ASSIGNMENT 3

TEAM ID	PNT2022TMID37533
STUDENT NAME	VENKATESH V
STUDENT REGISTER NUMBER	312019104063

QUESTION:

Create a Bucket in IBM object storage. Upload an 5 images to ibm object storage and make it public. write html code to displaying all the 5 images. Upload a css page to the object storage and use the same page in your HTML code. 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. Create Watson assistant service with 10 steps and use 3 conditions in it. Load that script in HTML page.

SOLUTION:

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>
                <script>
              window.watsonAssistantChatOptions = {
                integrationID: "6ab36d7d-b59e-4964-bc8a-44cb324ca125", // The ID of this
            integration.
                region: "au-syd", // The region your integration is hosted in.
```

```
    serviceInstanceID: "04f4c174-6106-47a5-8a6f-71ef403473e3", // 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>
            <img src="https://hospital-flask.s3.jp-tok.cloud-object-
storage.appdomain.cloud/{{row}}" width="150px"></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.route('/')
         def index():
           try:
                 files = cos.Bucket('hospital-flask').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=['POST'])
         def upload():
           name_file=request.form['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('hospital-flask', 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)
```