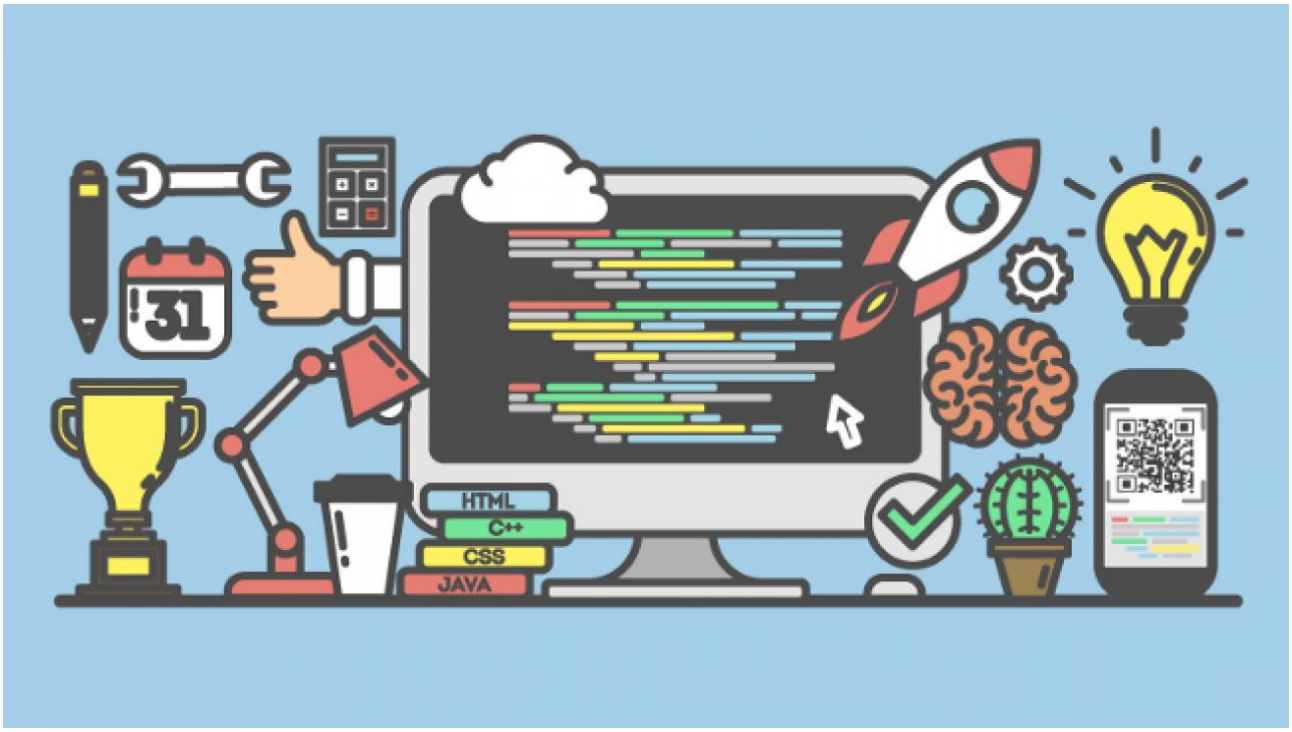


REVENUE/SALES COMPARISON AND ANALYSIS

COMPUTER SCIENCE PROJECT

(2021-2022)

DAV INTERNATIONAL SCHOOL

NAME: ANURAG KAVI SHANKAR THAKUR

CLASS: XI - A

ROLL NO: 6

INDEX

INTRODUCTION

PROGRAMMING IS THE PROCESS OF CREATING A SET OF INSTRUCTIONS THAT TELLS A COMPUTER HOW TO PERFORM A TASK. IN MY PROGRAM, I AM GOING TO ANALYSE THE DATA SET GIVEN BY THE USER, THEN ANALYSE IT TO DEPICT IT IN THE FORM OF A GRAPH. WITH THE HELP OF MY PROGRAM THE USER CAN ANALYSE ANY COMPANIE'S GROWTH OVER YEARS AND CAN EVEN COMPARE THE GROWTH RATE OF UP TO 5 DIFFERENT COMPANIES. THE GRAPHICAL REPRESENTATION MAKES IT EVEN EASIER TO FOR THE USER TO COMPARE AND ANALYSE. MY PROGRAM ANALYSES THE YEAR-WISE SALE OR REVENUE OF A COMPANY, THEREFORE INPUTTING MORE DATA WOULD BE GREAT. MY PROGRAM TAKES INPUT IN THE FORM OF AN EXCEL SHEET, AS INPUTTING SUCH A WIDE RANGE OF DATA STEP BY STEP DURING THE PROGRAM IS TIME-CONSUMING AND ALSO DATA IS USUALLY STORED IN THE FORM OF AN EXCEL SHEET.

PYTHON CONCEPTS USED

MY PROGRAM CONSISTS OF MANY PYTHON CONCEPTS, WHICH HELPED IN MAKING IT MORE EFFECTIVE IN TAKING THE INPUT AND READING IT.

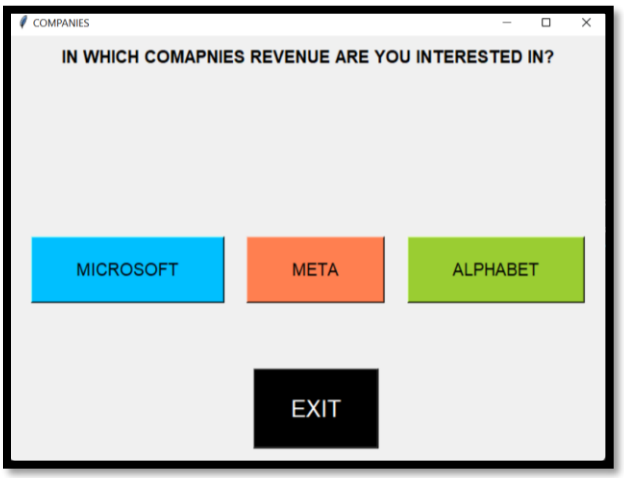
I USED CONDITIONAL STATEMENTS WHICH HAVE A HUGE EFFECT ON THE OUTPUT, AS CHOICES MADE BY THE USER DECIDE HOW THE PROGRAM WILL PROCEED FURTHER.

MOREOVER, I HAVE USED PYTHON LIBRARIES WHICH MAKE MY PROGRAM LOOK GOOD, USER-FRIENDLY, THUS, MAKING IT EASIER FOR USERS TO INTERACT WITH THE PROGRAM.

PYTHON LIBRARIES USED IN THE PROGRAM ARE:

1. TKINTER

TKINTER IS A PYTHON LIBRARY USED TO CREATE A GRAPHICAL USER INTERFACE(GUI). PYTHON CODE WHEN COMBINED WITH TKINTER PROVIDES A FAST AND EASY WAY FOR USERS TO INTERACT WITH THE PROGRAM.



EXAMPLE FROM MY PROGRAM

I HAVE USED TKINTER TO CREATE BUTTONS, FOR USERS TO MAKE CHOICES, AND SCREEN FOR INPUTTING THEIR FILE LOCATION AND DISPLAYING THE

GRAPHS. THIS LIBRARY HELPED ME IN MAKING MY PROGRAM LOOK BETTER AND EASY TO INTERACT WITH.

2. PANDAS

PANDAS IS A PYTHON LIBRARY, MAINLY USED FOR DATA ANALYSIS AND DATA MANIPULATION.

```
data=(pd.read_excel('Sales data.xlsx'))
```

EXAMPLE FROM MY PROGRAM

I HAVE USED PANDAS TO READ THE INPUT GIVEN AS EXCEL FILE BY THE USER. WITH THE HELP OF THIS LIBRARY I WAS ABLE TO LOCATE THE DATA IN EXCEL SHEET AND INPUT IT INTO THE GRAPH FOR ANALYSING AND COMPARING.

3. OS

THE OS MODULE IN PYTHON PROVIDES FUNCTIONS FOR INTERACTING WITH THE OPERATING SYSTEM.

I HAVE USED OS LIBRARY TO LOCATE AND TELL IF THE PROGRAM COULD FIND THE FILE IN THE SYSTEM OR NOT. I HAVE USED IT TO CHECK THE AVAILABILITY OF THE FILE ONLY.

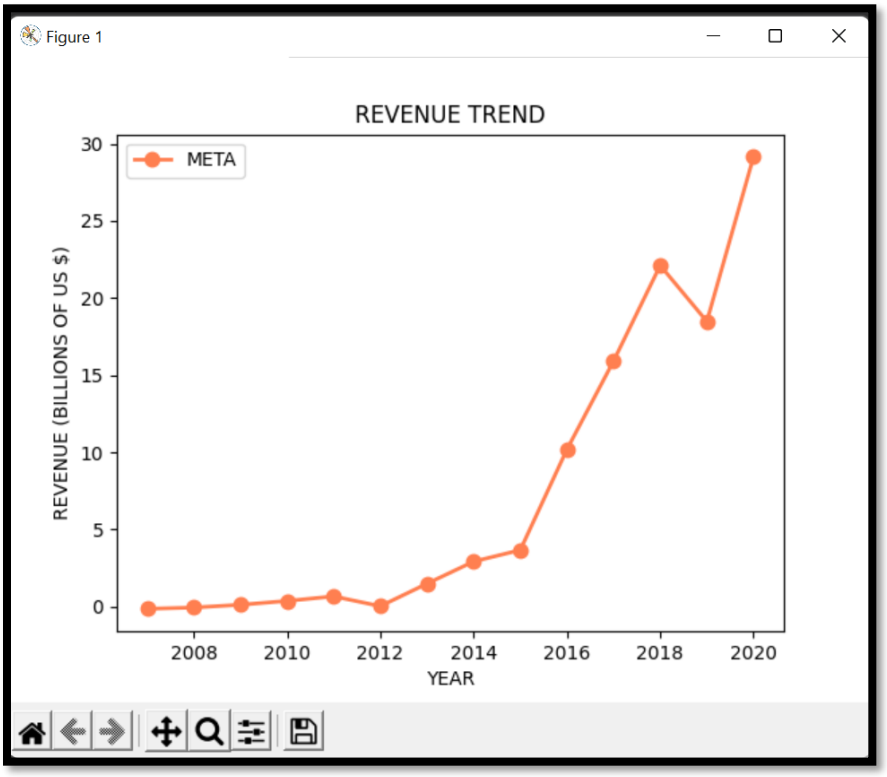
```
assert os.path.exists(path1), " I did not find your file at, "+str(path1)
f=open(path1)
print("Hooray! We found your file.")
```

EXAMPLE FROM MY PROGRAM

4. MATPLOTLIB

MATPLOTLIB IS A CROSS-PLATFORM, DATA VISUALIZATION AND GRAPHICAL PLOTTING LIBRARY FOR PYTHON.

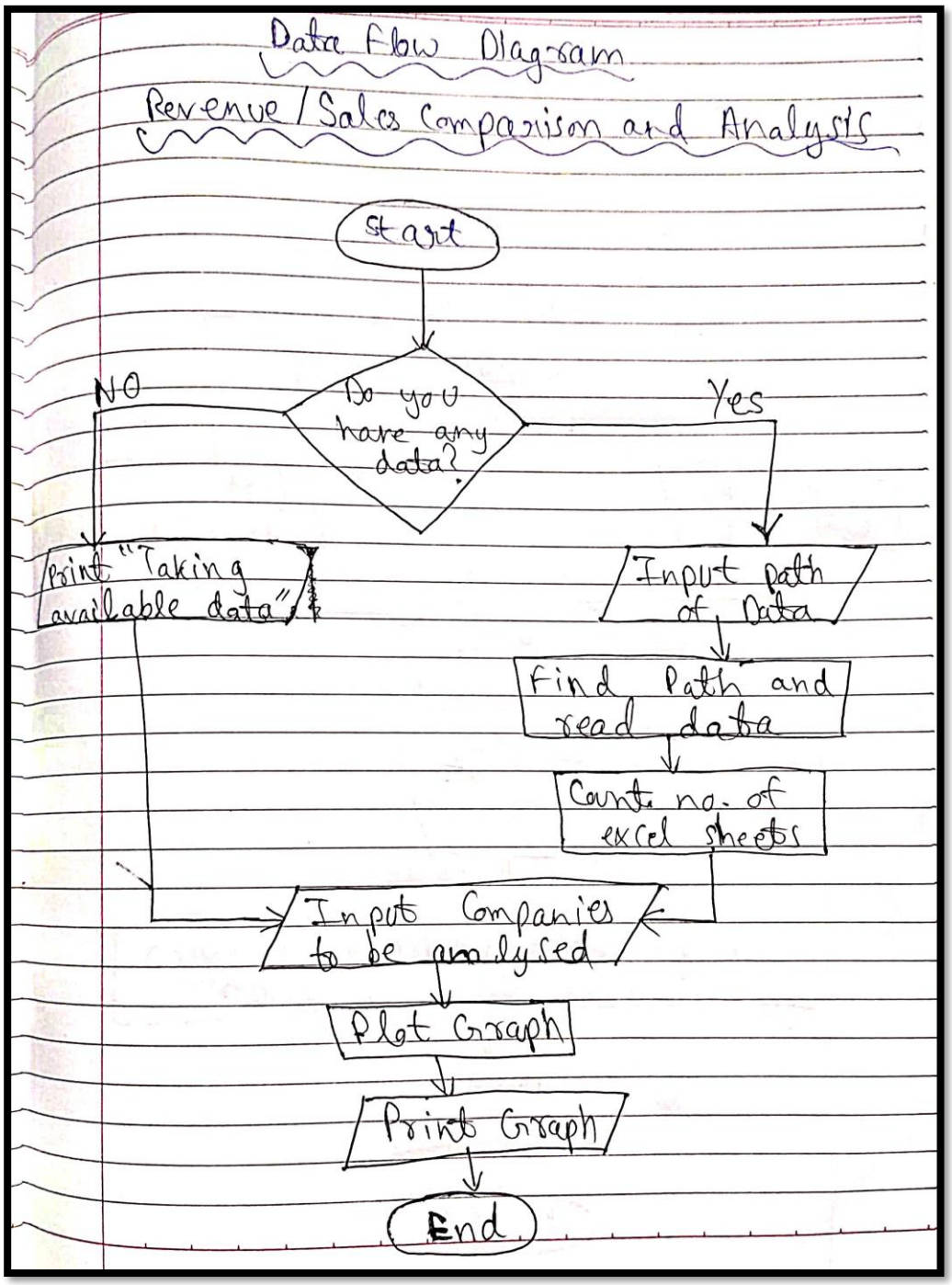
I HAVE USED MATPLOTLIB.PYPLOT MODULE OF MATPLOTLIB LIBRARY. IT IS USED FOR THE GRAPHICAL REPRESENTATION OF DATA, WHICH WAS THE MAIN OBJECTIVE OF MY PROJECT. IT ALSO HELPS TO REPRESENT MULTIPLE GRAPHS AT A TIME, WHICH PROVED TO BE VERY USEFUL FOR THE COMPARISON OF DIFFERENT COMPANIES.



EXAMPLE FROM MY PROGRAM

DATA FLOW DIAGRAM

A DATA FLOW DIAGRAM IS A LOGICAL FLOW OF THE CONTROL IN THE PROGRAM AND REPRESENTS THE FUNCTIONING OF A GRAPH. MY DATA FLOW DIAGRAM IS GIVEN BELOW.



SOURCE CODE

main.py

```
'''NAME : ANURAG KAVI SHANKAR TAHKUR
CLAS: XI A,  ROLL. NO : 6,
DAV INTERNATIONAL SCHOOL
COMPUTER SCIENCE PROJECT (2021-2022)'''

'''PROCESS TO INTALL A LIBRARY
1. RUN CMD AS ADMINISTRATOR
2. TYPE : "pip install <name of library>"
3. PRESS ENTER, THEN WAIT FOR IT TO INSTALL.'''

from tkinter import *
import pandas as pd
import matplotlib.pyplot as p
import os


l=[]
l_2=[]

root = Tk()
root.title("Excel Sheet")
root.geometry("600x350")

#Warning
label = Label(text = "**IF YOU DON'T HAVE ANY DATA, WE CAN PROVIDE YOU WITH THE AVAILABLE DATA**" ,bg= "white",padx = 50,font="arial 10 bold")
label.pack(side = BOTTOM)

#Question

quest= Label(text = "DO YOU HAVE ANY DATA OF YOUR OWN?", fg = "black", font = "arial 15 bold")
quest.pack(side=TOP, pady=10)

def click(event):
    text=event.widget.cget("text")
    l.append(text)
    if (l[0]=='YES'):
        print('''Your Excel Sheet must be of this form
```

main.py

```
def click(event):
    text=event.widget.cget("text")
    l.append(text)
    if (l[0]=='YES'):
        print('''Your Excel Sheet must be of this form
    ----------------------------------------
    ROW    TITLE ''')
        data=(pd.read_excel('Sales data.xlsx'))
        print(pd.DataFrame(data))


    elif (l[0]=="NO"):
        print('We have Sales Data of three comapnies: MICROSOFT, META, ALPHABET')
        print()
        comp=int(input('''Which one do you want
        1. MICROSOFT- *(TYPE 1)*
        1. ALPHABET- *(TYPE 2)*
        1. META- *(TYPE 3)*
        INPUT HERE : '''))

def yes():
    print(" SO, YOU HAVE CHOSEN TO INPUT YOUR DATA")
    print()
    print('''Your Excel Sheet must be of this form
----------------------------------------
ROW    TITLE ''')
    data=(pd.read_excel('Sales data.xlsx'))
    print(pd.DataFrame(data))
    root.destroy()
    print()
    print("***NOTE: THE PROGRAM CAN COMPARE ONLY 5 COMPANIES AT A TIME***")
    print()
    confirm=int(input("Enter 0 to continue after reading the instructions :"))
    if confirm!=0:
        print("Oops! You will have to run the program again and follow thw steps properly")
        print()
```

```
    if confirm!=0:
        print("Oops! You will have to run the program again and follow thw steps properly")
        print()
        print()
    else:
        path_root=Tk()
        path_root.title("PATH")
        path_root.geometry("600x400")

        #instruction
        inst= Label(path_root, text = '''***SAVE YOUR FILE IN THE GIVEN DIRECTORY
        'C:\\Users\\hp\\OneDrive - DAV INTERNATIONAL SCHOOL, KHARGHAR (1)\\Computer science\\Programs'
        PLEASE MAKE SURE THAT YOUR SHEET NAME AND HEADER AT 0 IS SAME AND HEAD AT 1 IS ANALYSIS ''', bg="white", fg="red", font="areal 9 bold")
        inst.pack(side=BOTTOM, pady=20)
        quest_2= Label(path_root, text="Enter name of your file with .xlsx at the end ", fg="black", font="areal 13")
        quest_2.pack(side=TOP,pady=20)
        print()




    #taking all the inputs from user
    def okay():
        print(" Path Taken. Now searching for data.....")
        path1=p_entry.get()
        path_root.destroy()
        assert os.path.exists(path1), " I did not find your file at, "+str(path1)
        f=open(path1)
        print("Hooray! We found your file.")

        #Now reading the file
        file=pd.ExcelFile(path1)
        sheets_1=file.sheet_names
        if len(sheets_1)==1:
            sheet_root=Tk()
            sheet_root.title("SHEET NAME")
            sheet_root.geometry("700x500")
```

```
            sheet_root.title("SHEET NAME")
            sheet_root.geometry("700x500")

            quest_4= Label(text = "IN WHICH COMAPNIES REVENUE ARE YOU INTERESTED IN?", fg = "black", font = "arial 15 bold")
            quest_4.pack(side=TOP, pady=10)

            def micro():
                print("Looks like you are interested in", sheets_1[0])
                print()

                #graph
                #using pandas to get specific rows now
                #x axis
                y_1=pd.read_excel(path1, sheet_name=sheets_1[0])
                year=y_1[sheets_1[0]].tolist()
                year=year[::-1]
                year.pop()

                #y axis
                r_1=pd.read_excel(path1, sheet_name=sheets_1[0])
                revenue=r_1["ANALYSIS"].tolist()
                revenue=revenue[::-1]
                revenue.pop()

                #making graph
                x=year
                y=revenue
                p.plot(x,y, color="deepskyblue", linewidth=2, marker='.',markersize=15, label=sheets_1[0])
                p.xlabel("YEAR")
                p.ylabel("REVENUE (BILLIONS OF US $)")
                p.title("REVENUE TREND")
                p.legend()
                p.show()


            def ex():
```

```
135
136             def ex():
137                 sheet_root.destroy()
138                 print('''THANK YOU
139         VISIT AGAIN''')
140
141             m= Button(sheet_root, text = sheets_1[0], padx= 40 , pady =20 ,command =micro , font="areal 15", bg="deepskyblue")
142             m.pack(side=LEFT,padx=25)
143             e= Button(sheet_root, text = " EXIT ",padx= 25 , pady =20 ,command =ex, font="areal 20", fg="white", bg="black")
144             e.pack(side=BOTTOM, pady=15)
145
146         if len(sheets_1)==2:
147             sheet_root=Tk()
148             sheet_root.title("SHEET NAME")
149             sheet_root.geometry("700x500")
150
151             quest_4= Label(text = "IN WHICH COMAPNIES REVENUE ARE YOU INTERESTED IN?", fg = "black", font = "arial 15 bold")
152             quest_4.pack(side=TOP, pady=10)
153
154             def micro():
155                 print("Looks like you are interested in", sheets_1[0])
156                 print()
157
158                 #graph
159                 #using pandas to get specific rows now
160                 #x axis
161                 y_1=pd.read_excel(path1, sheet_name=sheets_1[0])
162                 year=y_1[sheets_1[0]].tolist()
163                 year=year[::-1]
164                 year.pop()
165
166                 #y axis
167                 r_1=pd.read_excel(path1, sheet_name=sheets_1[0])
168                 revenue=r_1["ANALYSIS"].tolist()
169                 revenue=revenue[::-1]
170                 revenue.pop()
```

```
169                 revenue=revenue[::-1]
170                 revenue.pop()
171
172                 #making graph
173                 x=year
174                 y=revenue
175                 p.plot(x,y, color="deepskyblue", linewidth=2, marker='.',markersize=15, label=sheets_1[0])
176                 p.xlabel("YEAR")
177                 p.ylabel("REVENUE (BILLIONS OF US $)")
178                 p.title("REVENUE TREND")
179                 p.legend()
180                 p.show()
181
182
183             def alpha():
184                 print("Looks like you are interested in", sheets_1[1])
185                 print()
186
187                 #graph
188                 #using pandas to get specific rows now
189                 #x axis
190                 y_1=pd.read_excel(path1, sheet_name=sheets_1[1])
191                 year=y_1[sheets_1[1]].tolist()
192                 year=year[::-1]
193                 year.pop()
194                 #y axis
195                 r_1=pd.read_excel(path1, sheet_name=sheets_1[1])
196                 revenue=r_1["ANALYSIS"].tolist()
197                 revenue=revenue[::-1]
198                 revenue.pop()
199
200                 #making graph
201                 x=year
202                 y=revenue
203                 p.plot(x,y, color="greenyellow",linewidth=2, marker='.',markersize=15, label=sheets_1[1])
204                 p.xlabel("YEAR")
205                 p.ylabel("REVENUE (BILLIONS OF US $)")
```

```
            p.xlabel("YEAR")
            p.ylabel("REVENUE (BILLIONS OF US $)")
            p.title("REVENUE TREND")
            p.legend()
            p.show()


        def ex():
            sheet_root.destroy()
            print('''THANK YOU
    VISIT AGAIN''')

        m= Button(sheet_root, text = sheets_1[0], padx= 40 , pady =20 ,command =micro , font="areal 15", bg="deepskyblue")
        m.pack(side=LEFT,padx=25)
        a= Button(sheet_root, text = sheets_1[1], padx= 40 , pady =20 ,command =alpha, font="areal 15", bg="yellowgreen")
        a.pack(side=RIGHT,padx=25)
        e= Button(sheet_root, text = " EXIT ",padx= 25 , pady =20 ,command =ex, font="areal 20", fg="white", bg="black")
        e.pack(side=BOTTOM, pady=15)

    if len(sheets_1)==3:
        sheet_root=Tk()
        sheet_root.title("SHEET NAME")
        sheet_root.geometry("700x500")

        quest_4= Label(text = "IN WHICH COMAPNIES REVENUE ARE YOU INTERESTED IN?", fg = "black", font = "arial 15 bold")
        quest_4.pack(side=TOP, pady=10)

        def micro():
            print("Looks like you are interested in", sheets_1[0])
            print()

            #graph
            #using pandas to get specific rows now
            #x axis
            y_1=pd.read_excel(path1, sheet_name=sheets_1[0])
            year=y_1[sheets_1[0]].tolist()
```

```
            #x axis
            y_1=pd.read_excel(path1, sheet_name=sheets_1[0])
            year=y_1[sheets_1[0]].tolist()
            year=year[::-1]
            year.pop()

            #y axis
            r_1=pd.read_excel(path1, sheet_name=sheets_1[0])
            revenue=r_1["ANALYSIS"].tolist()
            revenue=revenue[::-1]
            revenue.pop()

            #making graph
            x=year
            y=revenue
            p.plot(x,y, color="deepskyblue",linewidth=2, marker='.',markersize=15, label=sheets_1[0])
            p.xlabel("YEAR")
            p.ylabel("REVENUE (BILLIONS OF US $)")
            p.title("REVENUE TREND")
            p.legend()
            p.show()


        def alpha():
            print("Looks like you are interested in", sheets_1[1])
            print()

            #graph
            #using pandas to get specific rows now
            #x axis
            y_1=pd.read_excel(path1, sheet_name=sheets_1[1])
            year=y_1[sheets_1[1]].tolist()
            year=year[::-1]
            year.pop()
            #y axis
            r_1=pd.read_excel(path1, sheet_name=sheets_1[1])
            revenue=r_1["ANALYSIS"].tolist()
```

```
                #y axis
                r_1=pd.read_excel(path1, sheet_name=sheets_1[1])
                revenue=r_1["ANALYSIS"].tolist()
                revenue=revenue[::-1]
                revenue.pop()

                #making graph
                x=year
                y=revenue
                p.plot(x,y, color="greenyellow",linewidth=2, marker='.',markersize=15,label=sheets_1[1])
                p.xlabel("YEAR")
                p.ylabel("REVENUE (BILLIONS OF US $)")
                p.title("REVENUE TREND")
                p.legend()
                p.show()


            def met():
                print("Looks like you are interested in", sheets_1[2])
                print()

                #graph
                #using openpyxl to get specific rows now
                y_1=pd.read_excel(path1, sheet_name=sheets_1[2])
                year=y_1[sheets_1[2]].tolist()
                year=year[::-1]
                year.pop()
                #y axis
                r_1=pd.read_excel(path1, sheet_name=sheets_1[2])
                revenue=r_1["ANALYSIS"].tolist()
                revenue=revenue[::-1]
                revenue.pop()

                #making graph
                x=year
                y=revenue
```

```
                #making graph
                x=year
                y=revenue
                p.plot(x,y, color="coral",linewidth=2, marker='.',markersize=15,label=sheets_1[2])
                p.xlabel("YEAR")
                p.ylabel("REVENUE (BILLIONS OF US $)")
                p.title("REVENUE TREND")
                p.legend()
                p.show()


            def ex():
                sheet_root.destroy()
                print('''THANK YOU
        VISIT AGAIN''')

            m= Button(sheet_root, text = sheets_1[0], padx= 40 , pady =20 ,command =micro , font="areal 15", bg="deepskyblue")
            m.pack(side=LEFT,padx=25)
            a= Button(sheet_root, text = sheets_1[1], padx= 40 , pady =20 ,command =alpha, font="areal 15", bg="yellowgreen")
            a.pack(side=RIGHT,padx=25)
            e= Button(sheet_root, text = " EXIT ",padx= 25 , pady =20 ,command =ex, font="areal 20", fg="white", bg="black")
            e.pack(side=BOTTOM, pady=15)
            me= Button(sheet_root, text = sheets_1[2], padx= 40 , pady= 20 ,command =met, font="areal 15", bg="coral")
            me.pack(side=BOTTOM, pady=62)

        if len(sheets_1)==4:
            sheet_root=Tk()
            sheet_root.title("SHEET NAME")
            sheet_root.geometry("700x500")

            quest_4= Label(text = "IN WHICH COMAPNIES REVENUE ARE YOU INTERESTED IN?", fg = "black", font = "arial 15 bold")
            quest_4.pack(side=TOP, pady=10)

            def micro():
                print("Looks like you are interested in", sheets_1[0])
                print()
```

```
337             def micro():
338                 print("Looks like you are interested in", sheets_1[0])
339                 print()
340 
341                 #graph
342                 #using pandas to get specific rows now
343                 #x axis
344                 y_1=pd.read_excel(path1, sheet_name=sheets_1[0])
345                 year=y_1[sheets_1[0]].tolist()
346                 year=year[::-1]
347                 year.pop()
348 
349                 #y axis
350                 r_1=pd.read_excel(path1, sheet_name=sheets_1[0])
351                 revenue=r_1["ANALYSIS"].tolist()
352                 revenue=revenue[::-1]
353                 revenue.pop()
354 
355                 #making graph
356                 x=year
357                 y=revenue
358                 p.plot(x,y, color="deepskyblue",linewidth=2, marker='.',markersize=15, label=sheets_1[0])
359                 p.xlabel("YEAR")
360                 p.ylabel("REVENUE (BILLIONS OF US $)")
361                 p.title("REVENUE TREND")
362                 p.legend()
363                 p.show()
364 
365 
366             def alpha():
367                 print("Looks like you are interested in", sheets_1[1])
368                 print()
369 
370                 #graph
371                 #using pandas to get specific rows now
372                 #x axis
373                 y_1=pd.read_excel(path1, sheet_name=sheets_1[1])
```

```
370                 #graph
371                 #using pandas to get specific rows now
372                 #x axis
373                 y_1=pd.read_excel(path1, sheet_name=sheets_1[1])
374                 year=y_1[sheets_1[1]].tolist()
375                 year=year[::-1]
376                 year.pop()
377                 #y axis
378                 r_1=pd.read_excel(path1, sheet_name=sheets_1[1])
379                 revenue=r_1["ANALYSIS"].tolist()
380                 revenue=revenue[::-1]
381                 revenue.pop()
382 
383                 #making graph
384                 x=year
385                 y=revenue
386                 p.plot(x,y, color="greenyellow",linewidth=2, marker='.',markersize=15,label=sheets_1[1])
387                 p.xlabel("YEAR")
388                 p.ylabel("REVENUE (BILLIONS OF US $)")
389                 p.title("REVENUE TREND")
390                 p.legend()
391                 p.show()
392 
393 
394             def met():
395                 print("Looks like you are interested in", sheets_1[2])
396                 print()
397 
398                 #graph
399                 #using openpyxl to get specific rows now
400                 y_1=pd.read_excel(path1, sheet_name=sheets_1[2])
401                 year=y_1[sheets_1[2]].tolist()
402                 year=year[::-1]
403                 year.pop()
404                 #y axis
405                 r_1=pd.read_excel(path1, sheet_name=sheets_1[2])
```

```
404                 #y axis
405                 r_1=pd.read_excel(path1, sheet_name=sheets_1[2])
406                 revenue=r_1["ANALYSIS"].tolist()
407                 revenue=revenue[::-1]
408                 revenue.pop()
409
410                 #making graph
411                 x=year
412                 y=revenue
413                 p.plot(x,y, color="coral",linewidth=2, marker='.',markersize=15,label=sheets_1[2])
414                 p.xlabel("YEAR")
415                 p.ylabel("REVENUE (BILLIONS OF US $)")
416                 p.title("REVENUE TREND")
417                 p.legend()
418                 p.show()
419
420
421             def beta():
422                 print("Looks like you are interested in", sheets_1[3])
423                 print()
424
425                 #graph
426                 #using openpyxl to get specific rows now
427                 y_1=pd.read_excel(path1, sheet_name=sheets_1[3])
428                 year=y_1[sheets_1[3]].tolist()
429                 year=year[::-1]
430                 year.pop()
431                 #y axis
432                 r_1=pd.read_excel(path1, sheet_name=sheets_1[3])
433                 revenue=r_1["ANALYSIS"].tolist()
434                 revenue=revenue[::-1]
435                 revenue.pop()
436
437                 #making graph
438                 x=year
439                 y=revenue
440                 p.plot(x,y, color="slategray",linewidth=2, marker='.',markersize=15,label=sheets_1[3])
441                 p.xlabel("YEAR")
442                 p.ylabel("REVENUE (BILLIONS OF US $)")
443                 p.title("REVENUE TREND")
444                 p.legend()
445                 p.show()
446
447
448             def ex():
449                 sheet_root.destroy()
450                 print('''THANK YOU
451         VISIT AGAIN''')
452
453             m= Button(sheet_root, text = sheets_1[0], padx= 40 , pady =20 ,command =micro , font="areal 15", bg="deepskyblue")
454             m.pack(side=LEFT,padx=25)
455             a= Button(sheet_root, text = sheets_1[1], padx= 40 , pady =20 ,command =alpha, font="areal 15", bg="yellowgreen")
456             a.pack(side=RIGHT,padx=25)
457             e= Button(sheet_root, text = " EXIT ",padx= 25 , pady =20 ,command =ex, font="areal 20", fg="white", bg="black")
458             e.pack(side=BOTTOM, pady=15)
459             me= Button(sheet_root, text = sheets_1[2], padx= 40 , pady= 20 ,command =met, font="areal 15", bg="coral")
460             me.pack(side=BOTTOM, pady=62)
461             n= Button(sheet_root, text = sheets_1[3], padx= 40 , pady =20 ,command =beta, font="areal 15", bg="slategray")
462             n.pack(side=RIGHT,padx=25)
463
464         if len(sheets_1)==5:
465             sheet_root=Tk()
466             sheet_root.title("SHEET NAME")
467             sheet_root.geometry("700x600")
468
469             quest_4= Label(text = "IN WHICH COMAPNIES REVENUE ARE YOU INTERESTED IN?", fg = "black", font = "arial 15 bold")
470             quest_4.pack(side=TOP, pady=10)
471
472             def micro():
```

```
472         def micro():
473             print("Looks like you are interested in", sheets_1[0])
474             print()
475 
476             #graph
477             #using pandas to get specific rows now
478             #x axis
479             y_1=pd.read_excel(path1, sheet_name=sheets_1[0])
480             year=y_1[sheets_1[0]].tolist()
481             year=year[::-1]
482             year.pop()
483 
484             #y axis
485             r_1=pd.read_excel(path1, sheet_name=sheets_1[0])
486             revenue=r_1["ANALYSIS"].tolist()
487             revenue=revenue[::-1]
488             revenue.pop()
489 
490             #making graph
491             x=year
492             y=revenue
493             p.plot(x,y, color="deepskyblue",linewidth=2, marker='.',markersize=15, label=sheets_1[0])
494             p.xlabel("YEAR")
495             p.ylabel("REVENUE (BILLIONS OF US $)")
496             p.title("REVENUE TREND")
497             p.legend()
498             p.show()
499 
500 
501         def alpha():
502             print("Looks like you are interested in", sheets_1[1])
503             print()
504 
505             #graph
506             #using pandas to get specific rows now
507             #x axis
```

```
506             #using pandas to get specific rows now
507             #x axis
508             y_1=pd.read_excel(path1, sheet_name=sheets_1[1])
509             year=y_1[sheets_1[1]].tolist()
510             year=year[::-1]
511             year.pop()
512             #y axis
513             r_1=pd.read_excel(path1, sheet_name=sheets_1[1])
514             revenue=r_1["ANALYSIS"].tolist()
515             revenue=revenue[::-1]
516             revenue.pop()
517 
518             #making graph
519             x=year
520             y=revenue
521             p.plot(x,y, color="greenyellow",linewidth=2, marker='.',markersize=15, label=sheets_1[1])
522             p.xlabel("YEAR")
523             p.ylabel("REVENUE (BILLIONS OF US $)")
524             p.title("REVENUE TREND")
525             p.legend()
526             p.show()
527 
528 
529         def met():
530             print("Looks like you are interested in", sheets_1[2])
531             print()
532 
533             #graph
534             #using openpyxl to get specific rows now
535             y_1=pd.read_excel(path1, sheet_name=sheets_1[2])
536             year=y_1[sheets_1[2]].tolist()
537             year=year[::-1]
538             year.pop()
539             #y axis
540             r_1=pd.read_excel(path1, sheet_name=sheets_1[2])
541             revenue=r_1["ANALYSIS"].tolist()
```

```
            r_1=pd.read_excel(path1, sheet_name=sheets_1[2])
            revenue=r_1["ANALYSIS"].tolist()
            revenue=revenue[::-1]
            revenue.pop()

            #making graph
            x=year
            y=revenue
            p.plot(x,y, color="coral",linewidth=2, marker='.',markersize=15,label=sheets_1[2])
            p.xlabel("YEAR")
            p.ylabel("REVENUE (BILLIONS OF US $)")
            p.title("REVENUE TREND")
            p.legend()
            p.show()


        def beta():
            print("Looks like you are interested in", sheets_1[2])
            print()

            #graph
            #using openpyxl to get specific rows now
            y_1=pd.read_excel(path1, sheet_name=sheets_1[3])
            year=y_1[sheets_1[3]].tolist()
            year=year[::-1]
            year.pop()
            #y axis
            r_1=pd.read_excel(path1, sheet_name=sheets_1[3])
            revenue=r_1["ANALYSIS"].tolist()
            revenue=revenue[::-1]
            revenue.pop()

            #making graph
            x=year
            y=revenue
            p.plot(x,y, color="slategray",linewidth=2, marker='.',markersize=15, label=sheets_1[3])
            p.xlabel("YEAR")
```

```
main.py > yes > okay
            p.plot(x,y, color="slategray",linewidth=2, marker='.',markersize=15, label=sheets_1[3])
            p.xlabel("YEAR")
            p.ylabel("REVENUE (BILLIONS OF US $)")
            p.title("REVENUE TREND")
            p.legend()
            p.show()


        def gama():
            print("Looks like you are interested in", sheets_1[4])
            print()

            #graph
            #using openpyxl to get specific rows now
            y_1=pd.read_excel(path1, sheet_name=sheets_1[4])
            year=y_1[sheets_1[4]].tolist()
            year=year[::-1]
            year.pop()
            #y axis
            r_1=pd.read_excel(path1, sheet_name=sheets_1[4])
            revenue=r_1["ANALYSIS"].tolist()
            revenue=revenue[::-1]
            revenue.pop()

            #making graph
            x=year
            y=revenue
            p.plot(x,y, color="magenta",linewidth=2, marker='.',markersize=15,label=sheets_1[4])
            p.xlabel("YEAR")
            p.ylabel("REVENUE (BILLIONS OF US $)")
            p.title("REVENUE TREND")
            p.legend()
            p.show()

        def ex():
            sheet_root.destroy()
```

```
609                 def ex():
610                     sheet_root.destroy()
611                     print('''THANK YOU
612             VISIT AGAIN''')
613 
614 
615                 e= Button(sheet_root, text = " EXIT ",padx= 30 , pady =20 ,command =ex, font="areal 20", fg="white", bg="black")
616                 e.pack(side=BOTTOM, padx=25, pady=15)
617                 me= Button(sheet_root, text = sheets_1[2], padx= 40 , pady= 20 ,command =met, font="areal 15", bg="coral")
618                 me.pack(side=RIGHT,padx=20, pady=20)
619                 n= Button(sheet_root, text = sheets_1[3], padx= 40 , pady =20 ,command =beta, font="areal 15", bg="slategray")
620                 n.pack(side=LEFT,padx=25, pady=30)
621                 n2= Button(sheet_root, text = sheets_1[4], padx= 40 , pady =20 ,command =gama, font="areal 15", bg="magenta")
622                 n2.pack(side=BOTTOM,padx=25, pady=30)
623                 m= Button(sheet_root, text = sheets_1[0], padx= 40 , pady =20 ,command =micro , font="areal 15", bg="deepskyblue")
624                 m.pack(side=TOP,pady=30)
625                 a= Button(sheet_root, text = sheets_1[1], padx= 40 , pady =20 ,command =alpha, font="areal 15", bg="yellowgreen")
626                 a.pack(side=BOTTOM,padx=35, pady=33)
627 
628 
629     p_entry=Entry(path_root, width=50)
630     p_entry.pack()
631     okay_1=Button(path_root, text="DONE", fg="black", bg="white", command=okay, font="areal 12")
632     okay_1.pack(pady=30)
633 
634 def no():
635     print(" TAKING AVAILABLE DATA TO PROCEED FROM HERE")
636     print()
637     root.destroy()
638 
639     #button for company
640     comp_root=Tk()
641     comp_root.title("COMPANIES")
642     comp_root.geometry("700x500")
643 
644     quest_3= Label(text = "IN WHICH COMAPNIES REVENUE ARE YOU INTERESTED IN?", fg = "black", font = "arial 15 bold")
645     quest_3.pack(side=TOP, pady=10)
```

```
644     quest_3= Label(text = "IN WHICH COMAPNIES REVENUE ARE YOU INTERESTED IN?", fg = "black", font = "arial 15 bold")
645     quest_3.pack(side=TOP, pady=10)
646 
647     def micro():
648         print("Looks like you are interested in MICROSOFT")
649         print()
650 
651         #graph
652         #using pandas to get specific rows now
653         #x axis
654         y_1=pd.read_excel('Sales data.xlsx', sheet_name="MICROSOFT")
655         year=y_1["MICROSOFT"].tolist()
656         year=year[::-1]
657         year.pop()
658         #y axis
659         r_1=pd.read_excel('Sales data.xlsx', sheet_name="MICROSOFT")
660         revenue=r_1["ANALYSIS"].tolist()
661         revenue=revenue[::-1]
662         revenue.pop()
663 
664         #making graph
665         x=year
666         y=revenue
667         p.plot(x,y, color="deepskyblue",linewidth=2, marker='.',markersize=15, label="MICROSOFT")
668         p.xlabel("YEAR")
669         p.ylabel("REVENUE (BILLIONS OF US $)")
670         p.title("REVENUE TREND")
671         p.legend()
672         p.show()
673 
674     def alpha():
675         print("Looks like you are interested in ALPHABET")
676         print()
677 
678         #graph
679         #using pandas to get specific rows now
```

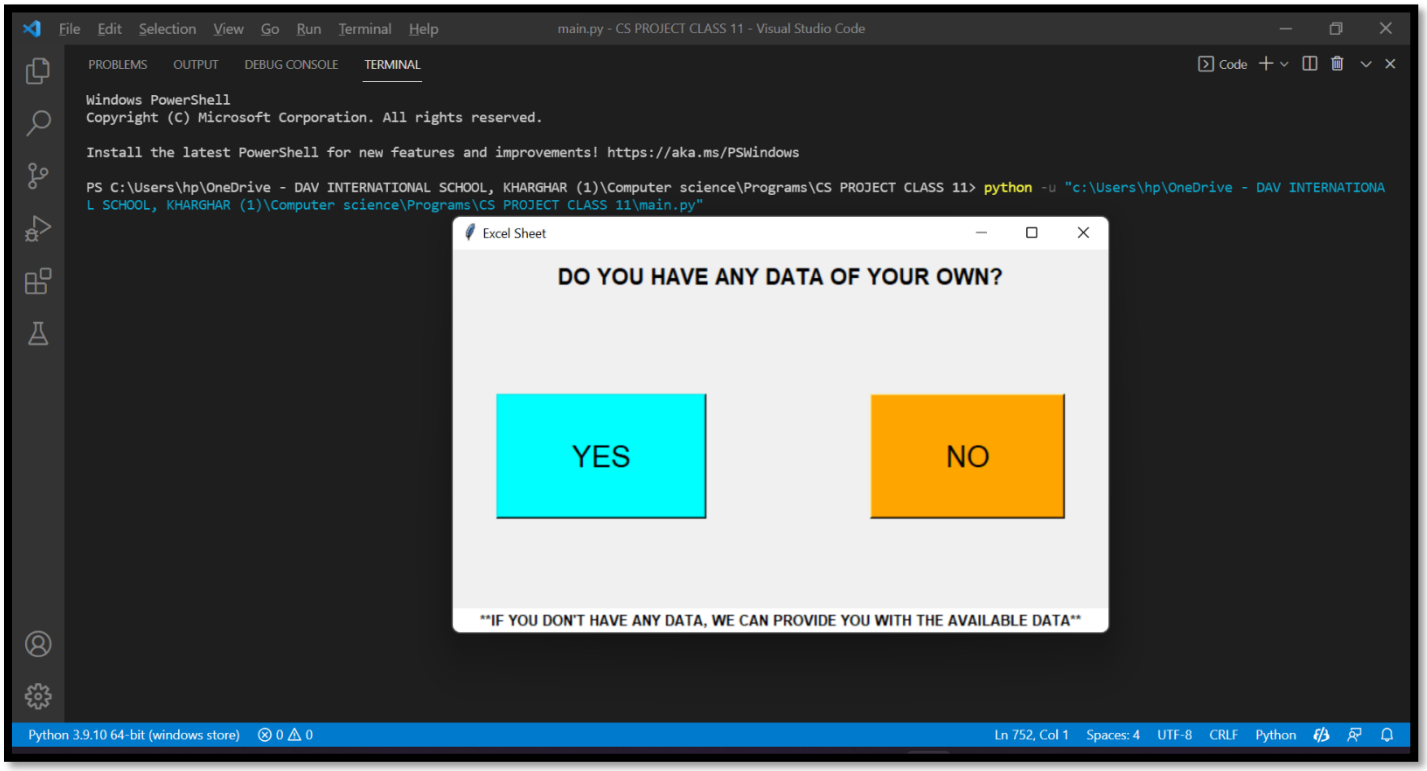
```
        #using pandas to get specific rows now
        #x axis
        y_1=pd.read_excel('Sales data.xlsx', sheet_name="ALPHABET")
        year=y_1["ALPHABET"].tolist()
        year=year[::-1]
        year.pop()
        #y axis
        r_1=pd.read_excel('Sales data.xlsx', sheet_name="ALPHABET")
        revenue=r_1["ANALYSIS"].tolist()
        revenue=revenue[::-1]
        revenue.pop()

        #making graph
        x=year
        y=revenue
        p.plot(x,y, color="greenyellow",linewidth=2, marker='.',markersize=15,label="ALPHABET")
        p.xlabel("YEAR")
        p.ylabel("REVENUE (BILLIONS OF US $)")
        p.title("REVENUE TREND")
        p.legend()
        p.show()

    def met():
        print("Looks like you are interested in META")
        print()

        #graph
        #using openpyxl to get specific rows now
        y_1=pd.read_excel('Sales data.xlsx', sheet_name="META")
        year=y_1["META"].tolist()
        year=year[::-1]
        year.pop()
        #y axis
        r_1=pd.read_excel('Sales data.xlsx', sheet_name="META")
        revenue=r_1["ANALYSIS"].tolist()
        revenue=revenue[::-1]
        revenue.pop()
```

```
        revenue.pop()

        #making graph
        x=year
        y=revenue
        p.plot(x,y, color="coral",linewidth=2, marker='.',markersize=15, label="META")
        p.xlabel("YEAR")
        p.ylabel("REVENUE (BILLIONS OF US $)")
        p.title("REVENUE TREND")
        p.legend()
        p.show()

    def ex():
        comp_root.destroy()
        print('''THANK YOU
        VISIT AGAIN''')

    m= Button(comp_root, text = " MICROSOFT ",padx= 40 , pady =20 ,command =micro , font="areal 15", bg="deepskyblue")
    m.pack(side=LEFT,padx=25)
    a= Button(comp_root, text = " ALPHABET ",padx= 40 , pady =20 ,command =alpha, font="areal 15", bg="yellowgreen")
    a.pack(side=RIGHT,padx=25)
    e= Button(comp_root, text = " EXIT ",padx= 25 , pady =20 ,command =ex, font="areal 20", fg="white", bg="black")
    e.pack(side=BOTTOM, pady=15)
    me= Button(comp_root, text = " META ",padx= 40 , pady= 20 ,command =met, font="areal 15", bg="coral")
    me.pack(side=BOTTOM, pady=62)

b1= Button(root, text = " YES ",padx= 50 , pady =30 ,command =yes, font="areal 20", bg="cyan")
b1.pack(side=LEFT,padx=40)
b1.bind("<Button-1>", click)

b2 = Button(root, text = " NO ",padx= 50 , pady = 30,command = no, font="areal 20", bg="orange")
b2.pack(side=RIGHT,padx=40)
b2.bind("<Button-1>", click)


root.mainloop()
```

```
root.mainloop()
print("DONE")
```

OUTPUT

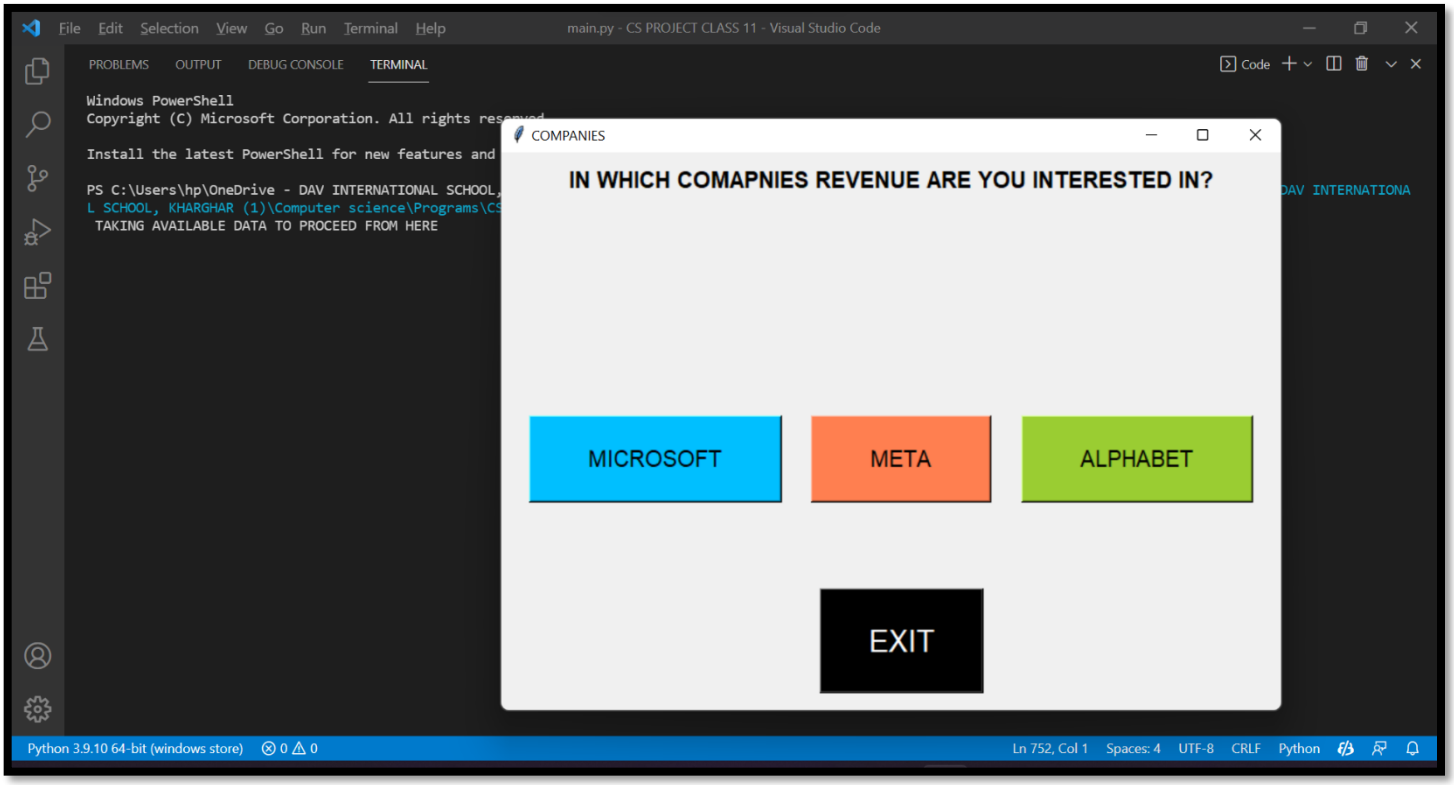
THE PROGRAM FIRST ASKS FOR THE USER IF THEY HAVE SOME DATA OR THEY ARE JUST TESTING THE PROGRAM.



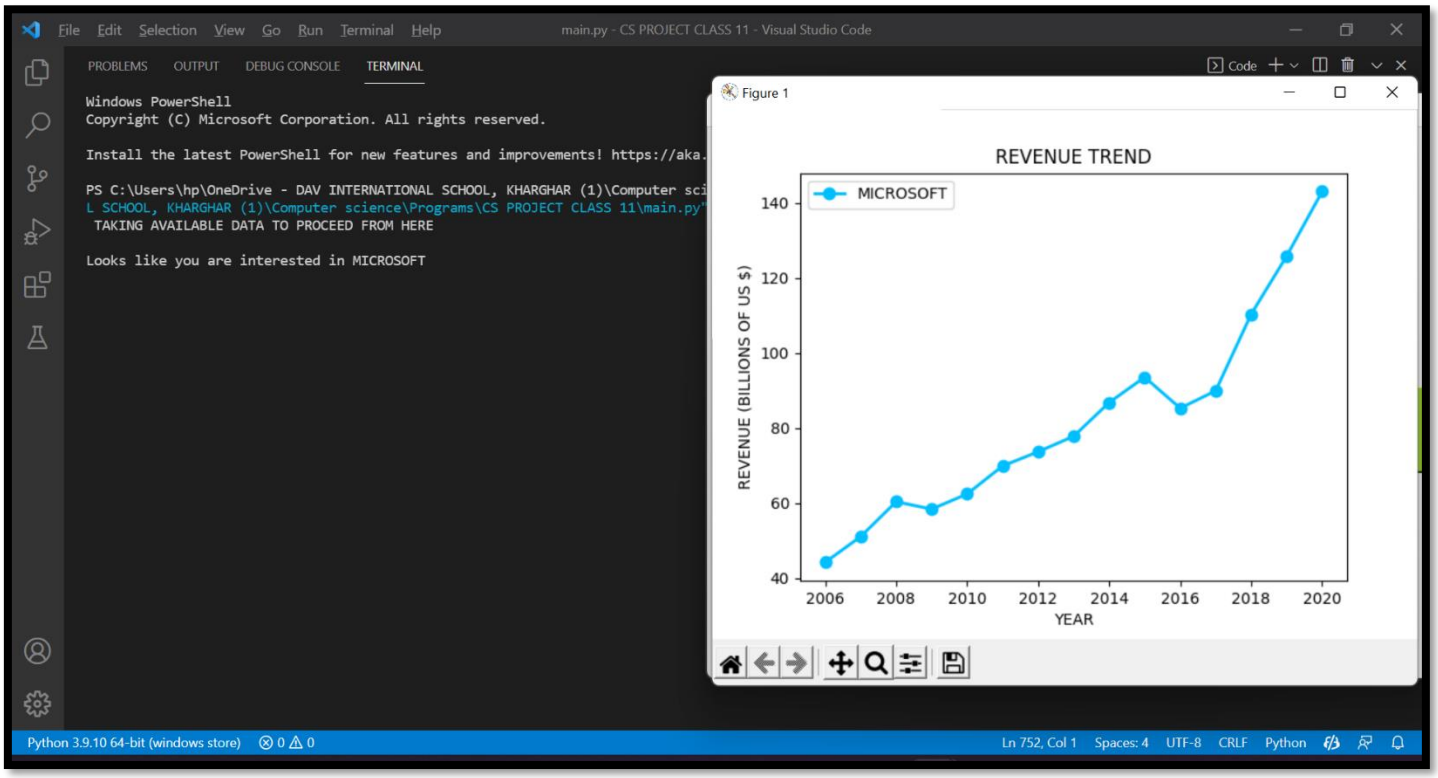
1. INPUTTING NO

NOW TO PROGRAM TAKES THE DATA AVAILABLE WITH HIM AND DISPLAYS THE COMPANY NAME. THE USER NOW CAN SELECT AN INDIVIDUAL OR MULTIPLE GRAPHS TO ANALYSE AND COMPARE.

HERE WE ARE SELECTING MICROSOFT FIRST.

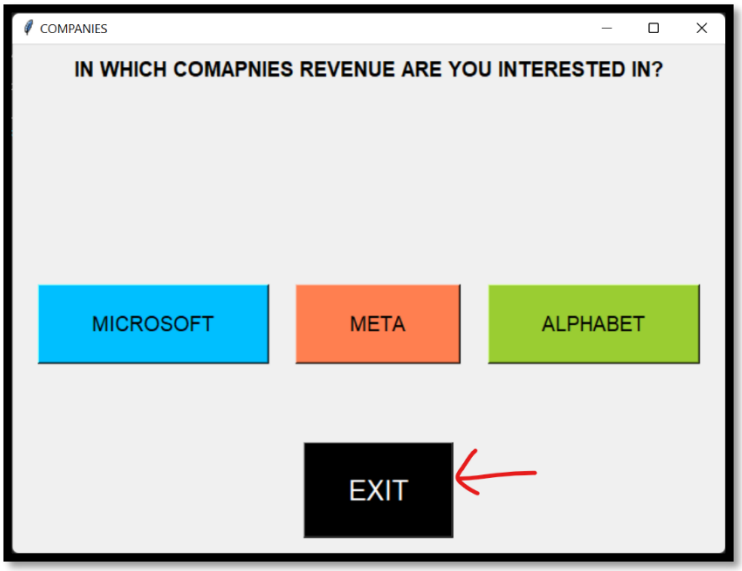


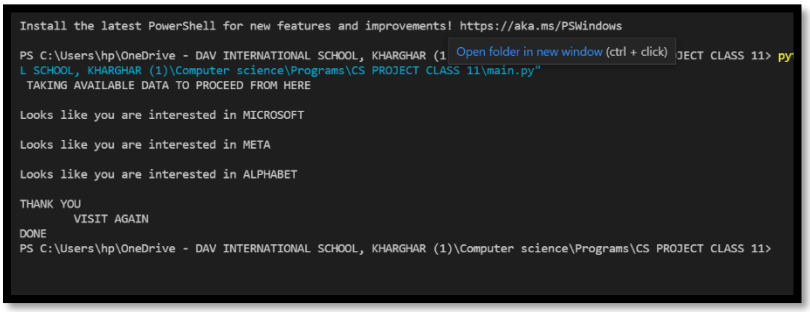
HERE WE ARE SELECTING MICROSFT FIRST AND THEN SELECTING ALL THE COMPANIES FOR COMPARISON.



FROM HERE WE CAN SEE THAT RATE OF GROWTH IS MAXIMUM IN ALPHABET AND THE HIGHEST REVENUE IN RECENT YEARS IS ALSO OF ALPHABET.

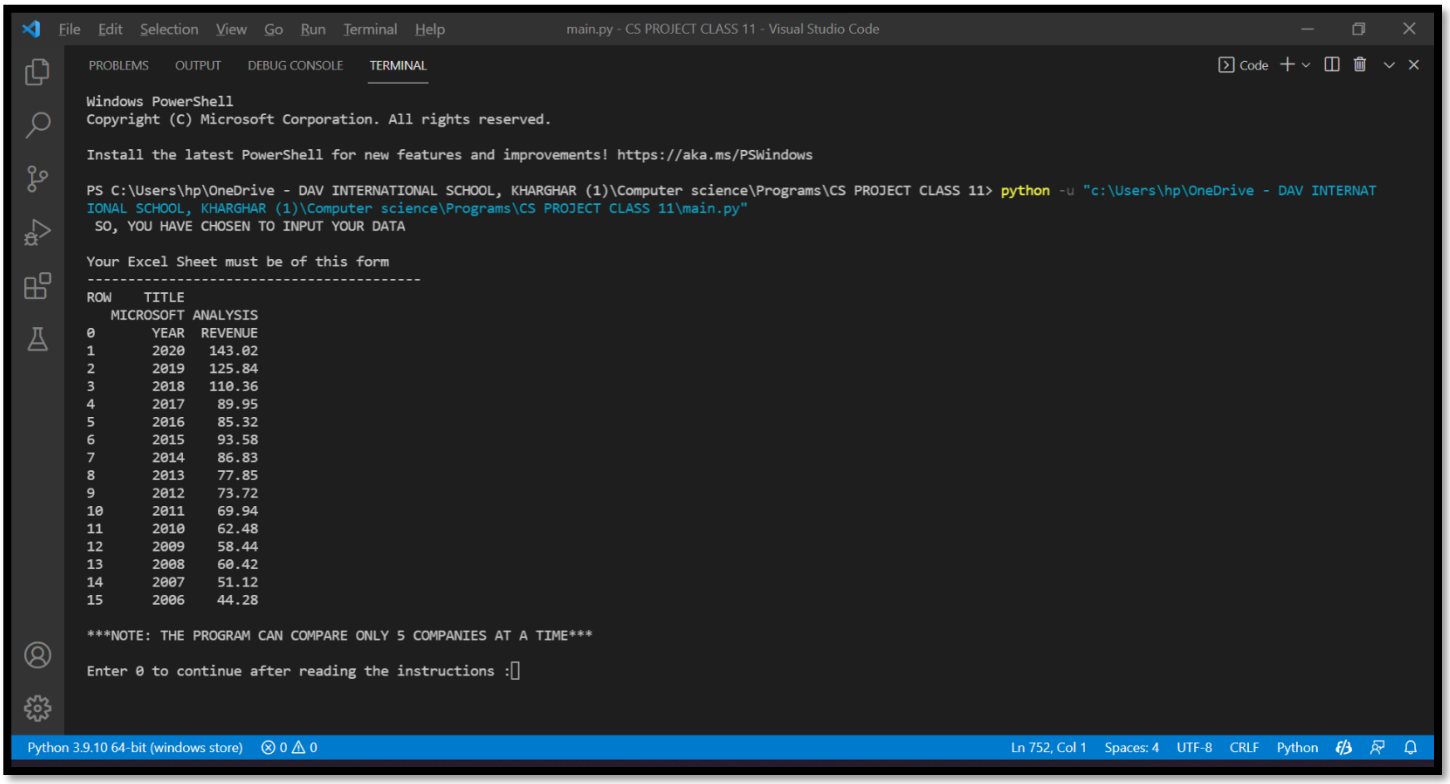
NOW CLICKING EXIT WILL CLOSE THE SCREEN AND THANK YOU WILL BE PRINTED.





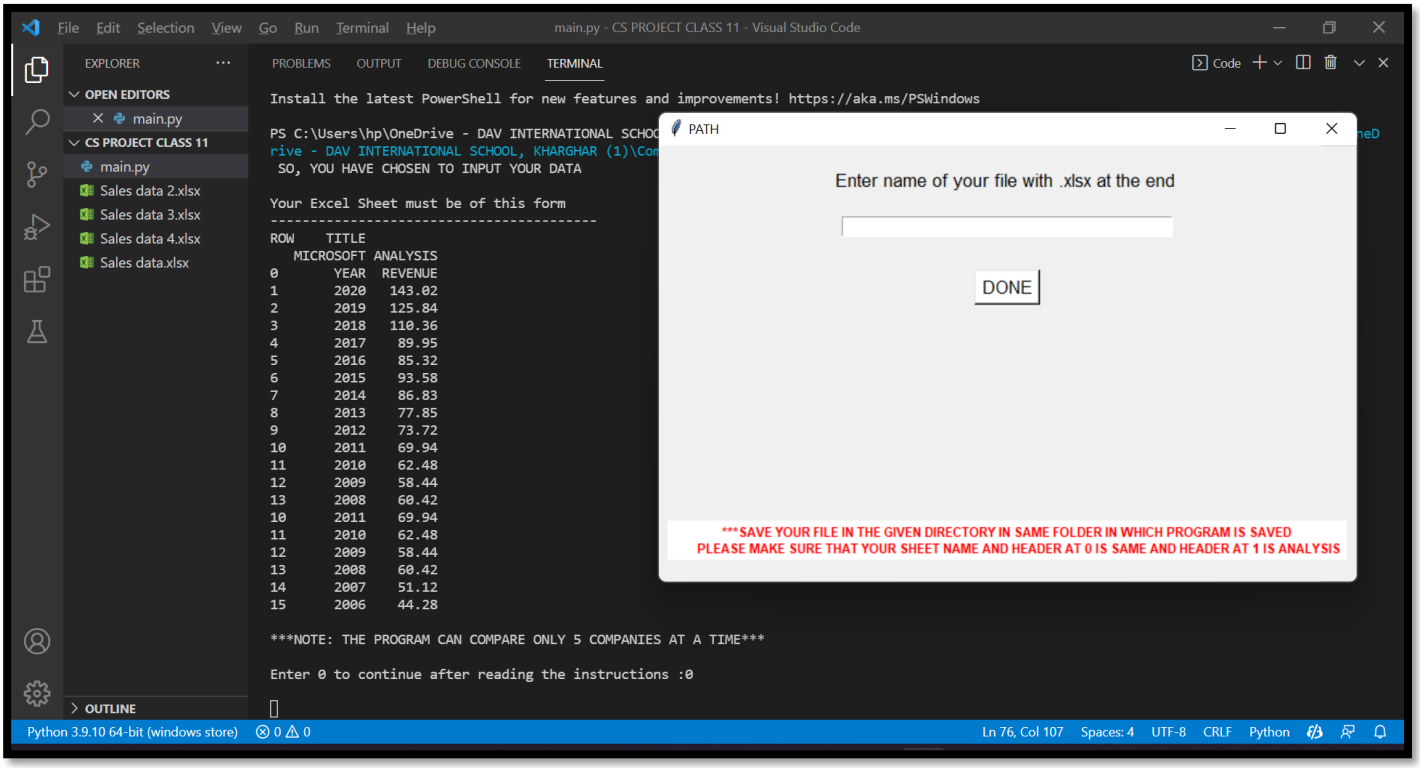
2. INPUTTING YES

IF WE CLICK YES, THAT MEANS WE HAVE OUR OWN DATA IN EXCEL SHEET, THE PROGRAM GIVES US SAMPLE DATA TO REFER TO WHILE MAKING A EXCEL SHEET FOR THIS PROGRAM. IT ALSO GIVES A WARNING TO NOT TO INPUT MORE THAN 5 COMPANIES.



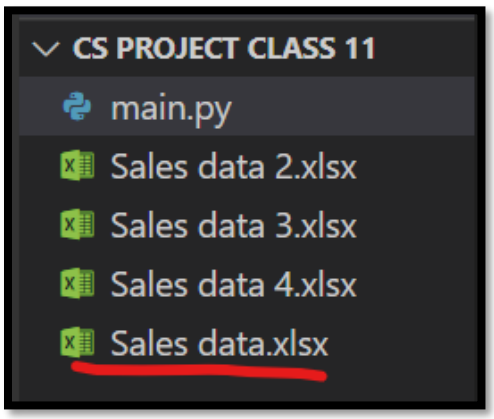
FROM HERE, WE ENTER 0 TO CONTINUE.

THIS TIME WE GET A SCREEN TO INPUT THE PATH OF THE EXCEL SHEET WITH AN IMPORTANT INSTRUCTION AT THE BELOW, WHICH IS TO BE FOLLOWED FOR THE PROGRAM TO RUN PERFECTLY.

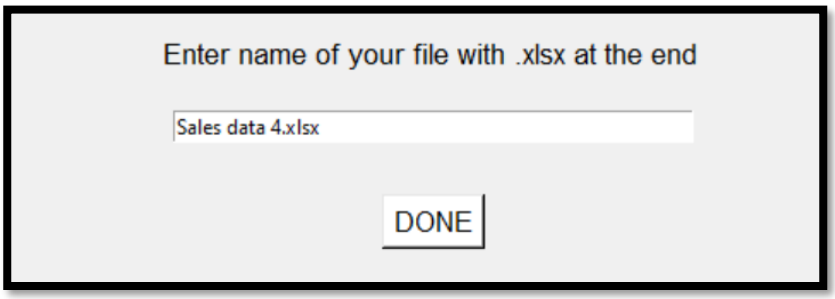


YOU CAN INPUT YOUR DATA BY FOLLOWING THE INSTRUCTIONS, ELSE **I HAVE GIVEN 3 DIFFERENT DATA SHEETS WITH THE PROGRAM IN THE GITHUB LINK AT THE END.** YOU CAN USE IT FOR TESTING THIS PART OF THE PROGRAM.

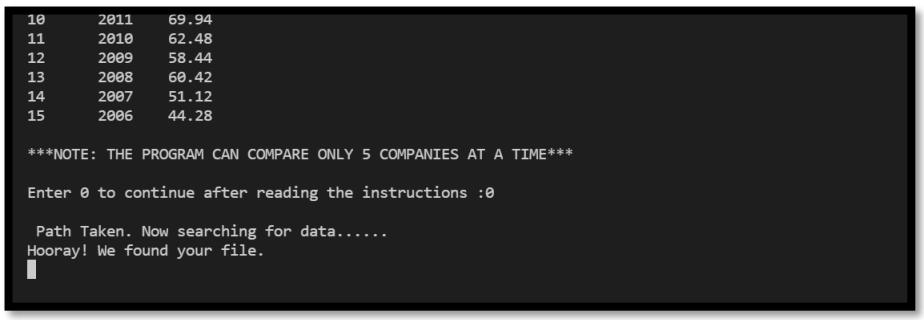
NOW FOR THE PROGRAM TO READ YOUR DATA, YOU WILL HAVE TO SAVE IT IN THE SAME FOLDER IN WHICH YOU HAVE SAVED THE PROGRAM. IF YOU WANT YOU CAN SAVE THESE THREE DATA ALSO, BUT **THE FILE 'Sales data.xlsx' IS MANDATORY TO SAVE.**



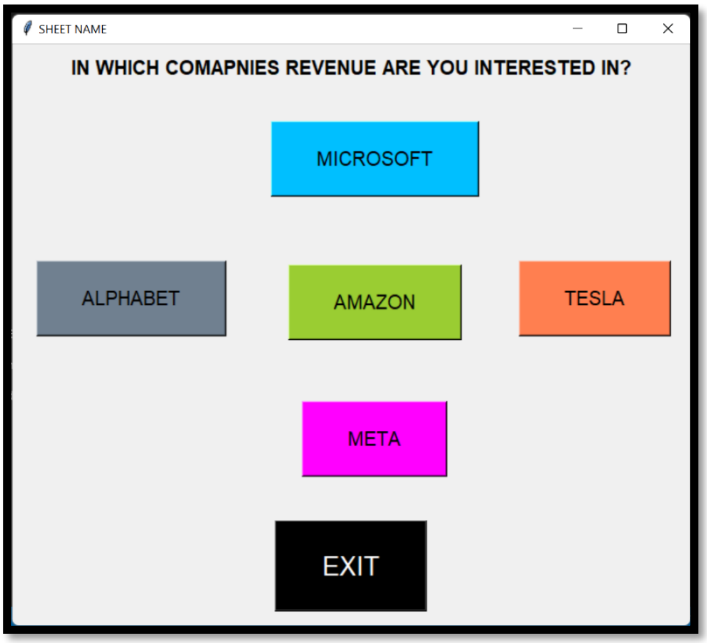
CHOOSING ONE RANDOM FILE FROM THESE 4, I AM SELECTING ‘Sales data 4.xlsx’.



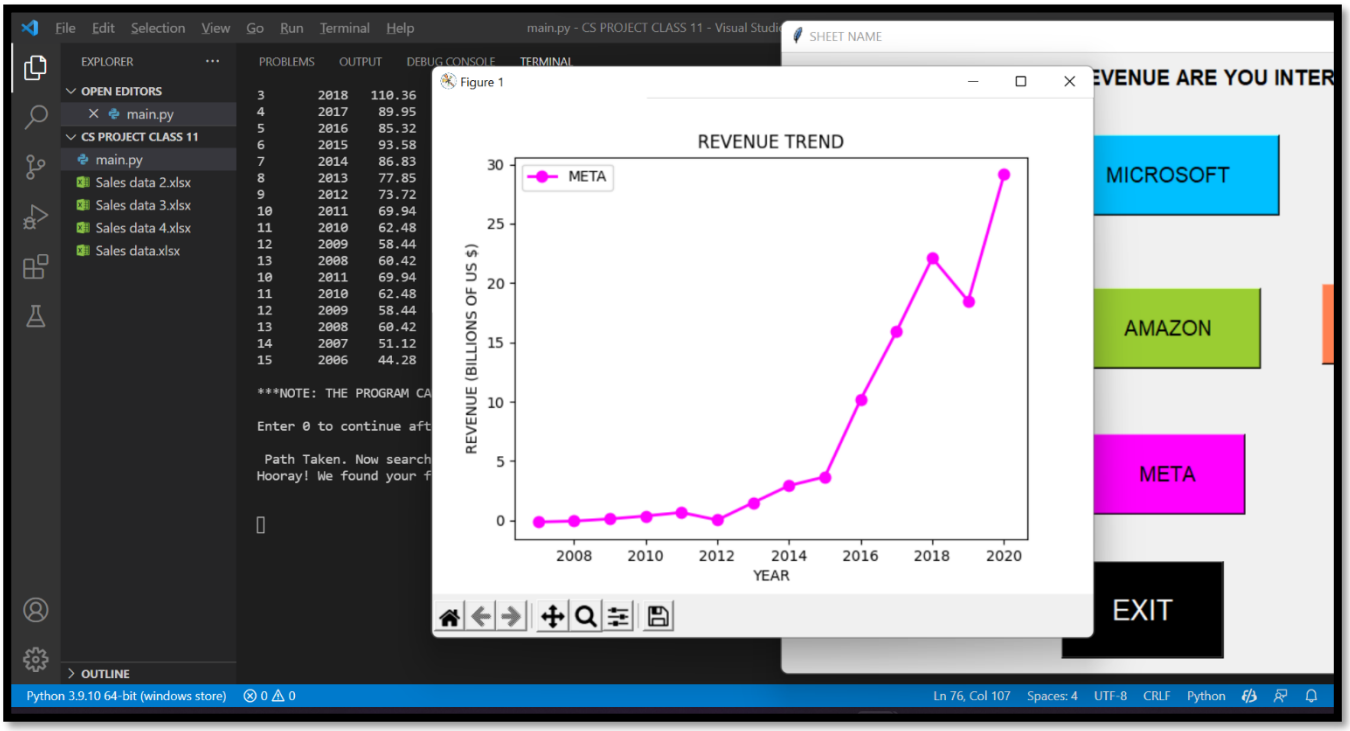
AS WE INPUT THE PATH, THE PROGRAM SEARCHES IT AND INFORMS US, IF IT WAS ABLE TO FIND IT OR NOT.



AS SOON AS IT FIND THE FILE, IT READS ALL THE SHEETS IN THE FILE AND ASKS US TO CHOOSE BETWEEN THE COMPANIES, WE CAN EVEN CHOOSE ALL FOR COMPARISON.



ACCORDING TO MY WISH, I AM SELECTING META.



THUS, WE GET THE GRAPH OF META'S REVENUE TREND IN PAST YEARS.

NOW IN ORDER TO COMPARE, I AM SELECTING ALL THE COMPANIES AND WE SEE A GRAPH IN WHICH ALL THE COMPANIES' REVENUE TRENDS ARE GIVEN AND WE CAN SEE THAT BOTH GROWTH RATES OVER THE YEARS AND REVENUE IN RECENT YEARS ARE DOMINATED BY AMAZON. SHOWING A LARGE DIFFERENCE FROM OTHERS IN COMPARISON.

