

REVENUE/SALES COMPARISON AND ANALYSIS

COMPUTER SCIENCE PROJECT

(2021-2022)

DAV INTERNATIONAL SCHOOL

NAME: ANURAG KAVI SHANKAR THAKUR

CLASS: XI - A

ROLL NO: 6

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.

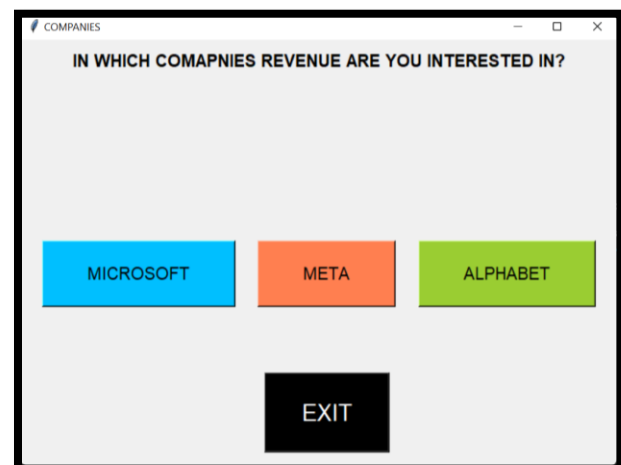
MOREOVER, I HAVE USED PYTHON LIBRARIES WHICH MAKE MY PROGRAM LOOK GOOD, USER-FRIENDLY, THUS, MAKING IT EASIER FOR USERS TO INTERACT WITH THE PROGRAM.

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.

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



EXAMPLE FROM MY PROGRAM

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.

I HAVE USED PANDAS TO READ THE INPUT GIVEN AS EXCEL FILE BY THE USER.

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

EXAMPLE FROM MY PROGRAM

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.

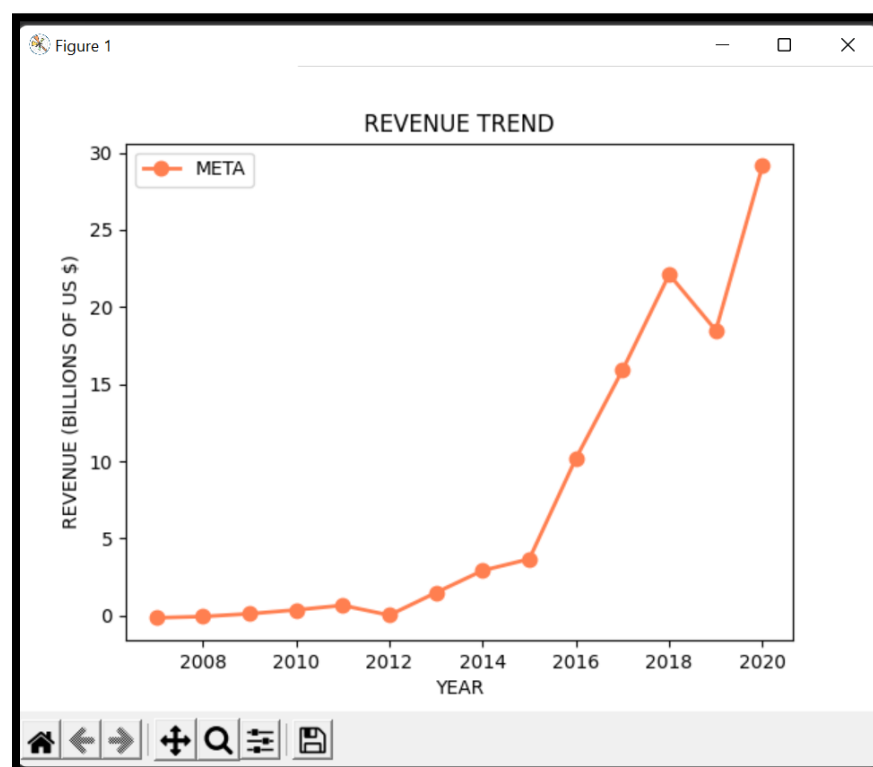
```
assert os.path.exists(path1), " I did not find your file at, "+str(path1)
f=open(path1)
print("Hooray! We found your file.")
```

EXAMPLE FROM MY PROGRAM

4. MATPLOTLIB

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

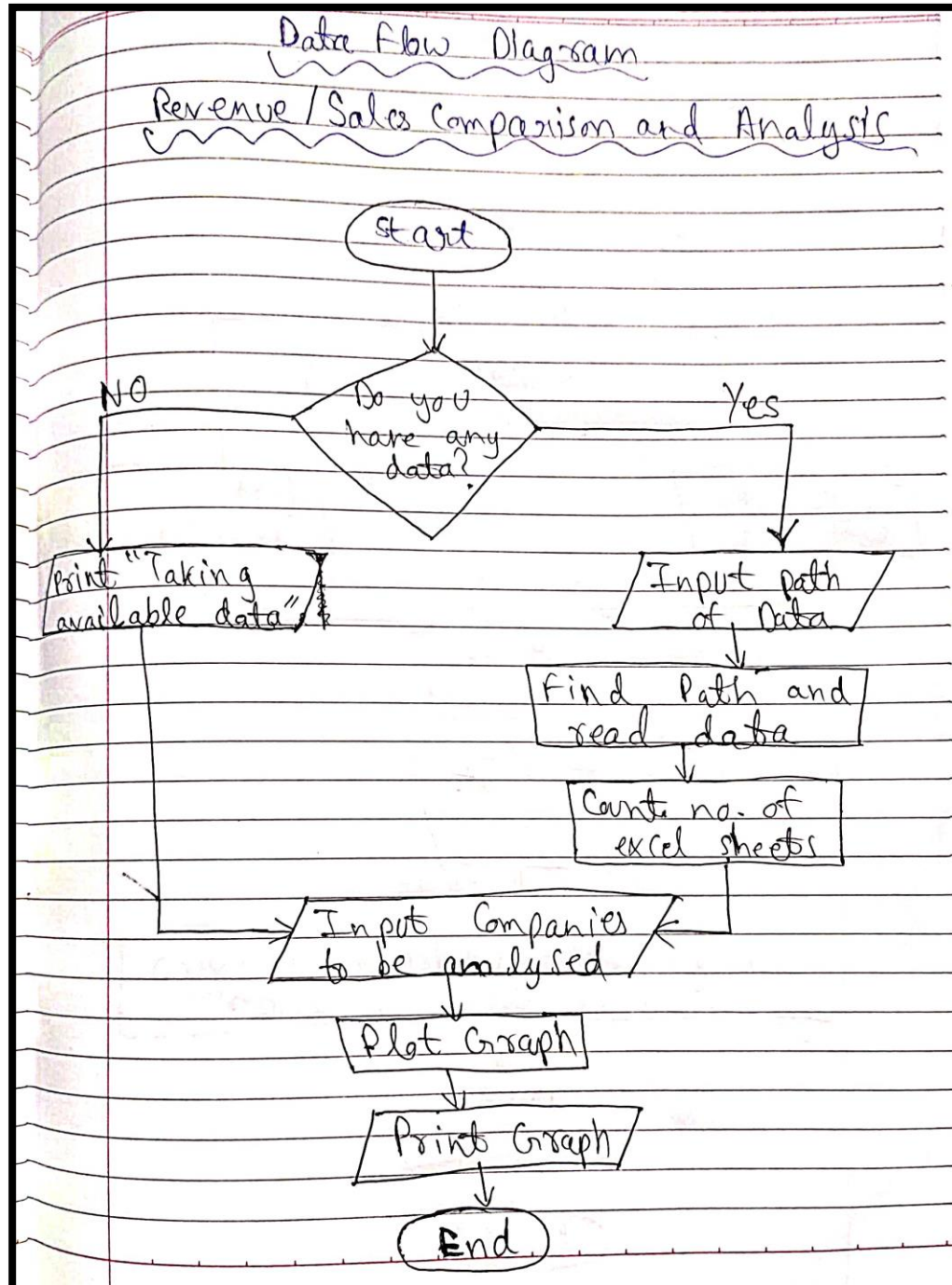
I HAVE USED MATPLOTLIB.PYPLOT MODULE OF MATPLOTLIB LIBRARY. IT IS USED FOR THE GRAPHICAL REPRESENTATION OF DATA, WHICH WAS THE MAIN OBJECTIVE OF MY PROJECT. IT ALSO HELPS TO REPRESENT MULTIPLE GRAPHS AT A TIME, WHICH PROVED TO BE VERY USEFUL FOR THE COMPARISON OF DIFFERENT COMPANIES.



