

Assignment -3
IBM Object Storage and IBM Watson Assistant

Assignment Date	08 November 2022
Student Name	Rajesh Kumar J
Student Roll Number	961819104069
Maximum Marks	2 Marks

Question 1:

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="wprEMAxjHj5sPI959wL_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_API_KEY_ID,

    ibm_service_instance_id=COS_INSTANCE_CRN,

    config=Config(signature_version="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"
```

```
with AsperaTransferManager(client) as transfer_manager:
```

```
    future = transfer_manager.download(bucket_name, object_name, download_filename)
```

```
    future.result()
```

```
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 = []
```

```
        print(files)
```

```
        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))
```

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

```
def index():
```

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

```
    return render_template('index.html', files = files)
```

```
if __name__ == '__main__':
```

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

Output:

Cloud Object Storage

Storage instances

Cloud Object Storage-bc

Buckets

Integrations

Endpoints

Usage details

Service credentials

Connections

Plan

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.

Search

Create bucket +

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

Question 2:

Upload 5 images to ibm object storage and make it public. write html code to displaying 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.watsonAssistantChatOptions = {
```

```
        integrationID: "8abe6aef-f2bb-4376-be93-7fda1857ec36", // The ID of this integration.
```

```
        region: "jp-tok", // The region your integration is hosted in.
```

```
        serviceInstanceID: "440aeee1-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 %}
```

```

<div style="border: 1px solid #E0E0E0;margin:10px;">

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

    </td>

</div>

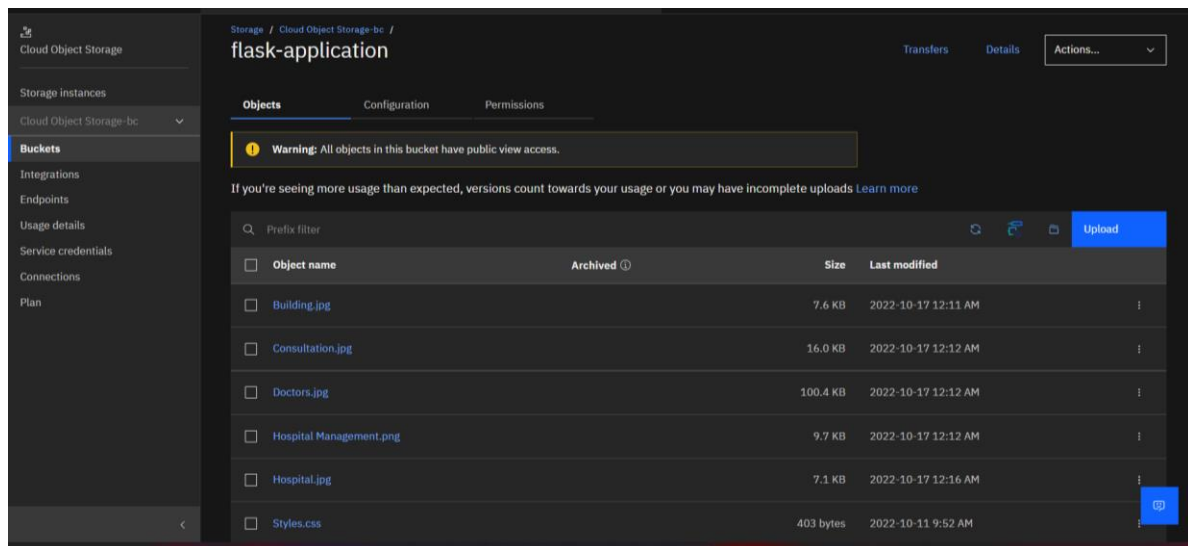
{% endfor %}

</body>

</html>

```

Output:



Question 3:

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;  
    display: 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;  
}
```