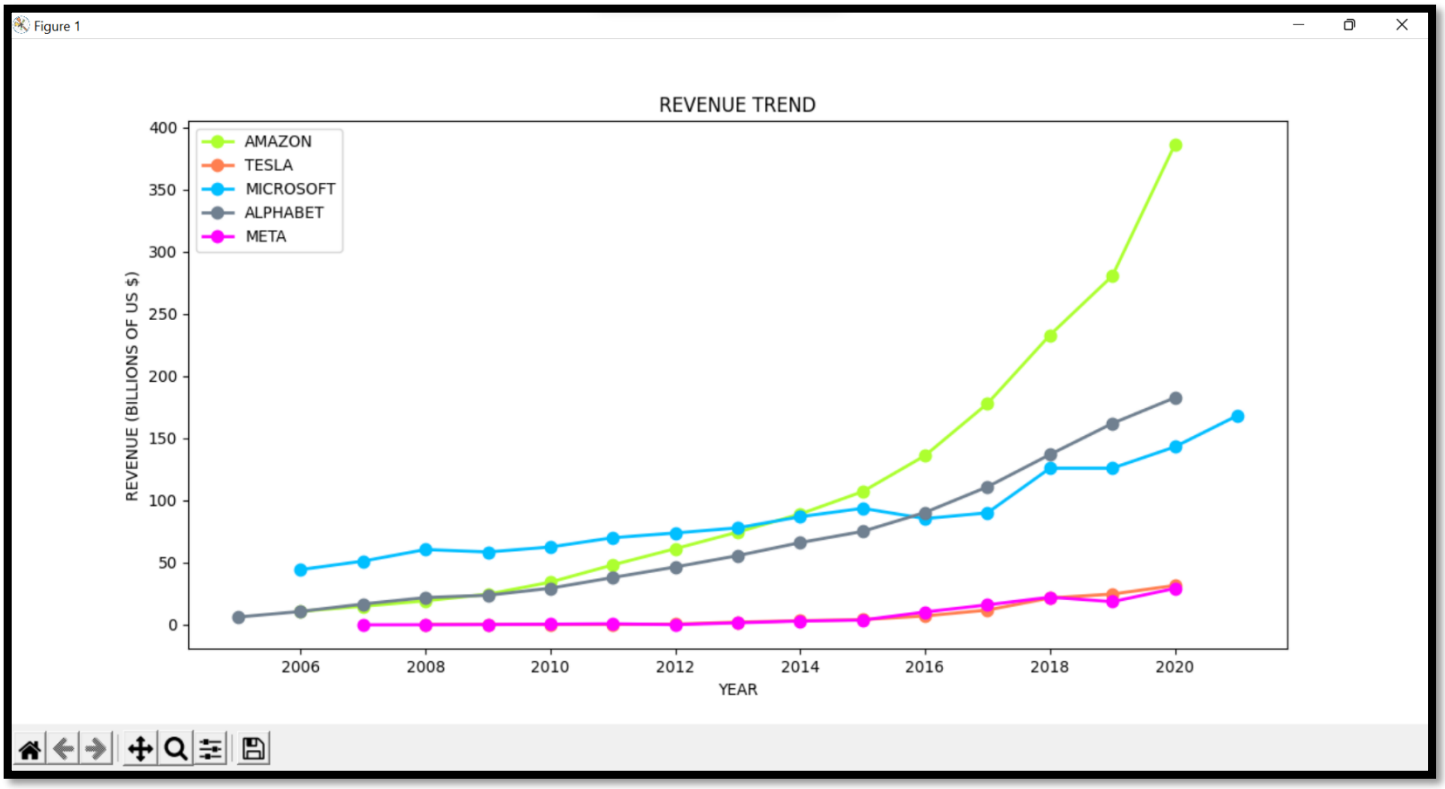
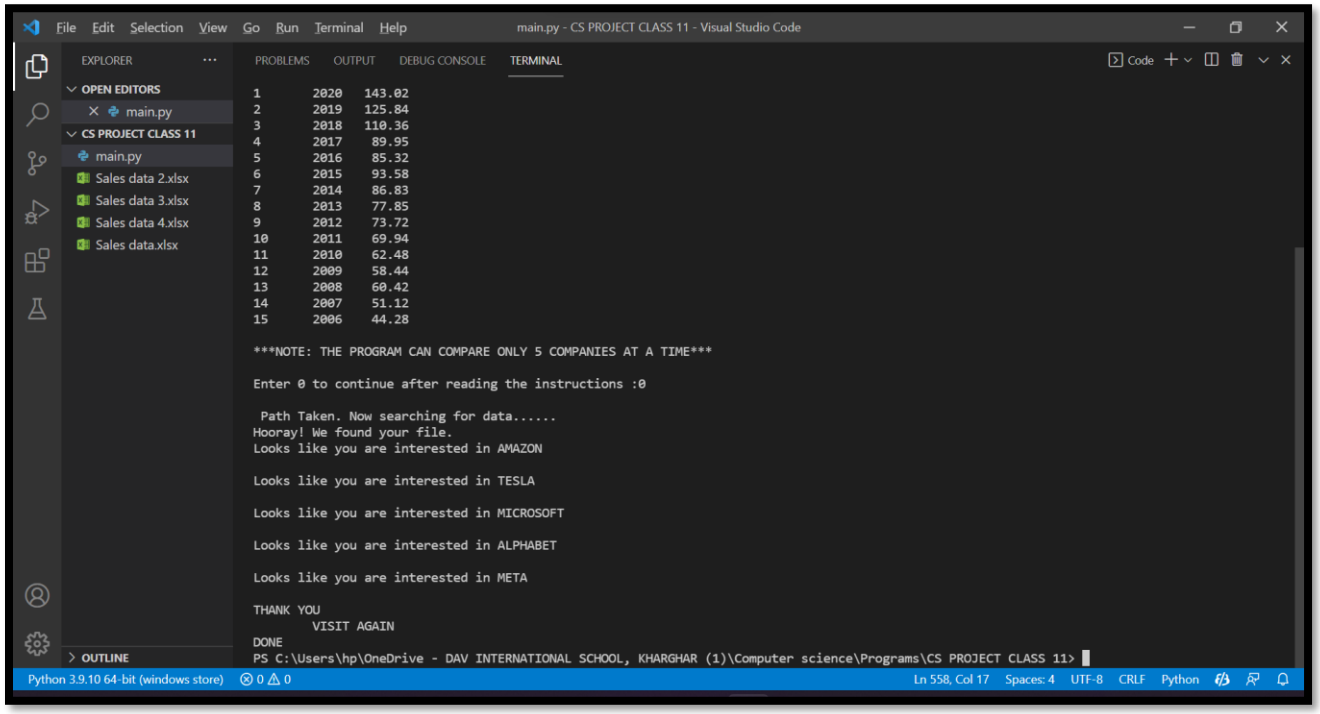


Figure 1
REVENUE TREND
AMAZON
TESLA
MICROSOFT
ALPHABET
META
REVENUE (BILLIONS OF US $)
YEAR
0
50
100
150
200
250
300
350
400
2006
2008
2010
2012
2014
2016
2018
2020


File Edit Selection View Go Run Terminal Help
main.py - CS PROJECT CLASS 11 - Visual Studio Code
EXPLORER
OPEN EDITORS
main.py
CS PROJECT CLASS 11
main.py
Sales data 2.xlsx
Sales data 3.xlsx
Sales data 4.xlsx
Sales data.xlsx
OUTLINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1 2020 143.02
2 2019 125.84
3 2018 110.36
4 2017 89.95
5 2016 85.32
6 2015 93.58
7 2014 86.83
8 2013 77.85
9 2012 73.72
10 2011 69.94
11 2010 62.48
12 2009 58.44
13 2008 60.42
14 2007 51.12
15 2006 44.28
NOTE: THE PROGRAM CAN COMPARE ONLY 5 COMPANIES AT A TIME
Enter 0 to continue after reading the instructions :0
Path Taken. Now searching for data.....
Hooray! We found your file.
Looks like you are interested in AMAZON
Looks like you are interested in TESLA
Looks like you are interested in MICROSOFT
Looks like you are interested in ALPHABET
Looks like you are interested in META
THANK YOU
VISIT AGAIN
DONE
PS C:\Users\hp\OneDrive - DAV INTERNATIONAL SCHOOL, KHARGHAR (1)\Computer science\Programs\CS PROJECT CLASS 11>
Python 3.9.10 64-bit (windows store)
Ln 558, Col 17 Spaces: 4 UTF-8 CRLF Python

NOW AS WE CLICK ON EXIT, THE SCREEN CLOSES.

WITH THIS THE PROGRAM ENDS AND THE PURPOSE OF THE USER IS FULFILLED.

GITHUB LINK FOR EXCEL SHEETS AND CODE:

https://github.com/anuragthakur2102/CS-PROJECT-CLASS-11

NOTE: IF YOU WANT TO RN THE CODE YOU WILL HAVE TO DOWNLOAD AND KEEP THE FILE NAMED 'Sales data.xlsx' PRESENT IN THE GIVEN LINK.

LEARNINGS

I LEARNED MANY NEW CONCEPTS OF PYTHON AND GOT TO KNOW ABOUT NEW LIBRARIES, WHICH WHEN USED MADE MY PROJECT LOOK GOOD AND ALSO, I WAS ABLE TO APPLY THE CONCEPTS LEARNT IN THE CLASS THROUGH THIS PROJECT. THIS PROJECT PLAYED A CRUCIAL ROLE IN INCREASING MY INTEREST IN CODING AND PROGRAMMING. SOME IMPORTANT CONCEPTS I LEARNED ARE:

1. SCREENS AND BUTTONS:

YOU WOULD HAVE SEEN THAT I USED TKINTER A LOT. I MADE SCREENS AND BUTTONS, WHICH MADE MY PROGRAM BETTER IN INTERACTION WITH USERS. USERS LIKE CLICKING JUST A BUTTON RATHER THAN WRITING THE WHOLE THING. THEREFORE, TO MAKE THIS PROJECT

GREAT, I LEARNED TKINTER AND MADE USE OF IT. MOREOVER, BUTTONS HELPED ME IN PLOTTING MULTIPLE GRAPHS AT A TIME.

2. READING EXCEL SHEETS:

IF USERS WANTED TO ANALYSE THEIR REVENUE GROWTH OVER MANY YEARS, IT WOULD HAVE BEEN REALLY DIFFICULT TO TYPE EVERY DATA IN THE TERMINAL AND IF SOME DATA WAS MISTYPED, IT COULD HAVE RESULTED IN WRONG ANALYSIS, THUS, INPUTTING DATA IN EXCEL SHEET FORM WAS IMPORTANT BOTH FOR USER TO BE RELAXED AND FOR THE PROGRAM TO GET THE DATA CORRECTLY. MOREOVER, WHEN COMPARING DIFFERENT COMPANY'S REVENUE, EXCEL SHEET IS THE BEST FORM AS IT CAN STORE DIFFERENT COMPANY'S DATA IN DIFFERENT SHEETS. AND THEREFORE, I LEARNED PANDAS TO IMPORT DATA FROM EXCEL AND ALSO READ AND TAKE DATA FROM IT.

3. DEFINING COMMANDS:

PYTHON DEF KEYWORD IS USED TO DEFINE A FUNCTION. WHENEVER A USER CLICKS A BUTTON THERE IS A COMMAND PERFORMED BY THE PROGRAM. SINCE MY PROGRAM HAD SO MANY BUTTONS, I USED DEF FUNCTION TO PROCEED WITH THE PROGRAM AS PER THE CHOICE MADE BY THE USER.

THIS PROJECT GAVE ME A GREAT OPPORTUNITY TO LEARN NEW THINGS IN PYTHON AND PROVED TO BE A VERY GOOD OPTION FOR INCREASING MY INTEREST IN PROGRAMMING.

LIMITATIONS

SINCE I AM JUST A BEGINNER IN PROGRAMMING, I WAS NOT AWARE OF MANY MODULES AND LIBRARIES THAT COULD CONVERT 20 LINES OF CODE INTO A 1 LINE STATEMENT OR FUNCTION AND THUS THERE WERE SOME LIMITATIONS OF THE PROJECT. SOME OF THEM ARE LISTED BELOW:

1. MANY LIBRARIES AND MODULES HAVE BEEN USED THAT INCREASE THE COMPUTATION TIME OF THE CODE AND ALSO THE SYSTEM REQUIREMENTS FOR THE PROGRAM TO RUN.

2. USERS DON'T USUALLY READ THE INSTRUCTIONS AND THUS, DON'T SAVE THEIR FILES IN THE SAME FOLDER WHERE THE CODE IS SAVED. THIS RESULTS IN THE PROGRAM NOT BEING ABLE TO FIND THE FILE.

EXAMPLE FROM MY PROGRAM

3. IT IS A VERY BIG CODE OF 750 LINES WHICH RESULTED IN SLOW PROCESSING OF THE CODE AND BEING A BEGINNER, I HAD NO CLUE OF ANY MODULE OR FUNCTION THAT COULD SHORTEN THE NUMBER OF LINES.

REFERENCES

1. TKINTER DOCUMNETATION
2. PANDAS DOCUMENTATION
3. GEEKSFORGEEKS.COM
4. STACKOVERFLOW.COM
5. DATA-FLAIR.COM

THANK YOU