Summer Internship
Project Report
On
BackOffice Software for Stock Traders

Developed By: -
Divij Bhutani (17162121002)

Guided By:-
Prof. Nidhi Thacker (Internal)
Mr. Siddharth Shah (External)

Submitted to
Department of Computer Science & Engineering
Institute of Computer Technology

Year: 2019

GANPAT UNIVERSITY
INSTITUTE OF COMPUTER TECHNOLOGY

CERTIFICATE

This is to certify that the report entitled **"BackOffice Software for Stock Traders"** by Divij Bhutani (17162121002) of Ganpat University, towards the fulfillment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, is record of Summer Internship Project carried out by them in the CSE(BDA) Department.
The results/findings contained in this Project have not been submitted in part or full to any other University / Institute for award of any other Degree/Diploma.

Nidhi Thacker
Name & Signature of Internal Guide Nidhi

Name & Signature of Head of Department

Place: Ahmedabad
Date: 13.07.2019

LETTER OF JOINING

multitrade
Technology that enhances trade

Date:13/05/2019

Mr. Divij Bhutani

INTERNSHIP OFFER LETTER

Dear Divij,

Multitrade Softech Private Limited is pleased to offer you an educational internship opportunity as a **Jr. Developer (Intern).** You will report directly to project manager. This position is located in Ahmedabad.

As you will be receiving academic credit for this position, you will not be paid. Additionally, students do not receive benefits as part of their internship program.

For this position, your major duties will include developing BackOffice Software. Your schedule will be approximately 6 hours per day beginning from **13/05/2019.** Your assignment will conclude on 13/06/2019.

Congratulations and welcome to the team!

Sincerely,

For, Multitrade Softech Pvt. Ltd.

Mahesh Khunt
H R Manager
Dated: 13/05/2019

Agreed and Accepted

Divij Bhutani
Intern

ACKNOWLEDGEMENT

A major project is a golden opportunity for learning and self-development. I consider myself very lucky and honored to have so many wonderful people lead me through in completion of this project. First and foremost, I would like to thank **Prof. Dharmesh Darji,** Head of Department, Computer Science and Engineering, who gave us an opportunity to undertake this project. My grateful thanks to **Prof. Nidhi Thacker (Internal Guide) & Mr. Siddharth Shah (External Guide)** for their guidance in project work **BackOffice Software for Stock Traders**, who despite being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where would have been without his/her help. CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Divij Bhutani (17162121002)

ABSTRACT

This project is a back-office software which shows detail analysis of daily trades done in stock market by a particular trader. It generates a report displaying the number of shares sold and bought by a trader, net & gross profit, outstanding positions, brokerage charges, total turnover, etc. It requires a csv file (trade file) which is provided by the stock broker to trader on daily basis. This software aims at providing greater customer satisfaction and traders no longer need to calculate their net position daily which in turn reduce a lot of paper work and makes life easier for the traders.

INDEX

LIST OF TABLES

Table Name: **Page No.**

CHAPTER: 1 INTRODUCTION

I developed a back-office software for stock traders which shows a report displaying the number of shares sold and bought by a trader, net & gross profit, outstanding positions, brokerage charges, total turnover, etc. by reading a csv file which is provided by the stock broker to trader on daily basis. With the help of this software, traders can easily see detailed analysis of the work done by them.

CHAPTER: 2 PROJECT SCOPE

Previously, traders had to keep record of each and every trade, brokerage paid, net profit, etc. on paper, but with the help of this software one can know the amount of shares sold and bought, net & gross profit, outstanding positions, brokerage charges, total turnover, etc. All you need is a csv file (trade file) which you'll get from the broker. Importing this csv file into the software, it will generate a pdf report of the trader's net position on daily basis. It calculates the net & gross profit using various formulas inculcated into it with help of HTML code & Python Logic. This project is limited to only Windows Operating System.

CHAPTER: 3 SOFTWARE AND HARDWARE REQUIREMENTS

Minimum Hardware Requirements

Processor	2.0 GHz
RAM	256 MB
HDD	500 MB

Table 1.1 Minimum Hardware Requirements

Minimum Software Requirements

Operating System	Windows
Programming language	Python
Other tools & tech	Browser

Table 1.2 Minimum Software Requirements

CHAPTER: 4 IMPLEMENTATION DETAILS

My project was to make a back-office software for daily trades done in stock market by a particular trader. So, basically I was expected to calculate the net profit made by each trader in each symbol by reading the csv file. I did research and found that it would be easy if I worked on Python. So having started with Python, I learned how to read CSV file using Python. I managed to convert the CSV File Data into corresponding Python Dictionaries. I found that it would be easier to manipulate the data If I convert the data into Lists so I decided to drop the idea of dictionaries and decided I rather convert the data into list. On each day I had to spend several hours solving unwanted errors.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Server	UserID	ATINTrade	ATINOrde	OrderID	Exchange(	Exchange	OrderTim	Exchange(	Exchange	Exchange	SecurityID	Symbol	Expiry Dat	SecurityT	OptionTy	Side	OrderTyp	Quantity	PendingQ	Price
2	Default	M13	1	4	2	0HPYSW0(	34292	20190513-	20190513-	20190513-	TTSIM:A	CL CLV0	CL CLV0	########	FUTURE		Sell	Limit	1	1	
3	Default	M13	2	9	3	0HPYSW0(	34303	20190513-	20190513-	20190513-	TTSIM:A	CL CLX0	CL CLX0	########	FUTURE		Buy	Limit	1	0	
4	Default	M13	3	10	4	0HPYSW0(	34724	20190513-	20190513-	20190513-	TTSIM:A	CL CLV0	CL CLV0	########	FUTURE		Sell	Limit	1	1	
5	Default	M13	4	11	5	0HPYSW0(	34745	20190513-	20190513-	20190513-	TTSIM:A	CL CLX0	CL CLX0	########	FUTURE		Buy	Limit	1	0	
6	Default	M13	5	3	1	0HPYSW0(	35431	20190513-	20190513-	20190513-	TTSIM:A	CL CLV0	CL CLV0	########	FUTURE		Buy	Limit	1	1	
7	Default	M13	6	13	7	0HPYSW0:	35436	20190513-	25024	20190513-	TTSIM:A	CL CLX0	CL CLX0	########	FUTURE		Sell	Limit	1	1	
8	Default	M13	7	14	8	0HPYSW0:	35521	20190513-	20190513-	20190513-	TTSIM:A	CL CLV0	CL CLV0	########	FUTURE		Buy	Limit	1	0	
9	Default	M13	8	15	9	0HPYSW0:	35524	20190513-	20190513-	20190513-	TTSIM:A	CL CLX0	CL CLX0	########	FUTURE		Sell	Limit	1	0	
10	Default	M13	9	16	10	0HPYSW0:	38625	20190513-	20190513-	20190513-	TTSIM:A	CL CLV0	CL CLV0	########	FUTURE		Buy	Limit	1	1	
11	Default	M13	10	17	11	0HPYSW0:	38650	20190513-	20190513-	20190513-	TTSIM:A	CL CLX0	CL CLX0	########	FUTURE		Sell	Limit	1	1	
12	Default	M13	11	12	6	0HPYSW0:	40060	20190513-	20190513-	20190513-	TTSIM:A	CL CLV0	CL CLV0	########	FUTURE		Sell	Limit	1	0	
13	Default	A2	12	2	2	0HPYSW0(	40062	20190513-	20190513-	20190513-	TTSIM:A	CL CLV0	CL CLV0	########	FUTURE		Sell	Limit	1	0	
14	Default	A2	13	21	3	0HPYSW0:	40067	20190513-	20190513-	20190513-	TTSIM:A	CL CLX0	CL CLX0	########	FUTURE		Buy	Limit	1	0	
15	Default	M13	14	22	13	0HPYSW0:	40071	20190513-	20190513-	20190513-	TTSIM:A	CL CLX0	CL CLX0	########	FUTURE		Buy	Limit	1	0	
16	Default	M13	15	18	12	0HPYSW0:	48699	20190513-	20190513-	20190513-	TTSIM:A	CL CLV0	CL CLV0	########	FUTURE		Buy	Limit	1	1	
17	Default	M13	16	24	15	0HPYSW0:	48724	20190513-	20190513-	20190513-	TTSIM:A	CL CLX0	CL CLX0	########	FUTURE		Sell	Limit	1	0	
18	Default	M13	17	25	16	0HPYSW0:	48843	20190513-	20190513-	20190513-	TTSIM:A	CL CLV0	CL CLV0	########	FUTURE		Buy	Limit	1	0	
19	Default	M13	18	26	17	0HPYSW0:	48858	20190513-	25463	20190513-	TTSIM:A	CL CLX0	CL CLX0	########	FUTURE		Sell	Limit	1	1	
20	Default	M13	19	23	14	0HPYSW0:	52132	20190513-	20190513-	20190513-	TTSIM:A	CL CLV0	CL CLV0	########	FUTURE		Sell	Limit	1	1	

Thonny - C:\Users\admin\Desktop\Internship\dict.py @ 10 : 12

File Edit View Run Device Tools Help

pro.py morn.py divij.py dict.py home.py

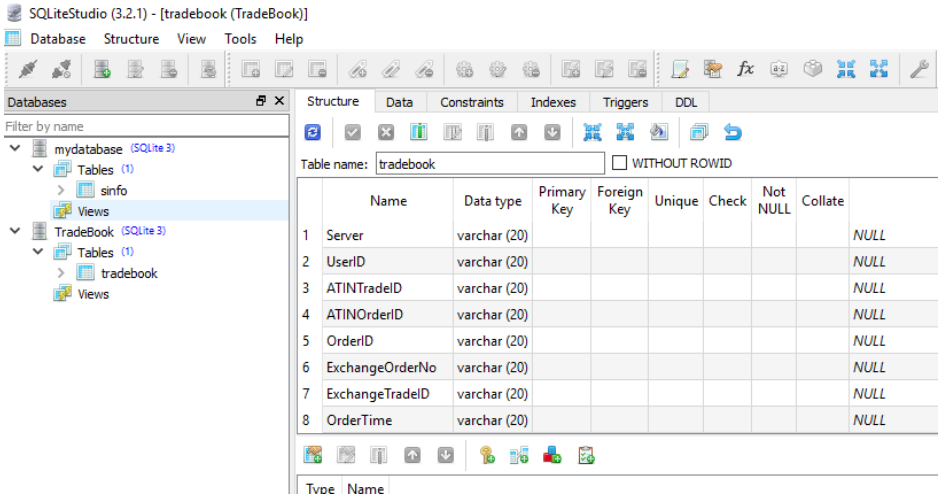
```
1  import csv
2  with open('TradeBook_All20190513.csv', 'r',encoding='utf-8',errors='ignc
3      reader = csv.reader(csvFile)
4      #reader = csv.reader(x.replace('\0', '') for x in csvFile)
5      dic={}
6      i=0
7      for row in reader:
8          i=i+1
9          dic[i] = {'Server': row[0], 'UserID': row[1], 'ATINTradeID': rou
10 print(dic[2])
```

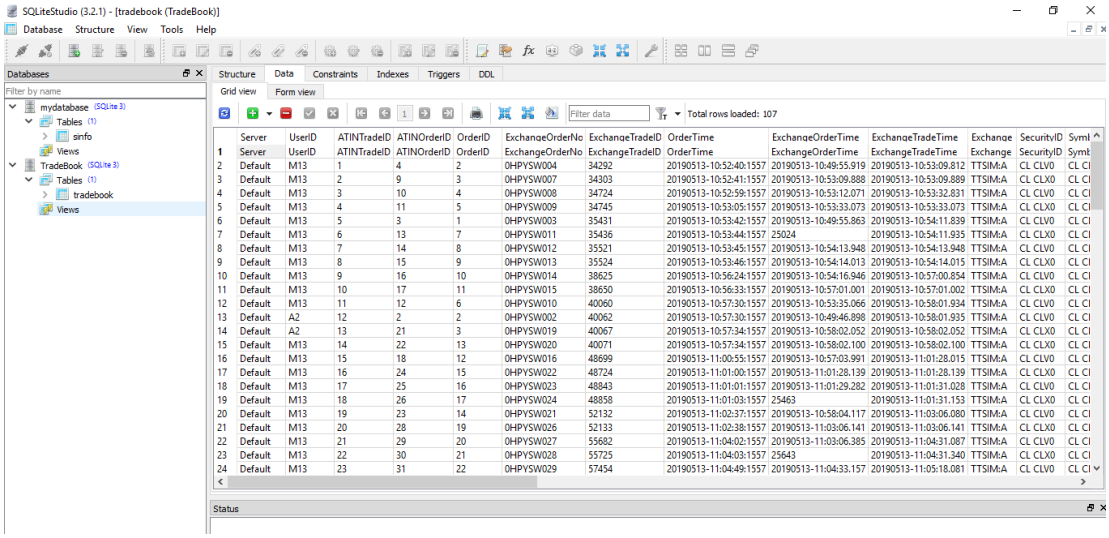
```
>>> %Run dict.py
{'Server': 'Default', 'UserID': 'M13', 'ATINTradeID': '1', 'ATINOrderID':
'4', 'OrderID': '2', 'ExchangeOrderNo': '0HPYSW004', 'ExchangeTradeID': '3
4292', 'OrderTime': '20190513-10:52:40:1557', 'ExchangeOrderTime': '201905
13-10:49:55.919', 'ExchangeTradeTime': '20190513-10:53:09.812', 'Exchange'
: 'TTSIM:A', 'SecurityID': 'CL CLV0', 'Symbol': 'CL CLV0', 'ExpiryDate': '
22-Sep-21', 'SecurityType': 'FUTURE', 'OptionType': '', 'Side': 'Sell', 'O
```

