

Not:Dosya adının altındaki kodlar o dosyaya aittir. Her dosya kodunun başlangıcı ve bitişi yıldızlar ile belirtilmiştir.

lib_face_reg.py

```
import pyqrcode
from pyzbar.pyzbar import decode
from PIL import Image
import png
import tkinter as tk
import sqlite3
import random
import time
db = sqlite3.connect('Library.db')
cr = db.cursor()
def crqrcode():
    bookname = kitapadientry.get()
    bookwriter = kitapyazarentry.get()
    booknum = kitapnumentry.get()
    bookedition = kitapbaskientry.get()
    qr = pyqrcode.create(bookname+"/"+bookwriter+"/"+booknum+"/"+bookedition)
    qr.png("{ }.png".format(bookname),scale=8)
    cr.execute("INSERT INTO Library VALUES(?,?,?,?,?,?,?)",(0,0,bookname,bookwriter,booknum,bookedition,1))
    db.commit()
    label = tk.Label(win,text="QRcode oluşturuldu ve veriler veri tabanına keydedildi.")
    label.place(x=100,y=300)
    breakpoint
    win = tk.Tk()
    win.geometry('500x500')
    win.title("QRcode Generator")
    kitapadi = tk.Label(win,text="Kitap Adı: ",font="Times 14 bold")
    kitapadi.place(x=100,y=100)
    kitapyazari = tk.Label(win,text="Kitap Yazarı: ",font="Times 14 bold")
    kitapyazari.place(x=100,y=150)
    kitapnum = tk.Label(win,text="Kitap Numarası: ",font="Times 14 bold")
    kitapnum.place(x=100,y=200)
    kitapbaski = tk.Label(win,text="Kitap Baskı Numarası: ",font="Times 14 bold")
    kitapbaski.place(x=100,y=250)
    kitapadientry = tk.Entry(win)
    kitapadientry.place(x=200,y=98)
    kitapyazarentry = tk.Entry(win)
    kitapyazarentry.place(x=225,y=150)
    kitapnumentry = tk.Entry(win)
    kitapnumentry.place(x=250,y=198)
    kitapbaskientry = tk.Entry(win)
    kitapbaskientry.place(x=300,y=248)
    qrcodebuton = tk.Button(win,text="QR oluştur",command=crqrcode)
    qrcodebuton.place(x=100,y=350)
    win.mainloop()
```

lib_qrcode_generator.py

```
import pyqrcode
from pyzbar.pyzbar import decode
from PIL import Image
import png
import tkinter as tk
import sqlite3
import random
import time
db = sqlite3.connect('Library.db')
cr = db.cursor()
def crqrcode():
    bookname = kitapadientry.get()
    bookwriter = kitapyazarentry.get()
    booknum = kitapnumentry.get()
    bookedition = kitapbaskientry.get()
    qr = pyqrcode.create(bookname+"/"+bookwriter+"/"+booknum+"/"+bookedition)
    qr.png("{} .png".format(bookname),scale=8)
    cr.execute("INSERT INTO Library VALUES(?,?,?,?,?,?)",(0,0,bookname,bookwriter,booknum,bookedition,1))
    db.commit()
    label = tk.Label(win,text="QRcode oluşturuldu ve veriler veri tabanına kaydedildi.")
    label.place(x=100,y=300)
    breakpoint
    win = tk.Tk()
    win.geometry('500x500')
    win.title("QRcode Generator")
    kitapadi = tk.Label(win,text="Kitap Adı: ",font="Times 14 bold")
    kitapadi.place(x=100,y=100)
    kitapyazari = tk.Label(win,text="Kitap Yazarı: ",font="Times 14 bold")
    kitapyazari.place(x=100,y=150)
    kitapnum = tk.Label(win,text="Kitap Numarası: ",font="Times 14 bold")
    kitapnum.place(x=100,y=200)
    kitapbaski = tk.Label(win,text="Kitap Baskı Numarası: ",font="Times 14 bold")
    kitapbaski.place(x=100,y=250)
    kitapadientry = tk.Entry(win)
    kitapadientry.place(x=200,y=98)
    kitapyazarentry = tk.Entry(win)
    kitapyazarentry.place(x=225,y=150)
    kitapnumentry = tk.Entry(win)
    kitapnumentry.place(x=250,y=198)
    kitapbaskientry = tk.Entry(win)
    kitapbaskientry.place(x=300,y=248)
    qrcodebuton = tk.Button(win,text="QR oluştur",command=crqrcode)
    qrcodebuton.place(x=100,y=350)
    win.mainloop()
```