```

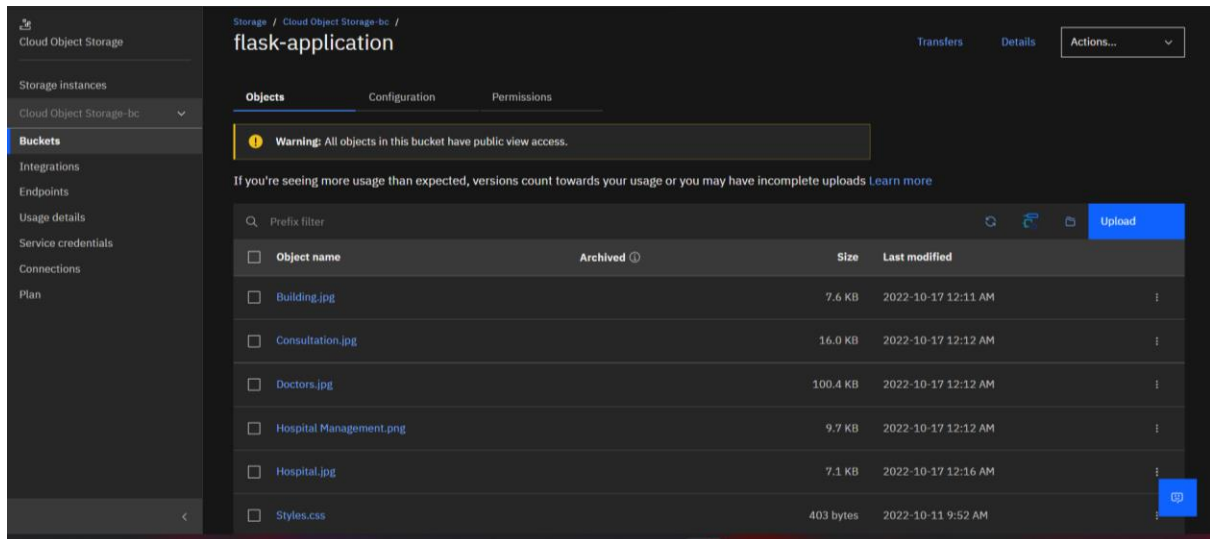
.leftcont {
    font-size: 17px;
}

.rightcont {
    font-size: 17px;
}

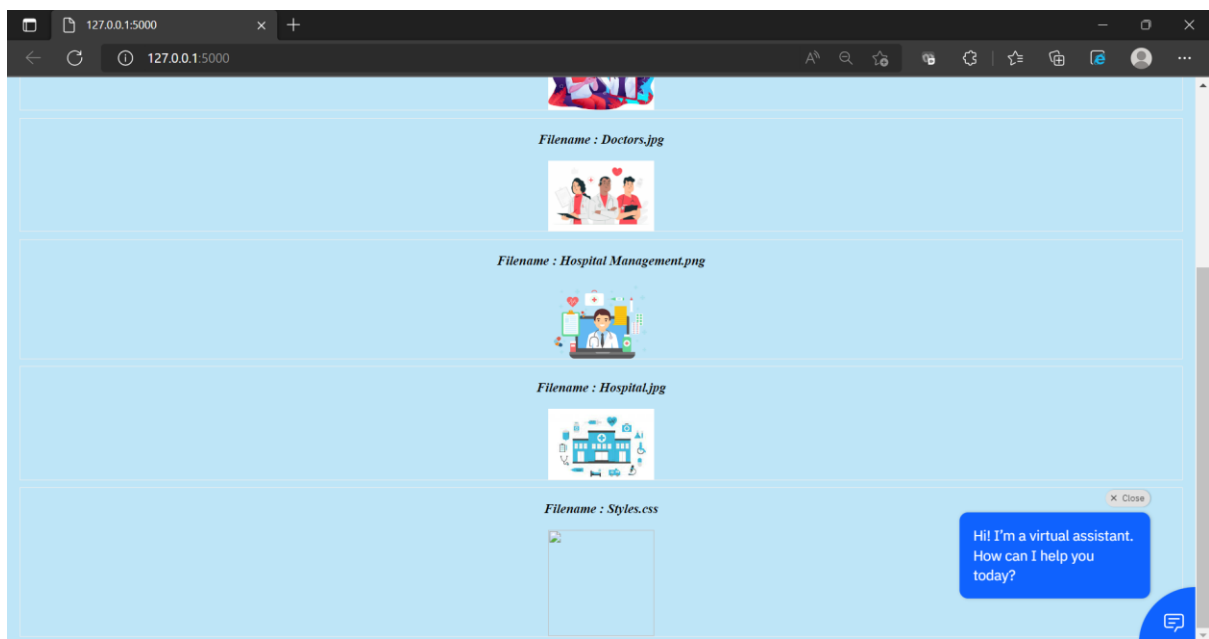
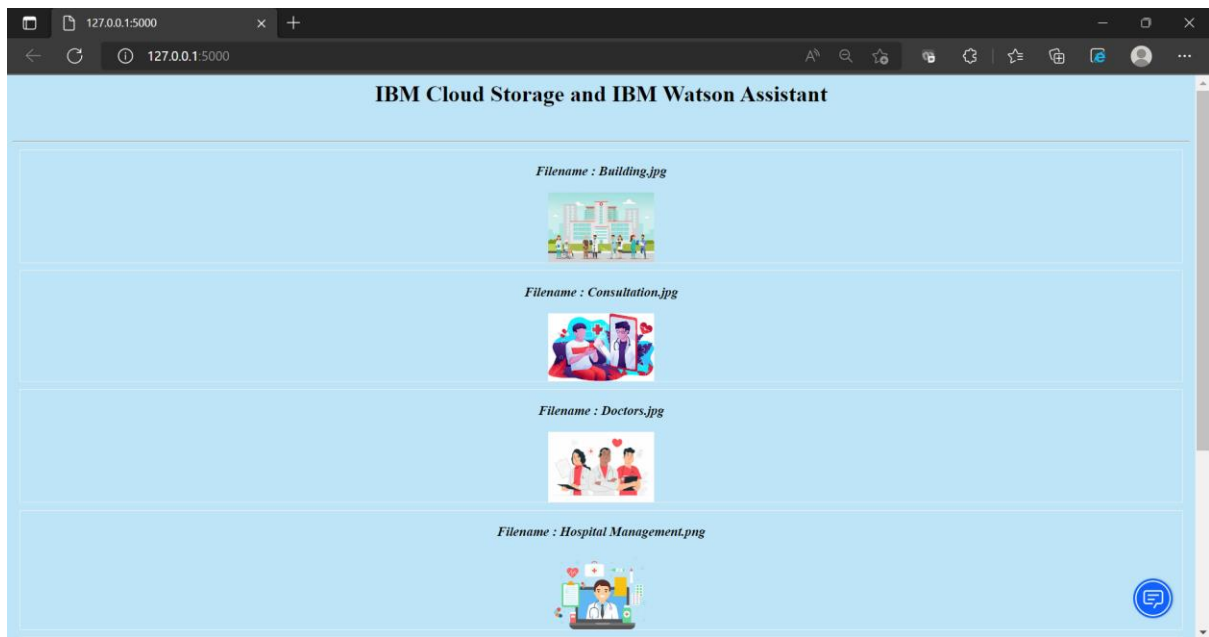
.othercont {
    display: flex;
}

```

Output:



Web Page:



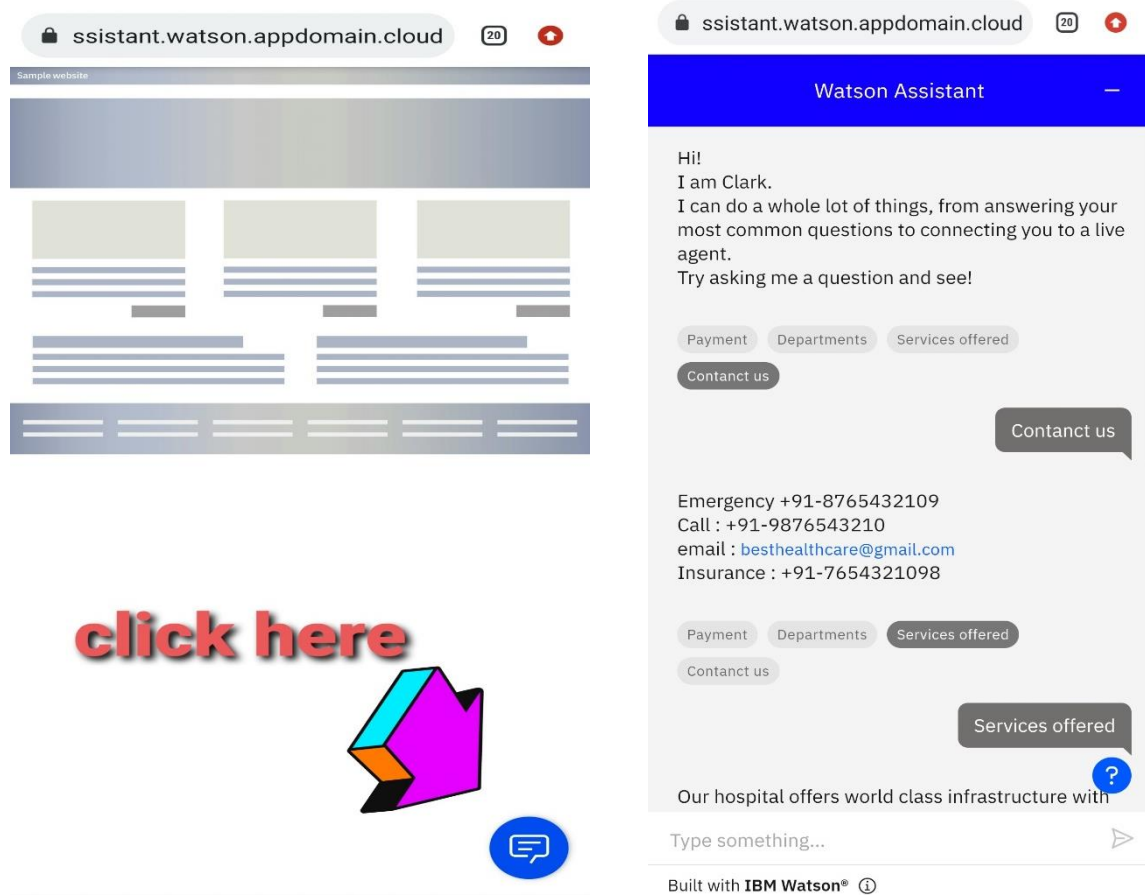
Question-4:

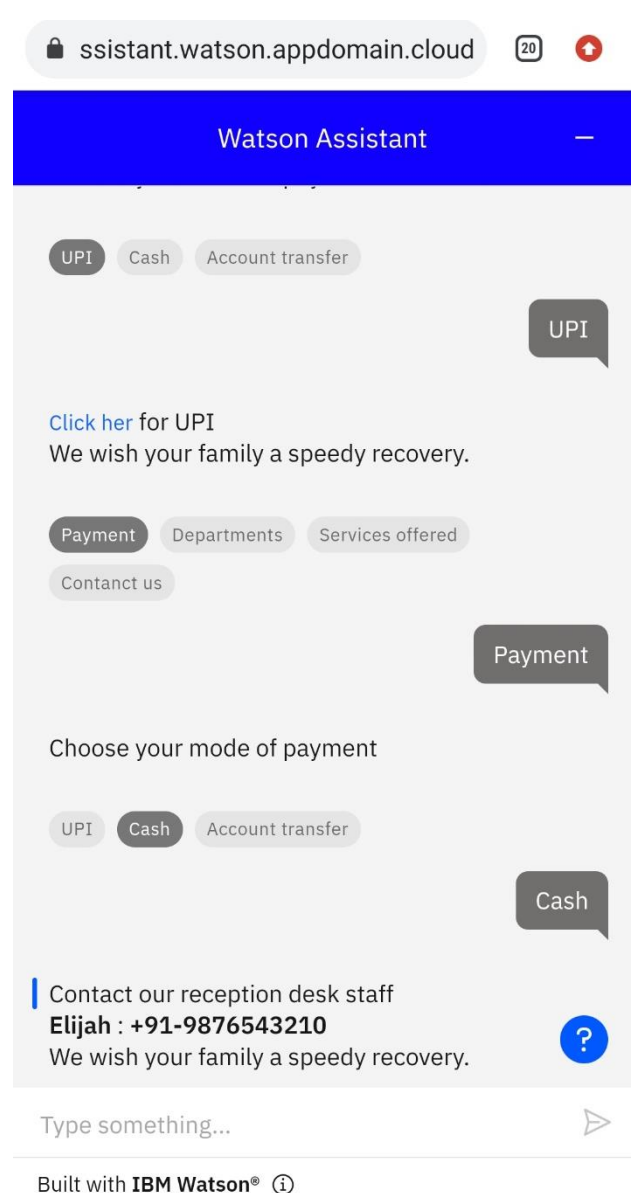
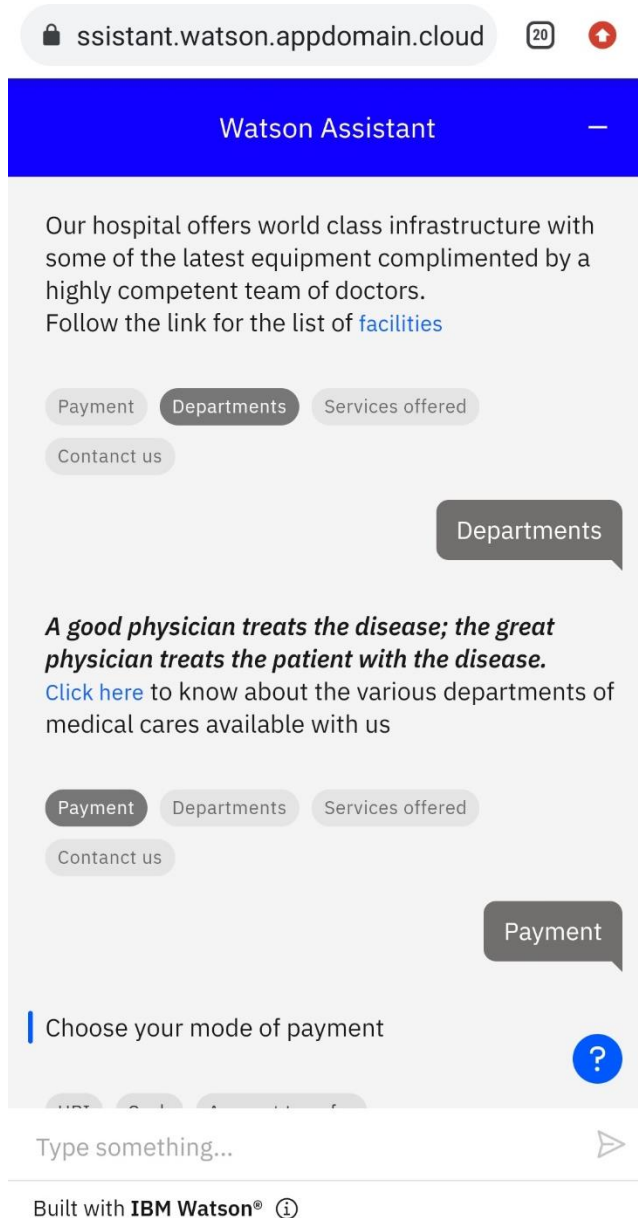
Design a chatbot using IBM Watson assistant for hospital. Ex: User comes with query to know the branches for that hospital in your city. Submit the web URL of that chat bot as a assignment.

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-440aaee1-fbb4-4c2b-8604-cb5d605fa157%3A%3Affa26b03-3295-42fd-a098-6723615c4e58&integrationID=a3e3b932-8ee4-43ee-9ccb-f3cfa9687188®ion=jp-tok&serviceInstanceID=440aaee1-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: "440aeee1-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:

Steps in payment

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/)

2 for Step 1 We wish your family a speedy recovery.

Go to action: What can you do?

1 == Account transfer

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

Go to action: What can you do?

1 == Cash

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

New step +

Steps in location

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 +

steps in What can you do

IBM Watson Assistant Lite Upgrade Clark ▾

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

The screenshot displays the Microsoft Bot Framework Designer interface for a bot named 'My Bot'. The current step is 'Step 2 is taken', which is a 'with conditions' step. The condition is defined as 'If All of this is true: 1. Choose your mode of payment is UPI'. The 'Add condition' dropdown is open, showing options: 'defined', 'UPI', 'Account transfer', and 'Cash'. The 'Assistant says' section shows the bot's response: 'Click her for 1. Choose your mode of payment. We wish your family a speedy recovery.' The 'And then' section is empty.

Step 2 is taken with conditions 1 condition

Conditions 1 condition

If All of this is true:

1. Choose your mode of payment is UPI

and Add condition +

New condition group +

defined

UPI

Account transfer

Cash

Assistant says

Click her for 1. Choose your mode of payment.

We wish your family a speedy recovery.

And then

Define customer response

Preview

Conditions in What can you do

- Departments
- Payment
- Services offered
- Contact us

The screenshot displays the IBM Watson Assistant Lite interface for configuring a dialog. The top bar shows 'IBM Watson Assistant Lite', 'Upgrade', and 'Clark'. The main workspace is titled 'What can you do?' and contains several panels:

- Conversation steps:** A list of steps for the dialog. Step 1 is 'Payment' (highlighted with a blue box). Step 2 is 'This step has no content'. Step 3 is 'Services offered'. Step 4 is 'Departments'. Step 5 is 'Contact us'. A 'New step +' button is at the bottom.
- Conditions:** A section for defining conditions. It shows 'Step 2 is taken with conditions'. A condition is defined: 'If All of this is true: 1. No response is Payment'. A dropdown menu is open, showing options: 'defined' (highlighted with a blue box), 'Departments', 'Payment', 'Services offered', and 'Contact us'. A 'New condition group +' button is also present.
- 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?'. A 'Define customer response' button is at the bottom.
- And then:** A section for defining the next action. It shows a 'Go to another action' button. Below it, a table shows the action: 'Goes to action' with the value 'Make a payment'.

A 'Preview' button is located at the bottom right of the interface.

Output:

