第一次大作业实验报告

PB20010382 王淏然

0.说明

本次大作业我写了两个小程序，其中第一个是练练手，关于获取科学家姓名，返回科学家的文献；
第二个小程序是最终的作业，关于股票数据的获取和展示。

1. 软件功能描述

程序1：
用户输入科学家姓名后，搜索爬取的数据，爬取的数据中包括科学家的前10篇文献的名称，右下角显示实时的时间
程序2：
用户输入股票代码CODE和对应公司name的名称，我们去爬取对应股票的数据，用户需要输入mission的编号 1_total assets 2_revenue 3_pe-ratio，用户点击search按钮进行搜索，点击save按钮会自动在用户桌面上生成一个csv文件，文件名为CODE_name.csv

1.1 程序一：科学家文献搜索

目的：为用户提供一个便捷的方式，输入科学家的姓名，快速获取该科学家的前十篇文献。
输入：用户输入的科学家姓名。
输出：显示科学家的前十篇文献的名称，并在右下角显示实时的时间。
使用场景：当用户需要快速了解某位科学家的研究成果时，可以使用此程序进行查询。

1.2 程序二：股票数据爬取与展示

目的：帮助用户快速获取特定股票的数据，并提供数据保存功能。
输入：用户输入的股票代码、公司名称和特定的任务编号（如总资产、营收、市盈率等）。
输出：展示对应股票的数据。用户可以选择保存此数据为CSV文件。
使用场景：当用户需要查找特定公司的股票数据时，或者需要保存某些股票数据进行后续分析时，可以使用此程序。

2. 软件设计

注意到qtdesigner可以很好的辅助我，比如把pushbutton换一个颜色和透明度，换一个背景；
比如把label的字体颜色换成白色，这样就可以在黑色背景上显示白色字体
显示实时时间的功能比较麻烦，需要自己在main_window.py里添加代码：

```
from PyQt5.QtCore import QTime, QTimer
self.Timer=QTimer(self)#括号内的self是必须的,不然会报错
self.Timer.timeout.connect(self.update_time)
self.Timer.start(1000)#1000是指1000ms即1秒
def update_time(self):#这个self是指类里面的函数
    curr_time=QTime.currentTime()
    self.timeLabel.setText(curr_time.toString('hh:mm:ss'))
```

2.1 class APP()

2.1.1 简介

class APP 是一个继承自 QWidget 的类，连接窗口main_window和自己写的函数，
对于程序一，实现了用户输入科学家姓名后，搜索爬取的数据，爬取的数据中包括科学家的前10篇文献的名称
对于程序二，实现了用户输入股票代码CODE和对应公司name的名称，我们去爬取对应股票的数据

2.1.2 类内函数

fetch_table_data

通过request,运用随机的headers,爬取网页数据，返回一个soup对象
同时根据网页的selector路径，返回一个target_data对象

fetch_table_data_alternative

与fetch_table_data类似，均运用了try,exception语句
加了一个mission_index的参数输入，用于拓展任务

save_to_csv

将爬取的数据存储为csv文件，文件名为CODE_name.csv
同时，用os获取桌面路径，通过pandas.DataFrame.to_csv()函数，将数据存储为csv文件

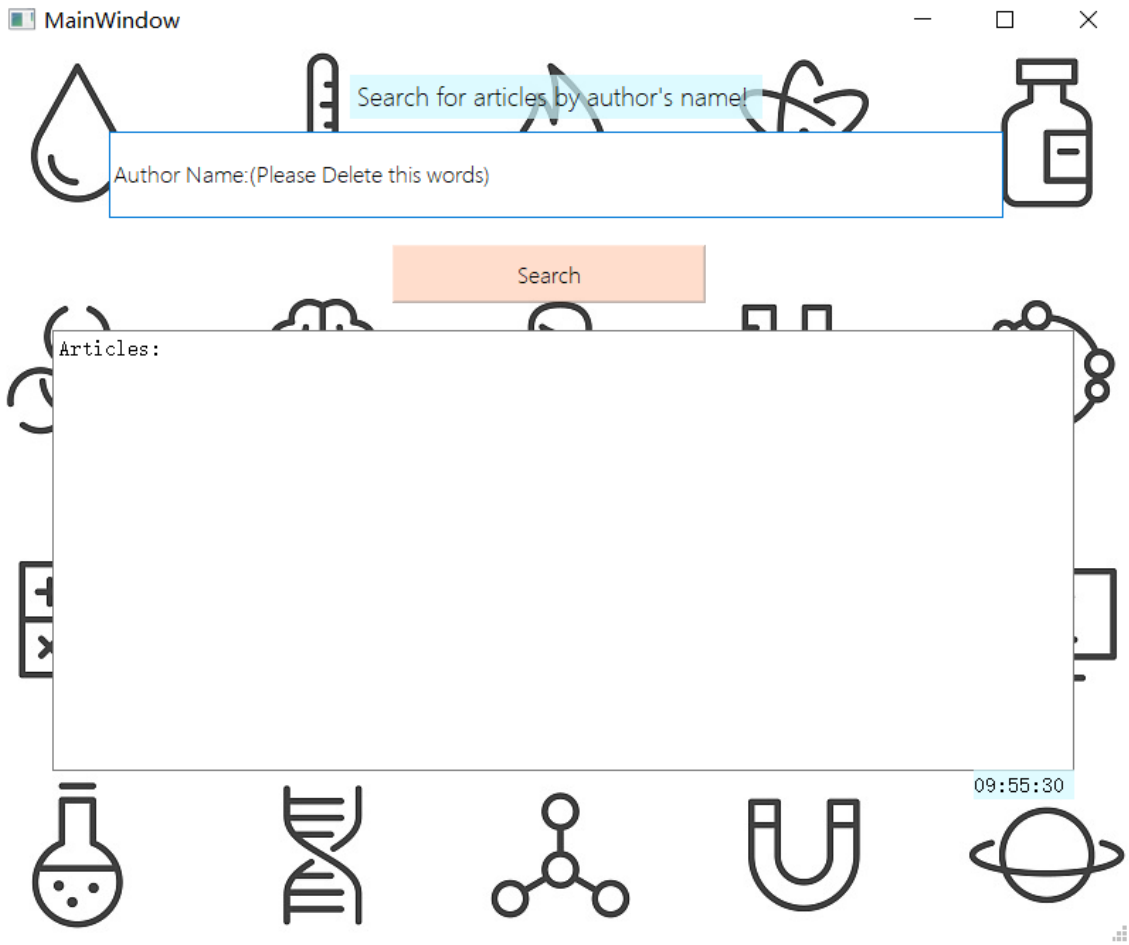
3. 设计过程中遇到的困难及解决方法

1.本来想做一个直接可以模拟点击网页中'download'按钮的应用程序，但是发现这个按钮是用js写的，所以没办法直接模拟点击，只能爬取数据后自己生成csv文件
2.在爬取数据中我发现，一些数据格内会有链接，但是我不知道怎么爬取这个链接，所以只能爬不带有链接的文本
3.我在第一个应用程序中写了显示实时时间的程序，后来发现没啥实际用处，不如save按钮实用，所以在这个应用程序中就没有写显示实时时间的程序了

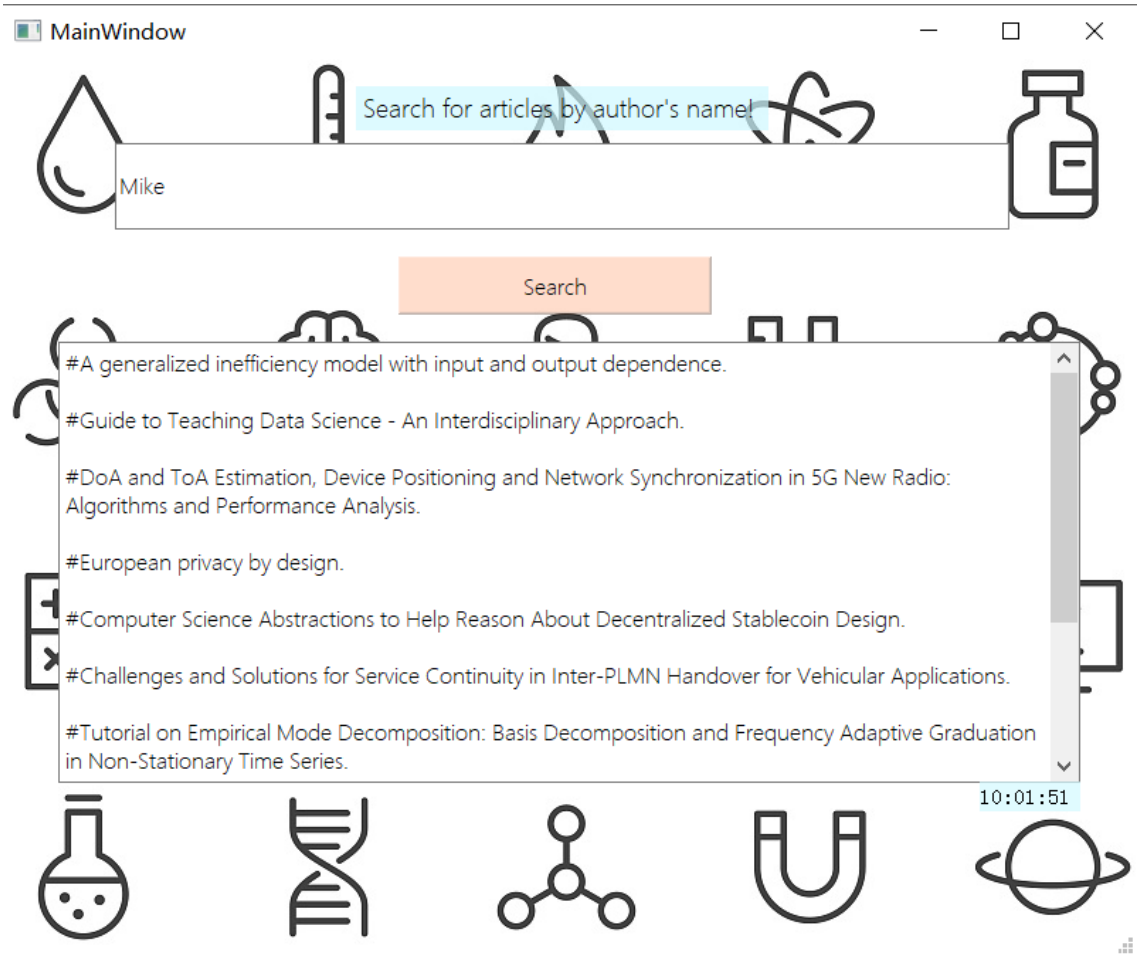
4. 图片和示例

程序一:

- **User Interface**

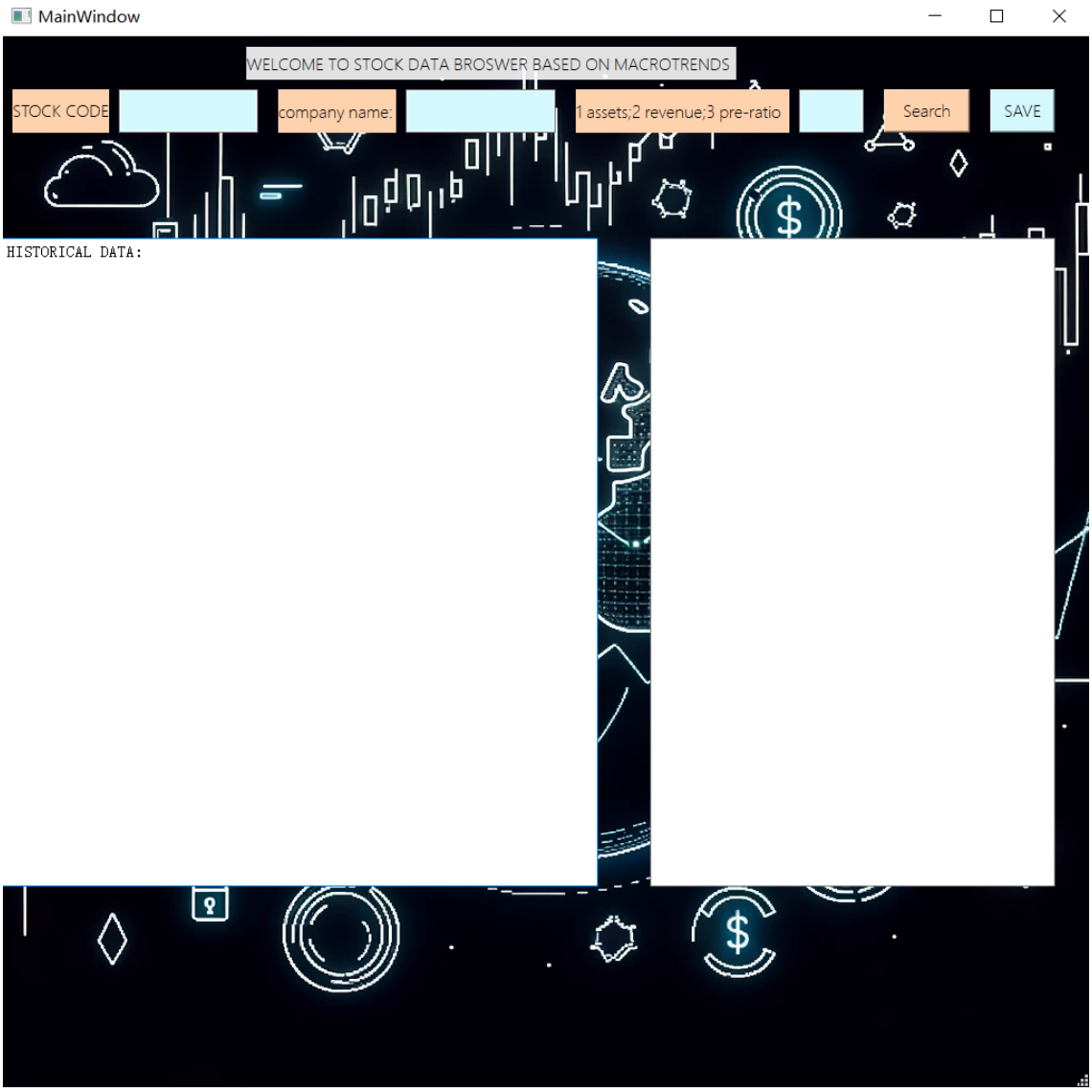


- **Example**

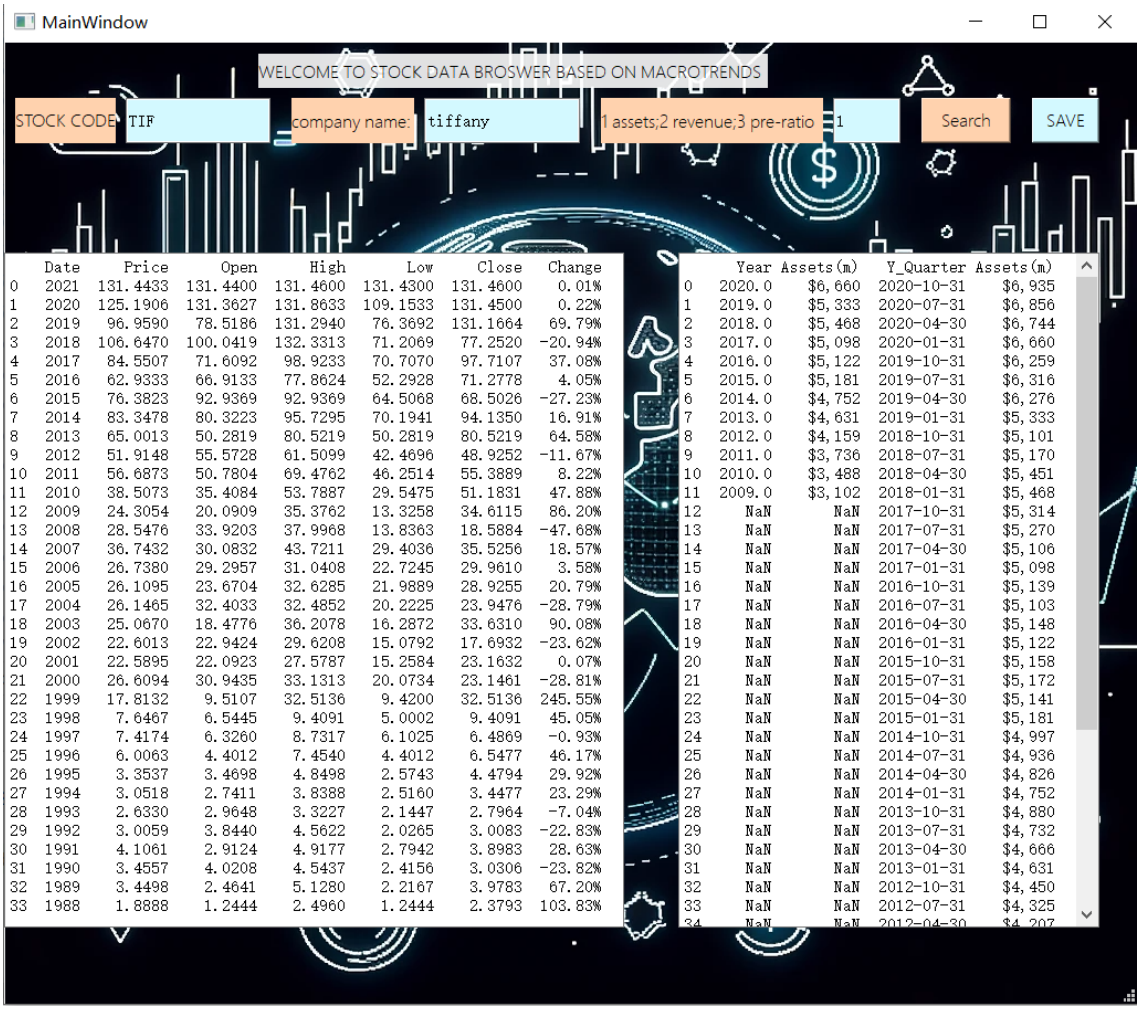

MainWindow
Search for articles by author's name!
Mike
Search
#A generalized inefficiency model with input and output dependence.
#Guide to Teaching Data Science - An Interdisciplinary Approach.
#DoA and ToA Estimation, Device Positioning and Network Synchronization in 5G New Radio: Algorithms and Performance Analysis.
#European privacy by design.
#Computer Science Abstractions to Help Reason About Decentralized Stablecoin Design.
#Challenges and Solutions for Service Continuity in Inter-PLMN Handover for Vehicular Applications.
#Tutorial on Empirical Mode Decomposition: Basis Decomposition and Frequency Adaptive Graduation in Non-Stationary Time Series.
10:01:51

程序二:

- **User Interface**



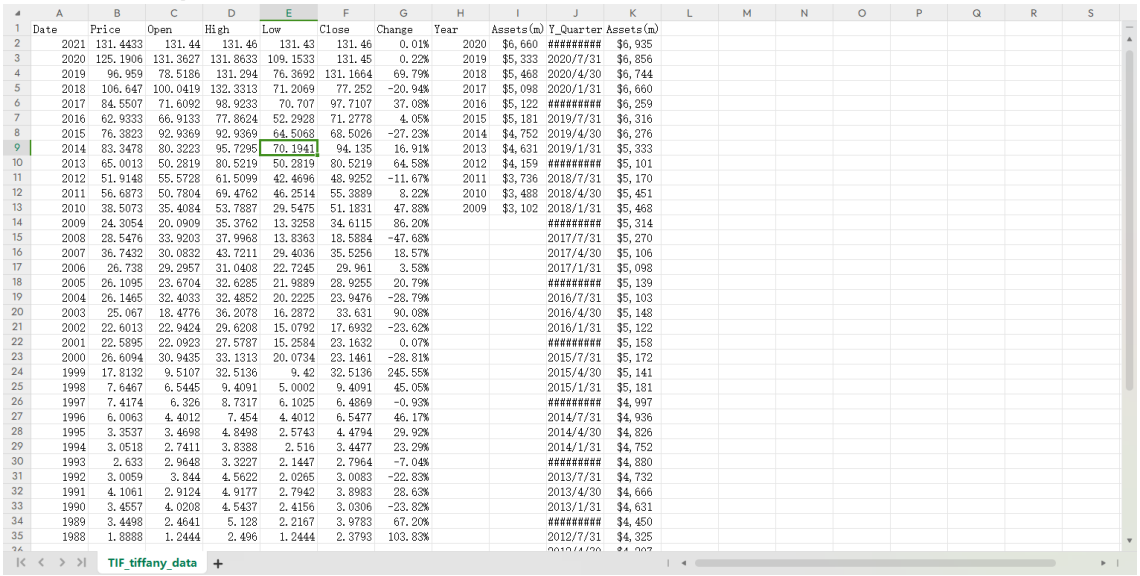
- **Example**



	Date	Price	Open	High	Low	Close	Change
0	2021	131.4433	131.4400	131.4600	131.4300	131.4600	0.01%
1	2020	125.1906	131.3627	131.8633	109.1533	131.4500	0.22%
2	2019	96.9590	78.5186	131.2940	76.3692	131.1664	69.79%
3	2018	106.6470	100.0419	132.3313	71.2069	77.2520	-20.94%
4	2017	84.5507	71.6092	98.9233	70.7070	97.7107	37.08%
5	2016	62.9333	66.9133	77.8624	52.2928	71.2778	4.05%
6	2015	76.3823	92.9369	92.9369	64.5068	68.5026	-27.23%
7	2014	83.3478	80.3223	95.7295	70.1941	94.1350	16.91%
8	2013	65.0013	50.2819	80.5219	50.2819	80.5219	64.58%
9	2012	51.9148	55.5728	61.5099	42.4696	48.9252	-11.67%
10	2011	56.6873	50.7804	69.4762	46.2514	55.3889	8.22%
11	2010	38.5073	35.4084	53.7887	29.5475	51.1831	47.88%
12	2009	24.3054	20.0909	35.3762	13.3258	34.6115	86.20%
13	2008	28.5476	33.9203	37.9968	13.8363	18.5884	-47.68%
14	2007	36.7432	30.0832	43.7211	29.4036	35.5256	18.57%
15	2006	26.7380	29.2957	31.0408	22.7245	29.9610	3.58%
16	2005	26.1095	23.6704	32.6285	21.9889	28.9255	20.79%
17	2004	26.1465	32.4033	32.4852	20.2225	23.9476	-28.79%
18	2003	25.0670	18.4776	36.2078	16.2872	33.6310	90.08%
19	2002	22.6013	22.9424	29.6208	15.0792	17.6932	-23.62%
20	2001	22.5895	22.0923	27.5787	15.2584	23.1632	0.07%
21	2000	26.6094	30.9435	33.1313	20.0734	23.1461	-28.81%
22	1999	17.8132	9.5107	32.5136	9.4200	32.5136	245.55%
23	1998	7.6467	6.5445	9.4091	5.0002	9.4091	45.05%
24	1997	7.4174	6.3260	8.7317	6.1025	6.4869	-0.93%
25	1996	6.0063	4.4012	7.4540	4.4012	6.5477	46.17%
26	1995	3.3537	3.4698	4.8498	2.5743	4.4794	29.92%
27	1994	3.0518	2.7411	3.8388	2.5160	3.4477	23.29%
28	1993	2.6330	2.9648	3.3227	2.1447	2.7964	-7.04%
29	1992	3.0059	3.8440	4.5622	2.0265	3.0083	-22.83%
30	1991	4.1061	2.9124	4.9177	2.7942	3.8983	28.63%
31	1990	3.4557	4.0208	4.5437	2.4156	3.0306	-23.82%
32	1989	3.4498	2.4641	5.1280	2.2167	3.9783	67.20%
33	1988	1.8888	1.2444	2.4960	1.2444	2.3793	103.83%

	Year	Assets(m)	Y_Quarter	Assets(m)
0	2020.0	$6,660	2020-10-31	$6,935
1	2019.0	$5,333	2020-07-31	$6,856
2	2018.0	$5,468	2020-04-30	$6,744
3	2017.0	$5,098	2020-01-31	$6,660
4	2016.0	$5,122	2019-10-31	$6,259
5	2015.0	$5,181	2019-07-31	$6,316
6	2014.0	$4,752	2019-04-30	$6,276
7	2013.0	$4,631	2019-01-31	$5,333
8	2012.0	$4,159	2018-10-31	$5,101
9	2011.0	$3,736	2018-07-31	$5,170
10	2010.0	$3,488	2018-04-30	$5,451
11	2009.0	$3,102	2018-01-31	$5,468
12	NaN	NaN	2017-10-31	$5,314
13	NaN	NaN	2017-07-31	$5,270
14	NaN	NaN	2017-04-30	$5,106
15	NaN	NaN	2017-01-31	$5,098
16	NaN	NaN	2016-10-31	$5,139
17	NaN	NaN	2016-07-31	$5,103
18	NaN	NaN	2016-04-30	$5,148
19	NaN	NaN	2016-01-31	$5,122
20	NaN	NaN	2015-10-31	$5,158
21	NaN	NaN	2015-07-31	$5,172
22	NaN	NaN	2015-04-30	$5,141
23	NaN	NaN	2015-01-31	$5,181
24	NaN	NaN	2014-10-31	$4,997
25	NaN	NaN	2014-07-31	$4,936
26	NaN	NaN	2014-04-30	$4,826
27	NaN	NaN	2014-01-31	$4,752
28	NaN	NaN	2013-10-31	$4,880
29	NaN	NaN	2013-07-31	$4,732
30	NaN	NaN	2013-04-30	$4,666
31	NaN	NaN	2013-01-31	$4,631
32	NaN	NaN	2012-10-31	$4,450
33	NaN	NaN	2012-07-31	$4,325
34	NaN	NaN	2012-04-30	$4,207

- **csv_file_example**



	A	B	C	D	E	F	G	H	I	J	K
1	Date	Price	Open	High	Low	Close	Change	Year	Assets(m)	Y_Quarter	Assets(m)
2	2021	131.4433	131.44	131.46	131.43	131.46	0.01%	2020	$6,660	#########	$6,935
3	2020	125.1906	131.3627	131.8633	109.1533	131.45	0.22%	2019	$5,333	2020/7/31	$6,856
4	2019	96.959	78.5186	131.294	76.3692	131.1664	69.79%	2018	$5,468	2020/4/30	$6,744
5	2018	106.647	100.0419	132.3313	71.2069	77.252	-20.94%	2017	$5,098	2020/1/31	$6,660
6	2017	84.5507	71.6092	98.9233	70.707	97.7107	37.08%	2016	$5,122	#########	$6,259
7	2016	62.9333	66.9133	77.8624	52.2928	71.2778	4.05%	2015	$5,181	2019/7/31	$6,316
8	2015	76.3823	92.9369	92.9369	64.5068	68.5026	-27.23%	2014	$4,752	2019/4/30	$6,276
9	2014	83.3478	80.3223	95.7295	70.1941	94.135	16.91%	2013	$4,631	2019/1/31	$5,333
10	2013	65.0013	50.2819	80.5219	50.2819	80.5219	64.58%	2012	$4,159	#########	$5,101
11	2012	51.9148	55.5728	61.5099	42.4696	48.9252	-11.67%	2011	$3,736	2018/7/31	$5,170
12	2011	56.6873	50.7804	69.4762	46.2514	55.3889	8.22%	2010	$3,488	2018/4/30	$5,451
13	2010	38.5073	35.4084	53.7887	29.5475	51.1831	47.88%	2009	$3,102	2018/1/31	$5,468
14	2009	24.3054	20.0909	35.3762	13.3258	34.6115	86.20%			#########	$5,314
15	2008	28.5476	33.9203	37.9968	13.8363	18.5884	-47.68%			2017/7/31	$5,270
16	2007	36.7432	30.0832	43.7211	29.4036	35.5256	18.57%			2017/4/30	$5,106
17	2006	26.738	29.2957	31.0408	22.7245	29.961	3.58%			2017/1/31	$5,098
18	2005	26.1095	23.6704	32.6285	21.9889	28.9255	20.79%			#########	$5,139
19	2004	26.1465	32.4033	32.4852	20.2225	23.9476	-28.79%			2016/7/31	$5,103
20	2003	25.067	18.4776	36.2078	16.2872	33.631	90.08%			2016/4/30	$5,148
21	2002	22.6013	22.9424	29.6208	15.0792	17.6932	-23.62%			2016/1/31	$5,122
22	2001	22.5895	22.0923	27.5787	15.2584	23.1632	0.07%			#########	$5,158
23	2000	26.6094	30.9435	33.1313	20.0734	23.1461	-28.81%			2015/7/31	$5,172
24	1999	17.8132	9.5107	32.5136	9.42	32.5136	245.55%			2015/4/30	$5,141
25	1998	7.6467	6.5445	9.4091	5.0002	9.4091	45.05%			2015/1/31	$5,181
26	1997	7.4174	6.326	8.7317	6.1025	6.4869	-0.93%			#########	$4,997
27	1996	6.0063	4.4012	7.454	4.4012	6.5477	46.17%			2014/7/31	$4,936
28	1995	3.3537	3.4698	4.8498	2.5743	4.4794	29.92%			2014/4/30	$4,826
29	1994	3.0518	2.7411	3.8388	2.516	3.4477	23.29%			2014/1/31	$4,752
30	1993	2.633	2.9648	3.3227	2.1447	2.7964	-7.04%			#########	$4,880
31	1992	3.0059	3.844	4.5622	2.0265	3.0083	-22.83%			2013/7/31	$4,732
32	1991	4.1061	2.9124	4.9177	2.7942	3.8983	28.63%			2013/4/30	$4,666
33	1990	3.4557	4.0208	4.5437	2.4156	3.0306	-23.82%			2013/1/31	$4,631
34	1989	3.4498	2.4641	5.128	2.2167	3.9783	67.20%			#########	$4,450
35	1988	1.8888	1.2444	2.496	1.2444	2.3793	103.83%			2012/7/31	$4,325

TIF_tiffany_data

5.个人总结和经验记录

上传文件至github

```
(Singapore VPN is expected)

cd C:/Users/WHR/ustc-whr.github.io

mkdir DL_PJ_1/TEST5
```

```
cp
D:/whr_laptop/2023Summer/TEST/4thGrade_a/DL_pragmatic/DL_PJ1/README_SUMMARY/README0.md DL_PJ_1
cp
D:/whr_laptop/2023Summer/TEST/4thGrade_a/DL_pragmatic/DL_PJ1/README_SUMMARY/img_3.png DL_PJ_1/TEST2
cp
D:/whr_laptop/2023Summer/TEST/4thGrade_a/DL_pragmatic/DL_PJ1/README_SUMMARY/SUMMARY.md DL_PJ_1/TEST5
cp
D:/whr_laptop/2023Summer/TEST/4thGrade_a/DL_pragmatic/DL_PJ1/TEST4/20231017_stock.py
D:/whr_laptop/2023Summer/TEST/4thGrade_a/DL_pragmatic/DL_PJ1/TEST4/main_window_2.py
D:/whr_laptop/2023Summer/TEST/4thGrade_a/DL_pragmatic/DL_PJ1/TEST4/STOCK_APP.png DL_PJ_1/TEST4

git add DL_PJ_1
git add DL_PJ_1/TEST5
git add DL_PJ_1/TEST4

git commit -m"Added to DL_PJ_1 folder"
git commit -m"Added to DL_PJ_1/TEST5 folder"
git commit -m"Added to DL_PJ_1/TEST4 folder"

git push origin main
```

用git bash 删除github中的文件（假设你已经克隆repository到本地）
导航到你的仓库目录：

```
cd C:/Users/WHR/ustc-whr.github.io
```

删除 .gitkeep 文件：

```
rm DL_PJ_1/.gitkeep
```

将更改添加到 Git：

```
git add DL_PJ_1/.gitkeep
```

提交你的更改：

```
git commit -m "删除 .gitkeep 文件"
```

将更改推送到 GitHub：

```
git push origin main
```

将.ui文件转化为.py文件

```
pyuic5 -o main_window_2.py main_window_2.ui
```

爬虫爬取数据

```
import requests
from bs4 import BeautifulSoup

url='your_url'
headers_list = [
        {
            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'},
        {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0)
Gecko/20100101 Firefox/89.0'},
        {
            'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'},
        {
            'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.1.1 Safari/605.1.15'},
        {
            'User-Agent': 'Mozilla/5.0 (iPhone; CPU iPhone OS 14_6 like Mac OS X)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0 Mobile/15E148
Safari/604.1'},
        {
            'User-Agent': 'Mozilla/5.0 (iPad; CPU OS 14_6 like Mac OS X)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0 Mobile/15E148
Safari/604.1'},
        {
            'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'},
        {'User-Agent': 'Mozilla/5.0 (Android 10; Mobile; rv:89.0) Gecko/89.0
Firefox/89.0'}
    ]

selected_headers = random.choice(headers_list)

response = requests.get(url, headers=selected_headers)

soup=BeautifulSoup(response.text,'html.parser')# html.parser解析器，速度快，容错能力强，但只支持HTML
或者soup=BeautifulSoup(response.text,'lxml')# lxml解析器速度快，容错能力强，支持HTML和XML，但需要安装C语言库
或者soup=BeautifulSoup(response.text,'html5lib')# html5lib解析器以浏览器的方式解析文档，生成HTML5格式的文档，但速度慢，容错能力强
或者soup=BeautifulSoup(response.text,'xml')# xml解析器速度快，唯一支持XML的解析器
或者soup=BeautifulSoup(response.content,'html.parser')# 用response.content解析网页内容,不需要指定编码，但是会丢失编码信息，可能会乱码

selector = 'your_selector'
target_data=soup.select_one(selector)
```

```
target_data.text.strip()# 去除空格和换行符
```

建立函数和ui界面的联系

步骤如下：

1.用qt designer设计界面：选取mian_window意味着这个界面是主界面，然后在右侧的object inspector中选取widget，然后在左侧的property editor中选取object name

2.将.ui文件转化为.py文件

3.直接调用.py文件中的object name即可

4.建立class，将object name作为参数传入class中，然后在class中定义函数，最后在主函数中调用class中的函数即可

5.如果有button,则需要在class中定义一个槽函数，然后将button的clicked信号与槽函数相连，这样当button被点击时，槽函数就会被调用

eg:self.pushButton_2.clicked.connect(self.save_to_csv)

6.运用try,exception语句，当try中的语句出现错误时，就会执行except中的语句

7.主函数：

```
if __name__=='__main__':

    app=QApplication(sys.argv)#from PyQt5.QtWidgets import QApplication

    myshow=APP()#自己定义的class

    myshow.show()#显示界面

    sys.exit(app.exec_())#app.exec_()表示程序一直执行，直到主窗口关闭，sys.exit()方法确保主循环安全退出
```