

**Assignment-3**  
**IBMObjectStorage and**  
**IBMWatsonAssistant**

AssignmentDate	28October2022
StudentName	S. Supriya
StudentRollNumber	961819104085
MaximumMarks	2Marks

**Question1:**

Create a Bucket in IBM Object storage.

**Solution:**

**ibmCloudTest.py**

```
import ibm_boto3

from ibm_botocore.client import Config, ClientError

from ibm_s3transfer.aspera.manager import AsperaTransferManager
from ibm_s3transfer.aspera.manager import AsperaConfig

from flask import Flask, render_template, url_for, request, redirect

COS_ENDPOINT = "https://s3.jp-tok.cloud-object-
storage.appdomain.cloud"
COS_API_KEY_ID = "wprEMAxjHj5sPI959w
L_3HJczOWRbYn52XUuLrDSJON"

COS_INSTANCE_CRN = "crn:v1:bluemix:public:cloud-object-
storage:global:a/602bcdcf9224f7b8c2e1aed60258292:b846251f-3216-44c8-b123-
4e13e3571cda::"

cos =

    ibm_boto3.resource("s3", ibm_api_key_id=COS_
        S_API_KEY_ID, ibm_service_instance_id=COS_
        INSTANCE_CRN, config=Config(signature_ver
            sion="oauth"), endpoint_url=COS_ENDPOINT
    )

ms_transfer_config = AsperaConfig(multi_session=2,
    multi_session_threshold_mb=100)
transfer_manager = AsperaTransferManager(client=client,
    transfer_config=ms_transfer_config)
```

```
app=Flask(name)
```

```
bucket_name = "flask-  
application"download_filename =  
"E:\IMS\static\css\Styles.css"object_name  
="Styles.css"
```

```
withAsperaTransferManager(client)astransfer_manager:  
    future=transfer_manager.download(bucket_name,object_name,download_filename)futur  
    e.result()
```

```
defget_item(bucket_name,item_name):  
    print("Retrievingitemfrombucket:{0},key:{1}".format(bucket_name,item_name))tr  
    y:  
        file = cos.Object(bucket_name,  
        item_name).get()print("FileContents:{0}".format(fil  
        e["Body"].read()))  
    exceptClientErrorasbe:  
        print("CLIENT ERROR:  
{0}\n".format(be))except Exception ase:  
        print("Unabletoretrievefilecontents:{0}".format(e))
```

```
defget_bucket_contents(bucket_name):  
    print("Retrievingbucketcontentsfrom:{0}".format(bucket_name))  
    try:  
        files=cos.Bucket(bucket_name).objects.  
        all()files_names=[]  
        print(files)  
        forfile  
            infiles:files_names.append  
            d(file.key)  
            print("Item:{0}({1}bytes)".format(file.key,file.size)  
        )returnfiles_names  
    exceptClientErrorasbe:  
        print("CLIENT ERROR:  
{0}\n".format(be))except Exception ase:
```

```
print("Unable to retrieve bucket contents:{}".format(e))
```

```
@app.route('/')
```

```
def index():
```

```
    files = get_bucket_contents('flask-
```

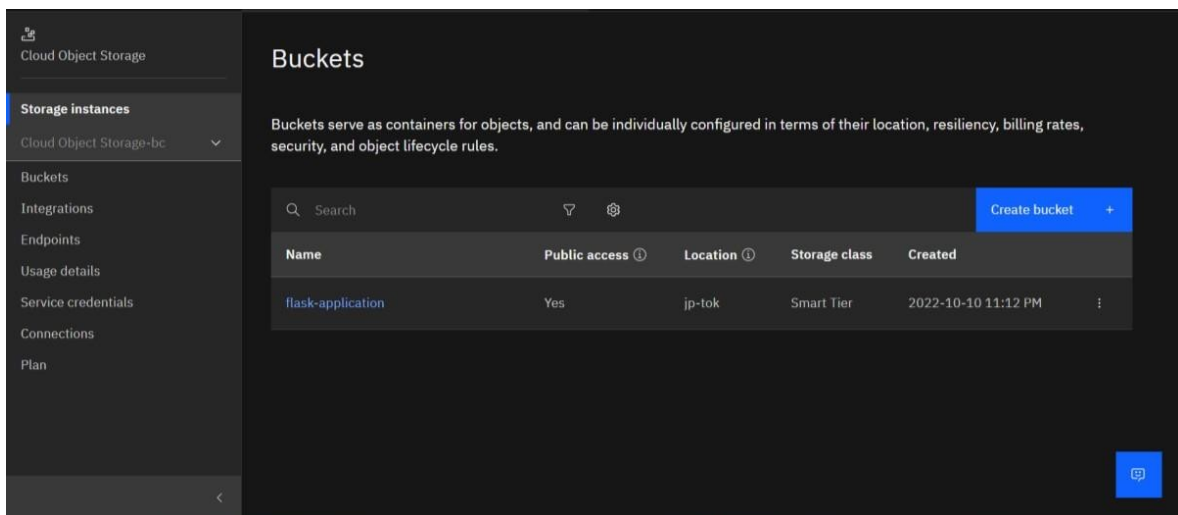
```
    application') return render_template('index.ht
```

```
    ml', files=files)
```

```
if name == '__main__': _
```

```
    app.run(debug=True)
```

## Output:



The screenshot shows the Google Cloud Platform console interface. On the left, the 'Storage instances' section is expanded, showing a list of storage instances. The main area displays the 'Buckets' page, which includes a search bar, a 'Create bucket' button, and a table of existing buckets.

Name	Public access	Location	Storage class	Created
flask-application	Yes	jp-tok	Smart Tier	2022-10-10 11:12 PM

## Question2:

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

## Solution:

### index.html

```
<h1>IBM Cloud Storage and IBM Watson Assistant</h1>
```

```
<br><hr>
```

```
{%block head%
```

```
<!--CSS-->
```

```
<link rel="stylesheet" href="{{url_for('static',filename='css/Styles.css')}}"/>
```

```
{%endblock%
```

```
<script>window.watsonAssistantChatOp
```

```
tions = {
```

```
  integrationID: "8abe6aef-f2bb-4376-be93-
```

```
  7fda1857ec36", // The ID of this integration. region: "jp-tok", // The region
```

```
  your integration is hosted in.
```

```
  serviceInstanceID: "440aaee1-fbb4-4c2b-8604-cb5d605fa157", // 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>
```

```
<!doctype html>
```

```
<html>
```

```
<body>
```

```
  {%for row in files%}
```

```

<divstyle="border:1pxsolid#EFEFEF;margin:10px;">

    <h3>Filename:{{row}}</h3>

    <imgsrc="https://flask-application.s3.jp-tok.cloud-object-
storage.appdomain.cloud/{{row}}"width="150px"></td>

</div>

{%endfor%}

</body>

</html>

```

## Output:

The screenshot shows the AWS Cloud Object Storage console for a bucket named 'flask-application'. The left sidebar contains navigation options like 'Storage instances', 'Cloud Object Storage-bc', 'Buckets', 'Integrations', 'Endpoints', 'Usage details', 'Service credentials', 'Connections', and 'Plan'. The main area shows the bucket's 'Objects' tab with a warning: 'Warning: All objects in this bucket have public view access.' Below the warning, there is a table of objects with columns for 'Object name', 'Archived', 'Size', and 'Last modified'. The objects listed are 'Building.jpg', 'Consultation.jpg', 'Doctors.jpg', 'Hospital Management.png', 'Hospital.jpg', and 'Styles.css'. An 'Upload' button is visible in the top right corner of the object list.

Object name	Archived	Size	Last modified
Building.jpg		7.6 KB	2022-10-17 12:11 AM
Consultation.jpg		16.0 KB	2022-10-17 12:12 AM
Doctors.jpg		100.4 KB	2022-10-17 12:12 AM
Hospital Management.png		9.7 KB	2022-10-17 12:12 AM
Hospital.jpg		7.1 KB	2022-10-17 12:16 AM
Styles.css		403 bytes	2022-10-11 9:52 AM

**Question3:**

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

**Solution:****Styles.css**

```
body{
    background-
    color:rgb(190,229,247);text-
    align:center;
    font-style:italic ;
}
.outercont
{
    height:
    100vh;width:
    100vw;displa
    y:flex;
    flex-direction:column;
    justify-content:    space-
    around;align-items: center;
}
p{
    font-size:20px;
}
h1{
    text-align:
    center;font-
    style:normal;
}
h2{
    text-align:center;
}
.cont {
    width:700px;
}
```

```

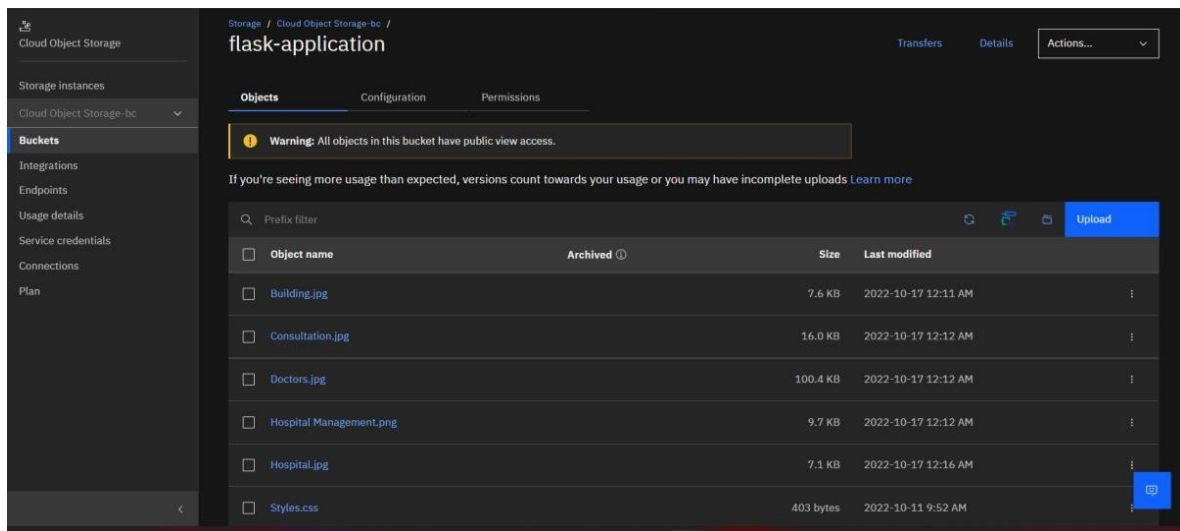
.leftcont{
    font-size:17px;
}

.rightcont{
    font-size:17px;
}

.othercont
{display:flex;
}

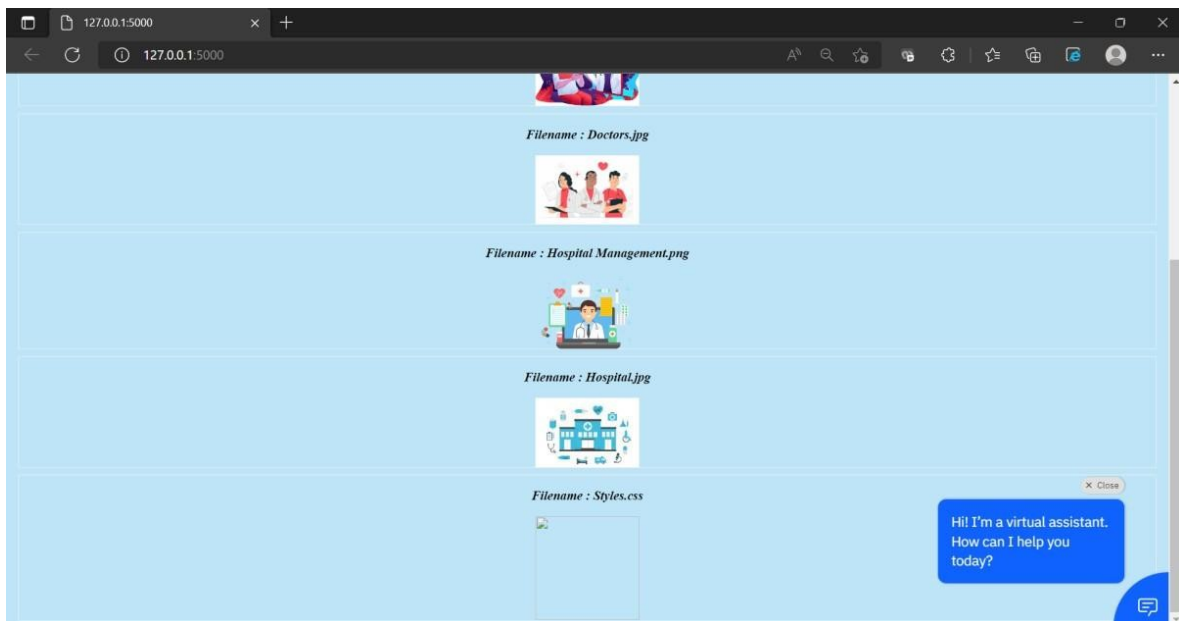
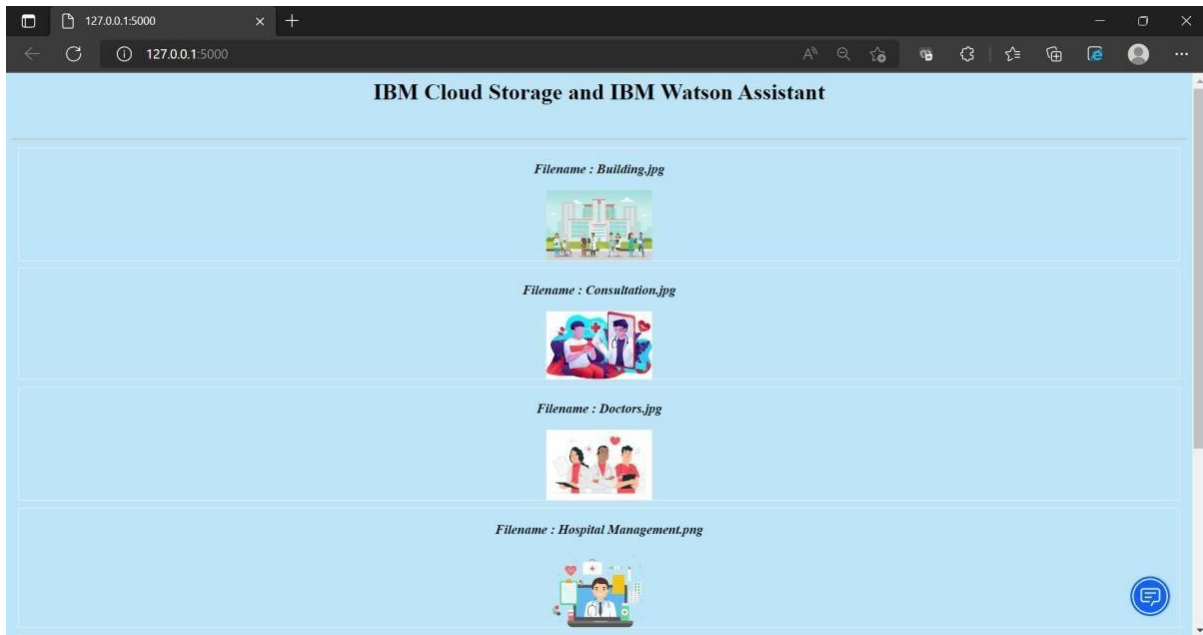
```

