Project: Investment Advisor

Team Members:

Aman Kumar

Arnold P Sunny

Shankar Pradhan

Sumit Roy

Links:

Google Sheet:

https://docs.google.com/spreadsheets/d/15fb9txIqcOIwysXVUSZy5kgQHeW5UFfZNUF_Cwl4RRc/edit?usp=sharing

Python File:

project.py

Library Used:

Pandas

Numpy

Gspread

Matplotlib

CODE:

```
import gspread
import pandas as pd
from oauth2client.service_account import ServiceAccountCredentials
from pprint import pprint as pp
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np


scope =
["https://spreadsheets.google.com/feeds",'https://www.googleapis.com/auth/spre
```

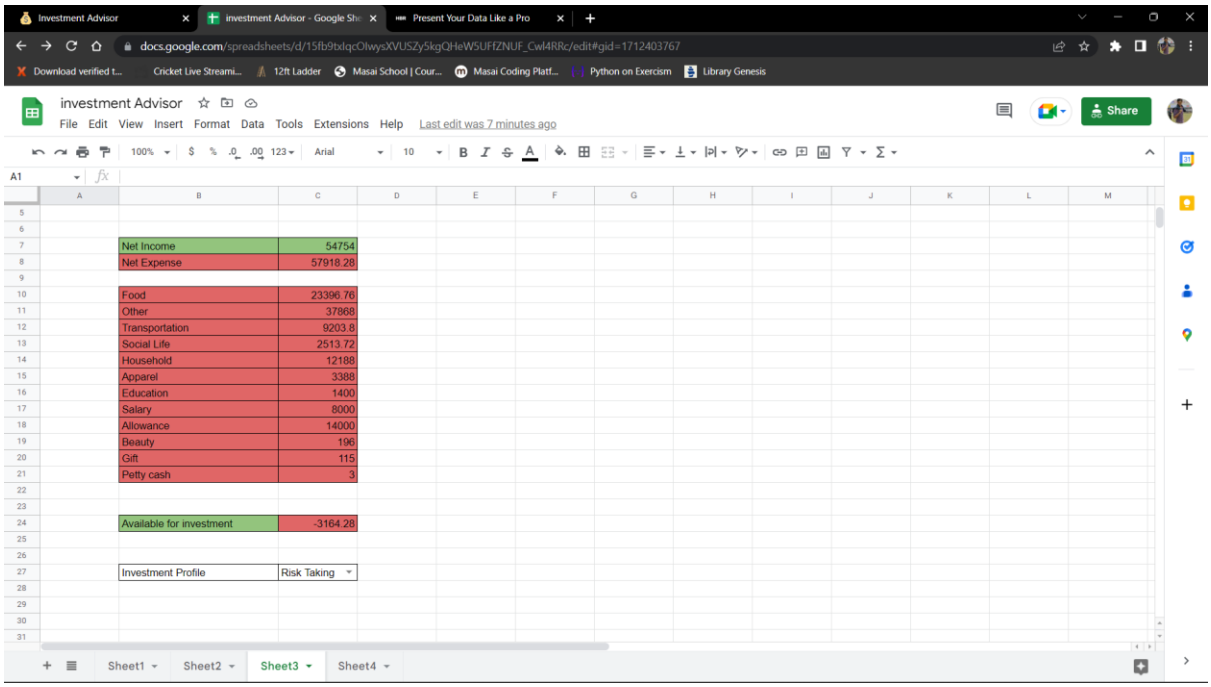
```
adsheets','https://www.googleapis.com/auth/drive.file","https://www.googleapis
.com/auth/drive"]

cred = ServiceAccountCredentials.from_json_keyfile_name("cred.json",scope)
client = gspread.authorize(cred)

sheet = client.open("investment Advisor").worksheet("Sheet1")
sheet2=client.open("investment Advisor").worksheet("Sheet2")
sheet3=client.open("investment Advisor").worksheet("Sheet3")
sheet4=client.open("investment Advisor").worksheet("Sheet4")

bse500=sheet.get_all_records()
inex=sheet2.get_all_values()
f_report1=sheet3.get_all_values()
f_report2=sheet4.get_all_values()


df_bse=pd.DataFrame(bse500)
df_inex=pd.DataFrame(inex)

#changing the datatype to float from str
df_bse["10-Year Return(%)"] = pd.to_numeric(df_bse["10-Year Return(%)"],
downcast="signed")

#filling the blank values with nan
df_bse['10-Year Return(%)'] = df_bse['10-Year Return(%)'].fillna(0)

# finding the differnece in bse500 in new column delta

df_bse['delta']=(df_bse['52 Week High'].sub(df_bse['Price'], axis =
0).div(df_bse['52 Week High']))
high_risk1= df_bse['Market Cap(Cr)']<2000
high_risk2=df_bse['delta']>0
high_risk3=(df_bse['10-Year Return(%)']< 8 )
# print(df_bse[high_risk3])
# print(df_bse[high_risk1])
# print(df_bse[high_risk2])

#merged all the dataframe
high_r=pd.merge(df_bse[high_risk1],df_bse[high_risk2])
high_risk=pd.merge(df_bse[high_r],df_bse[high_risk3])
# high_risk = df_bse.sort_values(high_risk['Dividend Per Share'],ascending ==
False).head(5)
# print (df_bse[high_risk])
if high_risk.empty:
    print("There are no companies who take High Risk")


#Risk Taking
```

```
risk1= df_bse['Market Cap(Cr)'].between(2000,5000)
risk2=df_bse['delta']>0
risk3=df_bse['10-Year Return(%)'].between(8,15)
# print(df_bse[risk1])
# print(df_bse[risk2])
# print(df_bse[risk3])
risk_r=df_bse[risk1].merge(df_bse[risk2], how = 'inner' ,indicator=False)
risk=risk_r.merge(df_bse[risk3], how = 'inner' ,indicator=False).head(5)
print(risk)
print("The Company Taking Risk are")
for i in range (4):
    print(risk.loc[i]['Company'])
