

|  |
| --- |
| Project Title: Develop a SEO Tool to Analyze Live Web Pages |
| MATHIVARUNI RAJAKUMAR |
|  |

# Project Title: Develop a SEO Tool to Analyze Live Web Pages

Introduction:

Search Engine Optimization (SEO) is an important aspect of a Web page to gain importance for a search engine to be able to display it earlier in the search list. The optimization is based on a lot of factors such as title, description, header tags and keyword density. Different search engines will have their own mechanisms that calculate the score of a keyword on the page and thus work out its ranking in the search order.

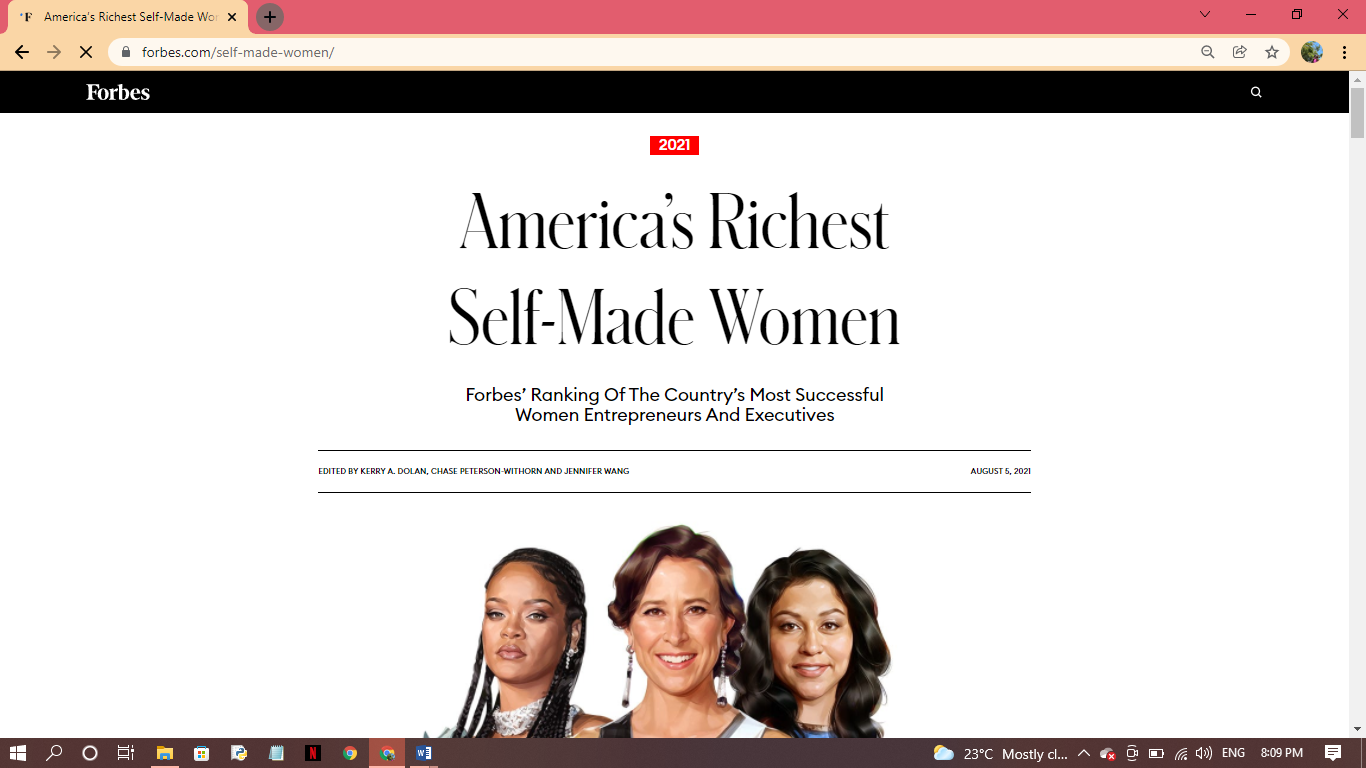
Project Requirements

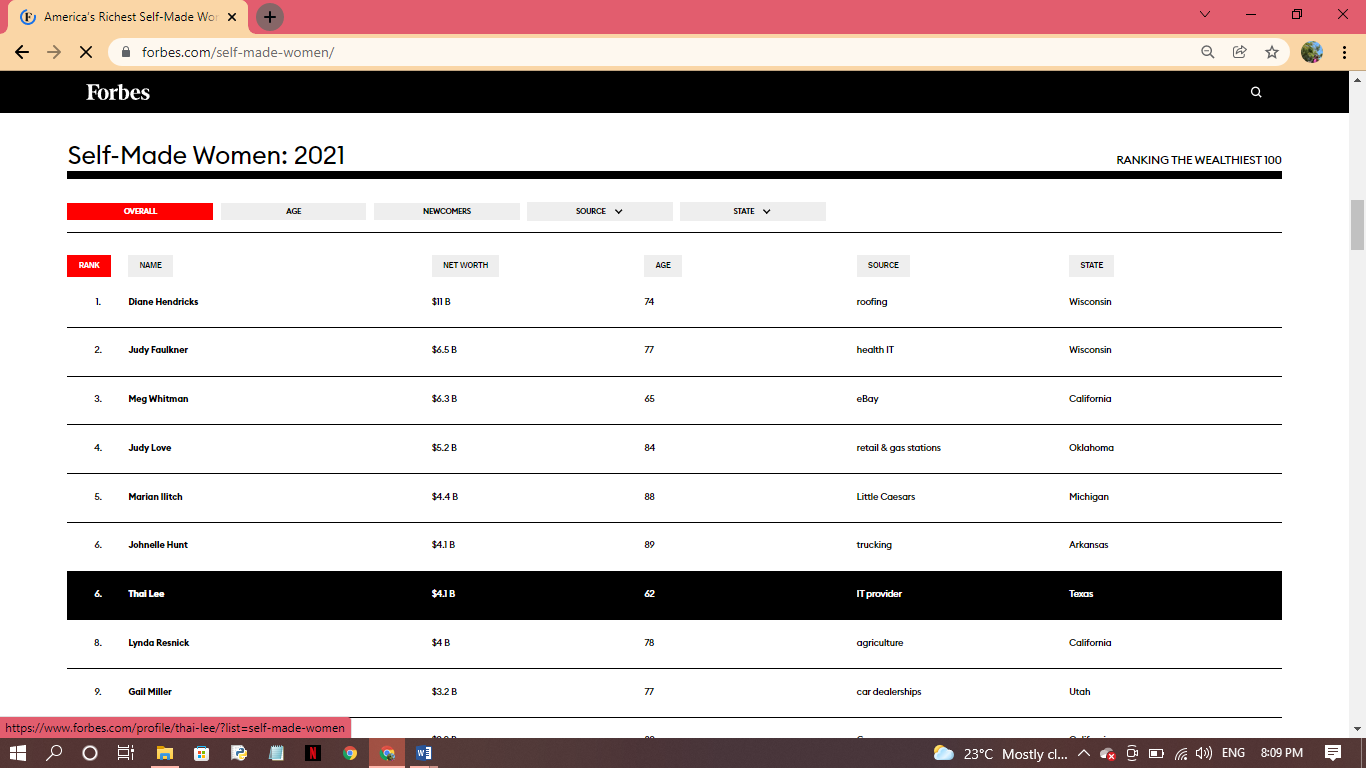
Python Version : python 3.x

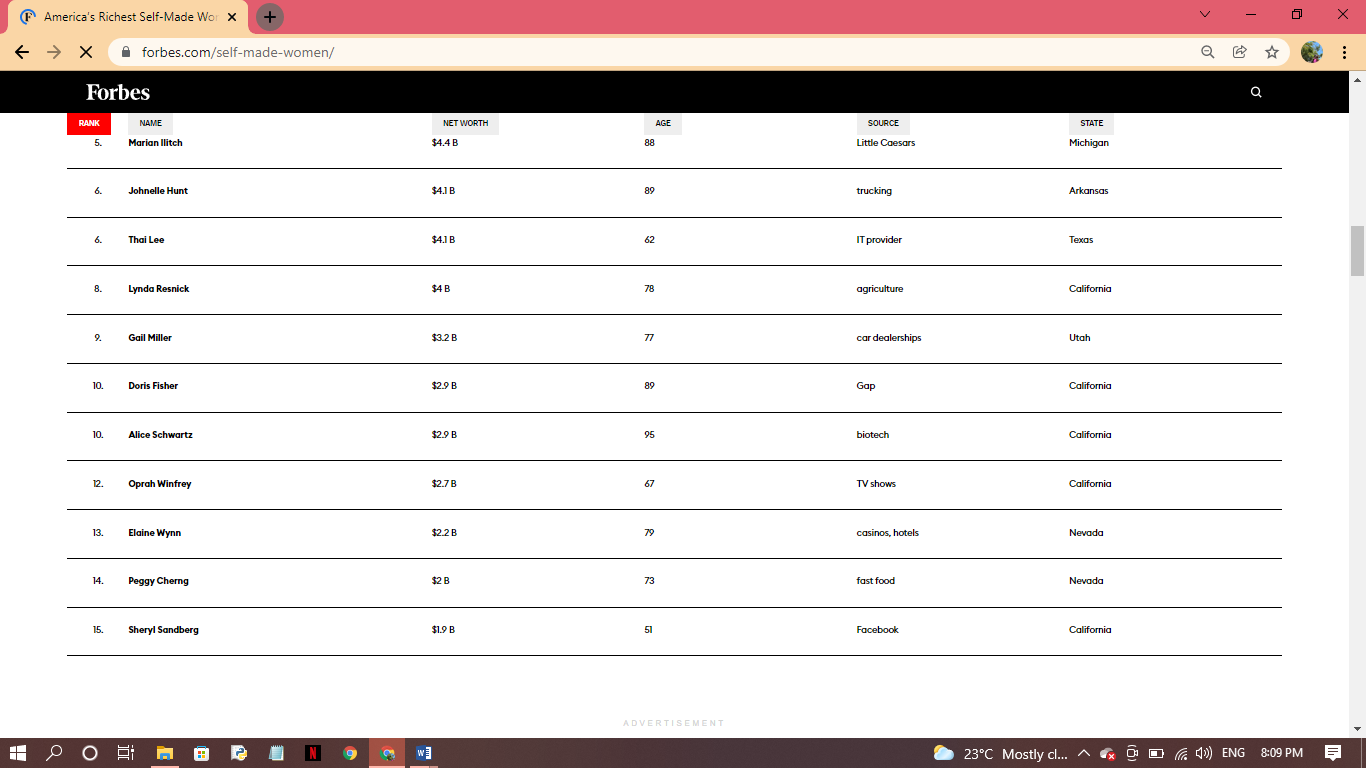
Web Scraping Tool : BeatuifulSoup4

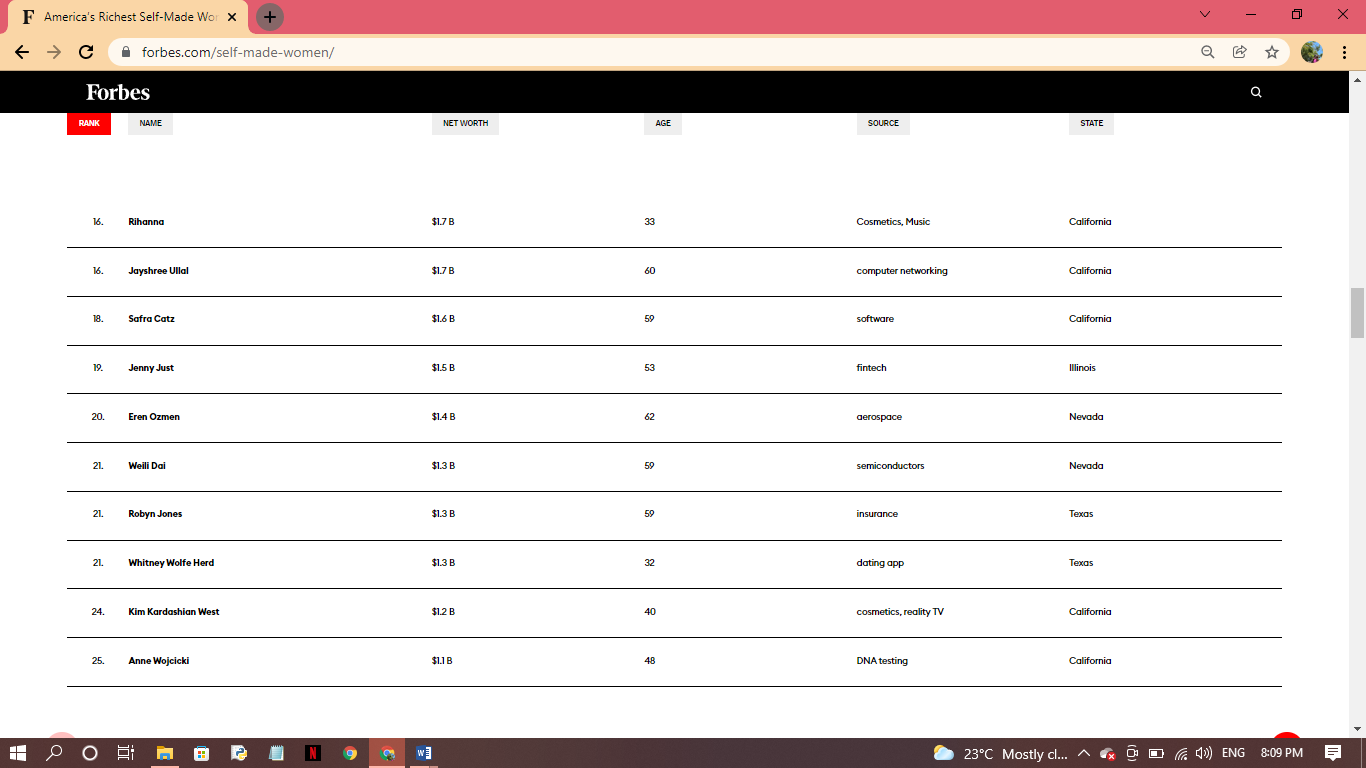
Database : SQLite

IDLE : PyCharm



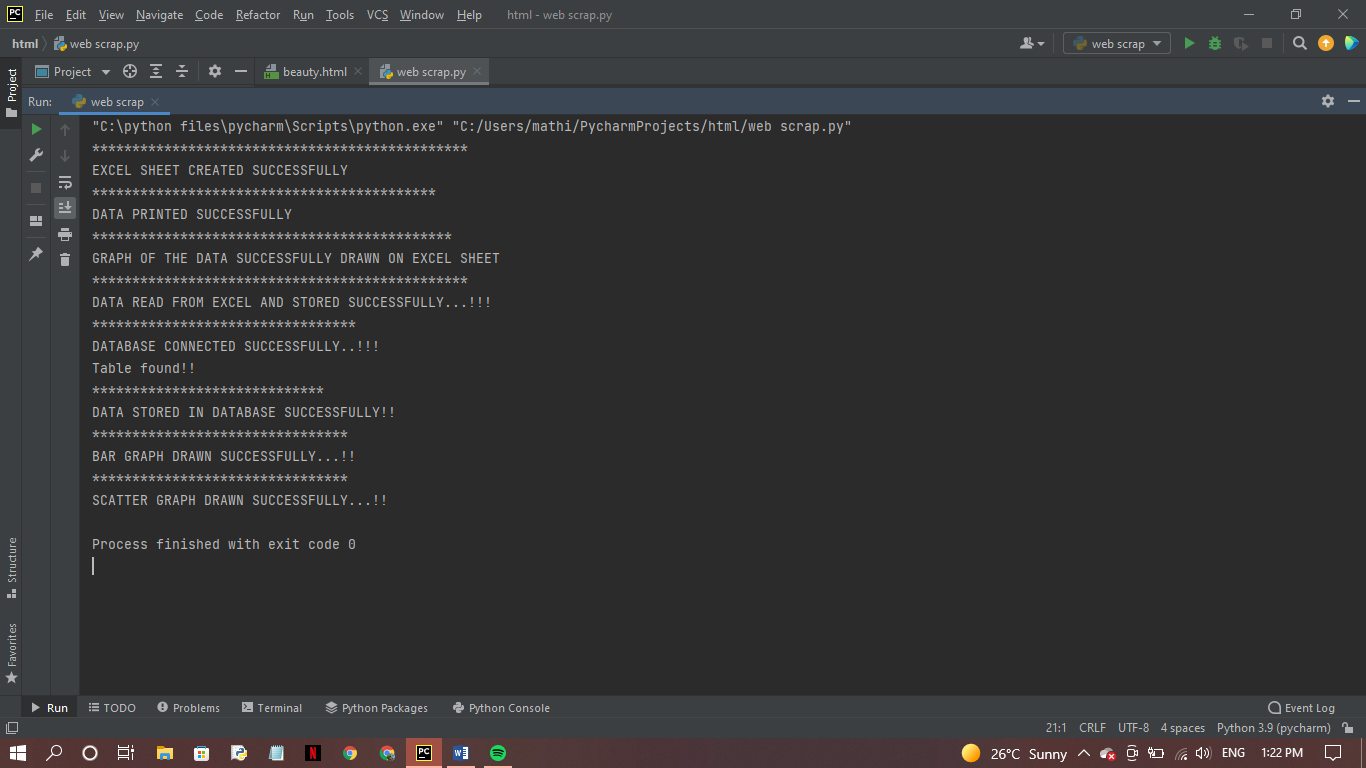


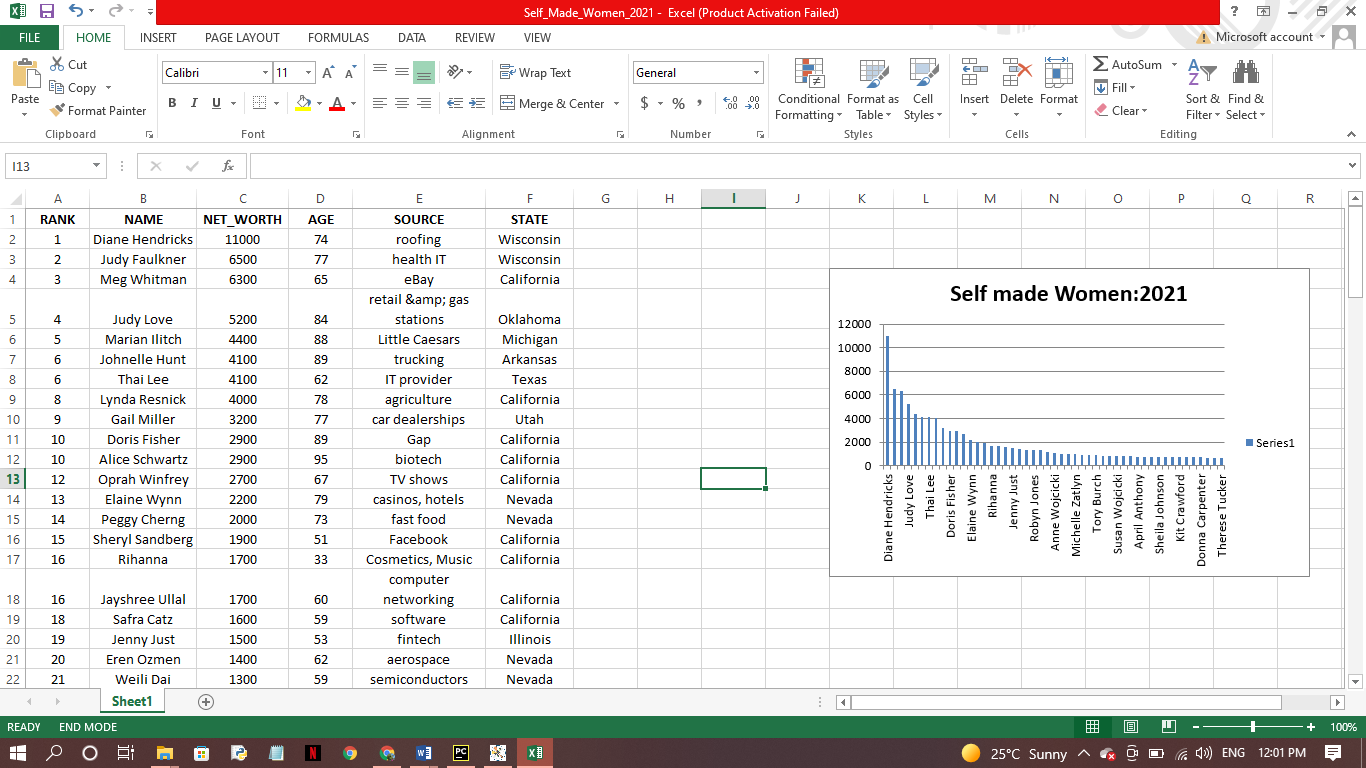


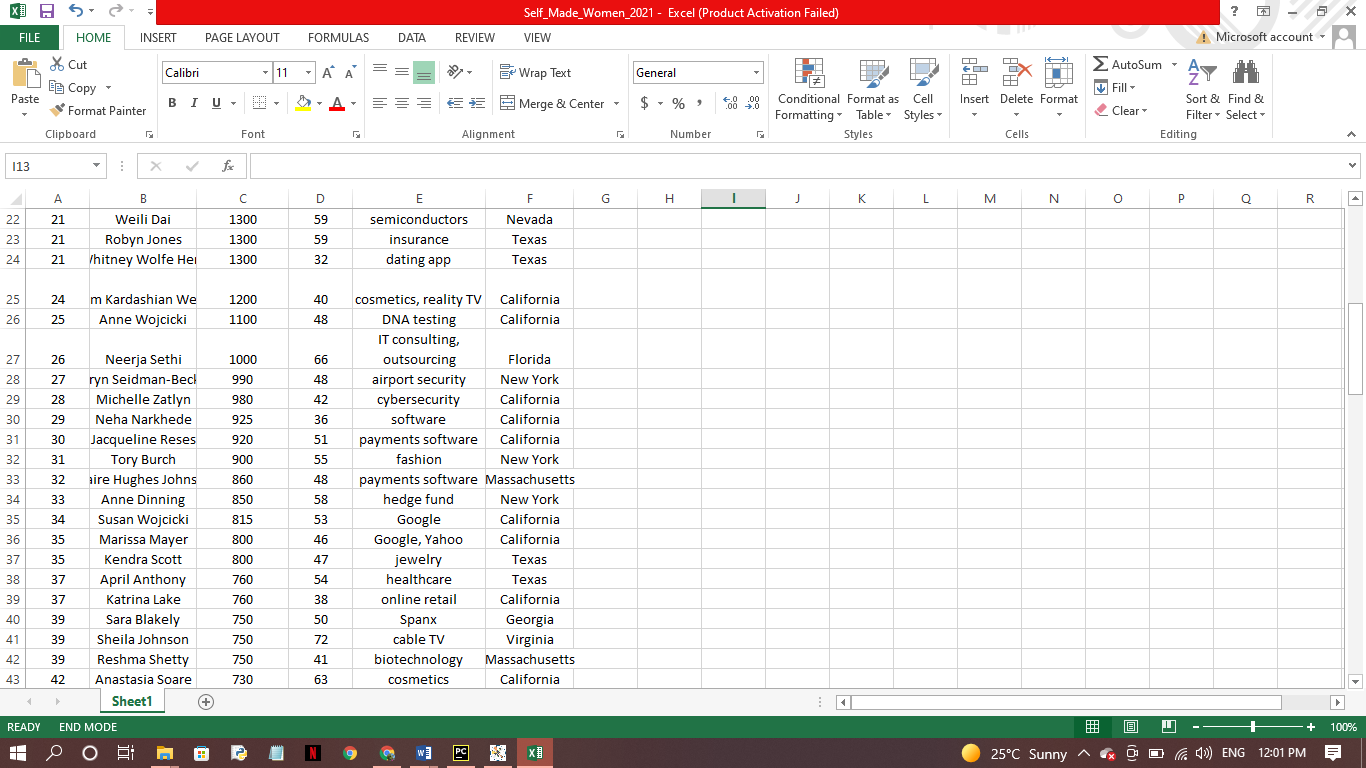


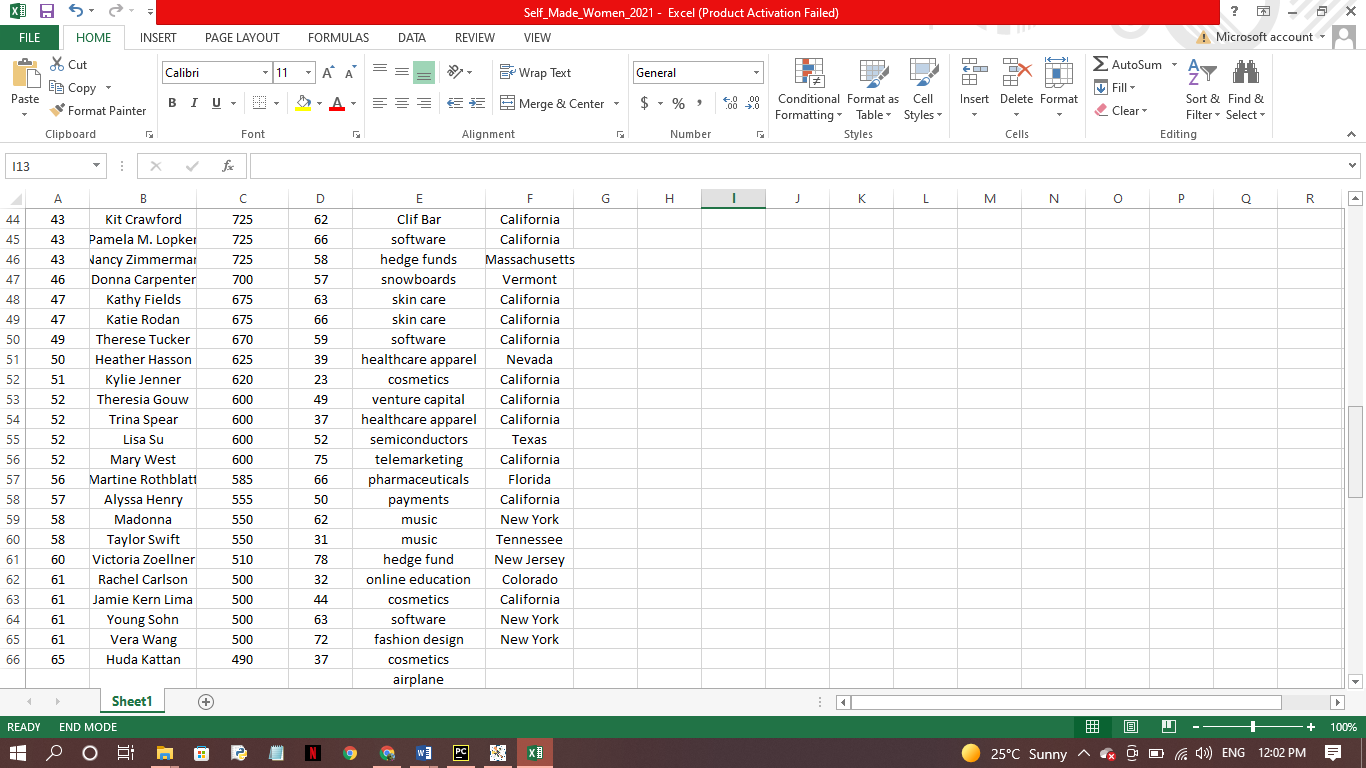
PROGRAM CODE:

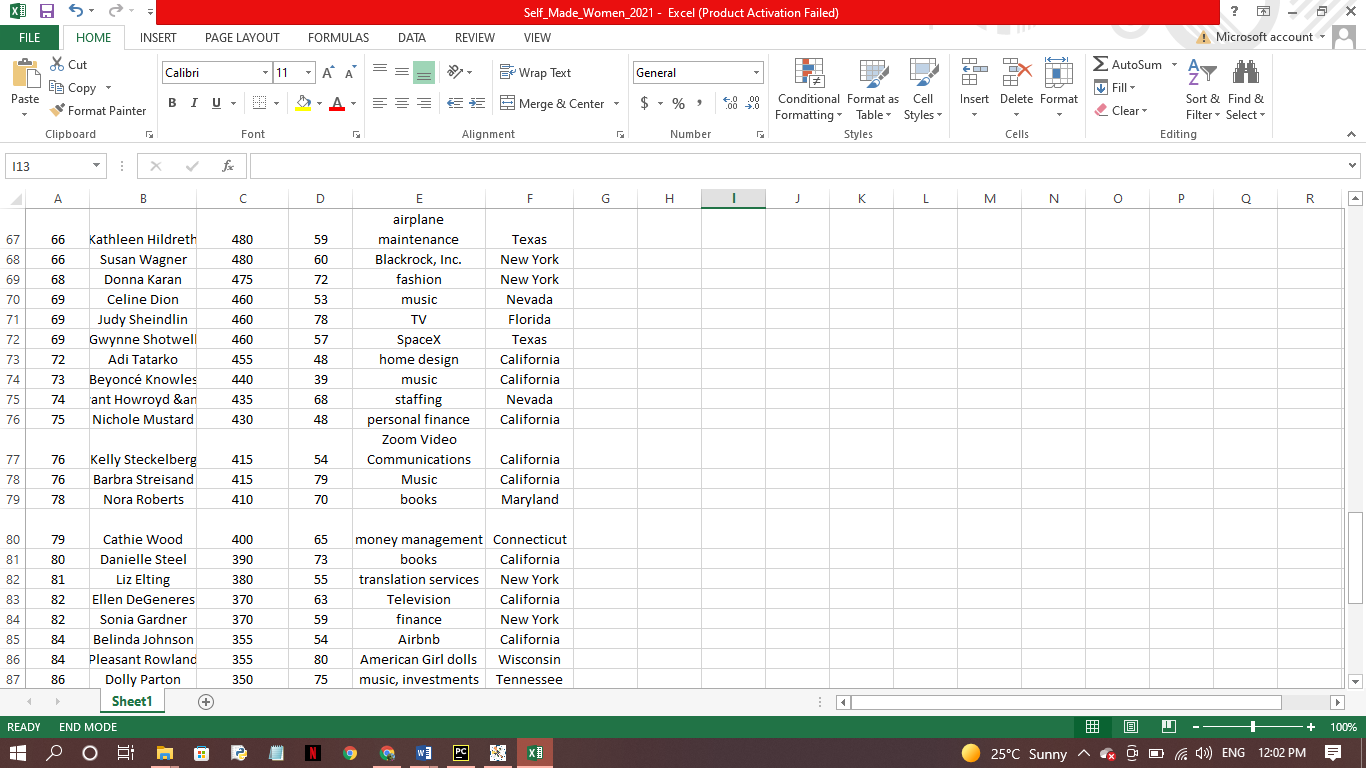
import sqlite3  
from matplotlib import pyplot as plt  
import bs4  
import requests  
import json  
import xlsxwriter  
import xlrd  
  
  
  
class Webscrap:  
 final\_data = []  
 read\_data=[]  
 def data\_extract(self,soup):  
 data = json.loads(str(soup))  
 data1 = data["personList"]["personsLists"]  
 # print(data1)  
 for dt in data1:  
 self.data\_one = [  
 dt.get("rank", ''),  
 dt.get("personName", ''),  
 dt.get("finalWorth", ''),  
 dt.get("age", ''),  
 dt.get("source", ''),  
 dt.get("state", '')  
 ]  
 # print(data\_one)  
 self.final\_data.append(self.data\_one)  
 # print(self.final\_data)  
  
 def excel\_data(self):  
 workbook=xlsxwriter.Workbook("Self\_Made\_Women\_2021.xlsx")  
 worksheet=workbook.add\_worksheet()  
 bold=workbook.add\_format({'bold':True})  
 worksheet.write('A1','RANK',bold)  
 worksheet.write('B1','NAME',bold)  
 worksheet.write('C1','NET\_WORTH',bold)  
 worksheet.write('D1','AGE',bold)  
 worksheet.write('E1','SOURCE',bold)  
 worksheet.write('F1','STATE',bold)  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("EXCEL SHEET CREATED SUCCESSFULLY")  
 row=1  
 col=0  
 for data1 in self.final\_data:  
 worksheet.write(row,col,data1[0])  
 worksheet.write(row,col+1,data1[1])  
 worksheet.write(row, col + 2, data1[2])  
 worksheet.write(row, col + 3, data1[3])  
 worksheet.write(row, col + 4, data1[4])  
 worksheet.write(row, col + 5, data1[5])  
  
 row+=1  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("DATA PRINTED SUCCESSFULLY")  
  
 chart1=workbook.add\_chart({'type':'column'})  
 chart1.add\_series({'categories':'=Sheet1!$B$2:$B$50','values':'=Sheet1!$C$2:$C$50'})  
 #chart1.add\_series({'categories':'=Sheet1!$B$2:$B$50','values':'=Sheet1!$C$2:$C$50'})  
 chart1.set\_title({'name':'Self made Women:2021'})  
 worksheet.insert\_chart('K4',chart1)  
 workbook.close()  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("GRAPH OF THE DATA SUCCESSFULLY DRAWN ON EXCEL SHEET")  
  
 def read\_excel(self):  
 wb=xlrd.open\_workbook("Self\_Made\_Women\_2021.xlsx")  
 worksheet= wb.sheet\_by\_name("Sheet1")  
 num\_rows=worksheet.nrows  
 num\_cols=worksheet.ncols  
  
 for current\_row in range(0,num\_rows,1):  
 row\_review=[]  
 for current\_col in range(0,num\_cols,1):  
 review=worksheet.cell\_value(current\_row,current\_col)  
 row\_review.append(review)  
 self.read\_data.append(row\_review)  
 #print(self.read\_data)  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("DATA READ FROM EXCEL AND STORED SUCCESSFULLY...!!!")  
  
 def data\_base(self):  
 db\_values=self.read\_data  
 con=sqlite3.connect("SELF\_MADE\_WOMEN\_2021.db")  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("DATABASE CONNECTED SUCCESSFULLY..!!!")  
 cur=con.cursor()  
 listofTables=cur.execute("""SELECT 'SELF\_MADE\_WOMEN\_2021' FROM sqlite\_master WHERE type='table'""").fetchall()  
  
 if listofTables==[]:  
 cur.execute('''CREATE TABLE SELF\_MADE\_WOMEN\_2021(  
 Rank INTEGER NOT NULL,  
 Person\_name TEXT,  
 Net\_worth INTEGER NOT NULL,  
 Age INTEGER NOT NULL,  
 Source TEXT,  
 State TEXT);''')  
  
 else:  
 print('Table found!!')  
  
 #print(db\_values)  
  
 cur.executemany("INSERT INTO SELF\_MADE\_WOMEN\_2021(Rank,Person\_name,Net\_worth,Age,Source,State) VALUES(?,?,?,?,?,?)",db\_values)  
 con.commit()  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("DATA STORED IN DATABASE SUCCESSFULLY!!")  
  
 def graph(self):  
 first\_plot=[dt[1] for dt in self.final\_data]  
 second\_plot=[dt[2] for dt in self.final\_data]  
 #print(first\_plot)  
 #print(second\_plot)  
 plt.bar(first\_plot,second\_plot,color='r')  
 plt.legend(["NAMES","NET\_WORTH"])  
 plt.show()  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("BAR GRAPH DRAWN SUCCESSFULLY...!!")  
  
 plt.scatter(first\_plot,second\_plot,label='cases',color='r')  
 plt.show()  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("SCATTER GRAPH DRAWN SUCCESSFULLY...!!")  
  
  
  
  
  
urllink="https://www.forbes.com/forbesapi/person/self-made-women/2021/position/true.json"  
header={  
'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/95.0.4638.69 Safari/537.36'  
}  
  
response=requests.get(url=urllink,headers=header)  
soup=bs4.BeautifulSoup(response.content,"html.parser")  
#print(soup.prettify())  
w=Webscrap()  
w.data\_extract(soup)  
w.excel\_data()  
w.read\_excel()  
w.data\_base()  
w.graph()

OUTPUT:

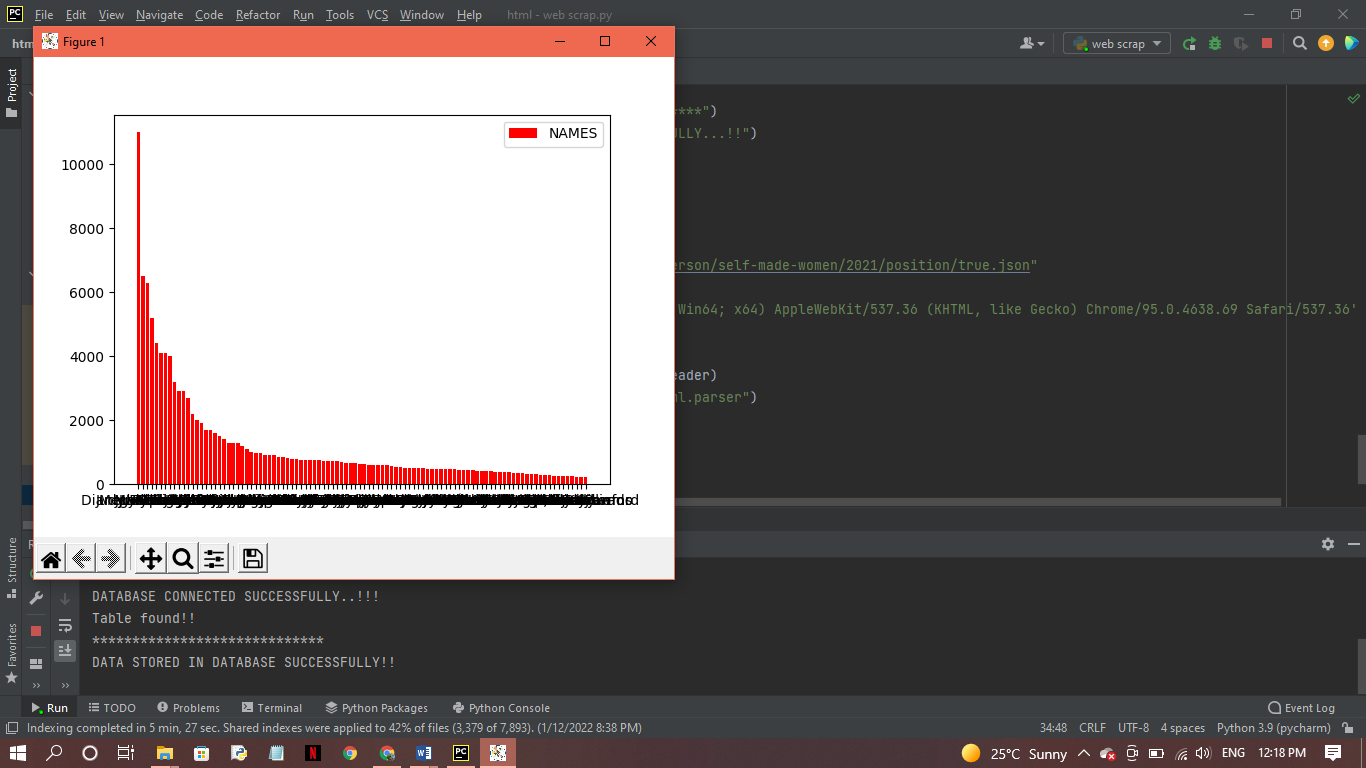
EXCEL SHEET OF ALL DATAS:



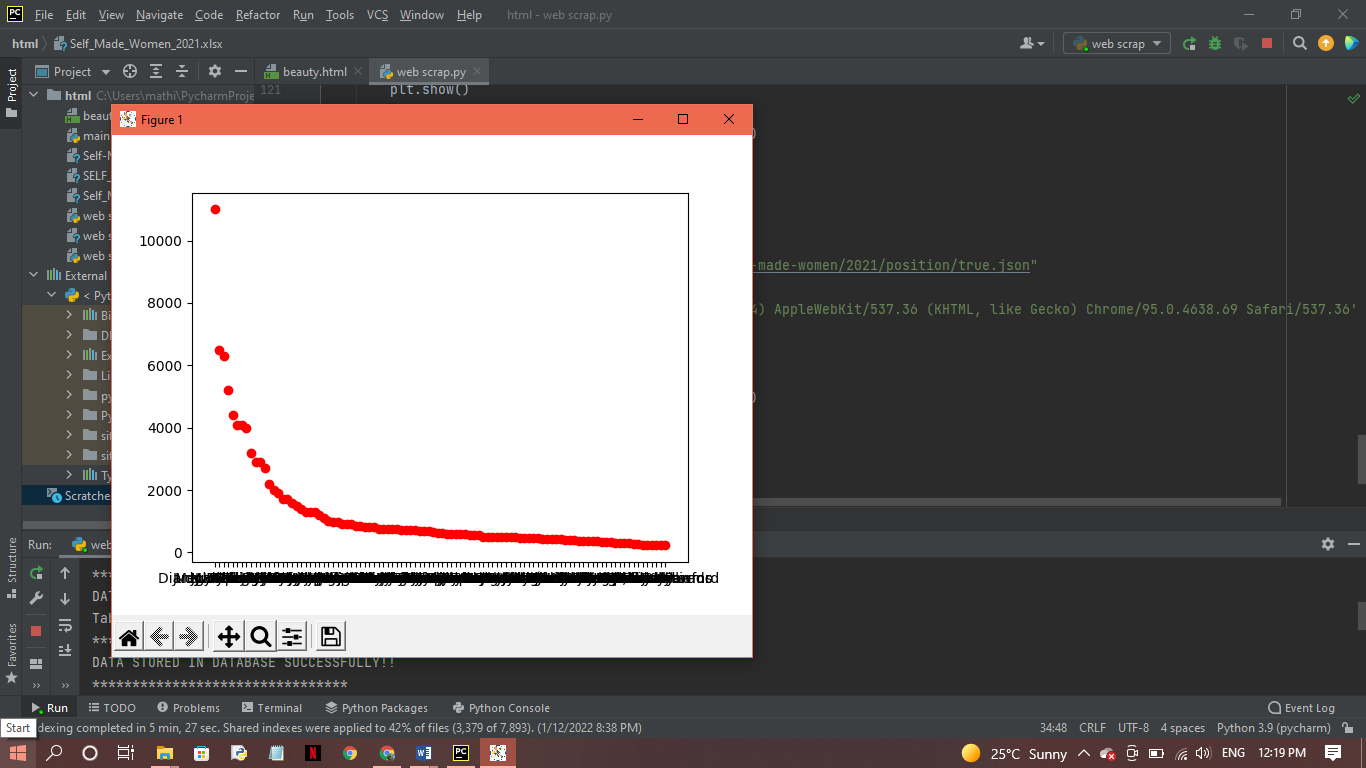




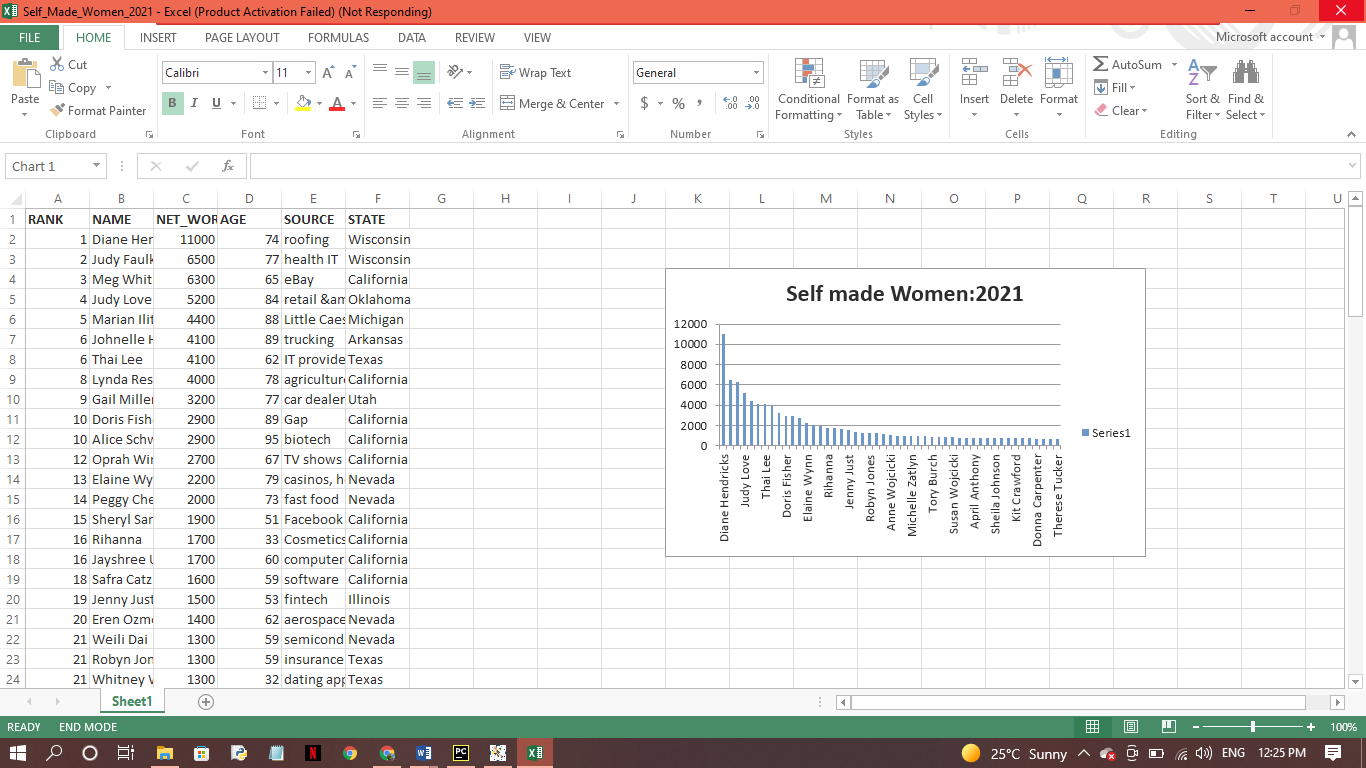
BAR GRAPH OF THE WOMEN AND THEIR NET WORTH:



SCATTER PLOT OF THE WOMEN AND THEIR NET WORTH:



EXCEL GRAPH:



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*