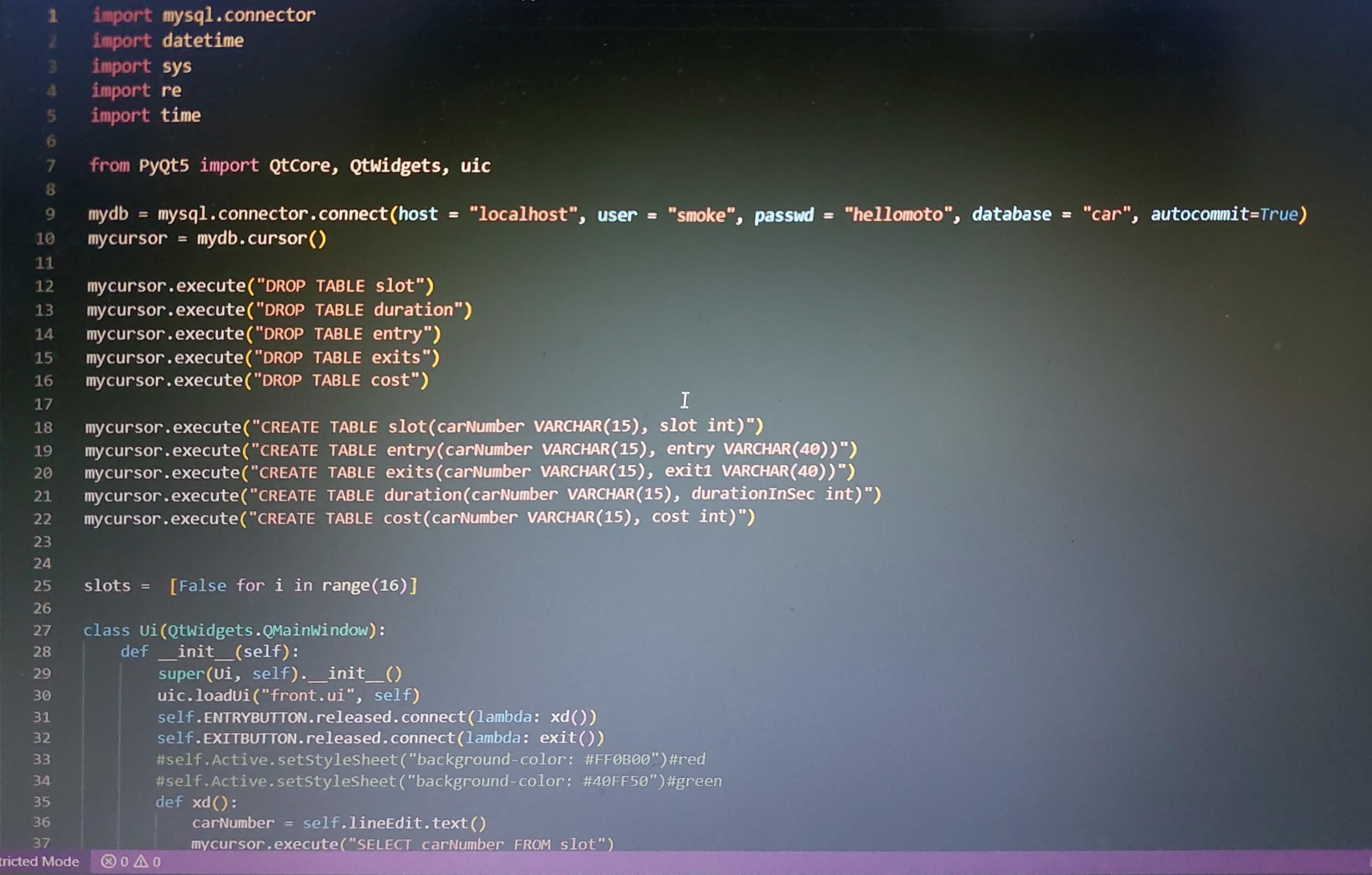
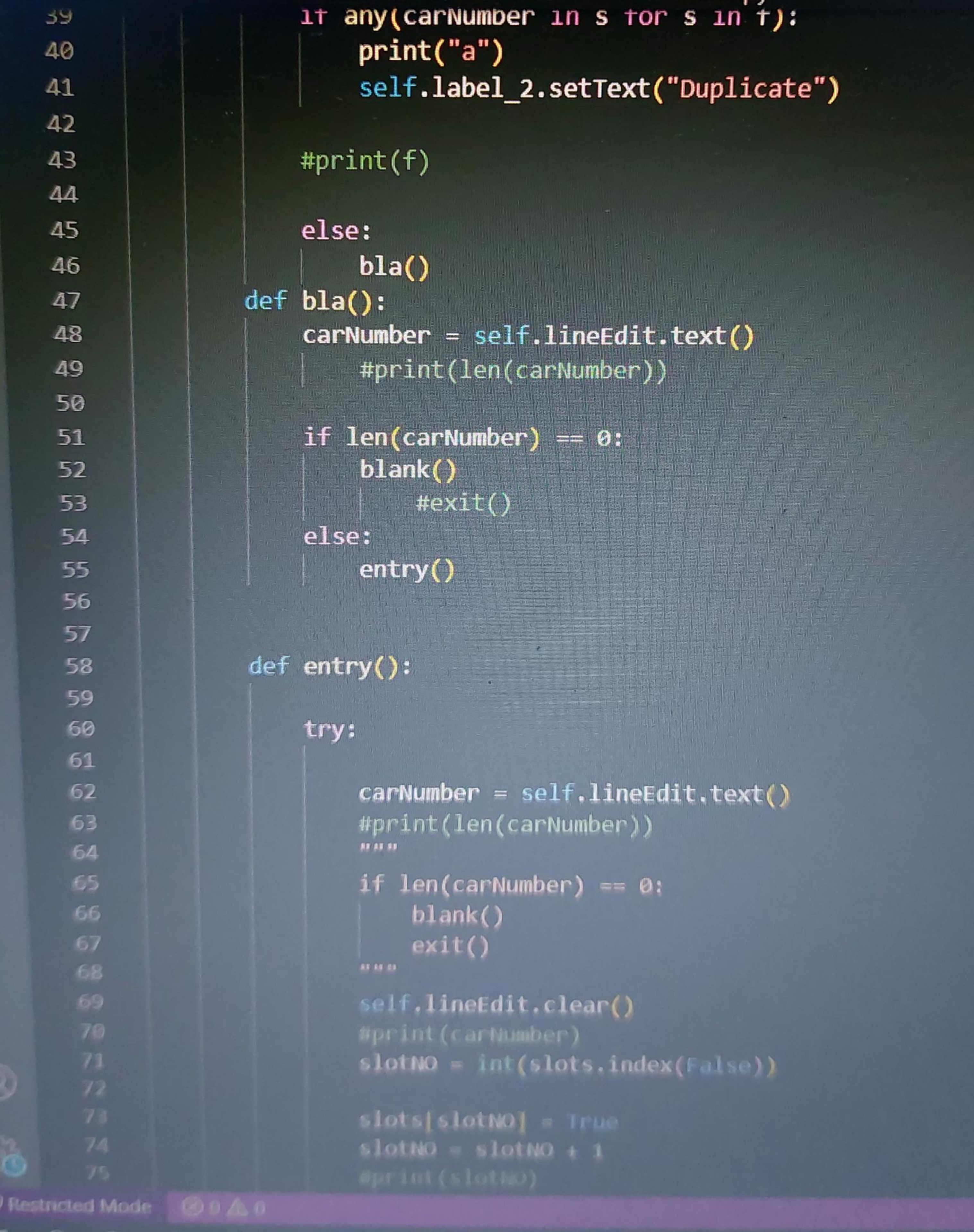
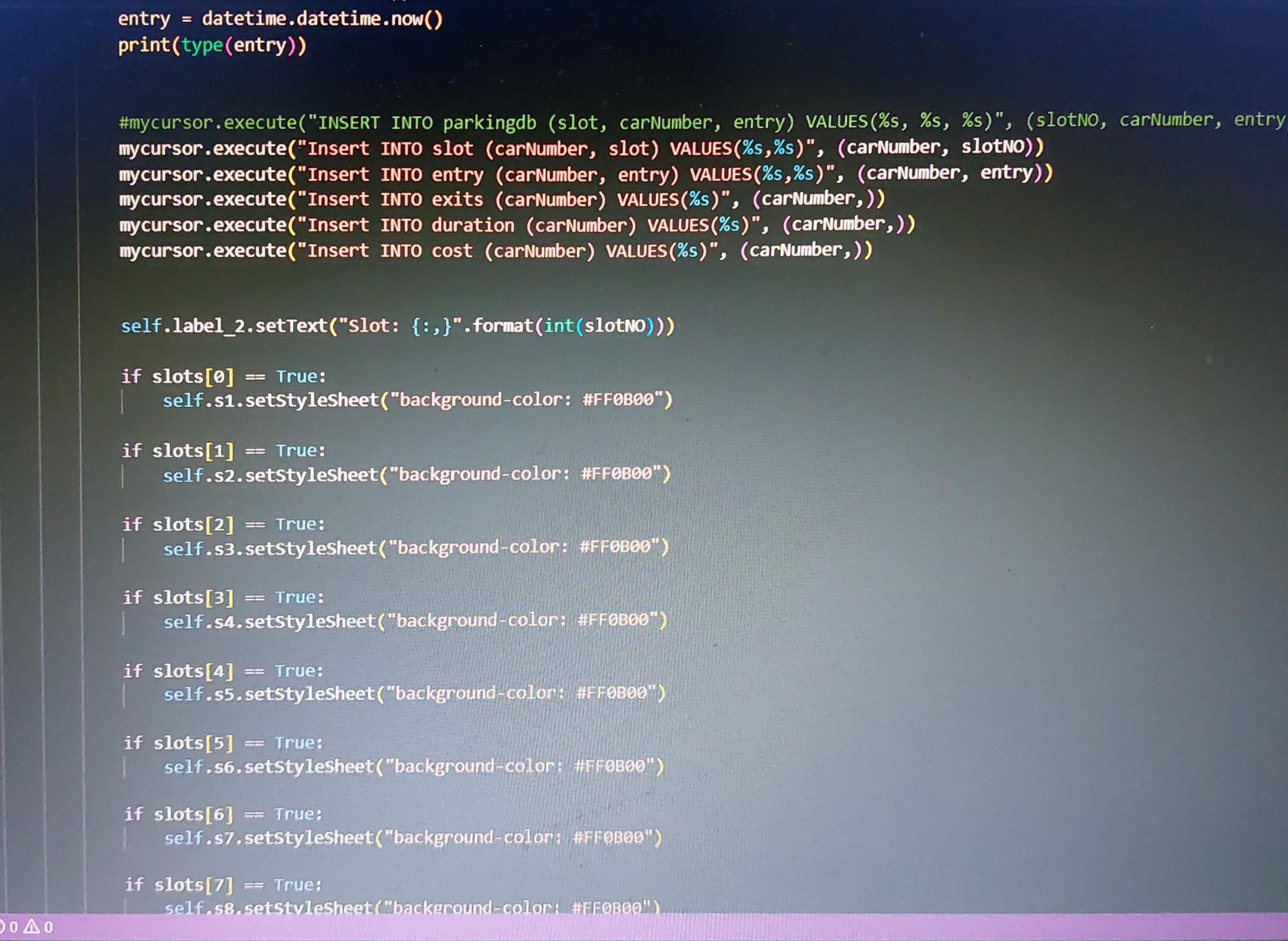


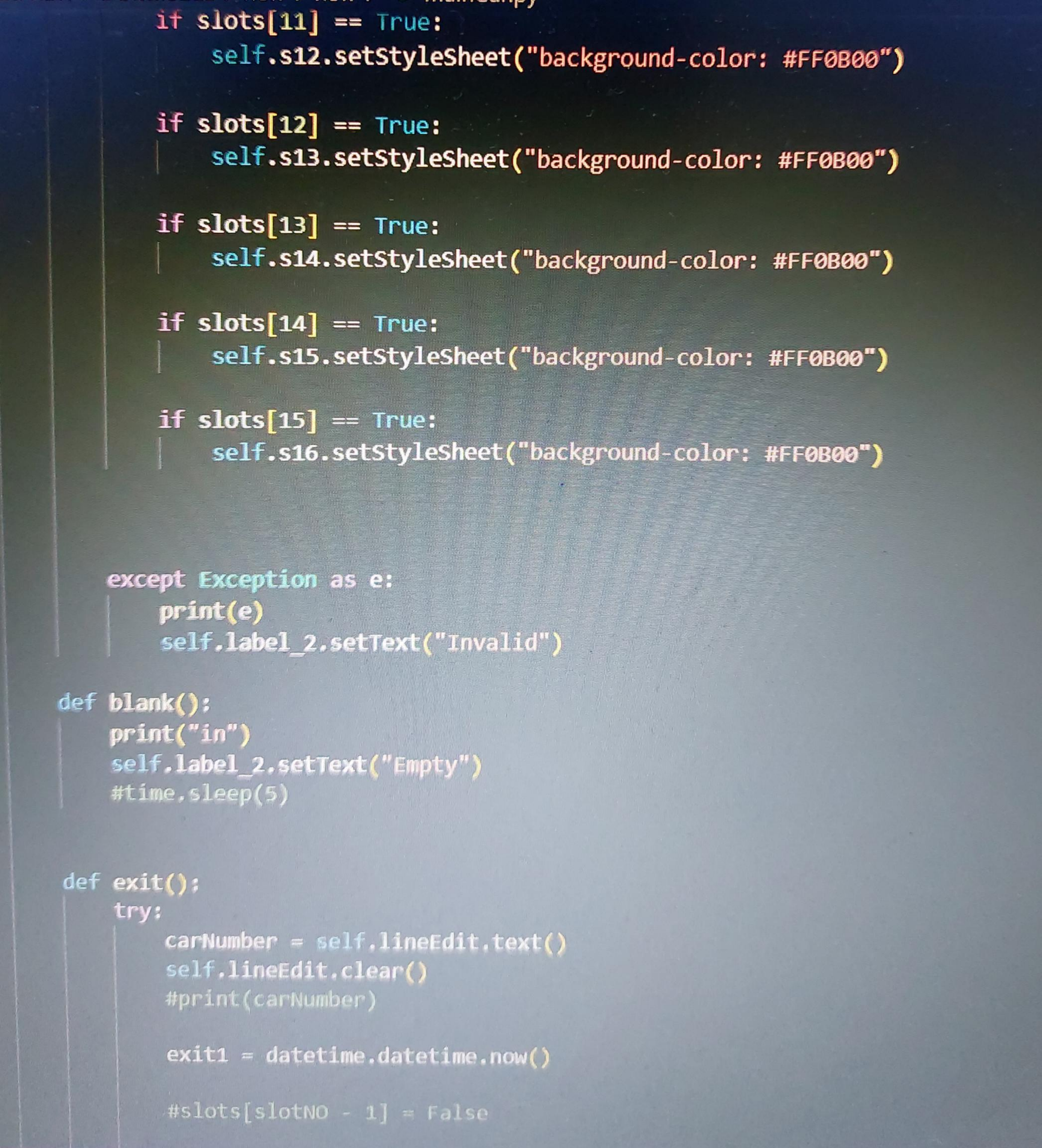
```python
import mysql.connector
import datetime
import sys
import re
import time

from PyQt5 import QtCore, QtWidgets, uic

mydb = mysql.connector.connect(host = "localhost", user = "smoke", passwd = "hellomoto", database = "car", autocommit=True)
mycursor = mydb.cursor()

mycursor.execute("DROP TABLE slot")
mycursor.execute("DROP TABLE duration")
mycursor.execute("DROP TABLE entry")
mycursor.execute("DROP TABLE exits")
mycursor.execute("DROP TABLE cost")

mycursor.execute("CREATE TABLE slot(carNumber VARCHAR(15), slot int)")
mycursor.execute("CREATE TABLE entry(carNumber VARCHAR(15), entry VARCHAR(40))")
mycursor.execute("CREATE TABLE exits(carNumber VARCHAR(15), exit1 VARCHAR(40))")
mycursor.execute("CREATE TABLE duration(carNumber VARCHAR(15), durationInSec int)")
mycursor.execute("CREATE TABLE cost(carNumber VARCHAR(15), cost int)")


slots =  [False for i in range(16)]

class Ui(QtWidgets.QMainWindow):
    def __init__(self):
        super(Ui, self).__init__()
        uic.loadUi("front.ui", self)
        self.ENTRYBUTTON.released.connect(lambda: xd())
        self.EXITBUTTON.released.connect(lambda: exit())
        #self.Active.setStyleSheet("background-color: #FF0B00")#red
        #self.Active.setStyleSheet("background-color: #40FF50")#green
        def xd():
            carNumber = self.lineEdit.text()
            mycursor.execute("SELECT carNumber FROM slot")
```



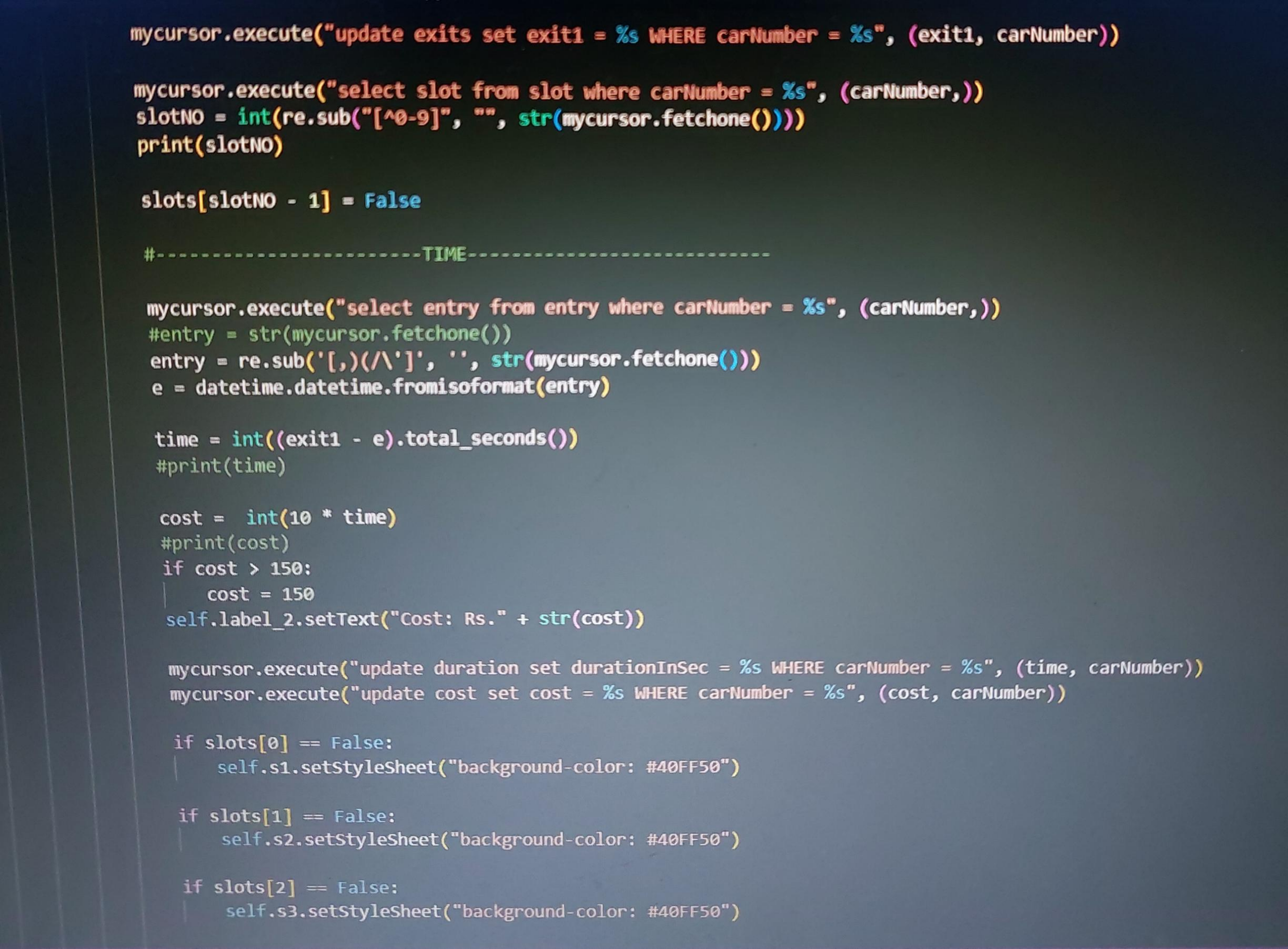
```
            if any(carNumber in s for s in f):
                print("a")
                self.label_2.setText("Duplicate")

            #print(f)

            else:
                bla()
        def bla():
            carNumber = self.lineEdit.text()
                #print(len(carNumber))

            if len(carNumber) == 0:
                blank()
                    #exit()
            else:
                entry()


        def entry():

            try:

                carNumber = self.lineEdit.text()
                #print(len(carNumber))
                """
                if len(carNumber) == 0:
                    blank()
                    exit()
                """
                self.lineEdit.clear()
                #print(carNumber)
                slotNO = int(slots.index(False))

                slots[slotNO] = True
                slotNO = slotNO + 1
                #print(slotNO)
```



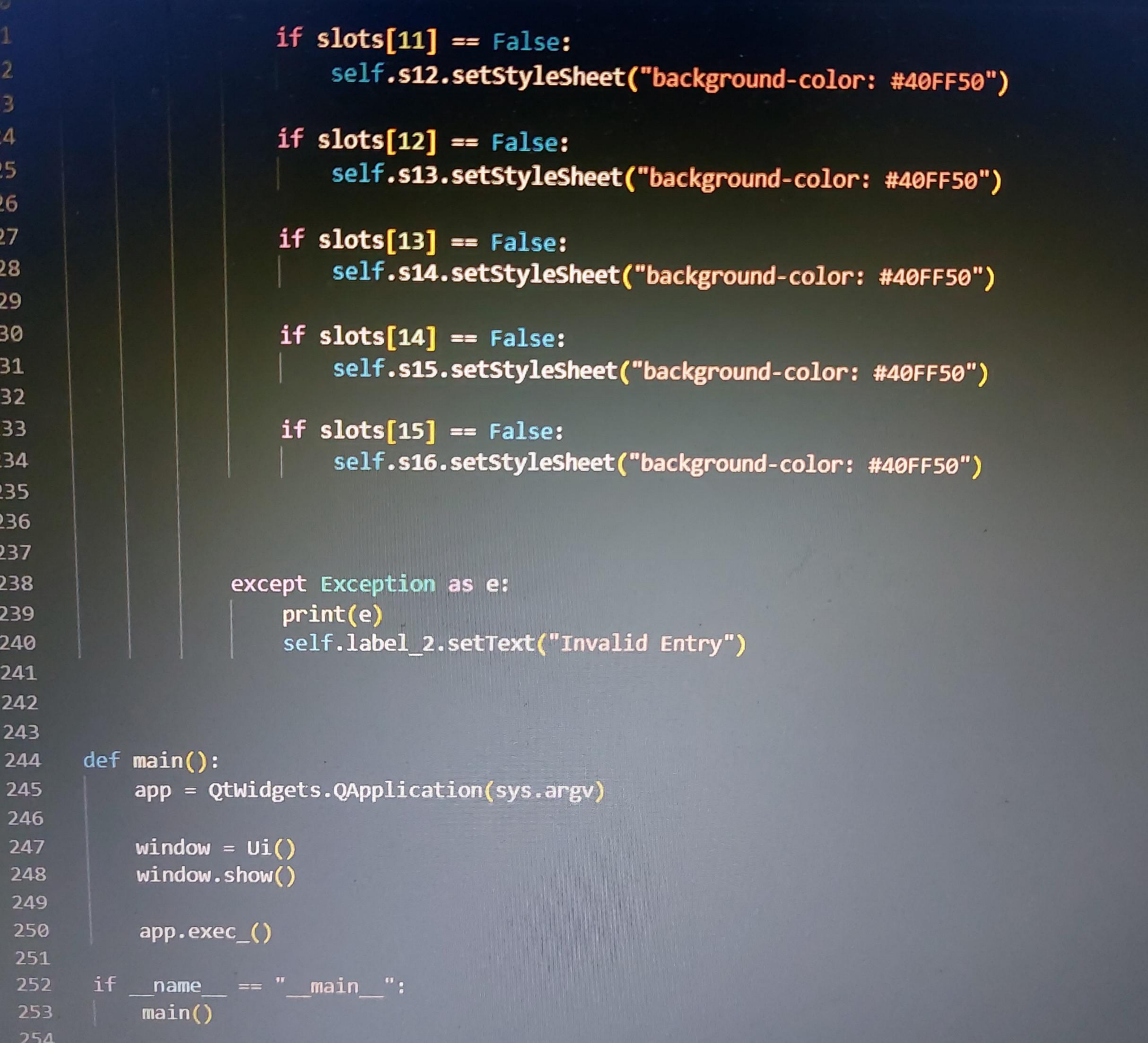
```
entry = datetime.datetime.now()
print(type(entry))


#mycursor.execute("INSERT INTO parkingdb (slot, carNumber, entry) VALUES(%s, %s, %s)", (slotNO, carNumber, entry
mycursor.execute("Insert INTO slot (carNumber, slot) VALUES(%s,%s)", (carNumber, slotNO))
mycursor.execute("Insert INTO entry (carNumber, entry) VALUES(%s,%s)", (carNumber, entry))
mycursor.execute("Insert INTO exits (carNumber) VALUES(%s)", (carNumber,))
mycursor.execute("Insert INTO duration (carNumber) VALUES(%s)", (carNumber,))
mycursor.execute("Insert INTO cost (carNumber) VALUES(%s)", (carNumber,))


self.label_2.setText("Slot: {:,}".format(int(slotNO)))

if slots[0] == True:
    self.s1.setStyleSheet("background-color: #FF0B00")

if slots[1] == True:
    self.s2.setStyleSheet("background-color: #FF0B00")

if slots[2] == True:
    self.s3.setStyleSheet("background-color: #FF0B00")

if slots[3] == True:
    self.s4.setStyleSheet("background-color: #FF0B00")

if slots[4] == True:
    self.s5.setStyleSheet("background-color: #FF0B00")

if slots[5] == True:
    self.s6.setStyleSheet("background-color: #FF0B00")

if slots[6] == True:
    self.s7.setStyleSheet("background-color: #FF0B00")

if slots[7] == True:
    self.s8.setStyleSheet("background-color: #FF0B00")
```



```
            if slots[11] == True:
                self.s12.setStyleSheet("background-color: #FF0B00")

            if slots[12] == True:
                self.s13.setStyleSheet("background-color: #FF0B00")

            if slots[13] == True:
                self.s14.setStyleSheet("background-color: #FF0B00")

            if slots[14] == True:
                self.s15.setStyleSheet("background-color: #FF0B00")

            if slots[15] == True:
                self.s16.setStyleSheet("background-color: #FF0B00")



        except Exception as e:
            print(e)
            self.label_2.setText("Invalid")

    def blank():
        print("in")
        self.label_2.setText("Empty")
        #time.sleep(5)


    def exit():
        try:
            carNumber = self.lineEdit.text()
            self.lineEdit.clear()
            #print(carNumber)

            exit1 = datetime.datetime.now()

            #slots[slotNO - 1] = False
```



```
mycursor.execute("update exits set exit1 = %s WHERE carNumber = %s", (exit1, carNumber))

mycursor.execute("select slot from slot where carNumber = %s", (carNumber,))
slotNO = int(re.sub("[^0-9]", "", str(mycursor.fetchone())))
print(slotNO)

slots[slotNO - 1] = False

#-----------------------TIME--------------------------

mycursor.execute("select entry from entry where carNumber = %s", (carNumber,))
#entry = str(mycursor.fetchone())
entry = re.sub('[,)(/\']', '', str(mycursor.fetchone()))
e = datetime.datetime.fromisoformat(entry)

time = int((exit1 - e).total_seconds())
#print(time)

cost =  int(10 * time)
#print(cost)
if cost > 150:
    cost = 150
self.label_2.setText("Cost: Rs." + str(cost))

mycursor.execute("update duration set durationInSec = %s WHERE carNumber = %s", (time, carNumber))
mycursor.execute("update cost set cost = %s WHERE carNumber = %s", (cost, carNumber))

if slots[0] == False:
    self.s1.setStyleSheet("background-color: #40FF50")

if slots[1] == False:
    self.s2.setStyleSheet("background-color: #40FF50")

if slots[2] == False:
    self.s3.setStyleSheet("background-color: #40FF50")
```



```
                if slots[11] == False:
                    self.s12.setStyleSheet("background-color: #40FF50")

                if slots[12] == False:
                    self.s13.setStyleSheet("background-color: #40FF50")

                if slots[13] == False:
                    self.s14.setStyleSheet("background-color: #40FF50")

                if slots[14] == False:
                    self.s15.setStyleSheet("background-color: #40FF50")

                if slots[15] == False:
                    self.s16.setStyleSheet("background-color: #40FF50")



            except Exception as e:
                print(e)
                self.label_2.setText("Invalid Entry")



def main():
    app = QtWidgets.QApplication(sys.argv)

    window = Ui()
    window.show()

    app.exec_()

if __name__ == "__main__":
    main()
```