Day 19 (28/08/23): (Threading, Auth token Generation, Encryption & Decryption)
Intermediate Level 2:

Multi-threading:
Multi-tasking in computer.

What is thread?
A program or processes which is a small unit is called a thread and it can run on our own schedule given on code compilations. Thread is used to run the defined code faster than def()

Code:

```
import threading
def sample1(num):
    print(f"h1 is {num}")
def sample2(name):
    print(f"my name is {name}")
t1=threading.Thread(target=sample1,args=(9,))
t2=threading.Thread(target=sample2,args=("Kumar",))
t1.start()
t2.start()
```

Output:
h1 is 9
my name is Kumar
Authentication Code Generation:

(Used to generate random Auth token code in Web URL. Use uuid4 module for uuid library.)

Code:

```
print("Authentication Token Generation:")
from uuid import uuid4 #uuid--> universal unique identifier
rand_token=uuid4()
print(f"Token->{rand_token}")
```

Output:
Authentication Token Generation:
Token->41fa65f9-f77a-4904-bb94-a5902a491fad

2) Encryption & Decryption:
Used to hide and unhide the password entered in the database.

Code:

```
print("Encryption: ")
from cryptography.fernet import Fernet
password="1234"
key=Fernet.generate_key()
fernet=Fernet(key)
enc_password=fernet.encrypt(password.encode())
print(enc_password)
```

a) Encryption: (pip install cyptography on terminal)
To hide the password by displaying random letters and numbers instead of password.
Use cryptograpgy.fernet library and import Fernet module to encode and decode the password.

Code:

```
print("Encryption: ")
from cryptography.fernet import Fernet
password="1234"
key=Fernet.generate_key()
fernet=Fernet(key)
enc_password=fernet.encrypt(password.encode())
print(enc_password)
```

Output:
b'gAAAAABk9_xq428PWVeCzNwm4f00_sC1kucpDt1gnjW-agIMFug8rWwB0fqAz-R-HEUH2AxBArk0GL3A771clS5AVQNEmCx11g=='

b) Decryption:
To unhide the password by changing the description and decode the encryption

Code:

```
dec_password=fernet.decrypt(enc_password).decode()
print(f"Original Password-> {dec_password}")
```

Output:
1234