    sheet4.update_cell(4+i,2,risk.loc[i]['Company'])

sheet4.update_cell(3,2,'Risk:::')

# Moderate Risk taking

moderate1= df_bse['Market Cap(Cr)'].between(2000,15000)
moderate2=df_bse['delta']>0
moderate3=df_bse['10-Year Return(%)'].between(15,20)
moderate_r=df_bse[moderate1].merge(df_bse[moderate2], how = 'inner'
,indicator=False)
moderate=moderate_r.merge(df_bse[moderate3], how = 'inner'
,indicator=False).head(5)
print(moderate)
print("The Company Taking Moderate Risk are")
for i in range (5):
    print(moderate.loc[i]['Company'])
    sheet4.update_cell(8+i,2,moderate.loc[i]['Company'])
sheet4.update_cell(7,2,'Moderate Risk:::')

#Low Risk

low1= df_bse['Market Cap(Cr)']>15000
low2=df_bse['delta']>0
low3=df_bse['10-Year Return(%)']>20
low_r=df_bse[low1].merge(df_bse[low2], how = 'inner' ,indicator=False)
low=low_r.merge(df_bse[low3], how = 'inner' ,indicator=False).head(5)
print(low)
print("The Company Taking Low Risk are")
for i in range (5):
    print(low.loc[i]['Company'])
    sheet4.update_cell(13+i,2,low.loc[i]['Company'])
sheet4.update_cell(12,2,'Low Risk:::')
```

```
##############################################

# Task 1


# df_inex["10-Year Return(%)"] = pd.to_numeric(df_inex["10-Year Return(%)"],
downcast="signed")
# print(df_inex.columns.tolist())

# Changed the header
df_inex=pd.DataFrame(inex)
header_row = df_inex.iloc[0]
df_inex2 = pd.DataFrame(df_inex.values[1:], columns=header_row)
df_inex2["Amount"] = pd.to_numeric(df_inex2["Amount"], downcast="signed")
# print(df_inex2)
inc=df_inex2.groupby(['Income/Expense'])[['Amount']].sum()
expense=inc.loc['Expense']['Amount']
income=inc.loc['Income']['Amount']
# print(expense)
# print(income)
inc2=df_inex2.groupby(['Category'])[['Amount']].sum()
# print(inc2)
Allowance=inc2.loc['Allowance']['Amount']
Apparel=inc2.loc['Apparel']['Amount']
Education=inc2.loc['Education']['Amount']
Food=inc2.loc['Food']['Amount']
Gift=inc2.loc['Gift']['Amount']
Other=inc2.loc['Other']['Amount']
Salary=inc2.loc['Salary']['Amount']
Pettycash=inc2.loc['Petty cash']['Amount']
Self_development=inc2.loc['Self-development']['Amount']
Social_Life=inc2.loc['Social Life']['Amount']
Transportation=inc2.loc['Transportation']['Amount']
Beauty=inc2.loc['Beauty']['Amount']
Household=inc2.loc['Household']['Amount']

# print(Allowance)
# print(Apparel)

sheet3.update_cell(7,3,income)
sheet3.update_cell(8,3,expense)
sheet3.update_cell(10,3,Food)
sheet3.update_cell(11,3,Other)
sheet3.update_cell(12,3,Transportation)
sheet3.update_cell(13,3,Social_Life)
sheet3.update_cell(14,3,Household)
sheet3.update_cell(15,3,Apparel)
sheet3.update_cell(16,3,Education)
sheet3.update_cell(17,3,Salary)
```

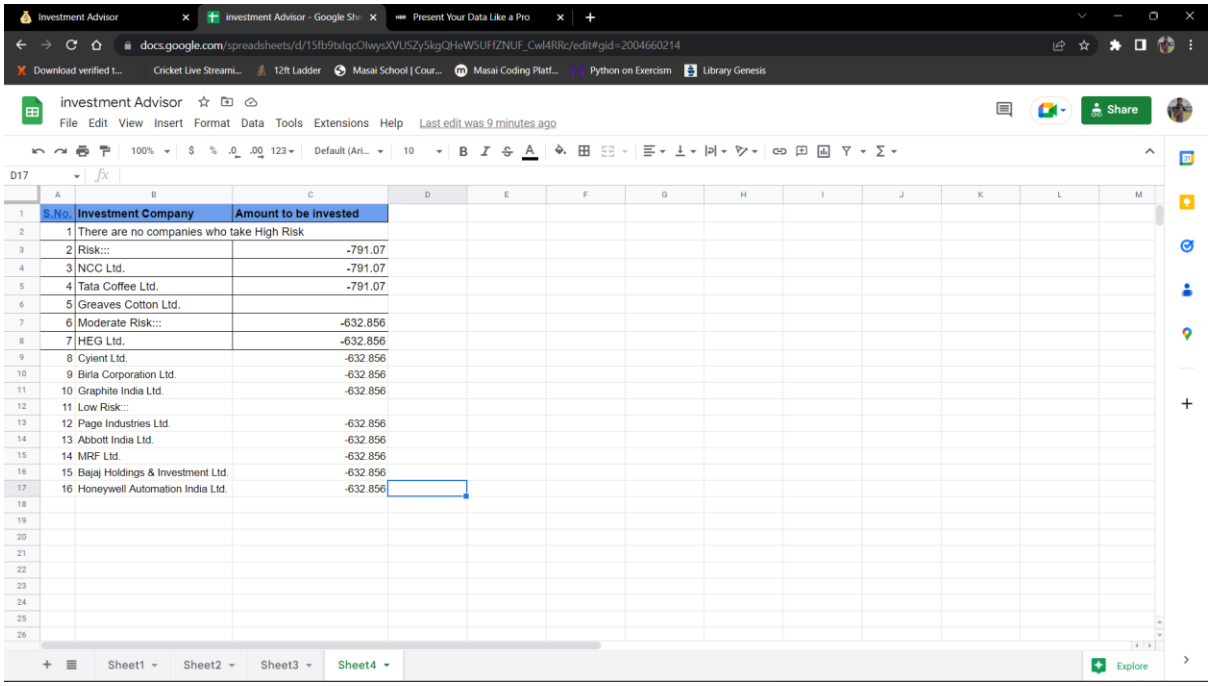
```
sheet3.update_cell(18,3,Allowance)
sheet3.update_cell(19,3,Beauty)
sheet3.update_cell(20,3,Gift)
sheet3.update_cell(21,3,Pettycash)


#
avaliable=inc.loc['Income']['Amount'].sub(inc.loc['Income']['Amount'],axis=0)
avaliable=(income-expense)
# print(avaliable)
sheet3.update_cell(24,3,avaliable)

sheet4.update_cell(3,3,avaliable/4)
sheet4.update_cell(4,3,avaliable/4)
sheet4.update_cell(4,3,avaliable/4)
sheet4.update_cell(5,3,avaliable/4)
sheet4.update_cell(7,3,avaliable/5)
sheet4.update_cell(8,3,avaliable/5)
sheet4.update_cell(9,3,avaliable/5)
sheet4.update_cell(10,3,avaliable/5)
sheet4.update_cell(11,3,avaliable/5)
sheet4.update_cell(13,3,avaliable/5)
sheet4.update_cell(14,3,avaliable/5)
sheet4.update_cell(15,3,avaliable/5)
sheet4.update_cell(16,3,avaliable/5)
sheet4.update_cell(16,3,avaliable/5)

################################
# df_bse=pd.DataFrame(bse500)
# print(df_bse.dtypes)
df_bse["Enterprise Value(Cr)"] = pd.to_numeric(df_bse["Enterprise Value(Cr)"],
downcast="signed")
enterprice=df_bse.groupby(['Sector'])[['Enterprise Value(Cr)']].median()

sect=df_bse.Sector.unique()
# print(sect)
xp=np.array(sect)
yp=np.array(enterprice)

plt.plot(xp, yp)
plt.show()
# print(sect)
# print(enterprice)

# industry=df_bse.Industry.unique()
# df_bse["3-Year Return"] = pd.to_numeric(df_bse["3-Year Return"],
downcast="signed")
```

Final Report:



investment Advisor

Net Income	54754
Net Expense	57918.28
Food	23396.76
Other	37868
Transportation	9203.8
Social Life	2513.72
Household	12188
Apparel	3388
Education	1400
Salary	8000
Allowance	14000
Beauty	196
Gift	115
Petty cash	3
Available for investment	-3164.28
Investment Profile	Risk Taking



investment Advisor

S.No.	Investment Company	Amount to be invested
1	There are no companies who take High Risk	
2	Risk:::	-791.07
3	NCC Ltd.	-791.07
4	Tata Coffee Ltd.	-791.07
5	Greaves Cotton Ltd.	
6	Moderate Risk:::	-632.856
7	HEG Ltd.	-632.856
8	Cyient Ltd.	-632.856
9	Birla Corporation Ltd.	-632.856
10	Graphite India Ltd.	-632.856
11	Low Risk:::	
12	Page Industries Ltd.	-632.856
13	Abbott India Ltd.	-632.856
14	MRF Ltd.	-632.856
15	Bajaj Holdings & Investment Ltd.	-632.856
16	Honeywell Automation India Ltd.	-632.856

Plotted a graph to get the relation between Market cap and Sector: