ASSIGNMENT 3

|  |  |
| --- | --- |
| BATCH NUMBER | B3-3M5E |
| STUDENT NAME | GOKUL SRI RAM S |
| STUDENT REGISTER NUMBER | 921319104044 |

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

|  |
| --- |
| <!DOCTYPEhtml> |
|  | <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> |

|  |
| --- |
| from flask import Flask, render\_template, request, redirect, url\_for, session |
| jhhjjhhhj |  | 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) | |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |