

### Assignment 3

NAME	MANIKANDAN.P
TEAM ID	PNT2022TMID43190
ROLL NO	714219106004
Project Title	Inventory Management System for Retailers

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

The screenshot displays the IBM Watson Assistant Lite web interface. The browser address bar shows the URL: `au-syd.assistant.watson.cloud.ibm.com/cm%3Av1%3Abluemix%3Apublic%3Aconversation%3Aau-syd%3Aa%2Fb63c687f12374d47be36aad8fd99bca%3A30d0b70...`. The interface includes a top navigation bar with 'IBM Watson Assistant Lite', 'Upgrade', and 'Assistant for h...'. A 'Learning center' link is also present.

The main workspace shows a conversation flow with 6 steps. The first step is 'Can I find out if my loved one is a patient at a particular...'. The subsequent steps are:

- Step 3: 'What kinds of information can I learn from a hospital when my loved one was unconscious...'
- Step 4: 'When I am a patient at a medical facility...'
- Step 5: 'I do not want to share my information with anyone - not even my closest family members. Can I...'
- Step 6: 'Can the hospital tell my member of the clergy (e.g. priest, rabbi, etc.) my room number?'

Each step has a 'Continue to next step' button. A 'New step +' button is located at the bottom left of the flow.

On the right side, the 'Step 7 is taken' section shows the assistant's response to the input 'without conditions'. The response is:

*I would like to list my name and room number, but not my medical condition and religious affiliation, in my hospital's directory. Can I limit the type of information included in my directory listing?*

Below the response, there is a 'Define customer response' button. The 'And then' section shows a 'Continue to next step' button.

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

The top screenshot shows the IBM Watson Assistant Lite interface. The conversation flow is as follows:

- Step 3: Can I find out if my loved one is a patient at a particular hospital when my loved one was unconscious...
- Step 4: When I am a patient at a medical facility...
- Step 5: I do not want to share my information with anyone - not even my closest family members. Can I...  
Assistant says: *I do not want to share my information with anyone - not even my closest family members. Can I request to have my information excluded from my hospital's directory?*
- Step 6: Can the hospital tell my member of the clergy (e.g. priest, rabbi, etc.) my room number?
- Step 7: I would like to list my name and room number, but not my medical condition and religious affiliation...

The bottom screenshot shows the IBM Watson Assistant Lite interface. The conversation flow is as follows:

- Customer starts with: Can I find out if my loved one is a patient at a particular...
- Step 1: A: Generally, yes. Under the HIPAA Privacy Rule, if you ask for a loved one by name, a hospital or...
- Step 2: What other kinds of information can a hospital disclose about my loved one?
- Step 3: What kinds of information can I learn from a hospital when my loved one was unconscious...
- Step 7: I would like to list my name and room number, but not my medical condition and religious affiliation, in my hospital's directory. Can I limit the type of information included in my directory listing?

## Index.html

```
<!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>Home</title>

<link rel="stylesheet" href="{url_for('redirect_to',link='https://s3.jp-tok.cloud-object-
storage.appdomain.cloud/cloudbucket/assign3.css')}" type="text/css">

<script> window.watsonAssistantChatOptions = { integrationID: "14b83b8f-3dfd-405f-
9520b550092892aa", // The ID of this integration. region: "us-south", // The region your
integration is hosted in. serviceInstanceID: "6e95bee9-8d0b-49f6-8a2f-4125fb3a7945", // 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> </head>

<body>

    <form action="/uploader" method="POST" enctype="multipart/form-data">

        <input type="text" placeholder="Enter file name" name="filename" />

        <br />

        <br />

        <input type="file" name="file" />

        <br />

        <br />

        <input type="submit" />
    </form>

<br/>

```

```

<br/>

<br/>

{% for row in files %}

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

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

        </td>

    </div>

{% endfor %}

</body> </html>

```

## App.py

```

import io from
flask import

Flask,redirect,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="" COS_INSTANCE_CRN=""

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__)
```

```
@app.r
```

```
oute('/') )
```

```
def index():
```

```
try:
```

```
    files =
```

```
    cos.Bucket('cloudbucket').objects.
```

```
    all() files_names = [] for file in
```

```
    files: files_names.append(file.key)
```

```
    print(file) print("Item: {0} ({1}
```

```
    bytes)".format(file.key, file.size))
```

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

```
except ClientError as be: print("CLIENT
```

```
ERROR: {0}\n".format(be)) return
```

```
render_template('index.html') except
```

```
Exception as e:
```

```
    print("Unable to retrieve bucket contents:
```

```
    {0}".format(e)) return render_template('index.html')
```

```
@app.route('/uploader',methods=['P
```

```
OST']) def upload():
```

```
    name_file=request.for
```

```
    m['filename'] f =
```

```
request.files['file'] try: part_size
```

```
= 1024 * 1024 * 5
```

```
file_threshold = 1024 * 1024 * 15
```

```
transfer_config = ibm_boto3.s3.transfer.TransferConfig( multipart_threshold=file_threshold,  
    multipart_chunksize=part_size
```

```
)
```

```
content = f.read()
```

```
cos.Object('cloudbucket', name_file).upload_fileobj(  
    Fileobj=io.BytesIO(content),  
    Config=transfer_config  
)
```

```
return redirect(url_for('index'))
```

```
except ClientError as be:
```

```
    print("CLIENT ERROR:
```

```
    {0}\n".format(be)) return
```

```
    redirect(url_for('index'))
```

```
except Exception as e:
```

```
    print("Unable to complete multi-part upload:
```

```
    {0}".format(e)) return redirect(url_for('index'))
```

```
if __name__=='_main_':
```

```
    app.run(host='0.0.0.0',port=8080,debug=True
```