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 X
main.py > ...
1  '''NAME : ANURAG KAVI SHANKAR TAHKUR
2  CLAS: XI A, ROLL. NO : 6,
3  DAV INTERNATIONAL SCHOOL
4  COMPUTER SCIENCE PROJECT (2021-2022)'''
5
6  '''PROCESS TO INTALL A LIBRARY
7  1. RUN CMD AS ADMINISTRATOR
8  2. TYPE : "pip install <name of library>"
9  3. PRESS ENTER, THEN WAIT FOR IT TO INSTALL.'''
10
11 from tkinter import *
12 import pandas as pd
13 import matplotlib.pyplot as p
14 import os
15
16
17 l=[]
18 l_2=[]
19
20 root = Tk()
21 root.title("Excel Sheet")
22 root.geometry("600x350")
23
24 #Warning
25 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")
26 label.pack(side = BOTTOM)
27
28 #Question
29
30 quest= Label(text = "DO YOU HAVE ANY DATA OF YOUR OWN?", fg = "black", font = "arial 15 bold")
31 quest.pack(side=TOP, pady=10)
32
33 def click(event):
34     text=event.widget.cget("text")
35     l.append(text)
36     if (l[0]=='YES'):
37         print('Your Excel Sheet must be of this form

```

```

main.py > ...
33 def click(event):
34     text=event.widget.cget("text")
35     l.append(text)
36     if (l[0]=='YES'):
37         print('Your Excel Sheet must be of this form
38         -----
39         ROW    TITLE ')
40         data=(pd.read_excel('Sales data.xlsx'))
41         print(pd.DataFrame(data))
42
43
44     elif (l[0]=="NO"):
45         print('We have Sales Data of three comapnies: MICROSOFT, META, ALPHABET')
46         print()
47         comp=int(input('Which one do you want
48         1. MICROSOFT- *(TYPE 1)*
49         1. ALPHABET- *(TYPE 2)*
50         1. META- *(TYPE 3)*
51         INPUT HERE : '))
52
53 def yes():
54     print(" SO, YOU HAVE CHOSEN TO INPUT YOUR DATA")
55     print()
56     print('Your Excel Sheet must be of this form
57     -----
58     ROW    TITLE ')
59     data=(pd.read_excel('Sales data.xlsx'))
60     print(pd.DataFrame(data))
61     root.destroy()
62     print()
63     print("*****NOTE: THE PROGRAM CAN COMPARE ONLY 5 COMPANIES AT A TIME****")
64     print()
65     confirm=int(input("Enter 0 to continue after reading the instructions :"))
66     if confirm!=0:
67         print("Oops! You will have to run the program again and follow thw steps properly")
68         print()

```

```

66 if confirm!=0:
67     print("Oops! You will have to run the program again and follow thw steps properly")
68     print()
69     print()
70 else:
71     path_root=Tk()
72     path_root.title("PATH")
73     path_root.geometry("600x400")
74
75     #instruction
76     inst= Label(path_root, text = '''***SAVE YOUR FILE IN THE GIVEN DIRECTORY
77     'C:\\Users\\hp\\OneDrive - DAV INTERNATIONAL SCHOOL, KHARGHAR (1)\\Computer science\\Programs'
78     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")
79     inst.pack(side=BOTTOM, pady=20)
80     quest_2= Label(path_root, text="Enter name of your file with .xlsx at the end ", fg="black", font="areal 13")
81     quest_2.pack(side=TOP, pady=20)
82     print()
83
84
85
86
87 #taking all the inputs from user
88 def okay():
89     print(" Path Taken. Now searching for data.....")
90     path1=p_entry.get()
91     path_root.destroy()
92     assert os.path.exists(path1), " I did not find your file at, "+str(path1)
93     f=open(path1)
94     print("Hooray! We found your file.")
95
96     #Now reading the file
97     file=pd.ExcelFile(path1)
98     sheets_1=file.sheet_names
99     if len(sheets_1)==1:
100         sheet_root=Tk()
101         sheet_root.title("SHEET NAME")
102         sheet_root.geometry("700x500")

```

```

101         sheet_root.title("SHEET NAME")
102         sheet_root.geometry("700x500")
103
104         quest_4= Label(text = "IN WHICH COMAPNIES REVENUE ARE YOU INTERESTED IN?", fg = "black", font = "arial 15 bold")
105         quest_4.pack(side=TOP, pady=10)
106
107         def micro():
108             print("Looks like you are interested in", sheets_1[0])
109             print()
110
111             #graph
112             #using pandas to get specific rows now
113             #x axis
114             y_1=pd.read_excel(path1, sheet_name=sheets_1[0])
115             year=y_1[sheets_1[0]].tolist()
116             year=year[::-1]
117             year.pop()
118
119             #y axis
120             r_1=pd.read_excel(path1, sheet_name=sheets_1[0])
121             revenue=r_1["ANALYSIS"].tolist()
122             revenue=revenue[::-1]
123             revenue.pop()
124
125             #making graph
126             x=year
127             y=revenue
128             p.plot(x,y, color="deepskyblue", linewidth=2, marker='.', markersize=15, label=sheets_1[0])
129             p.xlabel("YEAR")
130             p.ylabel("REVENUE (BILLIONS OF US $)")
131             p.title("REVENUE TREND")
132             p.legend()
133             p.show()
134
135         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()
171

```

```

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
195             #y axis
196             r_1=pd.read_excel(path1, sheet_name=sheets_1[1])
197             revenue=r_1["ANALYSIS"].tolist()
198             revenue=revenue[::-1]
199             revenue.pop()
200
201             #making graph
202             x=year
203             y=revenue
204             p.plot(x,y, color="greenyellow",linewidth=2, marker='.',markersize=15, label=sheets_1[1])
205             p.xlabel("YEAR")
206             p.ylabel("REVENUE (BILLIONS OF US $)")

```

```

204     p.xlabel("YEAR")
205     p.ylabel("REVENUE (BILLIONS OF US $)")
206     p.title("REVENUE TREND")
207     p.legend()
208     p.show()
209
210
211     def ex():
212         sheet_root.destroy()
213         print('THANK YOU
214 VISIT AGAIN')
215
216     m= Button(sheet_root, text = sheets_1[0], padx= 40 , pady =20 ,command =micro , font="areal 15", bg="deepskyblue")
217     m.pack(side=LEFT, padx=25)
218     a= Button(sheet_root, text = sheets_1[1], padx= 40 , pady =20 ,command =alpha , font="areal 15", bg="yellowgreen")
219     a.pack(side=RIGHT, padx=25)
220     e= Button(sheet_root, text = " EXIT ", padx= 25 , pady =20 ,command =ex, font="areal 20", fg="white", bg="black")
221     e.pack(side=BOTTOM, pady=15)
222
223     if len(sheets_1)==3:
224         sheet_root=tk()
225         sheet_root.title("SHEET NAME")
226         sheet_root.geometry("700x500")
227
228         quest_4= Label(text = "IN WHICH COMAPNIES REVENUE ARE YOU INTERESTED IN?", fg = "black", font = "arial 15 bold")
229         quest_4.pack(side=TOP, pady=10)
230
231         def micro():
232             print("Looks like you are interested in", sheets_1[0])
233             print()
234
235             #graph
236             #using pandas to get specific rows now
237             #x axis
238             y_1=pd.read_excel(path1, sheet_name=sheets_1[0])
239             year=y_1[sheets_1[0]].tolist()
240             year=year[:-1]
241             year.pop()

```

```

237     #x axis
238     y_1=pd.read_excel(path1, sheet_name=sheets_1[0])
239     year=y_1[sheets_1[0]].tolist()
240     year=year[:-1]
241     year.pop()
242
243     #y axis
244     r_1=pd.read_excel(path1, sheet_name=sheets_1[0])
245     revenue=r_1["ANALYSIS"].tolist()
246     revenue=revenue[:-1]
247     revenue.pop()
248
249     #making graph
250     x=year
251     y=revenue
252     p.plot(x,y, color="deepskyblue",linewidth=2, marker='.', markersize=15, label=sheets_1[0])
253     p.xlabel("YEAR")
254     p.ylabel("REVENUE (BILLIONS OF US $)")
255     p.title("REVENUE TREND")
256     p.legend()
257     p.show()
258
259
260     def alpha():
261         print("Looks like you are interested in", sheets_1[1])
262         print()
263
264         #graph
265         #using pandas to get specific rows now
266         #x axis
267         y_1=pd.read_excel(path1, sheet_name=sheets_1[1])
268         year=y_1[sheets_1[1]].tolist()
269         year=year[:-1]
270         year.pop()
271         #y axis
272         r_1=pd.read_excel(path1, sheet_name=sheets_1[1])
273         revenue=r_1["ANALYSIS"].tolist()

```

```

271 #y axis
272 r_1=pd.read_excel(path1, sheet_name=sheets_1[1])
273 revenue=r_1["ANALYSIS"].tolist()
274 revenue=revenue[::-1]
275 revenue.pop()
276
277 #making graph
278 x=year
279 y=revenue
280 p.plot(x,y, color="greenyellow",linewidth=2, marker='.',markersize=15,label=sheets_1[1])
281 p.xlabel("YEAR")
282 p.ylabel("REVENUE (BILLIONS OF US $)")
283 p.title("REVENUE TREND")
284 p.legend()
285 p.show()
286
287
288 def met():
289     print("Looks like you are interested in", sheets_1[2])
290     print()
291
292     #graph
293     #using openpyxl to get specific rows now
294     y_1=pd.read_excel(path1, sheet_name=sheets_1[2])
295     year=y_1[sheets_1[2]].tolist()
296     year=year[::-1]
297     year.pop()
298     #y axis
299     r_1=pd.read_excel(path1, sheet_name=sheets_1[2])
300     revenue=r_1["ANALYSIS"].tolist()
301     revenue=revenue[::-1]
302     revenue.pop()
303
304     #making graph
305     x=year
306     y=revenue
307     p.plot(x,y, color="greenyellow",linewidth=2, marker='.',markersize=15,label=sheets_1[2])

```

```

304 #making graph
305 x=year
306 y=revenue
307 p.plot(x,y, color="coral",linewidth=2, marker='.',markersize=15,label=sheets_1[2])
308 p.xlabel("YEAR")
309 p.ylabel("REVENUE (BILLIONS OF US $)")
310 p.title("REVENUE TREND")
311 p.legend()
312 p.show()
313
314
315 def ex():
316     sheet_root.destroy()
317     print(''''THANK YOU
318 VISIT AGAIN''')
319
320 m= Button(sheet_root, text = sheets_1[0], padx= 40 , pady =20 ,command =micro , font="areal 15", bg="deepskyblue")
321 m.pack(side=LEFT,padx=25)
322 a= Button(sheet_root, text = sheets_1[1], padx= 40 , pady =20 ,command =alpha , font="areal 15", bg="yellowgreen")
323 a.pack(side=RIGHT,padx=25)
324 e= Button(sheet_root, text = " EXIT ",padx= 25 , pady =20 ,command =ex, font="areal 20", fg="white", bg="black")
325 e.pack(side=BOTTOM, pady=15)
326 me= Button(sheet_root, text = sheets_1[2], padx= 40 , pady= 20 ,command =met, font="areal 15", bg="coral")
327 me.pack(side=BOTTOM, pady=62)
328
329 if len(sheets_1)==4:
330     sheet_root=tk()
331     sheet_root.title("SHEET NAME")
332     sheet_root.geometry("700x500")
333
334     quest_4= Label(text = "IN WHICH COMAPNIES REVENUE ARE YOU INTERESTED IN?", fg = "black", font = "arial 15 bold")
335     quest_4.pack(side=TOP, pady=10)
336
337 def micro():
338     print("Looks like you are interested in", sheets_1[0])
339     print()
340

```

```

337
338 def micro():
339     print("Looks like you are interested in", sheets_1[0])
340     print()
341
342     #graph
343     #using pandas to get specific rows now
344     #x axis
345     y_1=pd.read_excel(path1, sheet_name=sheets_1[0])
346     year=y_1[sheets_1[0]].tolist()
347     year=year[:-1]
348     year.pop()
349
350     #y axis
351     r_1=pd.read_excel(path1, sheet_name=sheets_1[0])
352     revenue=r_1["ANALYSIS"].tolist()
353     revenue=revenue[:-1]
354     revenue.pop()
355
356     #making graph
357     x=year
358     y=revenue
359     p.plot(x,y, color="deepskyblue",linewidth=2, marker='.',markersize=15, label=sheets_1[0])
360     p.xlabel("YEAR")
361     p.ylabel("REVENUE (BILLIONS OF US $)")
362     p.title("REVENUE TREND")
363     p.legend()
364     p.show()
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
378     #y axis
379     r_1=pd.read_excel(path1, sheet_name=sheets_1[1])
380     revenue=r_1["ANALYSIS"].tolist()
381     revenue=revenue[:-1]
382     revenue.pop()
383
384     #making graph
385     x=year
386     y=revenue
387     p.plot(x,y, color="greenyellow",linewidth=2, marker='.',markersize=15,label=sheets_1[1])
388     p.xlabel("YEAR")
389     p.ylabel("REVENUE (BILLIONS OF US $)")
390     p.title("REVENUE TREND")
391     p.legend()
392     p.show()
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
405     #y axis
406     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     n.plot(x,y, color="slategray", linewidth=2, marker='.', markersize=15, label=sheets_1[3])

```

```

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

```

```

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()
542     revenue=revenue[::-1]
543     revenue.pop()

```

```

540 r_1=pd.read_excel(path1, sheet_name=sheets_1[2])
541 revenue=r_1["ANALYSIS"].tolist()
542 revenue=revenue[::-1]
543 revenue.pop()
544
545 #making graph
546 x=year
547 y=revenue
548 p.plot(x,y, color="coral",linewidth=2, marker='.',markersize=15,label=sheets_1[2])
549 p.xlabel("YEAR")
550 p.ylabel("REVENUE (BILLIONS OF US $)")
551 p.title("REVENUE TREND")
552 p.legend()
553 p.show()
554
555
556 def beta():
557     print("Looks like you are interested in", sheets_1[2])
558     print()
559
560     #graph
561     #using openpyxl to get specific rows now
562     y_1=pd.read_excel(path1, sheet_name=sheets_1[3])
563     year=y_1[sheets_1[3]].tolist()
564     year=year[::-1]
565     year.pop()
566     #y axis
567     r_1=pd.read_excel(path1, sheet_name=sheets_1[3])
568     revenue=r_1["ANALYSIS"].tolist()
569     revenue=revenue[::-1]
570     revenue.pop()
571
572     #making graph
573     x=year
574     y=revenue
575     p.plot(x,y, color="slategray",linewidth=2, marker='.',markersize=15, label=sheets_1[3])
576     p.xlabel("YEAR")

```

```

main.py > yes > okay
575 p.plot(x,y, color="slategray",linewidth=2, marker='.',markersize=15, label=sheets_1[3])
576 p.xlabel("YEAR")
577 p.ylabel("REVENUE (BILLIONS OF US $)")
578 p.title("REVENUE TREND")
579 p.legend()
580 p.show()
581
582
583 def gama():
584     print("Looks like you are interested in", sheets_1[4])
585     print()
586
587     #graph
588     #using openpyxl to get specific rows now
589     y_1=pd.read_excel(path1, sheet_name=sheets_1[4])
590     year=y_1[sheets_1[4]].tolist()
591     year=year[::-1]
592     year.pop()
593     #y axis
594     r_1=pd.read_excel(path1, sheet_name=sheets_1[4])
595     revenue=r_1["ANALYSIS"].tolist()
596     revenue=revenue[::-1]
597     revenue.pop()
598
599     #making graph
600     x=year
601     y=revenue
602     p.plot(x,y, color="magenta",linewidth=2, marker='.',markersize=15,label=sheets_1[4])
603     p.xlabel("YEAR")
604     p.ylabel("REVENUE (BILLIONS OF US $)")
605     p.title("REVENUE TREND")
606     p.legend()
607     p.show()
608
609 def ex():
610     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
680

```



```

679     #using pandas to get specific rows now
680     #x axis
681     y_1=pd.read_excel('Sales data.xlsx', sheet_name="ALPHABET")
682     year=y_1["ALPHABET"].tolist()
683     year=year[:-1]
684     year.pop()
685     #y axis
686     r_1=pd.read_excel('Sales data.xlsx', sheet_name="ALPHABET")
687     revenue=r_1["ANALYSIS"].tolist()
688     revenue=revenue[:-1]
689     revenue.pop()
690
691     #making graph
692     x=year
693     y=revenue
694     p.plot(x,y, color="greenyellow",linewidth=2, marker='.',markersize=15,label="ALPHABET")
695     p.xlabel("YEAR")
696     p.ylabel("REVENUE (BILLIONS OF US $)")
697     p.title("REVENUE TREND")
698     p.legend()
699     p.show()
700
701     def met():
702         print("Looks like you are interested in META")
703         print()
704
705         #graph
706         #using openpyxl to get specific rows now
707         y_1=pd.read_excel('Sales data.xlsx', sheet_name="META")
708         year=y_1["META"].tolist()
709         year=year[:-1]
710         year.pop()
711         #y axis
712         r_1=pd.read_excel('Sales data.xlsx', sheet_name="META")
713         revenue=r_1["ANALYSIS"].tolist()
714         revenue=revenue[:-1]
715         revenue.pop()

```

```

715     revenue.pop()
716
717     #making graph
718     x=year
719     y=revenue
720     p.plot(x,y, color="coral",linewidth=2, marker='.',markersize=15, label="META")
721     p.xlabel("YEAR")
722     p.ylabel("REVENUE (BILLIONS OF US $)")
723     p.title("REVENUE TREND")
724     p.legend()
725     p.show()
726
727     def ex():
728         comp_root.destroy()
729         print('THANK YOU
730         VISIT AGAIN')
731
732     m= Button(comp_root, text = " MICROSOFT ",padx= 40 , pady =20 ,command =micro , font="areal 15", bg="deepskyblue")
733     m.pack(side=LEFT,padx=25)
734     a= Button(comp_root, text = " ALPHABET ",padx= 40 , pady =20 ,command =alpha, font="areal 15", bg="yellowgreen")
735     a.pack(side=RIGHT,padx=25)
736     e= Button(comp_root, text = " EXIT ",padx= 25 , pady =20 ,command =ex, font="areal 20", fg="white", bg="black")
737     e.pack(side=BOTTOM, pady=15)
738     me= Button(comp_root, text = " META ",padx= 40 , pady= 20 ,command =met, font="areal 15", bg="coral")
739     me.pack(side=BOTTOM, pady=62)
740
741     b1= Button(root, text = " YES ",padx= 50 , pady =30 ,command =yes, font="areal 20", bg="cyan")
742     b1.pack(side=LEFT,padx=40)
743     b1.bind("<Button-1>", click)
744
745     b2 = Button(root, text = " NO ",padx= 50 , pady = 30,command = no, font="areal 20", bg="orange")
746     b2.pack(side=RIGHT,padx=40)
747     b2.bind("<Button-1>", click)
748
749
750     root.mainloop()

```

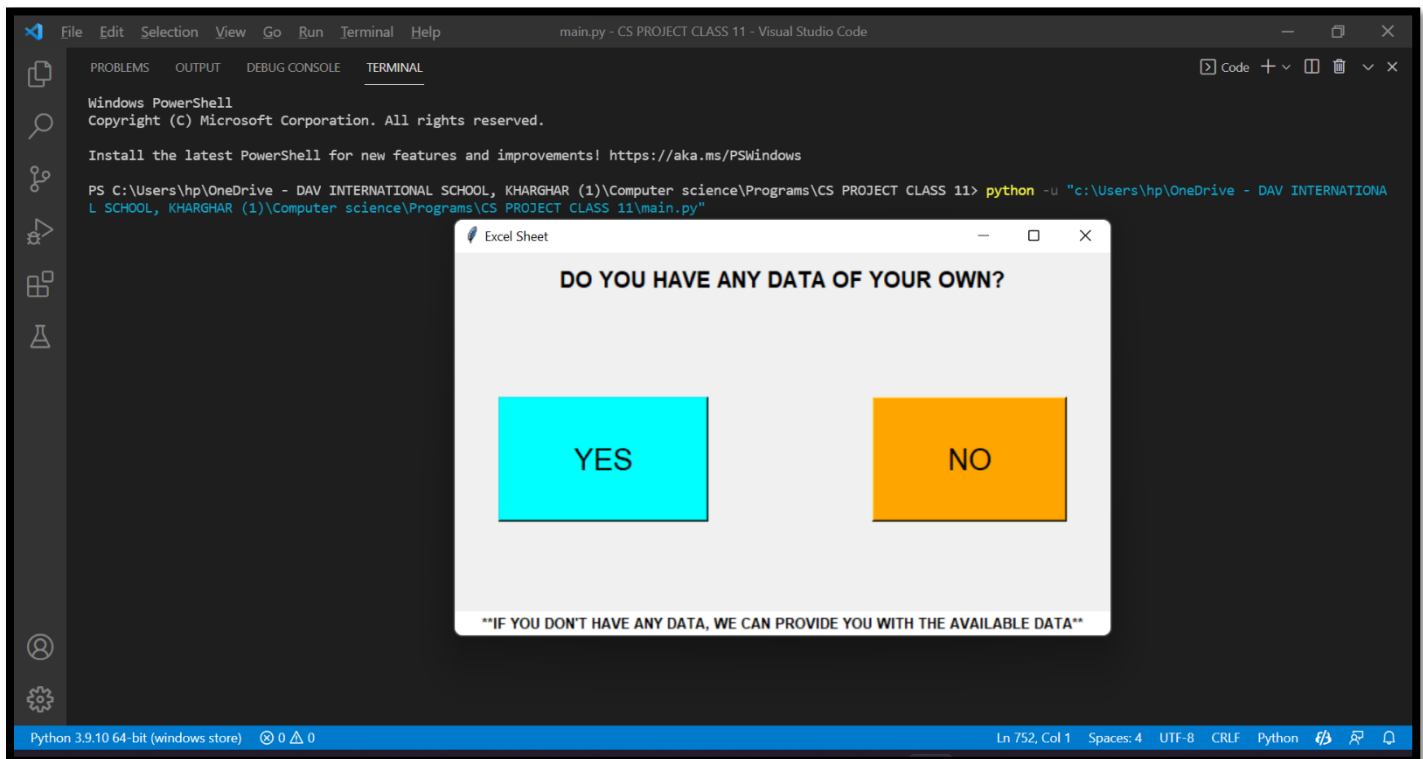
```

