TICKET VERIFICATION:

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

authenticator = BasicAuthenticator('apikey-v2-2vji5kegi5d7aaygtccqtt616y5u2yva2mpawe7ps7gp','15bc42554c2a4881a11889247e6ab66e')

service = CloudantV1(authenticator=authenticator)
print(service)
service.set_service_url('https://apikey-v2-2vji5kegi5d7aaygtccqtt616y5u2yva2mpawe7ps7gp:15bc42554c2a4881a11889247e6ab66e@58980990-cf73-48b6-bbee-db90262aae86-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:
        a=obj.data.decode('UTF-8')
        cv2.putText(frame, "Ticket", (50, 50), font, 2,(255, 0, 0),
3)
        try:
            response = service.get_document(
                  db='booking-table',
                  doc_id = a
                  ).get_result()
            print("Passenger Name\t Ticket Count \t Date \t Train
NO")
            print(response["pname"]+"\t
\t"+str(response["ticketcount"])+"\t"+response["date"]+"\t"+
str(response["trainNumber"]))
            time.sleep(5)
        except Exception as e :
             print(e)
             print("Not a Valid Ticket")
             time.sleep(5)
    cv2.imshow("Frame", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
client.disconnect()
```

GPS SCRIPT

```
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "kapgrv",
        "typeId": "ibm-iot",
        "deviceId": "10",
    },
    "auth": {
        "token": "12345678"
    }

}

train_detial = {12632: [[13.067439,80.237617],
              [8.735640, 77.708161],
              [8.738075, 77.708259],
              [8.741219, 77.708476],
              [8.743633, 77.708743],
              [8.746420, 77.708911],
              [8.748563, 77.709001],
              [8.750921, 77.709120],
              [8.753201, 77.709243],
              [8.755357, 77.709329],
              [8.757710, 77.709419],
              [8.760355, 77.709870],
              [8.762889, 77.710737],
              [8.764104, 77.711175],
              [8.765409, 77.711589],
              [8.766494, 77.711936],
              [8.767979, 77.712454],
              [8.769694, 77.713414],
              [8.770419, 77.713885],
              [8.771009, 77.714328],
              [8.773284, 77.716484],
              [8.774433, 77.717908],
              [8.776211, 77.720094],
              [8.777619, 77.721823],
              [8.779270, 77.723824],
              [8.781513, 77.726591],
              [8.783392, 77.728780],
              [8.783897, 77.729422],
              [8.784331, 77.729961]
              ], 12765: [12.972442,77.580643], 14375: [9.939093,78.121719]}


def myCommandCallback(cmd):
    print("" % cmd.data['command'])
    m=cmd.data['command']
```

```
def getDetials(trainNo):
    return train_detial[trainNo]

def getTemperature():
    return random.randint(0,50)

def pub(data):
    client.publishEvent(eventId="status", msgFormat="json",
data=myData, qos=0, onPublish=None)
    print("Published data successfully: %s",myData)

client = wiotp.sdk.device.DeviceClient(config=myConfig,
logHandlers=None)
client.connect()

trainNo = int(input("Enter TrainNo:"))
trainData=getDetials(trainNo)
lat=trainData[0]
lon=trainData[1]

while True:
    for val in trainData:
```

```
myData={'trainNo':trainNo,'lat':round(val[0],7),'long':round(val[1],7),'engine_temp':getTemperature()}
        pub(myData)
        time.sleep(3)
        client.commandCallback = myCommandCallback
client.disconnect()
```