```
from tkinter import *
from tkinter import ttk
import yfinance as yf
import tkinter as tk
import datetime
from datetime import date, timedelta
import os
import matplotlib.pyplot as plt
from tkinter import messagebox as tmsg
import sys

def quit():
    sys.exit()

def clear_screen():
    for widget in screen.winfo_children():
        widget.destroy()


def line_chart():
    plt.plot(data["Date"],data["Close"],color='red',label="Close")
    plt.plot(data["Date"],data["Low"],color='yellow',label="Low")
    plt.plot(data["Date"],data["Open"],color='green',label="Open")
    plt.plot(data["Date"],data["High"],color='orange',label="high")
    plt.xlabel("Date")
    plt.ylabel("Stock Prices")
    plt.title("Stock analysis for "+symbol)
    plt.legend()
    plt.show()


def candlestick_chart():
    symbol_small=data
    green_df=symbol_small[symbol_small["Close"]>symbol_small["Open"]].copy()
    green_df["Height"]=green_df["Close"]-green_df["Open"]
    red_df=symbol_small[symbol_small["Close"]<symbol_small["Open"]].copy()
    red_df["Height"]=red_df["Open"]-red_df["Close"]
    plt.vlines(x=green_df["Date"],ymin=green_df["Low"],ymax=green_df["High"],c
olor="green")
    plt.vlines(x=red_df["Date"],ymin=red_df["Low"],ymax=red_df["High"],color="
orangered")
    plt.bar(x=green_df["Date"],height=green_df["Height"],bottom=green_df["Open
"],color="green",label="Close")
    plt.bar(x=red_df["Date"],height=red_df["Height"],bottom=red_df["Close"],co
lor="orangered",label="Open")
    plt.xlabel("Date")
    plt.ylabel("Stock Prices")
    plt.title("Stock Analysis for "+symbol)
    plt.show()
```

```
def scatter_chart():
    plt.scatter(data["Date"],data["Close"],color='yellow')
    plt.scatter(data["Date"],data["Low"],color='yellow',label="Low")
    plt.scatter(data["Date"],data["Open"],color='green',label="Open")
    plt.scatter(data["Date"],data["High"],color='orange',label="high")
    plt.xlabel("Date")
    plt.ylabel("Stock Prices")
    plt.title("Stock analysis for "+symbol)
    plt.legend()
    plt.show()



def graph():
    clear_screen()

    global bg1,bg2,bg3

    Label(screen,text="Please Select Your Preferable Graph From Which You Want
To Analyse Your Stock",font="Calibri 24 bold",bg=colour).pack(side=TOP)
    bg1 = PhotoImage(file = "line chart crop.png")
    bg2 = PhotoImage(file = "candlestick.png")
    bg3 = PhotoImage(file = "scatter-chart-example.png")

    label1 = Label(screen, image = bg1,height=300,width=350).place(x = 50, y =
160)
    a=Button(screen,text="Line Chart",font="Calibri 16
bold",command=line_chart,width=10,height=1).place(x=160,y=500)

    label2 = Label( screen, image = bg2,height=300,width=350)
    label2.place(x = 500, y = 160)
    b=Button(screen,text="CandleStick Chart",font="Calibri 16
bold",width=20,height=1,command=candlestick_chart)
    b.place(x=585,y=500)


    label3= Label( screen, image = bg3,height=300,width=350)
    label3.place(x = 950, y = 160)
    c=Button(screen,text="Scatter Chart",font="Calibri 16
bold",width=10,height=1,command=scatter_chart)
    c.place(x=1080,y=500)



    Button(screen,text="Back",font="Calibri 16
bold",width=10,height=2,command=stock_screen).pack(anchor="ne")
```

```
    Button(screen,text="Quit",font="Calibri 16
bold",width=10,height=2,command=quit).place(x=1230,y=600)


def check_graph():
    print("working")
    s1=s.get()
    d1=d.get()
    if s1=="" or d1=="":
        print("No")
        tmsg.showerror("Error","Please Enter both fields")
    else:
        print("Yes")
        graph()



def stock_screen():
    clear_screen()
    def fetch_stock_history():

        global data
        global symbol

        symbol = symbol_entry.get()
        days = int(days_entry.get())

        today = date.today()
        end_date = today.strftime("%Y-%m-%d")

        start_date = today - timedelta(days)
        start_date = start_date.strftime("%Y-%m-%d")

        data = yf.download(symbol, start=start_date, end=end_date,
progress=False)
        data["Date"] = data.index

        data = data[["Date", "Open", "High", "Low", "Close", "Adj Close",
"Volume"]]
        data.reset_index(drop=True, inplace=True)

        result_text.delete(1.0, tk.END)
        result_text.insert(tk.END, data.to_string(index=False))

    global s
    global d
    screen.title("Stock Analysis")
    s=StringVar()
```

```
    d=StringVar()
    symbol_label = ttk.Label(screen, text="Enter Symbol of
Stock:",font="Calibri 16 bold",background=colour,foreground="blue")
    symbol_label.pack()

    symbol_entry = Entry(screen,textvariable=s)
    symbol_entry.pack()

    Label(screen,text="",bg=colour).pack()

    days_label = ttk.Label(screen, text="Enter Number of Days for Stock
History:",font="Calibri 16 bold",background=colour,foreground="blue")
    days_label.pack()

    days_entry = Entry(screen,textvariable=d)
    days_entry.pack()

    fetch_button = ttk.Button(screen, text="Fetch Stock History",
command=fetch_stock_history)
    fetch_button.pack()

    v=Scrollbar(screen, orient='vertical')
    v.pack(side=RIGHT, fill='y')

    result_text = tk.Text(screen, height=25, width=120, bg=colour, fg="black",
yscrollcommand=v.set)
    v.config(command=result_text.yview)
    result_text.pack()

    Label(screen,text="",bg=colour).pack()
    graph_label=tk.Label(screen,text="If You Want To Analyse Your Stock
Graphically Click On Ok Button",font="Calibri 16
bold",background=colour,foreground="blue")
    graph_label.pack()
    Label(screen,text="",bg=colour).pack()
    ok_button=tk.Button(screen,text="Ok",command=check_graph,height=1,width=10
,font="Calibi 16 bold")
    ok_button.pack()

def login_user(username,password):
    username=username.get()
    password=password.get()

    username_entry1.delete(0,END)
    password_entry1.delete(0,END)

    if username == "" or password == "":
        tmsg.showerror("Error", "Please enter both username and password.")
```

```
    else:
        if not "Credentials.txt" in os.listdir():
            file = open("Credentials.txt", "w")
            file.close()
            tmsg.showerror("Error","Invalid Username or Password")
        else:
            with open("Credentials.txt", "r") as file:
                for line in file:
                    stored_username, stored_password = line.strip().split(",")
                    if username == stored_username and password ==
stored_password:
                        tmsg.showinfo("Success", "Login successful!")
                        stock_screen()
                        return
                tmsg.showerror("Error", "Invalid username or password.")


def back_mainscreen():
    clear_screen()
    screen.title("Welcome Page")
    Label(screen,text="Welcome",bg="grey",font="Calibri 24
bold",width=300,height=2).pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="",bg=colour).pack()
    Button(screen,text="Login",font="Calibri 16
bold",width=30,height=2,command=login).pack()
    Label(screen,text="",bg=colour).pack()
    Button(screen,text="Register",font="Calibri 16
bold",width=30,height=2,command=register).pack()

def login() :
    clear_screen()

    global username_entry1
    global password_entry1

    screen.title("Login")
    screen.geometry(dimensions)

    username_verify=StringVar()
    password_verify=StringVar()

    Label(screen,text="Please Enter Details Below To Login",font="Calibri 24
bold",bg=colour,fg="black").pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="Username *",font="Calibri 16 bold",bg=colour).pack()
    username_entry1=Entry(screen,textvariable=username_verify)
```

```
    username_entry1.pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="Password *",font="Calibri 16 bold",bg=colour).pack()
    password_entry1=Entry(screen,textvariable=password_verify,show="*")
    password_entry1.pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="",bg=colour).pack()
    Button(screen,text="Login",font="Calibri 16
bold",width=10,height=1,command=lambda:login_user(username_verify,password_ver
ify)).pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="",bg=colour).pack()
    Button(screen,text="Back",font="Calibri 16
bold",width=10,height=1,command=back_mainscreen).pack()



def register_user(username, password) :
    if username=="" or password=="" :
        tmsg.showerror("Error","Please Enter all the fields" )
    else:
        with open("Credentials.txt", "a") as f:
            f.write(f"{username},{password}\n")
            print("Saved")
        Label(screen,text="Registration Success",font="Calibri 16
bold",fg="green",bg=colour).pack()

    username_entry.delete(0,END)
    password_entry.delete(0,END)
    confirm_entry.delete(0,END)



def check_unique(username, password):

    if not "Credentials.txt" in os.listdir():
        file = open("Credentials.txt", "w")
        file.close()

    with open("Credentials.txt", "r") as f:
        content = f.readlines()
        existing_usernames = []
        for items in content:
            existing_usernames.append(items.strip().split(",")[0])
        if username not in existing_usernames:
            register_user(username, password)
        else:
```

```
            tmsg.showwarning("Username Already Taken","Username Already
Taken!")

def check_password(password,confirm_password,username):
    username=username.get()
    password=password.get()
    confirm_password=confirm_password.get()
    if password==confirm_password:
        check_unique(username,password)
    else:
        tmsg.showerror("Error","Password and Confirm Password not matching")

def register() :
    clear_screen()
    global username_entry
    global password_entry
    global confirm_entry

    global username
    global password

    screen.title("Register")
    screen.geometry(dimensions)

    username=StringVar()
    password=StringVar()
    confirm_password=StringVar()

    Label(screen,text="Please Enter Details Below",font="Calibri 24
bold",fg="black",bg=colour).pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="Username *",font="Calibri 16 bold",bg=colour).pack()
    username_entry=Entry(screen,textvariable=username)
    username_entry.pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="Password *",font="Calibri 16 bold",bg=colour).pack()
    password_entry=Entry(screen,textvariable=password,show="*")
    password_entry.pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="Confirm Password *",font="Calibri 16
bold",bg=colour).pack()
    confirm_entry=Entry(screen,textvariable=confirm_password,show="*")
    confirm_entry.pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="",bg=colour).pack()
    Button(screen,text="Register",font="Calibri 16
bold",width=10,height=1,command=lambda:
check_password(password,confirm_password,username)).pack()
```

```
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="",bg=colour).pack()
    Button(screen,text="Back",font="Calibri 16
bold",width=10,height=1,command=back_mainscreen).pack()

def mainscreen() :
    global screen
    screen=Tk()
    screen.geometry(dimensions)
    screen.title("Welcome Page")
    Label(screen,text="Welcome",bg="grey",font="Calibri 24
bold",width=300,height=2).pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="",bg=colour).pack()
    Label(screen,text="",bg=colour).pack()
    Button(screen,text="Login",font="Calibri 16
bold",width=30,height=2,command=login).pack()
    Label(screen,text="",bg=colour).pack()
    Button(screen,text="Register",font="Calibri 16
bold",width=30,height=2,command=register).pack()
    screen.configure(background=colour)
    photo=PhotoImage(file="Stock Market Icon Crop.png")
    screen.iconphoto(False,photo)
    screen.resizable(0,0)
    screen.mainloop()



dimensions="1350x700"
colour="Aqua"


mainscreen()
```