

**NATIONAL UNIVERSITY OF COMPUTER AND  
EMERGING SCIENCES  
PROGRAM: SOFTWARE ENGINEERING**



***SOFTWARE CONSTRUCTION AND DEVELOPMENT  
LAB***

**LAB TASK 07**

**SUBMITTED BY:**

**Name: Ahmed Ali**

**Roll No: 22P-9318**

**Section: BS(SE)-5B**

**INSTRUCTOR NAME: Sir Saood Sarwar**

**A DEPARTMENT OF COMPUTER SCIENCE**

## **Service Account JSON: Do not submit the actual JSON file. Ensure you understand how to use it securely.**

### **ANSWER:**

To ensure I use the Firebase Service Account JSON securely Some practices are:

#### **1) Do Not Hardcode Sensitive Information:**

- Never hardcode the Service Account JSON credentials directly in your code. Instead, store it in a secure location, such as on the server or in environment variables.

#### **2) Secure File Permissions:**

- Ensure that the file containing the JSON credentials is not accessible by unauthorized users. Set restrictive file permissions on the file, so only your application can read it.

#### **3) Use Secret Management Services:**

- If possible, use secret management services such as **AWS Secrets Manager**, **Google Secret Manager**, or **PythonAnywhere environment variables** to store and retrieve your sensitive credentials dynamically at runtime.

#### **4) Rotate Credentials Regularly:**

- Periodically rotate your service account credentials to reduce the risk of misuse if they are ever exposed.

#### **5) Restrict Service Account Permissions:**

- Assign only the necessary permissions to your Firebase service account. Use the principle of **least privilege** to minimize the potential impact if the credentials are compromised.
-

**Deployment Steps: Provide a brief document outlining the steps you took to deploy the Flask app on PythonAnywhere.**

**Step 1:** Create an account on PythonAnywhere and log in.

**Step 2:** Go to the "**Web**" section and create a new web app.

**Step 3:** Select **Flask** as your web framework and choose Python version (matching the version you're developing with).

**Step 4:** In the web app configuration, under the "Code" section, set the path to your Flask app's app.py file (e.g., /home/ahmed121176/mysite/app.py).

**Step 5:** Install required dependencies using the "**Bash console**" in PythonAnywhere:  
*pip install firebase-admin xmltodict*

**Step 6:** Place your Firebase Service Account JSON file (p-9318-ahmed-bse-5b-project-firebase-adminsdk-f7jn6-13df3b6df6.json) in the correct directory (e.g., /home/ahmed121176/mysite/).

**Step 7:** Reload the web app from the "**Web**" section in PythonAnywhere to apply the changes.

**Step 8:** Test it using the provided code in instructions in Jupyter Notebook.

---

## Successful response from the Jupyter notebook.


```
[53]: import requests
# XML data to be sent
xml_data = """<?xml version="1.0" encoding="UTF-8"?>
<user>
<name>John Doe</name>
<email>john.doe@example.com</email>
</user>"""
# Your Flask server URL on PythonAnywhere
url = 'https://ahmed121176.pythonanywhere.com/receive-data'
# Send the request to the Flask server
response = requests.post(url, data=xml_data, headers={'Content-Type': 'application/xml'})
# Check if the response is successful
if response.status_code in [200, 201]:
    try:
        # Print the response from the server
        print(response.json())
    except ValueError as e:
        print("Error decoding JSON:", e)
        print("Response content:", response.text)
else:
    print("Error:", response.status_code)
    print("Response content:", response.text)

{'email': 'john.doe@example.com', 'id': 'UL7jZWNOGwsmQysN1KMv', 'name': 'John Doe'}
```

## Entry in Firestore Database under the users collection.

ase 22p-9318-Ahmed-BSE-5B-Project Cloud Firestore

Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing. [Configure App Check](#)

Overview 

th Gemini

s

e Database


ies

ucts

opment tools

ionth) Upgrade

Panel view

users > zs5GCIIuFV3x6.. 

(default)	users	zs5GCIIuFV3x6ryLUpB
<a href="#">+ Start collection</a>	<a href="#">+ Add document</a>	<a href="#">+ Start collection</a>
users >	<div>V1AFYNqhEYDL4aUUL0nK</div> <div>VQgBzCCwt6MbtMM7X5H1</div> <div>W1IT4dreInA1RJSpNjKE</div> <div>WqFyWl3wtM13GhtrqIBi</div> <div>Zm8MtByN2Ddp4nGNVn1M</div> <div>aSLaZAF4weA7WXAdufF5</div> <div>ht0TAFE22pXp50lGkDfe</div> <div>iEDyvEjyM0wjklNrIzNS</div> <div>mPKGGEtWz4k5o5RRIwBE</div> <div>v7Su78Jdg9hNk19HSZ6j</div> <div>vJKSW2H6h0V6cSk1qK20</div> <div>zMY8rmSceSzYizJf19sH</div> <div>zdCKEaeHmXZAtm7UuZkc</div> <div>zs5GCIIuFV3x6ryLUpB &gt;</div>	<a href="#">+ Add field</a> <div>email: "john.doe@example.com"</div> <div>name: "John Doe"</div>