750     root.mainloop()
751     print("DONE")

```

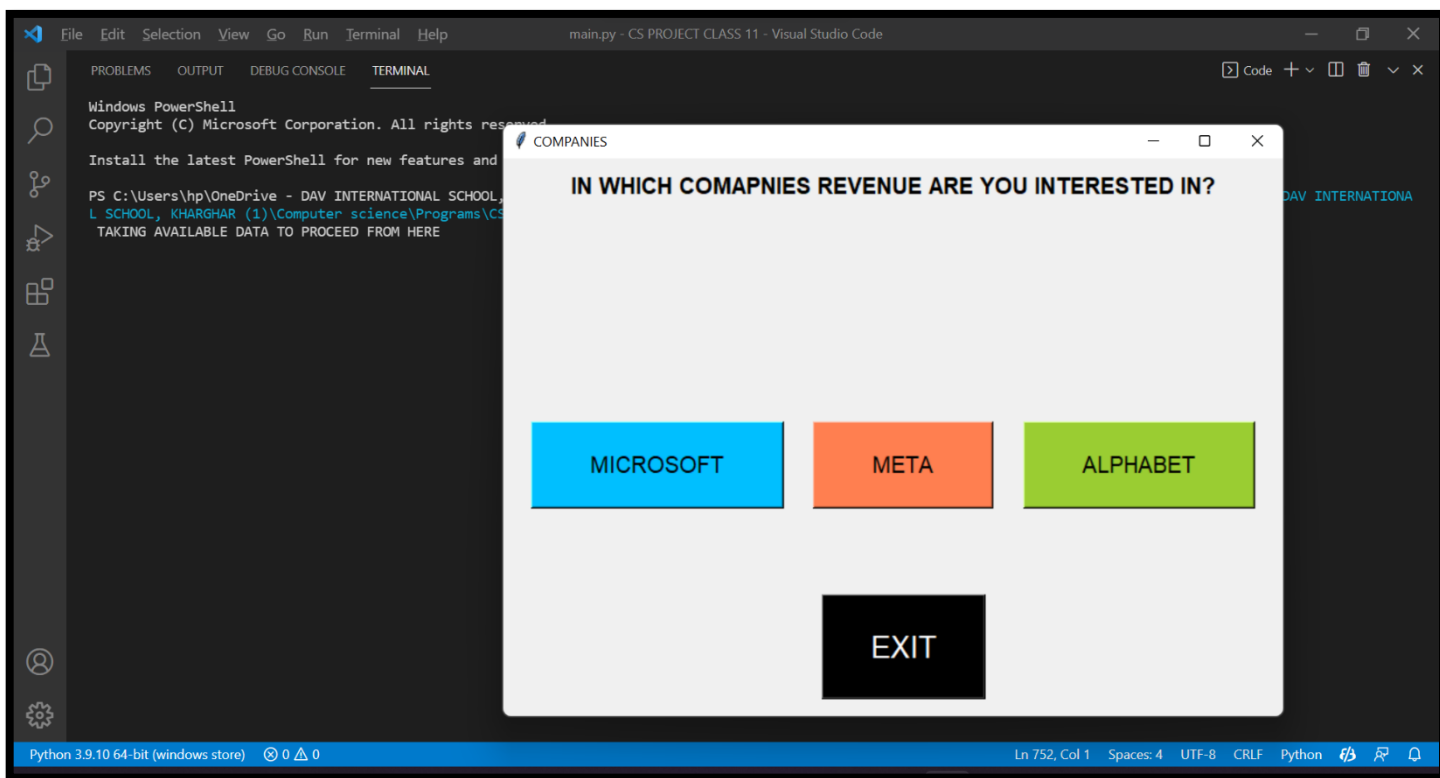
OUTPUT

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

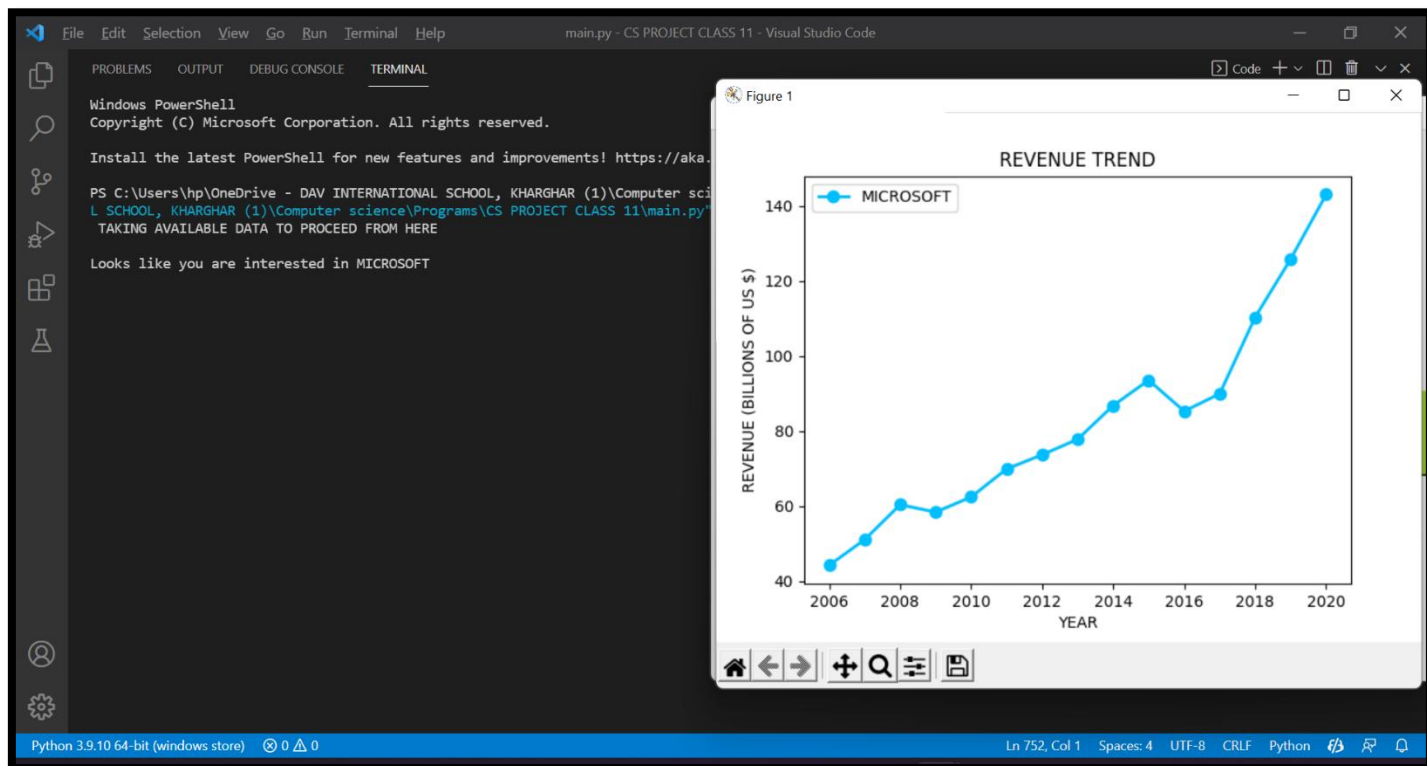


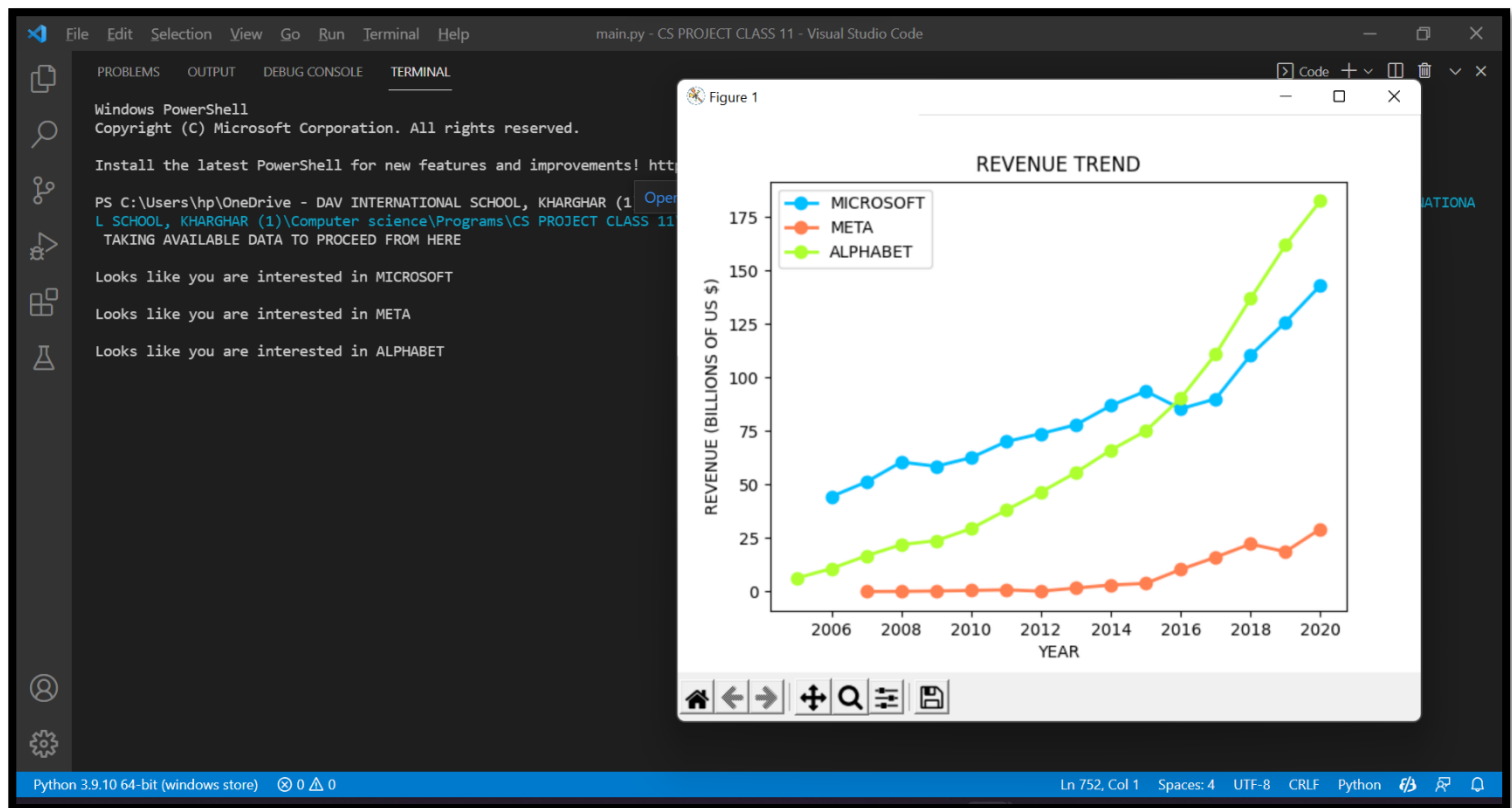
1. INPUTTING NO

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 MICROSOFT 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.

COMPANIES

IN WHICH COMAPNIES REVENUE ARE YOU INTERESTED IN?

MICROSOFT META ALPHABET

EXIT

```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\hp\OneDrive - DAV INTERNATIONAL SCHOOL, KHARGHAR (1) Open folder in new window (ctrl + click) :JECT CLASS 11> py
L SCHOOL, KHARGHAR (1)\Computer science\Programs\CS PROJECT CLASS 11\main.py"
TAKING AVAILABLE DATA TO PROCEED FROM HERE

Looks like you are interested in MICROSOFT
Looks like you are interested in META
Looks like you are interested in ALPHABET

THANK YOU
VISIT AGAIN
DONE
PS C:\Users\hp\OneDrive - DAV INTERNATIONAL SCHOOL, KHARGHAR (1)\Computer science\Programs\CS PROJECT CLASS 11>
```

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.

```

File Edit Selection View Go Run Terminal Help
main.py - CS PROJECT CLASS 11 - Visual Studio Code

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\hp\OneDrive - DAV INTERNATIONAL SCHOOL, KHARGHAR (1)\Computer science\Programs\CS PROJECT CLASS 11> python -u "c:\Users\hp\OneDrive - DAV INTERNATIONAL SCHOOL, KHARGHAR (1)\Computer science\Programs\CS PROJECT CLASS 11\main.py"
SO, YOU HAVE CHOSEN TO INPUT YOUR DATA

Your Excel Sheet must be of this form
-----
ROW  TITLE
MICROSOFT ANALYSIS
0   YEAR  REVENUE
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

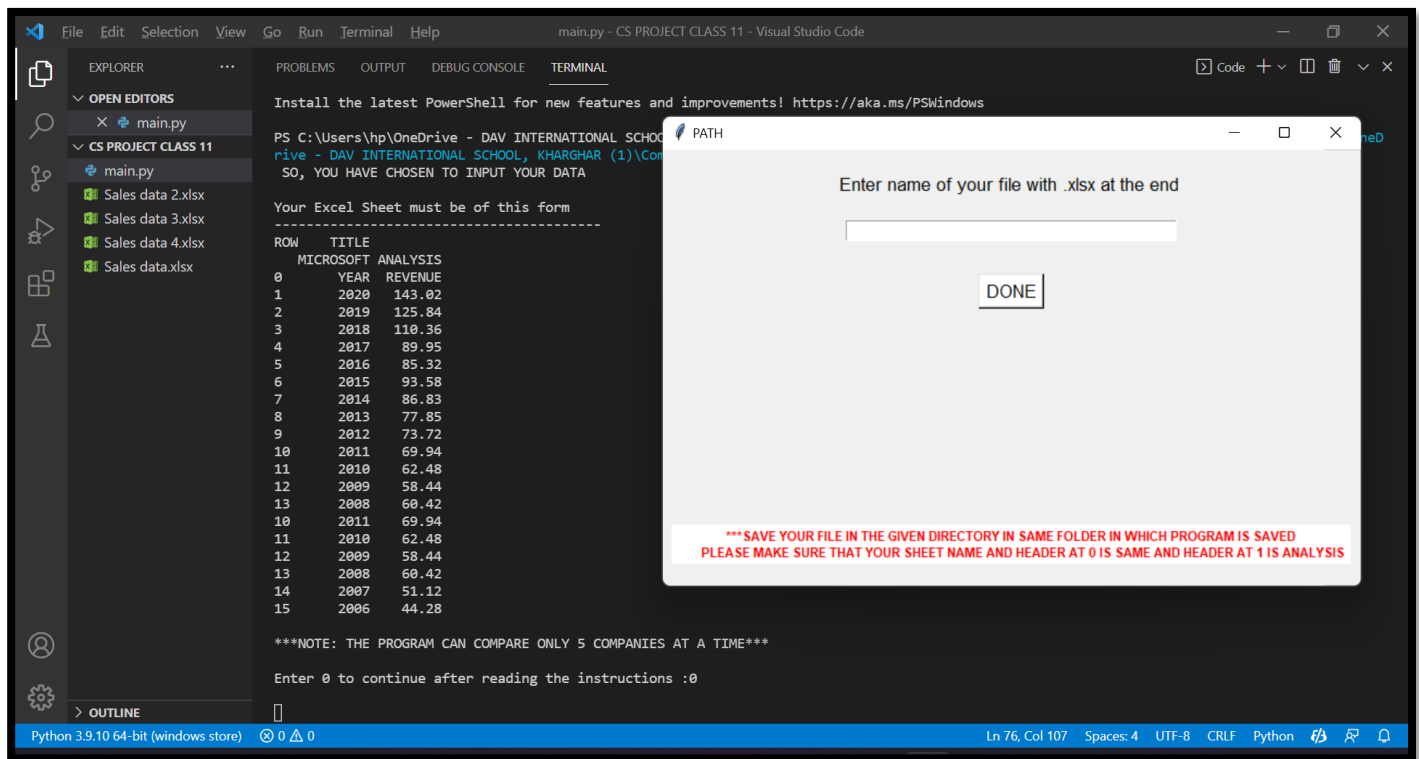
***NOTE: THE PROGRAM CAN COMPARE ONLY 5 COMPANIES AT A TIME***

Enter 0 to continue after reading the instructions :[]
Python 3.9.10 64-bit (windows store) 0 0 0
Ln 752, Col 1 Spaces: 4 UTF-8 CRLF Python

```

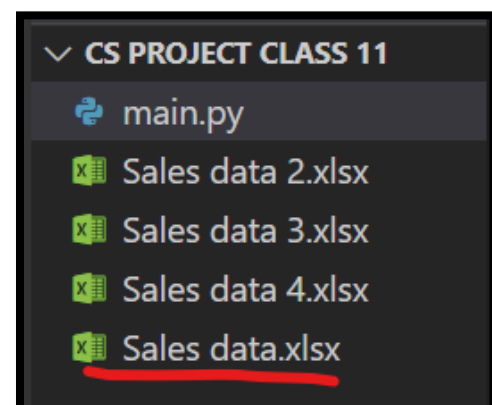
FROM HERE, WE ENTER 0 TO CONTINUE.

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.

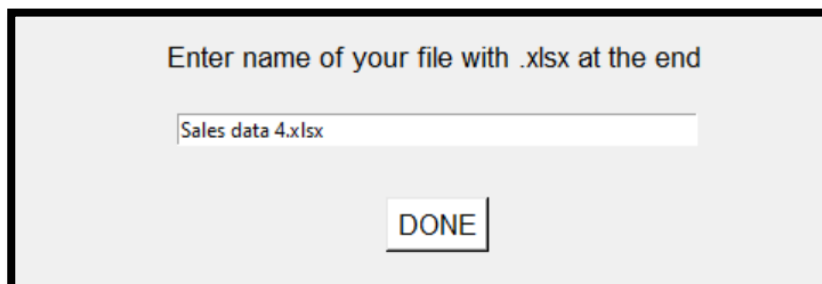


YOU CAN INPUT YOUR DATA BY FOLLOWING THE INSTRUCTIONS, ELSE I HAVE GIVEN 3 DIFFERENT DATA SHEETS WITH THE PROGRAM IN THE GITHUB LINK AT THE END. YOU CAN USE IT FOR TESTING THIS PART OF THE PROGRAM.

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.

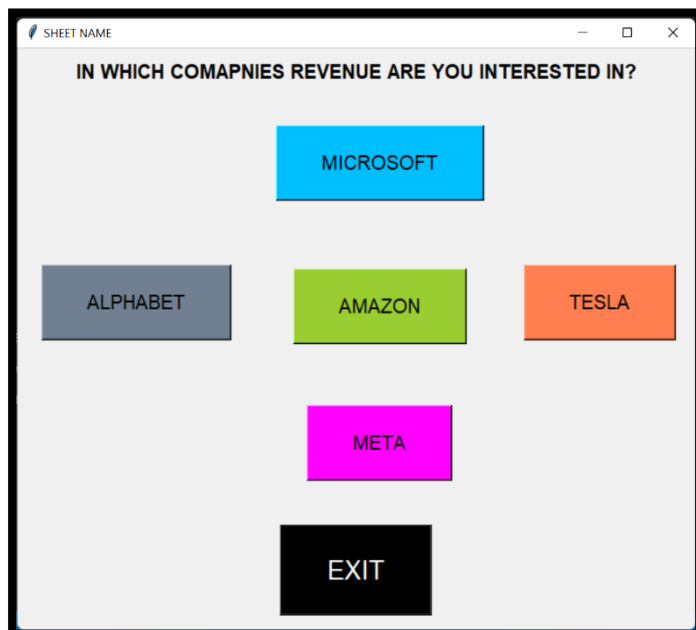
```
10      2011      69.94
11      2010      62.48
12      2009      58.44
13      2008      60.42
14      2007      51.12
15      2006      44.28

