# Python Flask and Google Firebase

---

## Task:

**Build a certificate management system with Python Flask. User should be able to personalize pre-designed certificate with their name. All These certificates must be securely stored in Firebase's Cloud Storage, while the application itself runs on a Flask backend server. For data management, use Firestore for text data and Cloud Storage for certificate images.**

## Steps to keep in mind while developing the application:

### STEP 1:

Create a GET request route in Flask that accepts a candidate's name and email from query parameters. Ensure that the candidate's name does not exceed 20 characters.

### STEP 2:

Using the provided sample certificate, add the candidate's name to the certificate. The Y-axis position, font size, and style are specified below. Center the candidate's name along the X-axis. You can use the OpenCV library for this task. A small code snippet for this will be provided in the attached file.

### STEP 3:

After adding the name to the certificate, upload the certificate to Firebase Bucket Storage. Additionally, store the candidate's name, email, and the certificate's storage link in the Firestore database.

### STEP 4:

Create a GET request route in Flask at /certificate/<certificate_id>. The certificate_id is the unique identifier for the certificate. When this route is accessed in a browser, it should open a page that extracts the certificate_id from the URL and uses the Firebase Bucket Storage public

URL to display the PDF. Use an <iframe> tag to embed the PDF by passing the Firebase public bucket link to the src attribute of the iframe.

```
<iframe src="<certificate_id>.pdf"></iframe>
```

## Important Instructions:

- Use of Python and Firebase is mandatory
- Please find the attached sample certificate that you need to use.

## Marking Scheme:

30m - Following good practices, Writing well-structured code with comments

30m - Building using the code snippets provided

40m - Fully functional program as mentioned in the problem statement

## Submission:

Final build is supposed to be pushed on GitHub.
A review will be done 1:1 and feedback will be given at the submission date.

**Final Submission date:** 20th June 2024