**Source code:**

**Code for Managing Inventory:**

import os

import fileinput

import sys

import cv2

import numpy as np

def abort():

CHOICE = int(input('Enter 98 to continue or 99 to exit: ')) == 98

menuDisplay() if CHOICE else sys.exit()

def menuDisplay():

print("\n-->Welcome to Inventory management menu<--\n")

print("(1) add new item")

print("(2) remove item")

print("(3) print list")

print("(4) exit")

CHOICE = int(input("Enter user choice : "))

menuSelection(CHOICE)

def menuSelection(CHOICE):

if CHOICE == 1:

addInventory()

elif CHOICE == 2:

removeInventory()

elif CHOICE == 3:

printInventory()

elif CHOICE != 4:

print("Enter Valid Choice")

menuDisplay()

sys.exit()

def addInventory():

print("scan your qr code when the camera asks")

print("Adding Inventory")

print("================")

def video\_reader():

cam = cv2.VideoCapture(0)

detector = cv2.QRCodeDetector()

while True:

\_, img = cam.read()

data, bbox, \_ = detector.detectAndDecode(img)

if data:

print(data)

with open('Inventory.txt', 'a') as f:

f.write(data + '\*\n')

break

cv2.imshow("img", img)

if cv2.waitKey(1) == ord("Q"):

break

cam.release()

cv2.destroyAllWindows()

video\_reader()

abort()

def removeInventory():

print("Removing Inventory")

print("==================")

def video\_reader():

cam = cv2.VideoCapture(0)

detector = cv2.QRCodeDetector()

while True:

\_, img = cam.read()

data, bbox, \_ = detector.detectAndDecode(img)

if data:

print(data)

with open('Inventory.txt', 'r') as f:

file = f.read()

file = list(file.split("\*\n"))

file.remove(data)

with open('Inventory.txt', 'w') as f:

f.truncate()

f.write("\*\n".join(file))

break

cv2.imshow("img", img)

if cv2.waitKey(1) == ord("Q"):

break

cam.release()

cv2.destroyAllWindows()

video\_reader()

abort()

def printInventory():

with open('Inventory.txt', 'r') as f:

InventoryFile = f.read()

InventoryFile = list(InventoryFile.split("\*\n"))

print('Current Inventory')

print('-----------------')

for item in InventoryFile:

if item != '':

item\_description, item\_quantity = item.rstrip("\n").split("\n")

print('Item: ', item\_description)

print('Quantity: ', item\_quantity.rstrip("\*"))

print('----------')

abort()

menuDisplay()

Code for uploading data in IBM Cloud:

import datetime

import ibm\_boto3

from ibm\_botocore.client import Config, ClientError

import sys

import time

from cloudant.client import Cloudant

from cloudant.error import CloudantException

from cloudant.result import Result, ResultByKey

COS\_ENDPOINT = "https://s3.jp-tok.cloud-object-storage.appdomain.cloud"

#s3.us-south.cloud-object-storage.appdomain.cloud" # Current list avaiable at https://control.cloud-object-storage.cloud.ibm.com/v2/endpoints

COS\_API\_KEY\_ID = "8MKJ8cj2FDFFxL5L9f4WE1HTq07nv-YRjJUT\_g7eFp9A" # eg "W00YiRnLW4a3fTjMB-oiB-2ySfTrFBIQQWanc--P3byk"

COS\_AUTH\_ENDPOINT = "https://iam.cloud.ibm.com/identity/token"

COS\_RESOURCE\_CRN = "crn:v1:bluemix:public:cloud-object-storage:global:a/aeda64267f3b41b2878d07b5ba0aff74:1ea4ecb2-490a-44f7-b328-82be1e8428c8::"

# Create resource

cos = ibm\_boto3.resource("s3",

ibm\_api\_key\_id=COS\_API\_KEY\_ID,

ibm\_service\_instance\_id=COS\_RESOURCE\_CRN,

ibm\_auth\_endpoint=COS\_AUTH\_ENDPOINT,

config=Config(signature\_version="oauth"),

endpoint\_url=COS\_ENDPOINT

)

def multi\_part\_upload(bucket\_name, item\_name, file\_path):

try:

print("Starting file transfer for {0} to bucket: {1}\n".format(item\_name, bucket\_name))

# set 5 MB chunks

part\_size = 1024 \* 1024 \* 5

# set threadhold to 15 MB

file\_threshold = 1024 \* 1024 \* 15

# set the transfer threshold and chunk size

transfer\_config = ibm\_boto3.s3.transfer.TransferConfig(

multipart\_threshold=file\_threshold,

multipart\_chunksize=part\_size

)

# the upload\_fileobj method will automatically execute a multi-part upload

# in 5 MB chunks for all files over 15 MB

with open(file\_path, "rb") as file\_data:

cos.Object(bucket\_name, item\_name).upload\_fileobj(

Fileobj=file\_data,

Config=transfer\_config

)

print("Transfer for {0} Complete!\n".format(item\_name))

except ClientError as be:

print("CLIENT ERROR: {0}\n".format(be))

except Exception as e:

print("Unable to complete multi-part upload: {0}".format(e))

multi\_part\_upload("malavika1234","Inventory.txt","C:/Users/vosha/Desktop/inventory.txt")