TITLE : SMART TOKEN LABELLING AND TRACEBILITY SYSTEM FOR CYBERSECURITY TOKENS

College name : JNN institute of engineering

Department : CSE (Cybersecurity)

Team members :

1.Vishnupriya

2.Monishree

3.Sandhiya

4.Keshini

5.Prasanna

**ABSTRACT:**

In cybersecurity environments, tokens such as access keys, API tokens, or authentication keys are widely used for secure communication and resource access. However, managing and verifying these tokens manually is error-prone and lacks traceability.

This project introduces a smart token labeling and traceability system that uses QR codes and a local database to ensure tokens can be securely generated, validated, tracked, and labeled visually. Each token is uniquely generated and embedded within a QR image, which is labeled with its current state (VALID, EXPIRED, or UNAUTHORIZED). The token data is stored in an SQLite database and validated using custom Python scripts.

TOOLS AND TECHNOLOGIES USED

Tool/Technology Purpose

Python 3 Programming language for entire project

SQLite3 Lightweight local database to store token details

Pillow (PIL) To draw labels on QR images

qrcode module for generating QR code images

VS Code Code editor used to write and run Python scripts

Terminal / CMD To execute the project modules step-by-stepMODULE 1:

Database Creation (create\_db.py)

Creates the database tokendata.db

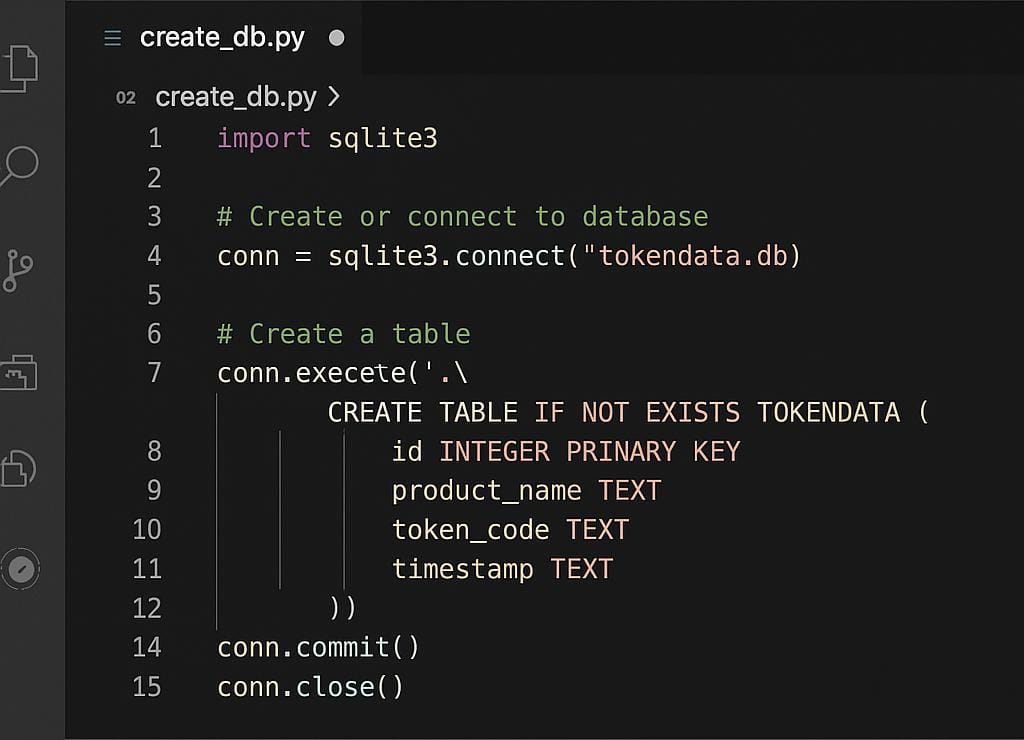
Defines the table tokens with fields:

1. Token\_id (Primary Key)

2.Issue\_date

3. Expiry\_date

4.Is\_authorized (1 = yes, 0 = no)

MODULE 2:

QR Code Generation (generate\_qr.py)

Takes input from the user to generate a new token ID

Saves the QR image with the token ID encoded inside it

Also inserts the full token details into the database



MODULE 3:

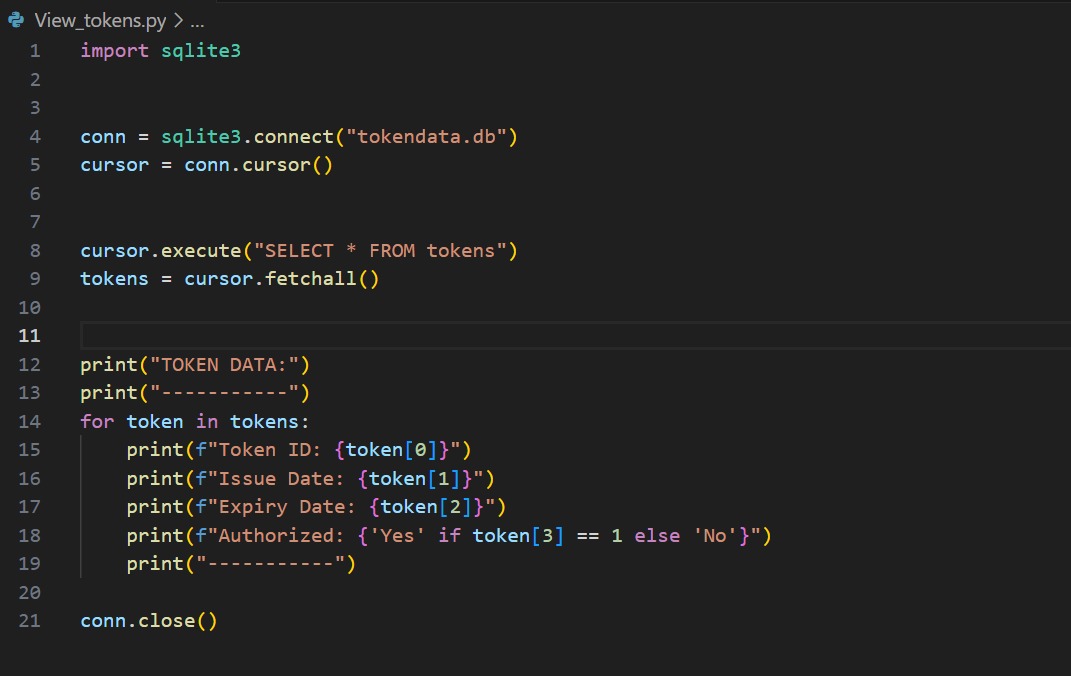
Labelling the token

We created the script label\_token.py that makes the QR code image with the status label like “valid “ in green or “invalid “ in red

MODULE 4 :

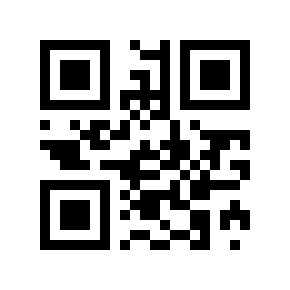
VIEWING THE TOKEN

As we developed a UI using view\_token.py to scan and check the token status visually for user interaction



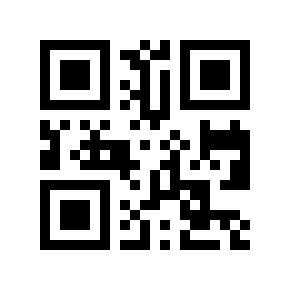
AUTHORIZED VALID TOKEN

When a token is marked as "VALID" in the database, our labeling script generates its corresponding QR code and overlays the text "VALID" directly onto the image. Using the Pillow library in Python, we draw green text at a fixed position to clearly indicate its authorization status. This makes validation visible.



AUTHORIZED EXPIRED TOKEN

If the token is found to be expired, the system reads the QR image and overlays the word "EXPIRED" in red text directly on the image using Pillow. This visual label provides instant clarity that the token is no longer authorized, based on status values fetched from the SQLite database.



WHY LABELING MATTERS:

Directly embedding token status into the image provides instant human-readable verification. Even without scanning, one can visually confirm if the token is usable. This is particularly helpful in manual checking processes, visitor passes, or physical checkpoints.

CONCLUSION:

In conclusion, this project presents a robust and efficient solution for managing cybersecurity tokens through smart labeling and traceability. By combining QR code technology with local database storage and custom Python validation scripts, the system enhances security, traceability, and usability. Its lightweight and offline-compatible design makes it ideal for practical deployment in various domains such as identity verification, license tracking, and secure API token management. This approach not only reduces human error but also provides a clear, visual status of each token, significantly improving the overall integrity and reliability of token-based systems.