**Task 1:**

Create a **Micro Service 1** with a face\_recognition API using open source face\_recognition module: <https://pypi.org/project/face-recognition/> and with the following specifications :

1. Accepts two images as input.
2. Generate encodings of each image using : face\_recognition.face\_encodings
3. Compare two face images using : face\_recognition.compare\_faces
4. Output the distance generated in the above step in a JSON.

**Note**: Preferable Language/Framework: python/Flask/Django

Or, you can use any language/framework of your choice based on face\_recognition module availability.

**Task 2:**

Create a **Micro Service 2** using any backend framework with which you are comfortable using and with the following specifications.

1. Create an API which accepts 2 face images as input
2. Make a call to the API created in **MicroService 1** to get the distance between above 2 face images
3. Output the distance received from **MicroService 1**.
4. Add all the images to any Cloud storage. (You can use free tier option to create a bucket)
5. Store all the transactions in a database(cloud or local) along with the timestamps of the requests and other requests/response details.

**Note:** Preferable Language - Golang

Be aware that you will be a step closer to the opportunity if you can use Cloud Services.

Cloud Preference Order - Azure Cloud, GCP, AWS

**Task 3:**

1. Create a Dockerfile for Micro Service 1
2. Create a Dockerfile for Micro Service 2
3. Create a docker-compose.yml file
4. When the above docker-compose is run, we should be able to send 2 images to Micro Service 2 and get the response.

**Task 5:**

1. Create a detailed documentation of the APIs created.