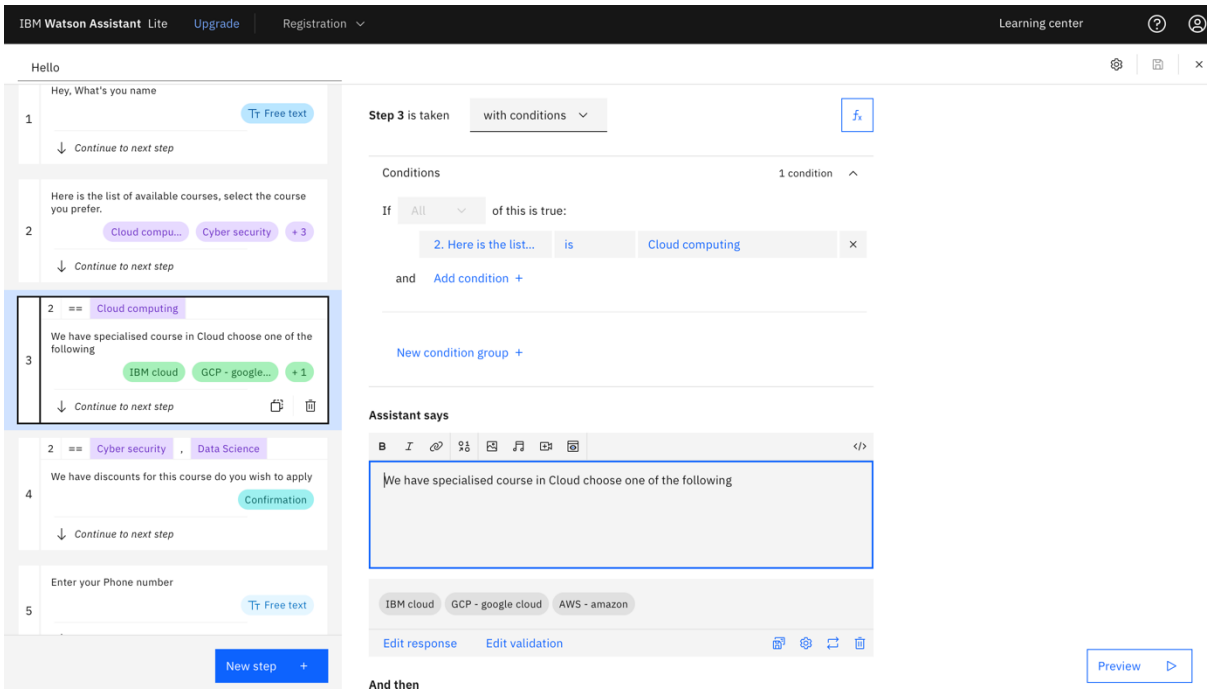
```html
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  </head>
  <body style="background-color: rgb(188, 188, 255)">
    <div>
      <h1>Hi register through IBM watson</h1>
      <p>Contact the chat bot to assist you register for course</p>
      <script>
        window.watsonAssistantChatOptions = {
          integrationID: "ae13e90e-151c-4463-8cfc-9c7b7d885c07", // The ID of this
integration.
          region: "eu-gb", // The region your integration is hosted in.
          serviceInstanceID: "eeb0f9b2-ba26-4bce-ad4d-08b36df6bcf6", // 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>
    </div>
  </body>
</html>
```

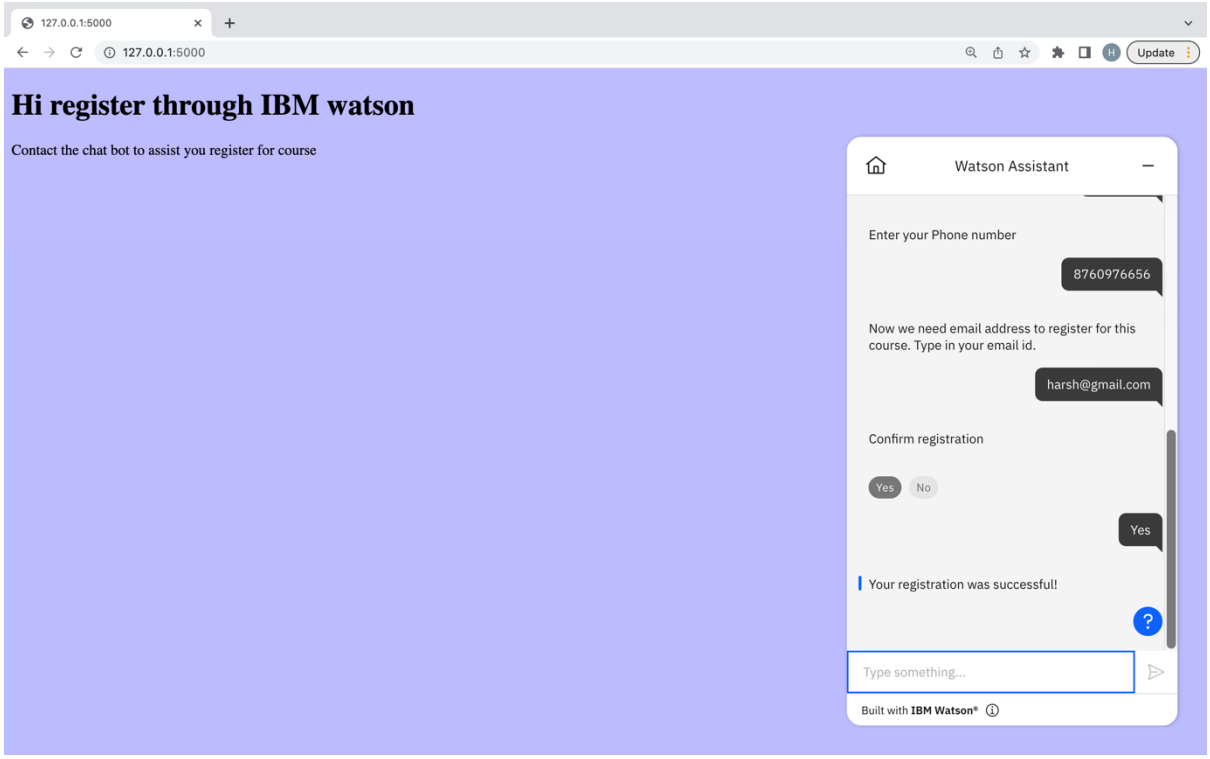
Output:


Watson Assistant
Hello
Hey, What's you name
Harsini
Here is the list of available courses, select the course you prefer.
Cloud computing
Cloud computing
We have specialised course in Cloud choose one of the following
IBM cloud
GCP - google cloud
AWS - amaz
Type something...
Built with IBM Watson®


Watson Assistant
IBM cloud
GCP - google cloud
AWS - amazon
IBM cloud
Enter your Phone number
8760976656
Now we need email address to register for this course. Type in your email id.
harsh@gmail.com
Confirm registration
Yes
No
Type something...
Built with IBM Watson®


Watson Assistant
Enter your Phone number
8760976656
Now we need email address to register for this course. Type in your email id.
harsh@gmail.com
Confirm registration
Yes
No
Yes
Your registration was successful!
Type something...
Built with IBM Watson®


IBM Watson Assistant Lite
Upgrade
Registration
Learning center
Hello
1
Hey, What's you name
Free text
Continue to next step
2
Here is the list of available courses, select the course you prefer.
Cloud compu...
Cyber security
+ 3
Continue to next step
3
2 == Cloud computing
We have specialised course in Cloud choose one of the following
IBM cloud
GCP - google...
+ 1
Continue to next step
4
2 == Cyber security , Data Science
We have discounts for this course do you wish to apply
Confirmation
Continue to next step
5
Enter your Phone number
Free text
New step
Step 3 is taken
with conditions
Conditions
1 condition
If All of this is true:
2. Here is the list...
is
Cloud computing
and
Add condition +
New condition group +
Assistant says
We have specialised course in Cloud choose one of the following
IBM cloud
GCP - google cloud
AWS - amazon
Edit response
Edit validation
And then
Preview


127.0.0.1:5000
127.0.0.1:5000
Update
Hi register through IBM watson
Contact the chat bot to assist you register for course
Watson Assistant
Enter your Phone number
8760976656
Now we need email address to register for this course. Type in your email id.
harsh@gmail.com
Confirm registration
Yes
No
Yes
Your registration was successful!
Type something...
Built with IBM Watson®