## Output:





## Web Page:



#### Question-4:

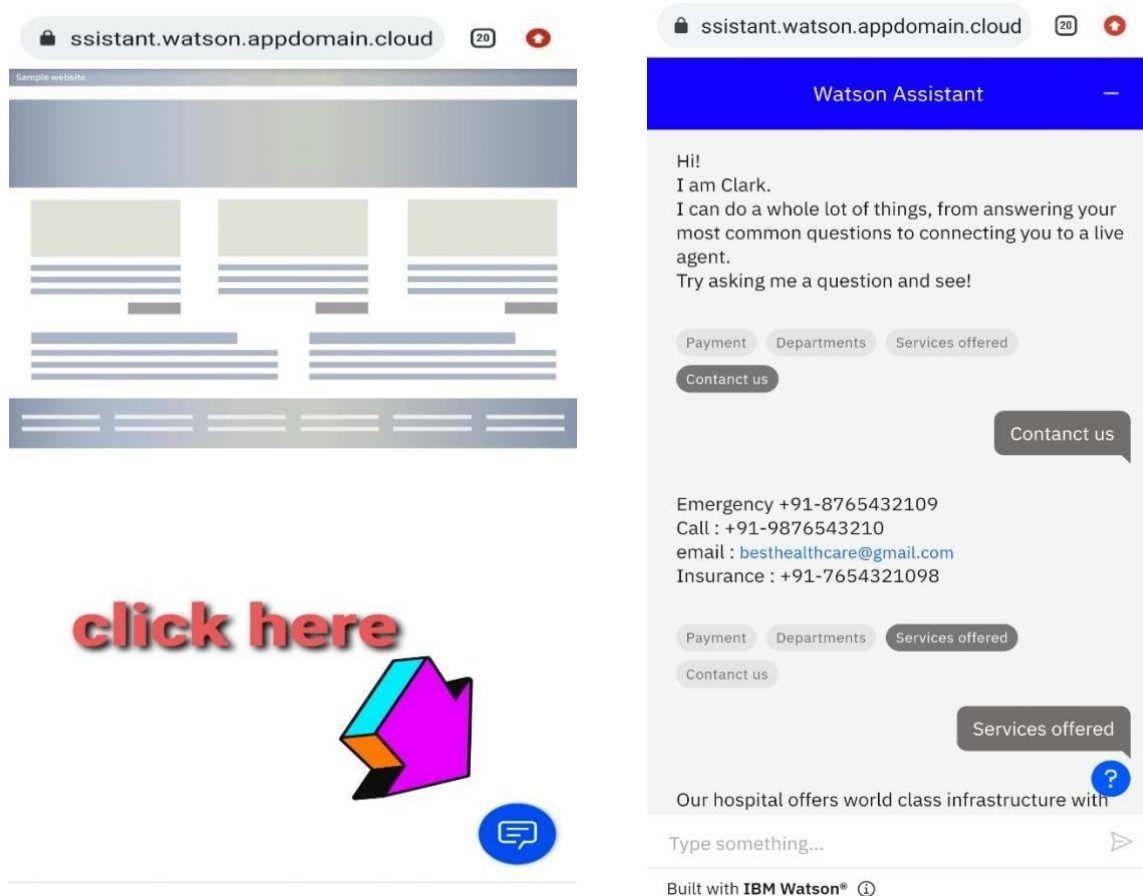
Design a chatbot

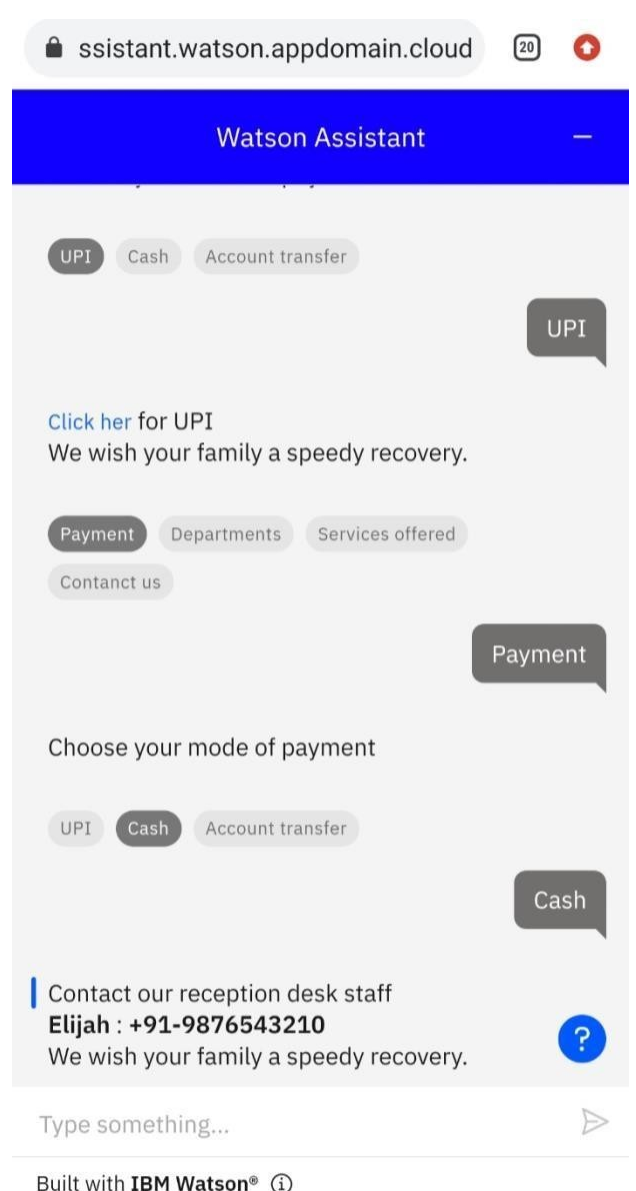
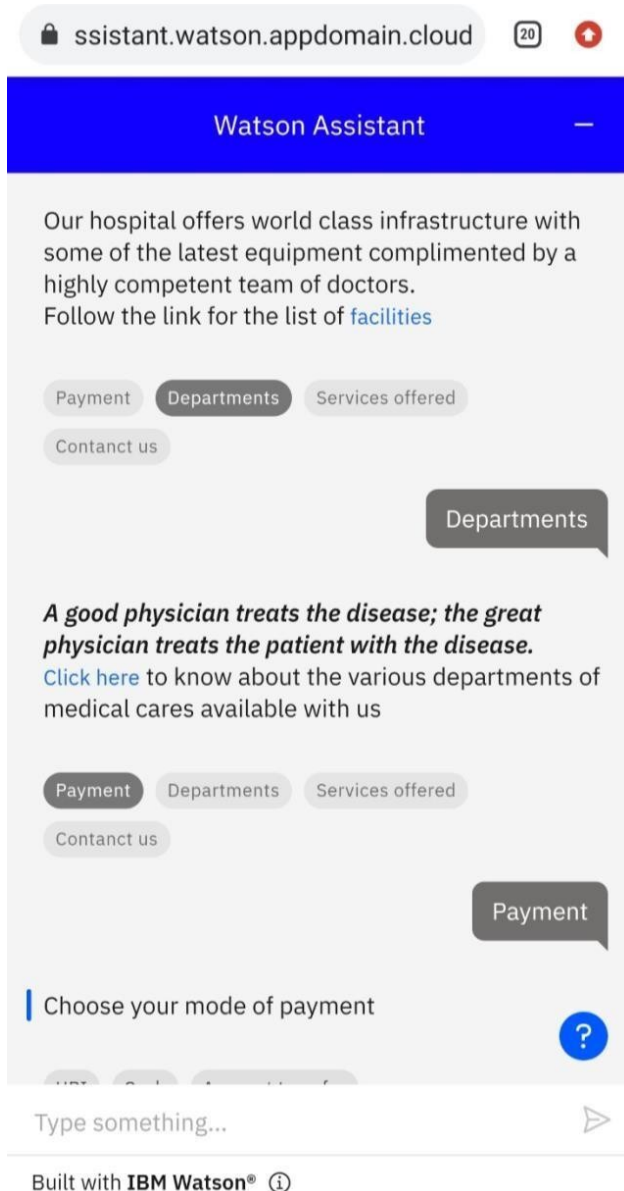
using IBM Watson Assistant for hospital. Ex: User comes with query to know the branches for that hospital in your city. Submit the web URL of that chat bot as a assignment.