lib_qrcode_get_reader.py

```
import pyqrcode
from pyzbar.pyzbar import decode
```

```

from PIL import Image
import png
import cv2
import os
import time
import sqlite3
db = sqlite3.connect('Library.db')
cr = db.cursor()
file = open(os.getcwd()+"/yoklama.txt","r",encoding="utf-8")
lines = file.readlines()
lines = lines[-1]
lines = lines.split("|")
alim_tarihi = lines[1]
alici_ismi = lines[0]
print(alici_ismi,alim_tarihi)
cap = cv2.VideoCapture(0)
cap.set(3,640)
cap.set(4,480)
camera = True
used_codes= []
while camera == True:
    success,frame = cap.read()
    for code in decode(frame):
        if code.data.decode("utf-8") not in used_codes:
            print(code.data.decode("utf-8"))
            data = code.data.decode("utf-8")
            data1 = data.split("/")
            data2 = data1[0]
            data3 = data1[3]
            #file.write(data)
            #file.write("\n")
            used_codes.append(code.data.decode("utf-8"))
            time.sleep(3)
            cr.execute("UPDATE Library set Alici_ismi = ? WHERE Kitap_adi = ? and kitap_baski_numarasi = ?",(alici_ismi,data2,data3,))
            cr.execute("UPDATE Library set Alim_Tarihi = ? WHERE Kitap_adi = ? and kitap_baski_numarasi = ?",(alim_tarihi,data2,data3,))
            cr.execute("UPDATE Library set Durum = ? WHERE Kitap_adi = ? and kitap_baski_numarasi = ?",(0,data2,data3,))
            db.commit()
        elif code.data.decode("utf-8") in used_codes:
            print("Bu kod daha önceden kullanılmıştır.")
            cv2.imshow("Code Scan",frame)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                exit()

```

lib_qrcode_give_reader.py

```

import pyqrcode
from pyzbar.pyzbar import decode
from PIL import Image

```

```

import png
import cv2
import os
import time
import sqlite3
check = 0
file = open(os.getcwd()+"/yoklama.txt","a",encoding="utf-8")
db = sqlite3.connect('Library.db')
cr = db.cursor()
cap = cv2.VideoCapture(0)
cap.set(3,640)
cap.set(4,480)
camera = True
used_codes= []
while camera == True:
    success,frame = cap.read()
    for code in decode(frame):
        if code.data.decode("utf-8") not in used_codes:
            print(code.data.decode("utf-8"))
            data = code.data.decode("utf-8")
            data1 = data.split("/")
            data2 = data1[0]
            data3 = data1[3]
            used_codes.append(code.data.decode("utf-8"))
            time.sleep(3)
            cr.execute("UPDATE Library set Alici_ismi = ? and Alim_Tarihi = ? WHERE Kitap_adi = ? and kitap_baski_
            _numarasi = ?",(0,0,data2,data3,))
            cr.execute("UPDATE Library set Alim_Tarihi = ? and Alici_ismi = ? WHERE Kitap_adi = ? and kitap_baski_
            _numarasi = ?",(0,0,data2,data3,))
            cr.execute("UPDATE Library set Alim_Tarihi = ? and Alici_ismi = ? and Durum = ? WHERE Kitap_adi = ?
            and kitap_baski_numarasi = ?",(0,0,1,data2,data3,))
            db.commit()
        elif code.data.decode("utf-8") in used_codes:
            print("Bu kod daha önceden kullanılmıştır.")
            cv2.imshow("Code Scan",frame)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                exit()

```

qr_olustur.bat

```

@echo
python C:\Users\user\Desktop\lib_codes\lib_qrcode_generator.py
timeout /t 5
echo "İşlem basariyla gerceklesti"
timeout /t 5

```

vermek.bat

```
@echo
python C:\Users\user\Desktop\lib_codes\lib_qrcode_give_reader.py
timeout /t 5
echo "İşlem basariyla gerceklesti"
```

almak.bat

```
@echo
python C:\Users\user\Desktop\lib_codes\lib_face_reg.py
timeout /t 5
python C:\Users\user\Desktop\lib_codes\lib_qrcode_get_reader.py
timeout /t 5
echo "İşlem basariyla gerceklesti"
```

install_lib.bat

```
@echo
pip install tk
pip install PyQRCode
pip install pyzbar
pip install Pillow
pip install pypng
pip install db-sqlite3
pip install random2
pip install times
pip install opencv-python
pip install face-recognition
pip install os-sys
pip install DateTime
```
