National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

SE-315: Cloud Computing

Lab 09: App Dev - Storing Image and Video Files in Cloud Storage

CLO3: **Distinguish** the various characteristics of public, private and hybrid cloud delivery models.

Date: 20.11.24

Lab 09: App Dev - Storing Image and Video Files in Cloud Storage

Introduction:

Cloud Storage allows world-wide storage and retrieval of any amount of data at any time. It can be used for a range of scenarios including serving website content, storing data for archival and disaster recovery, or distributing large data objects to users via direct download.

In this lab you'll configure an application to use Cloud Storage to store and retrieve application data. The application is an online Quiz, the data is the form data, including an image uploaded from user's local machine.

Lab Tasks

Go through the following link:

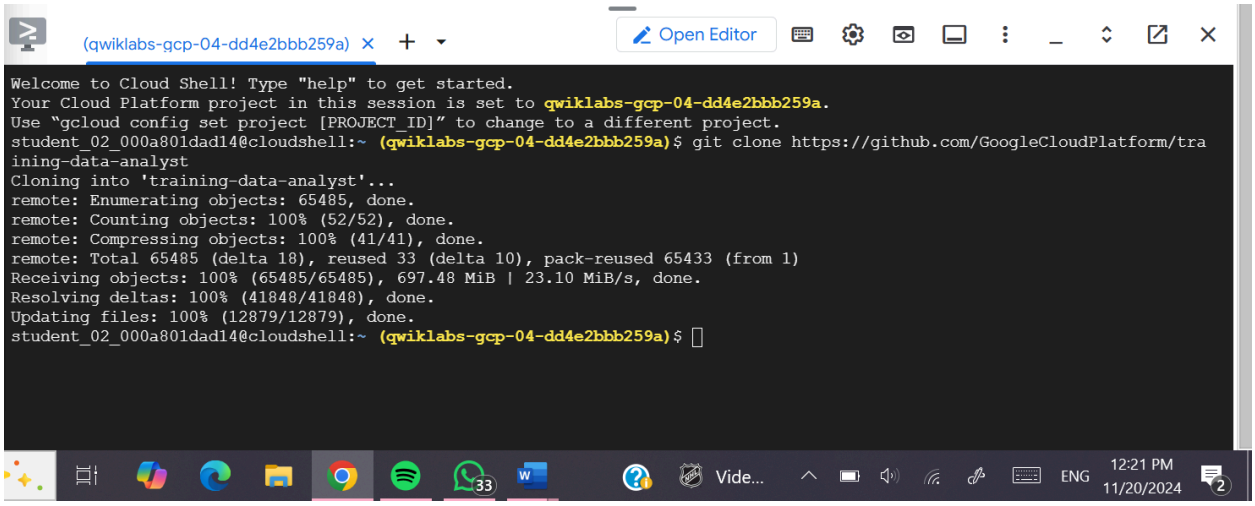
https://www.cloudskillsboost.google/focuses/1075?parent=catalog

which will take you to the 'App Dev: Storing Image and Video Files in Cloud Storage – Python' page. The list of tasks is given below. Make sure to take screenshots of each task as you will need to add them in the solution section given below.

Task 1. Prepare the quiz application

In this section, you access Cloud Shell, clone the git repository containing the Quiz application, and run the application.

Clone source code in Cloud Shell



Configure and run the quiz application

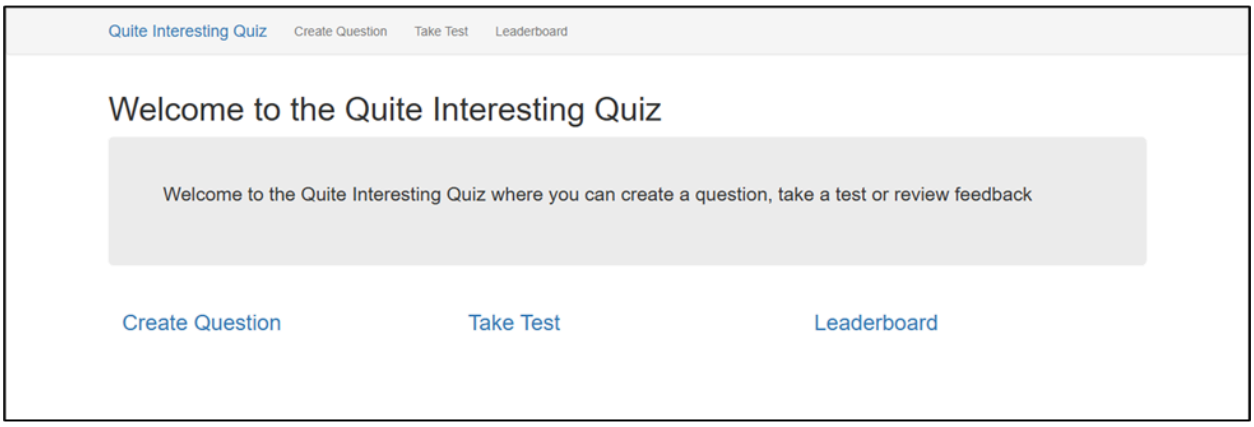
setting the region and configuring the application:

```
student_02_000a801dad14@cloudshell:~ (qwiklabs-gcp-04-dd4e2bbb259a)$ cd ~/training-data-analyst/courses/developingapps/python/cloudstorage/start
student_02_000a801dad14@cloudshell:~/training-data-analyst/courses/developingapps/python/cloudstorage/start (qwiklabs-gcp-04-dd4e2bbb259a)$ REGION=us-west1
sed -i s/us-central/$REGION/g prepare_environment.sh
student_02_000a801dad14@cloudshell:~/training-data-analyst/courses/developingapps/python/cloudstorage/start (qwiklabs-gcp-04-dd4e2bbb259a)$ . prepare_environment.sh
Creating Datastore/App Engine instance
You are creating an app for project [qwiklabs-gcp-04-dd4e2bbb259a].
WARNING: Creating an App Engine application for a project is irreversible and the region
cannot be changed. More information about regions is at
<https://cloud.google.com/appengine/docs/locations>.
```

Running the server:

```
art (qwiklabs-gcp-04-dd4e2bbb259a)$ python run_server.py
 * Serving Flask app 'quiz'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:8080
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 401-546-729
```

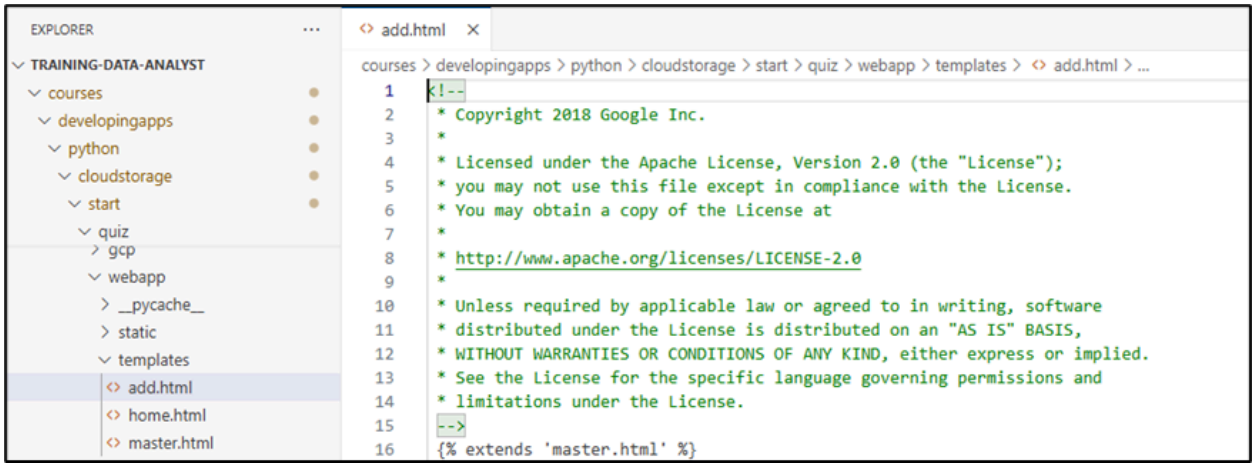
Review the quiz application



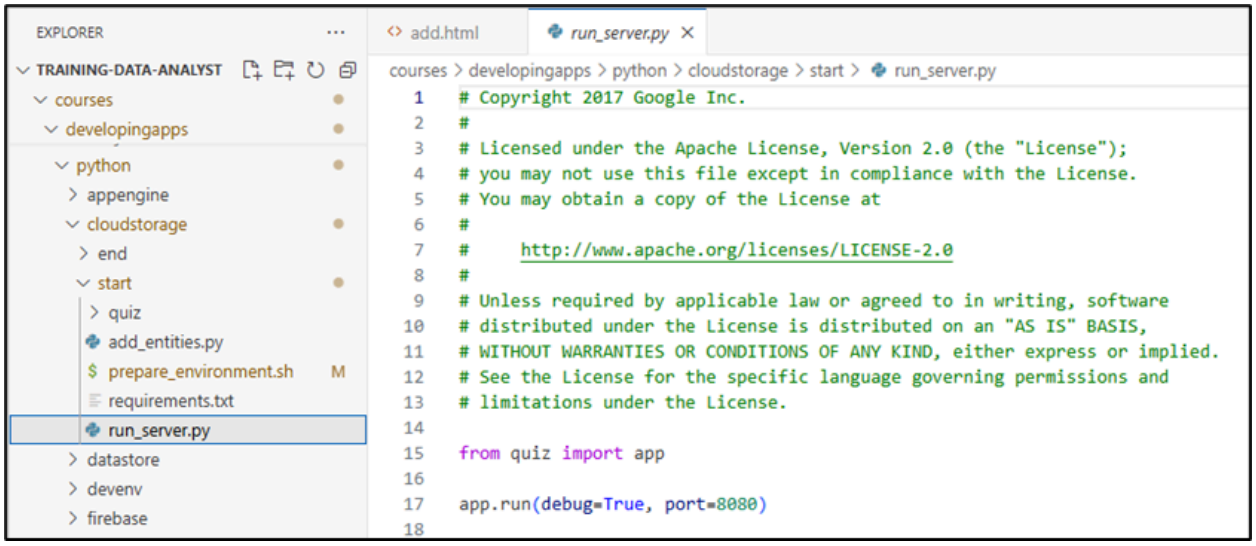
Task 2. Examine the quiz application code

Examine the application code

add.html file

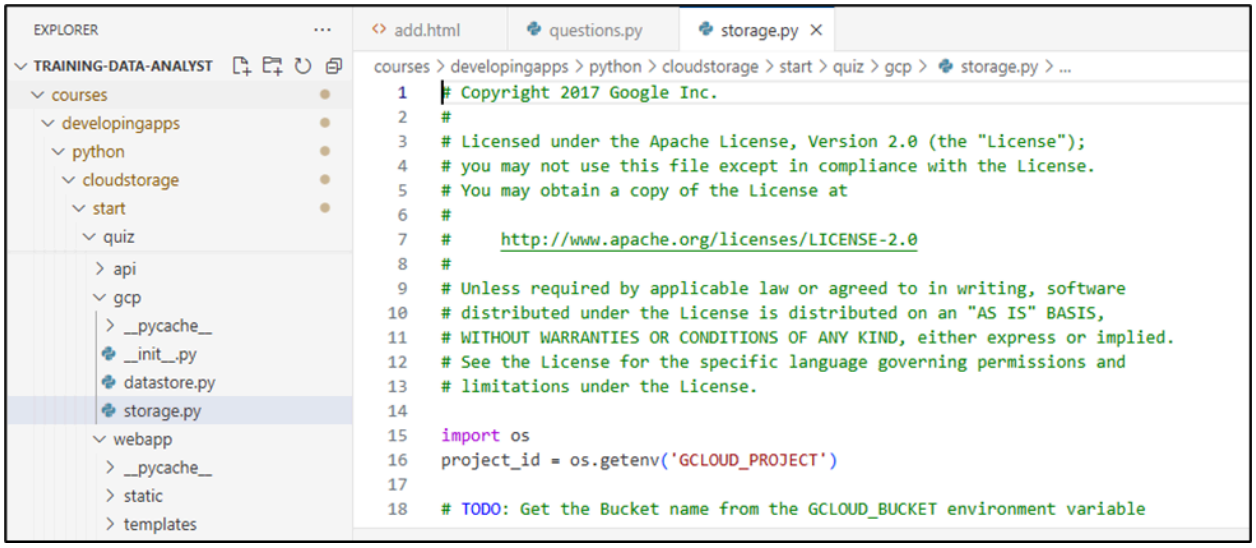


routes.py



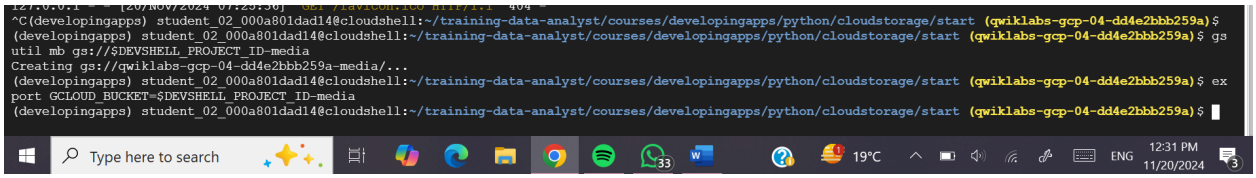
questions.py

storage.py



Task 3. Create a Cloud Storage Bucket

Create a Cloud Storage bucket named <Project ID>-media and export the Cloud Storage bucket name as an environment variable named GCLOUD_BUCKET



Task 4. Adding objects to Cloud Storage

Import and use the Python Cloud Storage module

Write code to send a file to Cloud Storage

storage.py:

```
15  import os
16  project_id = os.getenv('GCLOUD_PROJECT')
17
18  # TODO: Get the Bucket name from the
19  # GCLOUD_BUCKET environment variable
20  bucket_name = os.getenv('GCLOUD_BUCKET')
21  # END TODO
22
23  # TODO: Import the storage module
24  from google.cloud import storage
25  # END TODO
26
27  # TODO: Create a client for Cloud Storage
28  storage_client = storage.Client()
29  # END TODO
30
31  # TODO: Use the client to get the Cloud Storage bucket
32  bucket = storage_client.get_bucket(bucket_name)
33  # END TODO
34
35  """
36  Uploads a file to a given Cloud Storage bucket and returns the public url to the new object.
37  """
38  def upload_file(image_file, public):
39
40      # TODO: Use the bucket to get a blob object
41      blob = bucket.blob(image_file.filename)
42      # END TODO
43
44      # TODO: Use the blob to upload the file
45      blob.upload_from_string(
46          image_file.read(),
47          content_type=image_file.content_type)
48      # END TODO
49
50      # TODO: Make the object public
51      if public:
52          blob.make_public()
53      # END TODO
54
55      # TODO: Modify to return the blob's Public URL
56      return blob.public_url
57      # END TODO
```

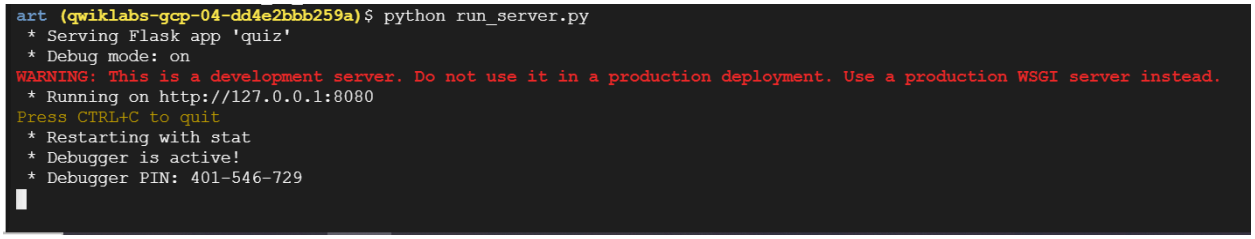
Write code to use the Cloud Storage functionality

question.py:

```
# TODO: Import the storage module
from quiz.gcp import storage, datastore
# END TODO

"""
uploads file into google cloud storage
- upload file
- return public_url
"""
def upload_file(image_file, public):
    if not image_file:
        return None

    # TODO: Use the storage client to Upload the file
    # The second argument is a boolean
    public_url = storage.upload_file(
       image_file,
       public
    )
    # END TODO

    # TODO: Return the public URL
    return public_url
    # END TODO
```

```
"""
uploads file into google cloud storage
- call method to upload file (public=true)
- call datastore helper method to save question
"""
def save_question(data, image_file):

    # TODO: If there is an image file, then upload it
    # And assign the result to a new Datastore
    # property imageUrl
    # If there isn't, assign an empty string

    if image_file:
        data['imageUrl'] = str(
                upload_file(image_file, True))
    else:
        data['imageUrl'] = u''

    # END TODO

    data['correctAnswer'] = int(data['correctAnswer'])
    datastore.save_question(data)
    return
```

Run the application and create a Cloud Storage object



Run the client application and test the Cloud Storage public URL

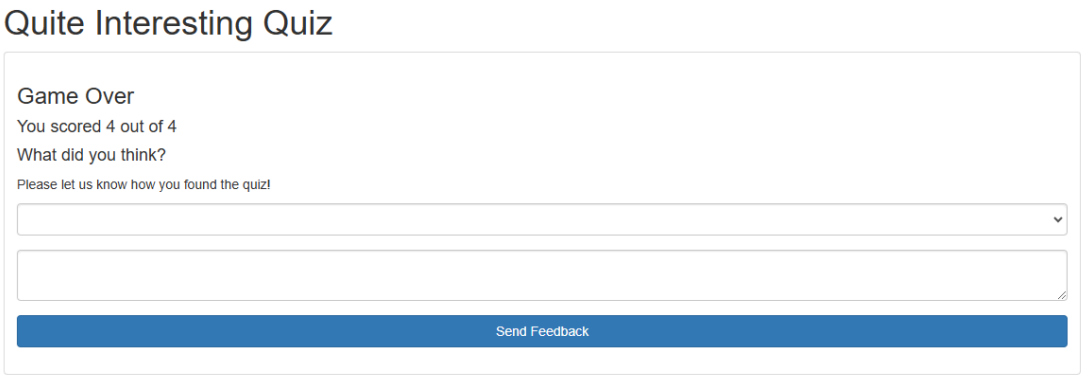
Cloud storage public link:
https://8080-cs-d6e38ec2-80e1-4e09-9d05-5c62a14245cb.ql-asia-east1-acey.cloudshell.dev/api/quizzes/gcp

Json file contents:


```
{
 "questions": [
  {
   "answer1": "Cloud Storage",
   "answer2": "Datastore",
   "answer3": "Big Table",
   "answer4": "All of the above",
   "author": "Nigel",
   "id": 5636645067948772,
   "imageUrl": "",
   "quiz": "gcp",
   "title": "Which GCP product is an Object Store?"
  },
  {
   "answer1": "Amazon",
   "answer2": "Google",
   "answer3": "IBM",
   "answer4": "Microsoft",
   "author": "Nigel",
   "id": 5634161670881280,
   "imageUrl": "",
   "quiz": "gcp",
   "title": "Which company runs GCP?"
  },
```

```
  {
    "answer1": "App Engine",
    "answer2": "Cloud Storage",
    "answer3": "Compute Engine",
    "answer4": "Container Engine",
    "author": "Saleha Zainab Fatima",
    "id": 5636645067948772,
    "imageUrl":
"https://storage.googleapis.com/qwiklabs-gcp-02-b8926891f9c3-media/Google_Cloud_Storage_logo.
png",
    "quiz": "gcp",
    "title": "Which product does this logo relate to?"
  },
  {
    "answer1": "Compute Engine",
    "answer2": "Datastore",
    "answer3": "Spanner",
    "answer4": "BigQuery",
    "author": "Nigel",
    "id": 5644004762845184,
    "imageUrl": "",
    "quiz": "gcp",
    "title": "Which GCP product is NoSQL?"
  } ] }
```


Return to the application home page and click the Take Test link



The end

App Dev: Storing Image and Video Files in Cloud Storage - Python.