

SPRINT-3

| | |
|--------------|--|
| Date | 15 NOVEMBER 2022 |
| Team ID | PNT2022TMID43868 |
| Project Name | Project – SMART SOLUTIONS FOR RAILWAYS |

PROCEDURE:

Step1: Develop a python script to scan the QR code

Step2: Connect the python code to IBM Cloudant using the credentials

Step3: Run the program

PYTHON SCRIPT TO SCAN QR 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-v2-125rwcp4ifi6zz2ly1cq0kakyjn98du2ysgc72h53lzi',  
'af693938842290ec2c254461754447b5') service =
```

```
CloudantV1(authenticator=authenticator)
```

```
service.set_service_url('https://apikey-v2-  
125rwcp4ifi6zz2ly1cq0kakyjn98du2ysgc72h53lzi:af693938842290ec2c254461754447b5@82d874994395-  
4f46-a190-6a186bee5051-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

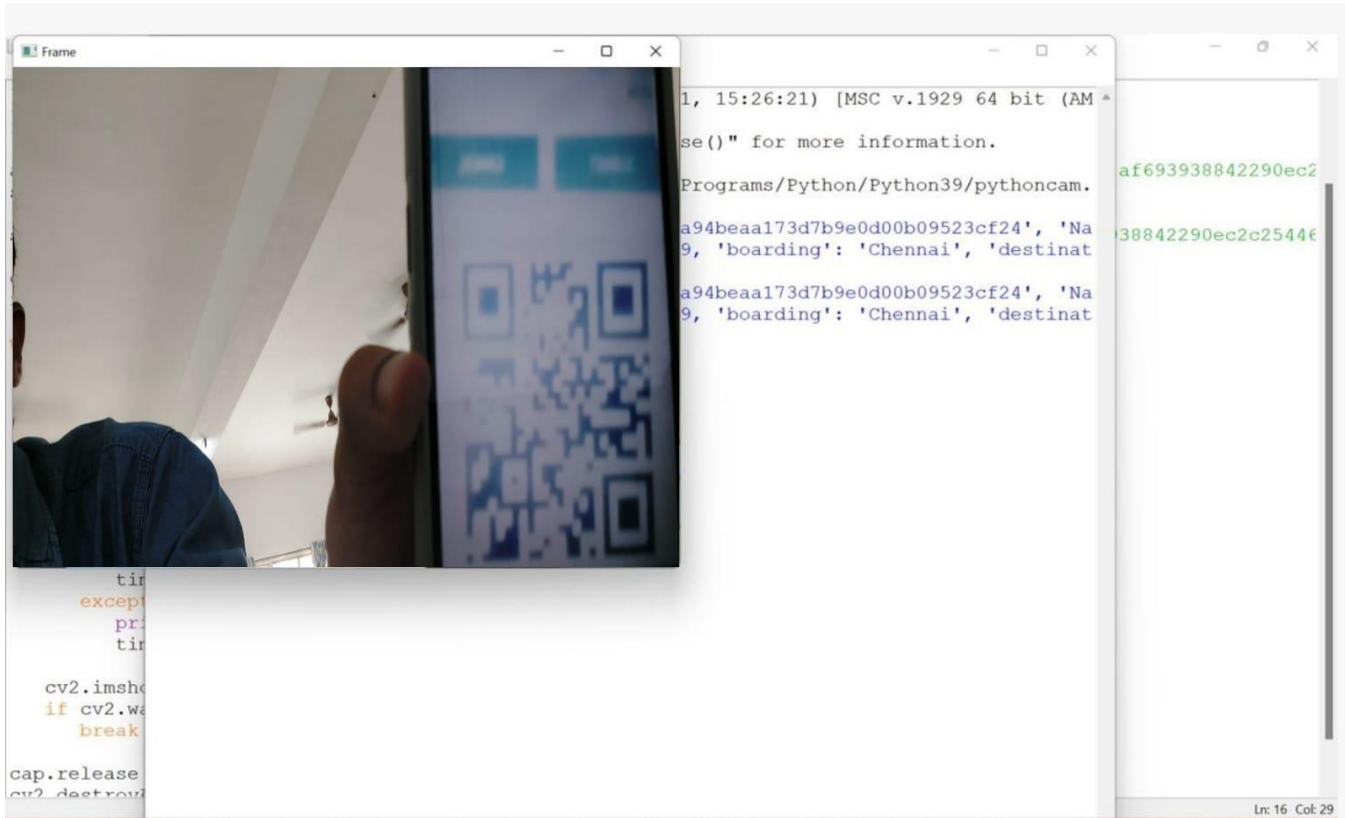
cv2.waitKey(1) & 0xFF ==ord('q'):

    break


cap.release() cv2.destroyAllWindows()

client.disconnect()
```

PYTHON CODE OUTPUT:



The screenshot displays a Python application for QR code detection. It consists of three main windows:

- Frame:** A video feed showing a hand holding a smartphone with a QR code on the screen.
- Terminal:** Shows the execution of the program. The output includes a timestamp (1, 15:26:21), a message about the camera (MSC v.1929 64 bit (AMD64)), and a list of detected QR codes with their data.
- Output Window:** Displays the detected QR codes and their data in a structured format.

The terminal output shows the following lines:

```
1, 15:26:21) [MSC v.1929 64 bit (AMD64)] for more information.  
Programs/Python/Python39/pythoncam.  
a94beaal73d7b9e0d00b09523cf24', 'Na  
9, 'boarding': 'Chennai', 'destinat  
a94beaal73d7b9e0d00b09523cf24', 'Na  
9, 'boarding': 'Chennai', 'destinat
```

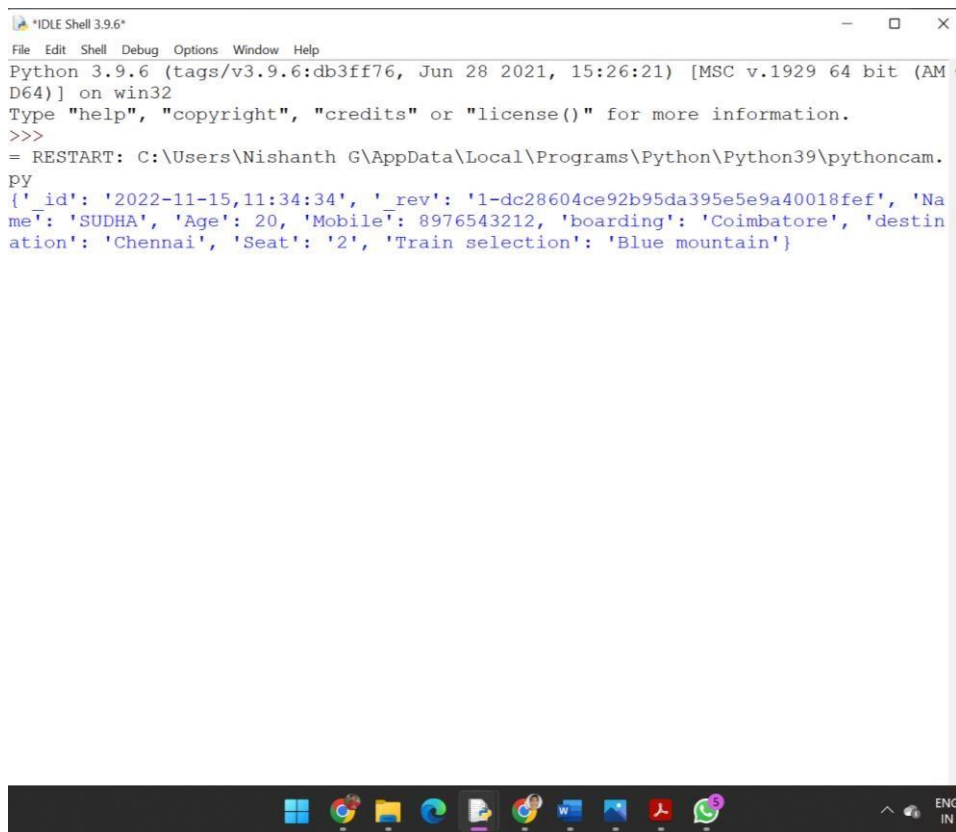
The output window shows the following lines:

```
af693938842290ec2  
38842290ec2c25446
```

The code in the Frame window is as follows:

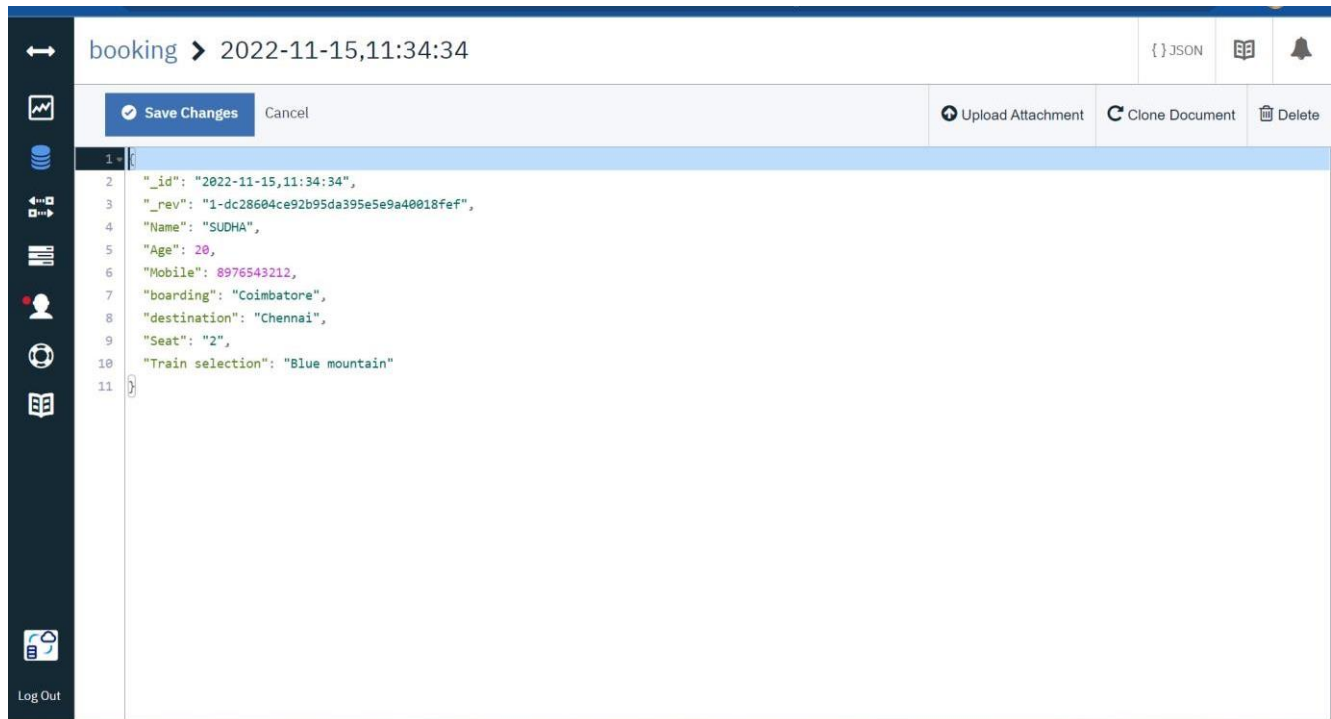
```
try:  
    except  
    print  
    tir  
  
    cv2.imshow  
    if cv2.wa  
        break  
  
cap.release  
cv2.destroy
```

QR CODE DETAILS:



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Nishanth G\AppData\Local\Programs\Python\Python39\pythoncam.py
{'_id': '2022-11-15,11:34:34', '_rev': '1-dc28604ce92b95da395e5e9a40018fef', 'Name': 'SUDHA', 'Age': 20, 'Mobile': 8976543212, 'boarding': 'Coimbatore', 'destination': 'Chennai', 'Seat': '2', 'Train selection': 'Blue mountain'}
```

DATA STORED IN CLOUDANT:



```
booking > 2022-11-15,11:34:34
[{"_id": "2022-11-15,11:34:34", "_rev": "1-dc28604ce92b95da395e5e9a40018fef", "Name": "SUDHA", "Age": 20, "Mobile": 8976543212, "boarding": "Coimbatore", "destination": "Chennai", "Seat": "2", "Train selection": "Blue mountain"}]
```