#### Solution:

<https://web-chat.global.assistant.watson.appdomain.cloud/preview.html?backgroundImageUrl=https%3A%2F%2Fjp-tok.assistant.watson.cloud.ibm.com%2Fpublic%2Fimages%2Fupx-440aeee1-fbb4-4c2b-8604-cb5d605fa157%3A%3Affa26b03-3295-42fd-a098-6723615c4e58&integrationID=a3e3b932-8ee4-43ee-9ccb-f3cfa9687188&region=jp-tok&serviceInstanceID=440aeee1-fbb4-4c2b-8604-cb5d605fa157>

#### Output:





**Question-5:**

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

**Solution:**

```
<script>window.watsonAssistantChatOptions = {
  integrationID: "8abe6aef-f2bb-4376-be93-7fda1857ec36", // The ID of this integration. region: "jp-tok", // The region
  your integration is hosted in.
  serviceInstanceID: "440aaee1-fbb4-4c2b-8604-cb5d605fa157", // 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>
```

## Steps:

### Stepsinpayment

Make a payment

Customer starts with:  
Complete payment

Conversation steps

1

Choose your mode of payment

UPI Account tran... + 1

Continue to next step

1 == UPI

[Click her](https://pay.google.com/intl/en\_in/about/) for Step 1 We wish your family a speedy recovery.

Go to action: What can you do?

1 == Account transfer

RTGS Account Number: 1234567890 IFSC code of the Branch :SBIN0981

Go to action: What can you do?

1 == Cash

Contact our reception desk staff Elijah : +91-9876543210 We wish your family a speedy recovery.

New step +

### Stepsinlocation

IBM Watson Assistant Lite Upgrade Clark

Location

Customer starts with:  
Can you please give me a list of the types of things you can he...

Conversation steps

1

We have our branch all around the world [Click here] (https://www.google.com/maps/search/best+health+...)

Go to action: What can you do?

New step +

stepsinWhat canyou do

IBM Watson Assistant LiteUpgradeClark ▾

What can you do?

Conversation steps

1

This step has no content

DepartmentsPayment+ 2

↓ Continue to next step

1 == Payment

2

This step has no content

Go to action: **Make a payment**

1 == Services offered

3

Our hospital offers world class infrastructure with some of the latest equipment complimented by a...

Re-ask previous step(s)

1 == Departments

4

A good physician treats the disease; the great physician treats the patient with the disease. [Click...

Re-ask previous step(s)

1 == Contact us

5

Emergency +91-8765432109 Call : +91-9876543210 email : besthealthcare@gmail.com...

Re-ask previous step(s)

New step +

## Conditions:

Conditions in payment

- UPI
- Account transfer
- Cash

Step 2 is taken

with conditions

fs

Conditions

1 condition

If

All

of this is true:

1. Choose your m...

is

UPI

X

and

Add condition +

New condition group +

defined

UPI

Account transfer

Cash

Assistant says

B

I

U

Y

Q

W

E

R

T

</>

Click [here](#) for 1. Choose your mode of payment

We wish your family a speedy recovery.

Define customer response

And then

Preview

>

## Conditions in What can you do

- Departments
- Payment
- Services offered
- Contact us

The screenshot displays the IBM Watson Assistant Lite interface, showing a conversation flow configuration. The interface is divided into several sections:

- Header:** IBM Watson Assistant Lite | Upgrade | Clark
- Left Panel (Conversation steps):** A list of steps for the conversation flow. Step 1 is highlighted, showing a condition "1 == Payment" and an action "Go to action: Make a payment".
- Right Panel (Conditions):** A section for defining conditions. It shows a condition "1. No response" with the operator "is" and the value "Payment". A dropdown menu is open, showing options: "defined", "Departments", "Payment", "Services offered", and "Contact us".
- Assistant says:** A section for defining the assistant's response. It shows a text input field with the placeholder "For example: What type of transfer would you like to make?".
- And then:** A section for defining the next action. It shows a dropdown menu with the option "Go to another action".
- Bottom:** A "New step" button and a "Preview" button.



## Output:

