Letter

Description automatically generated with low confidenceMAIN CODE

from tkinter import \*

from tkinter import messagebox

import time

import tkinter.font as font

from datetime import date

import cryptocompare

import datetime

import time

from datetime import date

from tkinter import \*

import tkinter.font as font

from tkinter import messagebox

import cryptocompare

import datetime

rooting =Tk()

rooting.title('Login to CrypToGen-Z')

rooting.geometry('850x650')

import mysql.connector

db = mysql.connector.connect(host='localhost', user='root', password='Arnavcool1', database='crypto')

cur=db.cursor()

#fg = '#03DAC6'

fg = '#FFFFFF'

rooting.configure(background=fg)

bg4 = PhotoImage(file='ool.png')

label=Label(rooting, image = bg4)

label.place(x=0, y=1)

#Time and Date

axess = False

t = time.localtime()

time\_now = time.strftime("%H.%M.%S", t)

d = date.today()

date\_now = d.strftime("%d-%b-%Y")

head = Label( text ='Welcome to CrypToGen-Z ',bg = fg, font=(None, 25, 'bold'))

head.place(x = 363, y = 50)

lab1 = Label( text = 'Enter User-ID (Email-Id): ',bg = fg, fg = '#646464', font=(None, 17))

lab1.place(x= 400, y= 160)

username = Entry(width=50, bg = fg, fg='#778899', font=(None, 15) )

username.place(x=400, y=200, width = 320)

lab2 = Label(text = 'Enter Password: ',bg = fg, fg = '#646464', font=(None, 17))

lab2.place(x= 400, y= 260)

password = Entry(width=50, bg = fg, fg='#778899', font=(None, 15))

password.place(x=400, y=300, width = 320)

lab4 = Label(text='Made by ¬© Arnav Suman | All Rights Reserved',bg = fg, font=(None, 12, 'bold'))

lab4.place(x=380, y=620)

def passworderror():

messagebox.showerror("PASSWORD ERROR ", "Please enter correct Password!")

def iderror():

messagebox.showerror("ID ERROR ", "Please enter correct USER-ID!")

query="select userid,password from userdata;"

cur.execute(query)

rows1222=cur.fetchall()

axess = False

u\_id=[]

u\_pa=[]

for i in rows1222:

u\_id.append(i[0])

u\_pa.append(i[1])

def pushq():

global axess, a, b

a=username.get()

b=password.get()

if a in u\_id:

clr\_1=True

else:

clr\_1=False

iderror()

if b in u\_pa:

clr\_2=True

else:

clr\_2=False

passworderror()

if clr\_1 and clr\_2 ==True:

axess = True

rooting.destroy()

def signup():

global axess, a, b

a=username.get()

a=str(a)

b=password.get()

b=str(b)

query1z="insert into userdata (userid, password, name, usd\_balance, crypto\_balance, history) values (%s, %s, %s, %s, %s, %s)"

val1z=(a, b,a, 1000, "{'BTC': 1.0, 'ETH': '3.0', 'SOL': '6.5'}", " BUY: 4.3BTC, SELL: 2.5BTC, BUY: 1.1BTC, SEll: 21.2XRP, SELL: 23.3DOGE, BUY: 2.3BTC, SELL: 11.2DOGE, BUY: 1.1BTC,")

cur.execute(query1z,val1z)

db.commit()

axess = True

rooting.destroy()

'''

--------------------

MYSQL DATA

create databse crypto

CREATE TABLE userdata (

userid varchar(100) primary key,

password varchar(100),

name varchar(255),

usd\_balance int,

crypto\_balance varchar(750),

history varchar(12000)

);

insert into userdata (userid, password, name, usd\_balance, crypto\_balance, history) values ('arnavsuman', 'test1','Arnav Suman', 1000, "{'BTC': 1.0, 'ETH': '3.0', 'DOGE': '12.0', 'XRP': '100.0', 'SOL': '6.5'}", " BUY: 4.3BTC, SELL: 2.5BTC, BUY: 1.1BTC, SEll: 21.2XRP, SELL: 23.3DOGE, BUY: 2.3BTC, SELL: 11.2DOGE, BUY: 1.1BTC,")

-----------------

'''

but = Button(text = 'Login', fg = 'black', bg = '#78ffd6',font=('Bahnschrift Light Condensed', 30), command = pushq).place(x = 450, y = 390)

butzx = Button(text = 'Sign Up', fg = 'black', bg = '#78ffd6',font=('Bahnschrift Light Condensed', 30), command = signup).place(x = 600, y = 390)

rooting.mainloop()

# LOGIN ENDS

#MAIN SCREEN

if axess == True:

root=Tk()

root.title('Welcome to CrypToGen-Z')

root.geometry('=1920x1080')

bg41 = PhotoImage(file='2.png')

label1=Label(image = bg41)

label1.place(x=410, y=100)

label1zx=Label(image = bg41)

label1zx.place(x=980, y=100)

querya="select \* from userdata where userid=(%s);"

us=(a,)

cur.execute(querya,us)

rows=cur.fetchall()

for i in rows:

user\_id\_str=list(i)[0]

password\_str=list(i)[1]

name\_str=list(i)[2]

usd\_balance\_str=list(i)[3]

crypto\_balance\_str=list(i)[4]

history\_str=list(i)[5]

bg='#2b263f' #2b263f

root.configure(background = bg)#0f1c4d #4380b0 #3a668b #172a41 #29445b #1b1626 #2b263f #a74e46

head = Label( text ='WELCOME ' + name\_str, bg=bg, fg = '#aba9b5', font = ('Bahnschrift Light Condensed', 25, 'bold'))

head.place(x = 10, y = 10)

wq=StringVar()

wq.set(str(usd\_balance\_str)+ ' $')

bbc = Label( textvariable=wq, bg = '#323150', fg = '#0c9d58', font = ('Bahnschrift Light Condensed', 25, 'bold')) .place(x = 1300, y = 10)

bbc33 = Label( text='$BALANCE', bg = '#323150', fg = '#aba9b5', font = ('Bahnschrift Light Condensed', 25, 'bold')) .place(x = 1130, y = 10)

qq=StringVar()

cost=[]

crypto\_balance\_str=eval(crypto\_balance\_str)

crypto\_balance\_str=dict(crypto\_balance\_str)

for i in crypto\_balance\_str.keys():

cc=i

asd=cryptocompare.get\_price(cc, currency='USD', full=True)

ppcoin=asd['RAW'][cc]['USD']['PRICE']

cost.append(float(crypto\_balance\_str[i])\*float(ppcoin))

sum=0

for i in cost:

sum+=i

tot\_sum=(round(sum,2))

qq.set(str(tot\_sum)+' $')

bbc = Label( textvariable=qq, bg = '#323150', fg = '#0c9d58', font = ('Bahnschrift Light Condensed', 25, 'bold')) .place(x = 930, y = 10)

bbc321 = Label( text='CRYPTO WORTH:', bg = '#323150', fg = '#aba9b5', font = ('Bahnschrift Light Condensed', 25, 'bold')) .place(x = 700, y = 10)

options = [

'BTC',

'ETH',

'SOL',

'LTC',

'XRP',

'DOGE',

'ADA',

'FTM',

'LUNA',

'DOT',

'SHIB',

'BNB',

'AVAX',

'MATIC',

'CRO',

'UST',

'ATOM',

'TRON',

'STX',

]

clicked = StringVar()

clicked.set(options[0])

lb555 = Label(text='Select Crypto: ', fg = 'white' , bg=bg, font = ('Bahnschrift Light Condensed', 15, ))

lb555.place(x = 170, y = 105)#'black'

drop = OptionMenu(root, clicked, \*options)

drop.place(x=293, y=105)

lb1aa\_string = StringVar()

lb1aa\_string.set('$BTC/USD')

lb1aa =Label(textvariable=lb1aa\_string, fg = '#0c9d58' ,bg=bg, font = ('Bahnschrift Light Condensed', 35, 'bold'))

lb1aa.place(x = 10, y = 60)

coin\_list\_name\_is=clicked.get()

def cc():

global appz

global coin\_list\_name\_is, appz, price\_today, crypto\_balance\_str

coin\_list\_name\_is=clicked.get()

for i in crypto\_balance\_str.keys():

if i ==coin\_list\_name\_is:

bal=crypto\_balance\_str[i]

break

else:

bal=0

coin\_list\_name\_is = clicked.get()

lb1aa\_string.set('$'+coin\_list\_name\_is+'/USD')

a=cryptocompare.get\_coin\_list(format=False)

for i in a:

if i ==coin\_list\_name\_is:

global cc\_na

cc\_na=a[i]['CoinName']

no.set(cc\_na)

appz= cryptocompare.get\_price(coin\_list\_name\_is, currency='USD', full=True)

nw.set(str(bal)+' '+coin\_list\_name\_is)

price\_today=appz['RAW'][str(coin\_list\_name\_is)]['USD']['PRICE']

prc3=appz['RAW'][coin\_list\_name\_is]['USD']['PRICE']

b0.set(str(prc3)+' $')

medw=appz['RAW'][coin\_list\_name\_is]['USD']['MEDIAN']

b1.set(str(medw)+' $')

vol\_2w4=appz['RAW'][coin\_list\_name\_is]['USD']['VOLUMEDAYTO']

zxe=str(vol\_2w4)

zxe=zxe[0:7]

b2.set(str(zxe)+' $')

pcccw=appz['RAW'][coin\_list\_name\_is]['USD']['CHANGE24HOUR']

pritdw=appz['RAW'][coin\_list\_name\_is]['USD']['PRICE']

pcccw=pcccw/pritdw

pcccw=str(pcccw)

pcccw=pcccw[0:7]

b3.set(pcccw+' %')

cgppw=appz['RAW'][coin\_list\_name\_is]['USD']['CHANGEPCTDAY']

cgppw=str(cgppw)

cgppw=cgppw[0:7]

b4.set(cgppw+' %')

mcapw=appz['RAW'][coin\_list\_name\_is]['USD']['MKTCAP']

b5.set(str(mcapw)+' $')

low24w=appz['RAW'][coin\_list\_name\_is]['USD']['LOW24HOUR']

b6.set(str(low24w)+ ' $')

photos = PhotoImage(file = "go.png")

butcc = Button(text = 'Go', fg = 'black', bg = bg,image = photos, font=('Bahnschrift Light Condensed', 25), command = cc).place(x = 360, y = 104)

cc\_na='Bitcoin'

na=Label(text='Crypto Name:', fg = '#9d9ea9' ,bg=bg, font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=20,y=160)

#2nd part od page

a1 = StringVar()

a1.set('')

a1.set('MEDIAN PRICE')

na11=Label(textvariable=a1, fg = 'white' ,bg=bg, font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=450,y=170)

a2 = StringVar()

a2.set('')

a2.set('VOLUME TRADED ($)')

na111=Label(textvariable=a2, fg = 'white' ,bg=bg, font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=450,y=240)

a3 = StringVar()

a3.set('')

a3.set('% CHANGE (24H)')

na1311=Label(textvariable=a3, fg = 'white' ,bg=bg, font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=450,y=320)

a4 = StringVar()

a4.set('')

a4.set('EXPT. % CHANGE (24H)')

na3111=Label(textvariable=a4, fg = 'white' ,bg=bg, font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=450,y=400)

a5 = StringVar()

a5.set('')

a5.set('TOTAL MARKET CAP')

na5111=Label(textvariable=a5, fg = 'white' ,bg=bg, font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=450,y=480)

a6 = StringVar()

a6.set('')

a6.set('24H LOW')

na66111=Label(textvariable=a6, fg = 'white' ,bg=bg, font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=450,y=560)

a0 = StringVar()

a0.set('')

a0.set('PRICE NOW')

na661131=Label(textvariable=a0, fg = 'white' ,bg=bg, font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=450,y=105)

apps= cryptocompare.get\_price(coin\_list\_name\_is, currency='USD', full=True)

prc2=apps['RAW'][coin\_list\_name\_is]['USD']['PRICE']

b0 = StringVar()

b0.set(str(prc2)+' $')

na1wb1a=Label(textvariable=b0, fg = '#0c9d58' ,bg='#323150', font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=750,y=105)

med=apps['RAW'][coin\_list\_name\_is]['USD']['MEDIAN']

b1 = StringVar()

b1.set(str(med)+' $')

na1b1a=Label(textvariable=b1, fg = '#0c9d58' ,bg='#323150', font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=750,y=170)

vol\_24=apps['RAW'][coin\_list\_name\_is]['USD']['VOLUMEDAYTO']

b2 = StringVar()

vol\_24=str(vol\_24)

vol\_24=vol\_24[0:7]

b2.set(str(vol\_24)+' $')

na1wb1a=Label(textvariable=b2, fg = 'white' ,bg='#323150', font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=750,y=240)

pccc=apps['RAW'][coin\_list\_name\_is]['USD']['CHANGE24HOUR']

pritd=apps['RAW'][coin\_list\_name\_is]['USD']['PRICE']

pccc=pccc/pritd

pccc=str(pccc)

pccc=pccc[0:7]

b3 = StringVar()

b3.set(pccc+' %')

na1b1dda=Label(textvariable=b3, fg = '#0c9d58' ,bg='#323150', font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=750,y=320)

cgpp=apps['RAW'][coin\_list\_name\_is]['USD']['CHANGEPCTDAY']

b4 = StringVar()

cgpp=str(cgpp)

cgpp=cgpp[0:7]

b4.set(cgpp+' %')

na1bd1a=Label(textvariable=b4, fg = '#0c9d58' ,bg='#323150', font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=750,y=400)

mcap=apps['RAW'][coin\_list\_name\_is]['USD']['MKTCAP']

b5 = StringVar()

b5.set(str(mcap)+' $')

na1fb1a=Label(textvariable=b5, fg = 'white' ,bg=b'#323150', font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=750,y=480)

low24=apps['RAW'][coin\_list\_name\_is]['USD']['LOW24HOUR']

b6 = StringVar()

b6.set(str(low24)+ ' $')

t33na1b1a=Label(textvariable=b6, fg = 'red' ,bg='#323150', font = ('Bahnschrift Light Condensed', 18, 'bold')).place(x=750,y=560)

def graph\_plotty(days):

import datetime

dates\_list=[]

open\_data=[]

high\_data = []

low\_data = []

close\_data =[]

from datetime import date as dt

today = dt.today()

d3 = today.strftime("%d/%m/%y")

from datetime import datetime as ddt

date\_time\_str = d3 + ' 00:00:00'

date\_time\_obj = ddt.strptime(date\_time\_str, '%d/%m/%y %H:%M:%S')

for u in cryptocompare.get\_historical\_price\_day(coin\_list\_name\_is, 'USD', days, toTs=date\_time\_obj): # limit is how many days you want to see +1

ts = u['time']

from datetime import datetime

# it will print dates

l1=datetime.utcfromtimestamp(ts).strftime('%Y-%m-%d %H:%M:%S')

dates\_list.append(l1)

open = u['open']

open\_data.append(open)

high =u['high']

high\_data.append(high)

low = u['low']

low\_data.append(low)

close = u['close']

close\_data.append(close)

new\_date=[]

for j in dates\_list:

x = datetime.fromisoformat(j)

new\_date.append(x)

import plotly.graph\_objects as go

fig = go.Figure(data=[go.Candlestick(x=new\_date,

open=open\_data, high=high\_data,

low=low\_data, close=close\_data)])

fig.show()

def graph\_finplot(days):

import finplot as fplt

import pandas as pd

new\_date=[]

open\_data=[]

high\_data=[]

low\_data=[]

close\_data=[]

final =[]

from datetime import date as dt

today = dt.today()

d3 = today.strftime("%d/%m/%y")

from datetime import datetime as ddt

date\_time\_str = d3 + ' 00:00:00'

date\_time\_obj = ddt.strptime(date\_time\_str, '%d/%m/%y %H:%M:%S')

for u in cryptocompare.get\_historical\_price\_day(coin\_list\_name\_is, 'USD', days, toTs=date\_time\_obj): # limit is how many days you want to see +1

ts = u['time']

open = u['open']

high =u['high']

low = u['low']

close = u['close']

volume=u['volumefrom']

amount = u['volumeto']

final.append([ts, open, high, low, close,volume, amount])

df = pd.DataFrame(final, columns='time open high low close volume amount'.split())

fplt.candlestick\_ochl(df[['time','open','close','high','low']])

fplt.show()

op\_list=['15 Days','1 Month','6 Months', '1 Year']

op\_c = StringVar()

op\_c.set(op\_list[1])

dropd = OptionMenu(root, op\_c, \*op\_list)

dropd.place(x=450, y=625)

def graph1():

days=op\_c.get()

if days==op\_list[0]:

days=16

elif days==op\_list[1]:

days=31

elif days==op\_list[2]:

days=181

elif days==op\_list[3]:

days=366

graph\_plotty(days)

def graph2():

days=op\_c.get()

if days==op\_list[0]:

days=16

elif days==op\_list[1]:

days=31

elif days==op\_list[2]:

days=181

elif days==op\_list[3]:

days=366

graph\_finplot(days)

burtc = Button(text = 'Plotty Web Graph', fg = 'black', bg = bg,font=('Bahnschrift Light Condensed', 25), command = graph1).place(x = 450, y = 680)

burtced = Button(text = 'FinPlot Graph', fg = 'black', bg = bg,font=('Bahnschrift Light Condensed', 25), command = graph2).place(x = 750, y = 680)

#2nd part of page

#3rd part of page

b211 = Label( text='TRANSACTION HISTORY', bg = '#323150', fg = '#aba9b5', font = ('Bahnschrift Light Condensed', 22, 'bold')) .place(x = 1060, y = 60)

history\_str1=history\_str.split(',')

history\_str2=history\_str1[0:-1]

fifth=StringVar()

fifth.set('1. '+history\_str2[-1])

b21r1 = Label( textvariable=fifth, bg = bg, fg = 'white', font = ('Bahnschrift Light Condensed', 18, 'bold')) .place(x = 1100, y = 105)

fourth=StringVar()

fourth.set('2. '+history\_str2[-2])

b21r331 = Label( textvariable=fourth, bg = bg, fg = 'white', font = ('Bahnschrift Light Condensed', 18, 'bold')) .place(x = 1100, y = 150)

third=StringVar()

third.set('3. '+history\_str2[-3])

b21r231 = Label( textvariable=third, bg = bg, fg = 'white', font = ('Bahnschrift Light Condensed', 18, 'bold')) .place(x = 1100, y = 195)

second=StringVar()

second.set('4. '+history\_str2[-4])

b21212r1 = Label( textvariable=second, bg = bg, fg = 'white', font = ('Bahnschrift Light Condensed', 18, 'bold')) .place(x = 1100, y = 240)

first=StringVar()

first.set('5. '+history\_str2[-5])

b212113r1 = Label( textvariable=first, bg = bg, fg = 'white', font = ('Bahnschrift Light Condensed', 18, 'bold')) .place(x = 1100, y = 285)

b22111 = Label( text='TRANSFER CRYPTO', bg = '#323150', fg = '#aba9b5', font = ('Bahnschrift Light Condensed', 28, 'bold')) .place(x = 1060, y = 370)

b2p1 = Label( text='TRANSFERING ACCOUNT ID', bg = bg, fg = 'white', font = ('Bahnschrift Light Condensed', 18, 'bold')) .place(x = 1040, y = 450)

ein3 = Entry(width=30, bg = 'white', font=(None, 15))

ein3.place(x=1045, y=490)

b21a = Label( text='SELECT CRYPTO', bg = bg, fg = 'white', font = ('Bahnschrift Light Condensed', 18, 'bold')) .place(x = 1040, y = 540)

b21a = Label( text='HOW MANY?', bg = bg, fg = 'white', font = ('Bahnschrift Light Condensed', 18, 'bold')) .place(x = 1040, y = 640)

ein32 = Entry(width=17, bg = 'white', font=(None, 15))

ein32.place(x=1180, y=640)

def abc():

to\_send\_user=str(ein3.get())

how\_coin\_send=float(ein32.get())

which\_coin=str(clickedlip.get())

for i in crypto\_balance\_str.keys():

if i ==which\_coin: #add what to do if i= not = coin

balance = float(crypto\_balance\_str[i])

if balance>how\_coin\_send:

balance=balance-how\_coin\_send

balance=round(balance,3)

crypto\_balance\_str[i]=balance

how\_coin\_send=str(how\_coin\_send)

history\_upd=history\_str+' TRANSFERRED: '+how\_coin\_send+which\_coin+','

query="update userdata set crypto\_balance = (%s), history =(%s) where userid=(%s);"

val=(str(crypto\_balance\_str),history\_upd, user\_id\_str)

cur.execute(query,val)

db.commit()

#how\_coin\_send bal

how=float(how\_coin\_send)

bals=bal

bals=bals-how

bals=round(bals,3)

nw.set(str(bals)+' '+which\_coin)

messagebox.showinfo("SOLD Coin", "Congrats you have Transferred the coins.")

#friend side

query="select crypto\_balance from userdata where userid=(%s);"

val=(to\_send\_user,)

cur.execute(query, val)

row=cur.fetchall()

row=str(row)

row=row[3:-4]

row=eval(row)

row=dict(row)

for i in row.keys():

if i ==which\_coin:

avail=float(row[i])

howocoin=float(how\_coin\_send)

avail=avail+howocoin

avail=str(avail)

row[i]=avail

row=str(row)

query="update userdata set crypto\_balance = (%s) where userid=(%s);"

val=(row, to\_send\_user)

cur.execute(query,val)

db.commit()

else:

messagebox.showerror("COINS TOO LOW", "You don't have enough coins to sell! Try a lower number.")

ss2wsa = Button(text = 'TRANSFER COINS', fg = 'black', bg = bg,font=('Bahnschrift Light Condensed', 25), command = abc).place(x = 1100, y = 700)

optionslip = [

'BTC',

'ETH',

'SOL',

'LTC',

'USDT',

'XRP',

'DOGE',

'ADA',

'FTM',

'LUNA',

'DOT',

'SHIB',

'BNB',

'AVAX',

'MATIC',

'CRO',

'UST',

'ATOM',

'TRON',

'STX',

'BFYC',]

clickedlip = StringVar()

clickedlip.set(options[0])

droplip = OptionMenu(root, clickedlip, \*optionslip)

droplip.place(x=1040, y=580)

#3rd part of page

no = StringVar()

no.set(cc\_na)

na1=Label(textvariable=no, fg = 'white' ,bg='#323150', font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=265,y=150)

for i in crypto\_balance\_str.keys():

if i ==coin\_list\_name\_is:

bal=crypto\_balance\_str[i]

break

else:

bal=0

nw= StringVar()

nw.set(str(bal)+' '+coin\_list\_name\_is)

ba=Label(textvariable=nw, fg = 'white' ,bg=bg, font = ('Bahnschrift Light Condensed', 20, 'bold')).place(x=255,y=200) # take balance of a specific crypto from sql znd paste

na111=Label(text='Crypto Balance:', fg = '#9d9ea9' ,bg=bg, font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=18,y=205)

opt\_lis=['SELL','BUY']

opt = StringVar()

opt.set(opt\_lis[0])

drop = OptionMenu(root, opt, \*opt\_lis)

drop.place(x=140, y=280)

b\_se = Label(text='Buy or Sell?', fg = '#9d9ea9' ,bg=bg, font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=18,y=280)

def sell\_button():

global bal, usd\_balance\_str, history\_str, crypto\_balance\_str

coin\_to\_sell = int(en3.get())

total\_money = coin\_to\_sell\*price\_today

coins\_sold=coin\_to\_sell

for i in crypto\_balance\_str.keys():

if i ==coin\_list\_name\_is:

bal=crypto\_balance\_str[i]

break

else:

bal=0

if coin\_to\_sell<float(bal):

bal=float(bal)-float(coins\_sold)

nw.set(str(round(float(bal),2))+' '+coin\_list\_name\_is)

usd\_balance\_str=usd\_balance\_str+total\_money

usd\_balance\_str=round(usd\_balance\_str,4)

crypto\_balance\_str[coin\_list\_name\_is]=str(round(float(bal),2))

history\_str11=history\_str+' SELL: '+str(coin\_to\_sell)+coin\_list\_name\_is+','

usd\_balance\_str=int(usd\_balance\_str)

wq.set(str(usd\_balance\_str)+ ' $')

query1="update userdata set usd\_balance = (%s), crypto\_balance = (%s), history =(%s) where userid=(%s);"

val1=(usd\_balance\_str,str(crypto\_balance\_str),history\_str, user\_id\_str)

cur.execute(query1,val1)

db.commit()

cost=[]

for i in crypto\_balance\_str.keys():

cc=i

asd=cryptocompare.get\_price(cc, currency='USD', full=True)

ppcoin=asd['RAW'][cc]['USD']['PRICE']

cost.append(float(crypto\_balance\_str[i])\*float(ppcoin))

sum=0

for i in cost:

sum+=i

tot\_sum=(round(sum,2))

qq.set(str(tot\_sum)+' $')

history\_str=history\_str11

history\_str1=history\_str.split(',')

history\_str2=history\_str1[0:-1]

fifth.set('1. '+history\_str2[-1])

fourth.set('2. '+history\_str2[-2])

third.set('3. '+history\_str2[-3])

second.set('4. '+history\_str2[-4])

first.set('5. '+history\_str2[-5])

messagebox.showinfo("SOLD Coin", "Congrats you have sold the coins.")

else:

messagebox.showerror("COINS TOO LOW", "You don't have enough coins to sell! Try a lower number.")

def buybutton():

global bal, usd\_balance\_str, history\_str, crypto\_balance\_str

coins\_bought = float(en3b.get())

total\_money=coins\_bought\*price\_today

for j in crypto\_balance\_str.keys():

if j ==coin\_list\_name\_is:

bal=crypto\_balance\_str[j]

break

else:

bal=0

if total\_money<usd\_balance\_str:

bal=float(bal)+float(coins\_bought)

usd\_balance\_str=usd\_balance\_str-total\_money

crypto\_balance\_str[coin\_list\_name\_is]=str(round(float(bal),2))

history\_str112=history\_str+' BUY: '+str(coins\_bought)+coin\_list\_name\_is+','

nw.set(str(round(float(bal),2))+' '+coin\_list\_name\_is)

usd\_balance\_str=int(usd\_balance\_str)

wq.set(str(usd\_balance\_str)+ ' $')

query1="update userdata set usd\_balance = (%s), crypto\_balance = (%s), history =(%s) where userid=(%s);"

val1=(usd\_balance\_str,str(crypto\_balance\_str),history\_str, user\_id\_str)

cur.execute(query1,val1)

db.commit()

for i in crypto\_balance\_str.keys():

cc=i

asd=cryptocompare.get\_price(cc, currency='USD', full=True)

ppcoin=asd['RAW'][cc]['USD']['PRICE']

cost.append(float(crypto\_balance\_str[i])\*float(ppcoin))

sum=0

for i in cost:

sum+=i

tot\_sum=(round(sum,2))

qq.set(str(tot\_sum)+' $')

history\_str=history\_str112

history\_str3=history\_str.split(',')

history\_str4=history\_str3[0:-1]

fifth.set('1. '+history\_str4[-1])

fourth.set('2. '+history\_str4[-2])

third.set('3. '+history\_str4[-3])

second.set('4. '+history\_str4[-4])

first.set('5. '+history\_str4[-5])

messagebox.showinfo("BOUGHT Coin", "Congrats you have bought the coins.")

else:

messagebox.showerror("USD BALANCE TOO LOW", "You don't have enough USD Balance to sell! Try to buy a lower number.")

def sellbuy():

text1=StringVar()

text2=StringVar()

text3=StringVar()

text4=StringVar()

if opt.get()=='SELL':

global coin\_list\_name\_is

text1.set('How many coins to sell?')

text2.set('Or how much of?')

text3.set('SELL SUMMARY')

text4.set(coin\_list\_name\_is)

low=appz['RAW'][coin\_list\_name\_is]['USD']['LOW24HOUR']

low=round(low,4)

vol=appz['RAW'][coin\_list\_name\_is]['USD']['VOLUME24HOUR']

vol=round(vol,4)

hig=appz['RAW'][coin\_list\_name\_is]['USD']['HIGH24HOUR']

hig=round(hig,4)

global en3, en31

wes=Label(textvariable=text1, fg = '#9d9ea9' ,bg='#323150', font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=20,y=350)

en3 = Entry(width=50, bg = 'white', font=(None, 15))

en3.place(x=230, y=350, width = 120)

sig=Label(textvariable=text4,bg=bg, fg = 'white', font = ('Bahnschrift Light Condensed', 20, 'bold')).place(x=360,y=350)

wes1=Label(textvariable=text2, fg = '#9d9ea9' ,bg='#323150', font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=20,y=410)

en31 = Entry(width=50, bg = 'white', font=(None, 15))

en31.place(x=230, y=410, width = 120)

sig1=Label(text='$',bg=bg, fg = 'white', font = ('Bahnschrift Light Condensed', 20, 'bold')).place(x=360,y=410)

ig1=Label(textvariable=text3,bg=bg, fg = 'white', font = ('Bahnschrift Light Condensed', 20, 'bold')).place(x=20,y=510)

pp=Label(text='24HR VOLUME', bg=bg, fg='white',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=20,y=570)

op=Label(text='OPEN PRICE', bg=bg, fg='white',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=20,y=610)

ppc=Label(text='TODAY HIGH', bg=bg, fg='white',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=20,y=650)

wsd=Label(text=str(price\_today)+' $', bg=bg, fg='#0c9d58',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=280,y=610)

wwsd=Label(text=str(hig)+' $', bg=bg, fg='#0c9d58',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=280,y=650)

wwdsd=Label(text=str(vol)+' '+ coin\_list\_name\_is, bg=bg, fg='white',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=265,y=570)

edfv=Button(text = 'PLACE ORDER', fg = 'black', bg = bg, font=('Bahnschrift Light Condensed', 25), command = sell\_button).place(x = 120, y = 700)

elif opt.get()=='BUY':

global en3b, en31b

text1.set('How many coins to buy?')

text2.set('Or how much of?')

text3.set('BUY SUMMARY')

low=appz['RAW'][str(coin\_list\_name\_is)]['USD']['LOWDAY']

low=round(low,4)

vol=appz['RAW'][str(coin\_list\_name\_is)]['USD']['VOLUME24HOUR']

vol=round(vol,4)

wes=Label(textvariable=text1, fg = '#9d9ea9' ,bg='#323150', font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=20,y=350)

en3b = Entry(width=50, bg = 'white', font=(None, 15))

en3b.place(x=230, y=350, width = 120)

sig=Label(text=coin\_list\_name\_is,bg=bg, fg = 'white', font = ('Bahnschrift Light Condensed', 20, 'bold')).place(x=360,y=350)

wes1=Label(textvariable=text2, fg = '#9d9ea9' ,bg='#323150', font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=20,y=410)

en31b = Entry(width=50, bg = 'white', font=(None, 15))

en31b.place(x=230, y=410, width = 120)

sig1=Label(text='$',bg=bg, fg = 'white', font = ('Bahnschrift Light Condensed', 20, 'bold')).place(x=360,y=410)

ig1=Label(textvariable=text3,bg=bg, fg = 'white', font = ('Bahnschrift Light Condensed', 20, 'bold')).place(x=20,y=510)

pp=Label(text='24HR VOLUME', bg=bg, fg='white',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=20,y=570)

op=Label(text='OPEN PRICE', bg=bg, fg='white',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=20,y=610)

ppc=Label(text='TODAY LOW', bg=bg, fg='white',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=20,y=650)

wsd=Label(text=str(price\_today)+' $', bg=bg, fg='#0c9d58',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=280,y=610)

wwsd=Label(text=str(low)+' $', bg=bg, fg='#0c9d58',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=280,y=650)

wwdsd=Label(text=str(vol)+' '+ coin\_list\_name\_is, bg=bg, fg='white',font = ('Bahnschrift Light Condensed', 15, 'bold')).place(x=280,y=570)

edfv=Button(text = 'PLACE ORDER', fg = 'black', bg = bg, font=('Bahnschrift Light Condensed', 25), command = buybutton).place(x = 120, y = 700)

photoxc = PhotoImage(file = "go.png")

butccxc = Button(text = 'Go', fg = 'black', bg = bg,image = photos, font=('Bahnschrift Light Condensed', 25), command = sellbuy).place(x = 220, y = 280)

root.mainloop()