

PROJECT DEVELOPMENT PHASE - SPRINT 4

Date	15 November 2022
Team ID	PNT2022TMID08702
Project Name	Smart Solutions for railways

SPRINT 4– Ticket Verification

Once the payment is successful, QR Code will be generated. The ticket collector can scan the generated QR Code from the user and verify it. By scanning the QR Code the user details will be displayed.

CODE:

```
import cv2
import numpy as np
import time
import pyzbar.pyzbar as pyzbar
from pyzbar.pyzbar import decode
from ibmcloudant.cloudant_v1 import CloudantV1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator

authenticator = BasicAuthenticator('apikey-v2-k0li2r9edsevyjyqtgzh0j653g1g81j24lfbe7hzz2h',
'71b524ba05a9592d11221ed267338f27')
service = CloudantV1(authenticator=authenticator)

service.set_service_url('https://apikey-v2-
k0li2r9edsevyjyqtgzh0j653g1g81j24lfbe7hzz2h:71b524ba05a9592d11221ed267338f27@af6aa2
e1-088e-4c99-960c-9faea5d31cf8-bluemix.cloudantnosqldb.appdomain.cloud')

cap= cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_PLAIN

while True:
    _, frame = cap.read()
    decodedObjects = pyzbar.decode(frame)
    for obj in decodedObjects:
        #print ("Data", obj.data)
        a=obj.data.decode('UTF-8')
        cv2.putText(frame, "Ticket", (50, 50), font, 2, (255, 0, 0), 3)
        #print (a)
```

try:

```
response = service.get_document(db='booking',doc_id = a).get_result()
print(response)
time.sleep(5)
```

except Exception as e:

```
print("NOT A VALID TICKER")
time.sleep(5)
```

```
cv2.imshow("Frame",frame)
```

```
if cv2.waitKey(1) & 0xFF ==ord('q'):
```

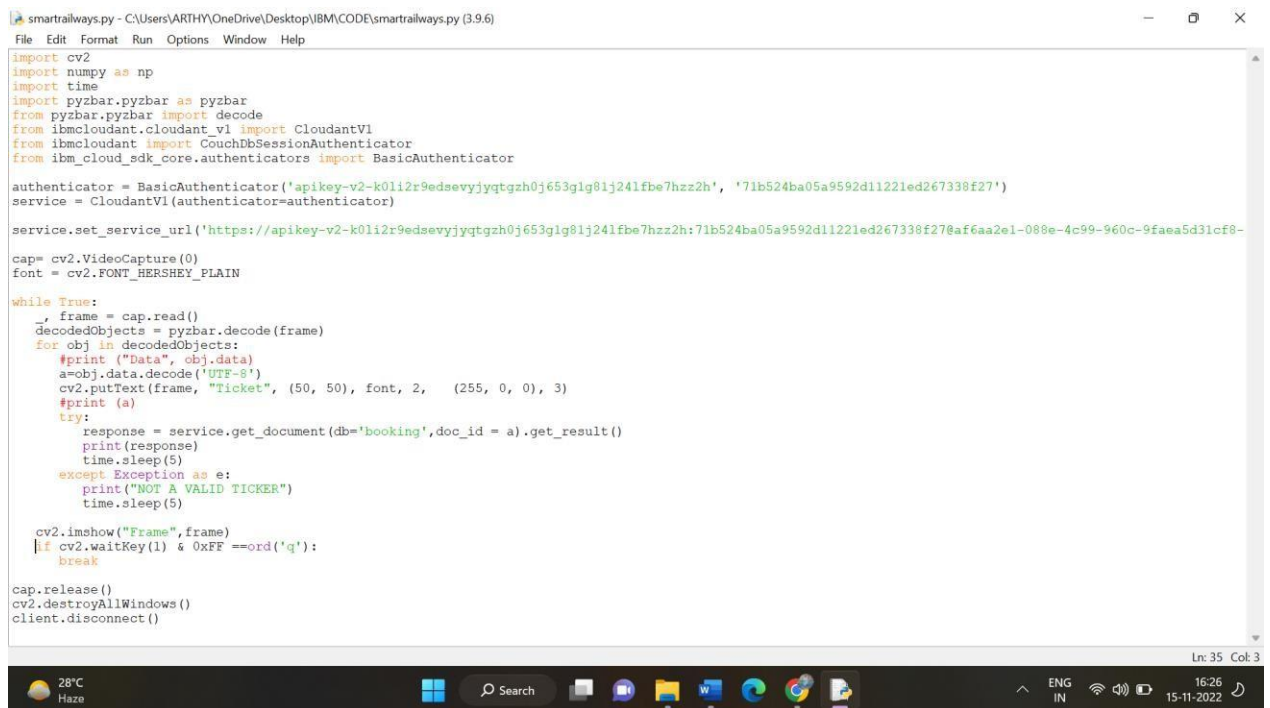
```
break
```

```
cap.release()
```

```
cv2.destroyAllWindows()
```

```
client.disconnect()
```

PYTHON CODE FOR QR CODE VERIFICATION:

A screenshot of a code editor window titled 'smartrailways.py - C:\Users\ARTHY\OneDrive\Desktop\IBM\CODE\smartrailways.py (3.9.6)'. The editor contains a Python script for QR code verification. The script imports necessary libraries like cv2, numpy, time, pyzbar, and IBM Cloudant SDK. It initializes a BasicAuthenticator and a CloudantV1 service with specific API keys and service URLs. The main logic is in a 'while True' loop that reads frames from a video capture, decodes QR codes using pyzbar, and checks if the decoded data is a valid 'Ticket'. If it is, it makes a GET request to the service to retrieve document details. If not, it prints an error message. The loop is broken when the user presses 'q'. Finally, it releases the video capture, destroys all windows, and disconnects the client. The Windows taskbar at the bottom shows the date as 15-11-2022 and time as 16:26.

```
import cv2
import numpy as np
import time
import pyzbar.pyzbar as pyzbar
from pyzbar.pyzbar import decode
from ibmcloudant.cloudant_v1 import CloudantV1
from ibmcloudant import CouchDbSessionAuthenticator
from ibm_cloud_sdk_core.authenticators import BasicAuthenticator

authenticator = BasicAuthenticator('apikey-v2-k01i2r9edsevyjyqtgzh0j653g1g81j241f8e7hzz2h', '71b524ba05a9592d11221ed267338f27')
service = CloudantV1(authenticator=authenticator)

service.set_service_url('https://apikey-v2-k01i2r9edsevyjyqtgzh0j653g1g81j241f8e7hzz2h:71b524ba05a9592d11221ed267338f27@af6aa2e1-088e-4c99-960c-9faea5d31cf8-

cap= cv2.VideoCapture(0)
font = cv2.FONT_HERSHEY_PLAIN

while True:
    frame = cap.read()
    decodedObjects = pyzbar.decode(frame)
    for obj in decodedObjects:
        #print ("Data", obj.data)
        a=obj.data.decode('UTF-8')
        cv2.putText(frame, "Ticket", (50, 50), font, 2, (255, 0, 0), 3)
        #print (a)
        try:
            response = service.get_document(db='booking',doc_id = a).get_result()
            print(response)
            time.sleep(5)
        except Exception as e:
            print("NOT A VALID TICKER")
            time.sleep(5)

    cv2.imshow("Frame", frame)
    if cv2.waitKey(1) & 0xFF ==ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
client.disconnect()
```

QR CODE VERIFICATION AND USER DETAILS STORED IN CLOUD