***NOTE: THE PROGRAM CAN COMPARE ONLY 5 COMPANIES AT A TIME***

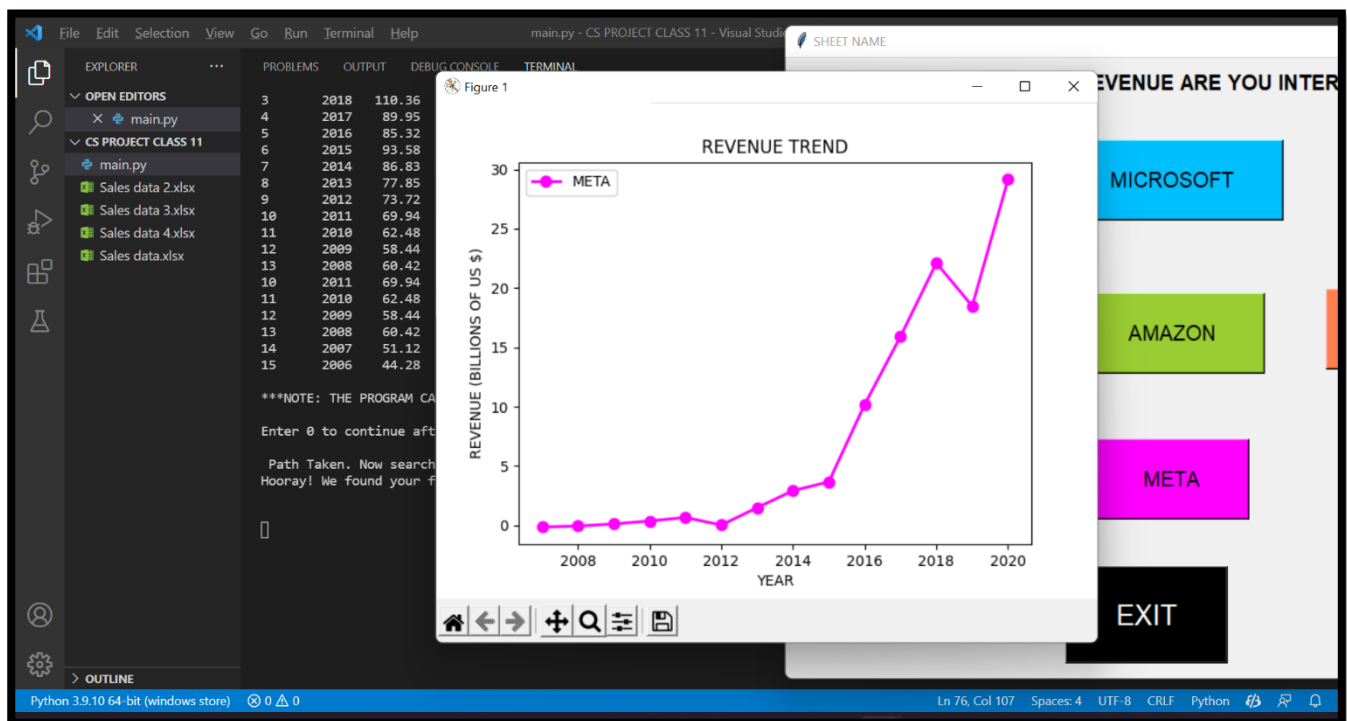
Enter 0 to continue after reading the instructions :0

Path Taken. Now searching for data.....
Hooray! We found your file.
█
```

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.

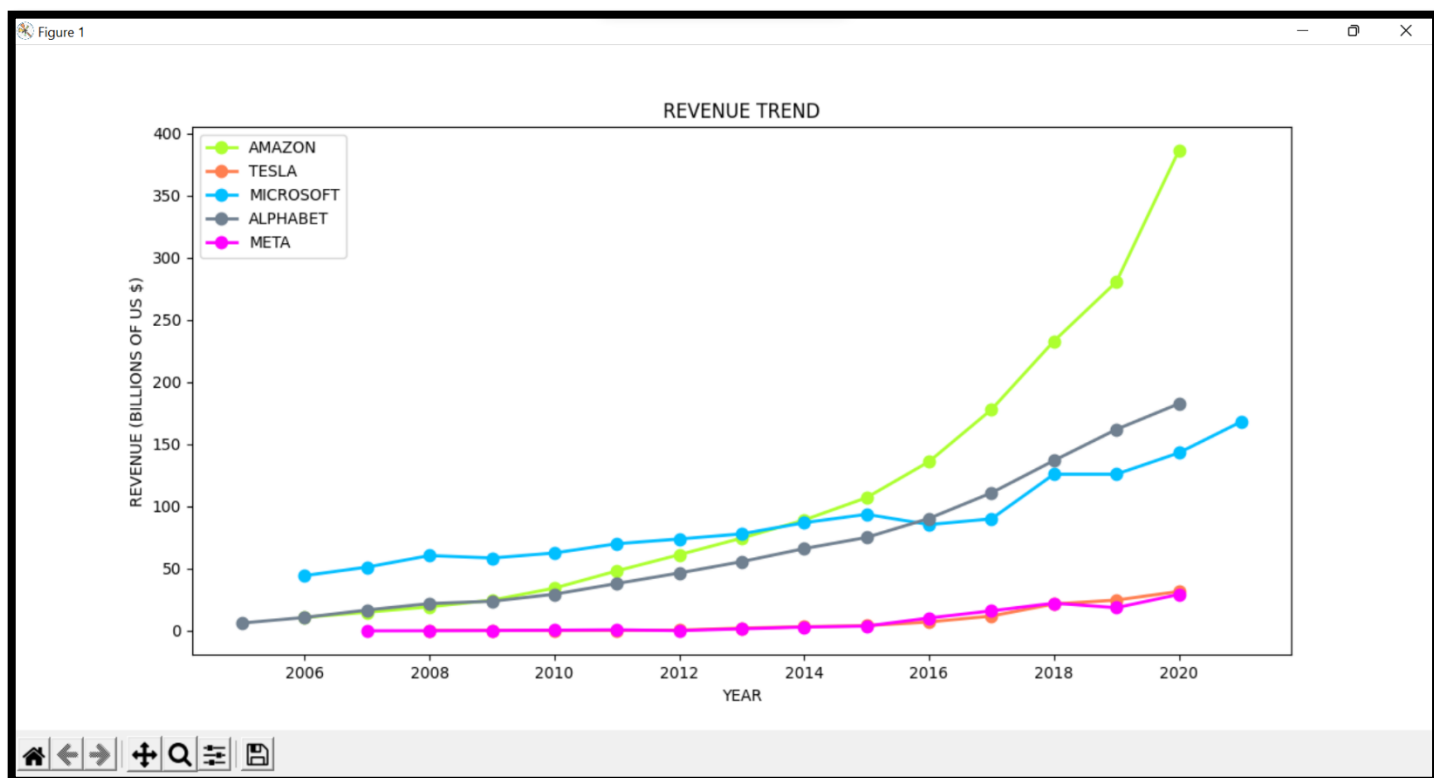


ACCORDING TO MY WISH, I AM SELECTING META.



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.



```

File Edit Selection View Go Run Terminal Help main.py - CS PROJECT CLASS 11 - Visual Studio Code
EXPLORER PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
OPEN EDITORS
  X main.py
  CS PROJECT CLASS 11
    main.py
    Sales data 2.xlsx
    Sales data 3.xlsx
    Sales data 4.xlsx
    Sales data.xlsx
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) 0 0 In 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 RUN 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.

*****SAVE YOUR FILE IN THE GIVEN DIRECTORY IN SAME FOLDER IN WHICH PROGRAM IS SAVED
PLEASE MAKE SURE THAT YOUR SHEET NAME AND HEADER AT 0 IS SAME AND HEADER AT 1 IS ANALYSIS**

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

THANK YOU