- Having explored various libraries, packages and soft wares of python from which we can design GUI, I came across ACL and page software which helps in designing python GUI using tkinter package of python.
- I found another GUI package named GTK of PyObject library. I found that GTK was more dynamic and is used for making more complex GUI whereas tkinter is used to make simple GUI. I looked various GTK programs online and learnt how to program GUI in GTK but when I wanted to run my GTK program on Anaconda, it said I need to import various header files, I found that GTK header files are not available in anaconda and when I tried pip installing them from conda prompt it said that these GTK packages are not available in anaconda channels. So I externally downloaded the GTK setup. I got an error while installing GTK externally, it said that GTK setup could not find my python path. So, I tried resolving this error by adding python path to environmental variables in advanced settings of This PC properties. After several attempts and still the GTK package having failed to work with any of my python Environment, I finally decided giving up on GTK and tried finding other alternatives.
- I talked my team lead through this and we came on a common conclusion that I must try PyQT for making GUI in python. So having downloaded PyQT package and QT designer I made a simple GUI table view display for my csv In the QT Table Widget. I also converted my csv file to dataframes in pandas library of python and tested several conditions over it, making significant progress in my main code also besides designing GUI. On each day I had to spend several hours solving unwanted errors like “OSError: [WinError 193] %1 is not a valid Win32 application” which is a major OS error which comes when you misplace any of your package DLL files or a collision of 32 bit vs 64 bit occurs. For resolving this I had to reinstall all my python environments including IDEs. I also prepared database using Sqlite 3 in python.

Database Part Code:

```
import sqlite3
#sqlite3.connect(:memory:)
db = sqlite3.connect('e:\TradeBook')
cursor = db.cursor()
cursor.execute('''create table tradebook(Server varchar(20), UserID varchar(20), ATINTradeID varchar(20), ATINOrderID varchar(20), OrderID varchar(20), ExchangeOrderNo varchar(20), ExchangeTradeID varchar(20), OrderTime varchar(20), ExchangeOrderTime varchar(20), ExchangeTradeTime varchar(20) ,Exchange varchar(20), SecurityID varchar(20), Symbol varchar(20), ExpiryDate varchar(20), SecurityType varchar(20), OptionType varchar(20), Side varchar(20), OrderType varchar(20), Quantity varchar(20), PendingQuantity varchar(20), Price varchar(20), Strikeprice varchar(20), ClientID varchar(20), ReferenceText varchar(20), ManagerID varchar(20), MemberID varchar(20), StrategyID varchar(20), CTCLID varchar(20), ProductType varchar(20), OpenClose varchar(20), Multiplier varchar(20), Pancard varchar(20), TerminalInfo varchar(20), AlgoID varchar(20), AlgoCategory varchar(20), ParticipantID varchar(20), Amount varchar(20), Id varchar(20))''')
db.commit()
print('New Table created...')
```

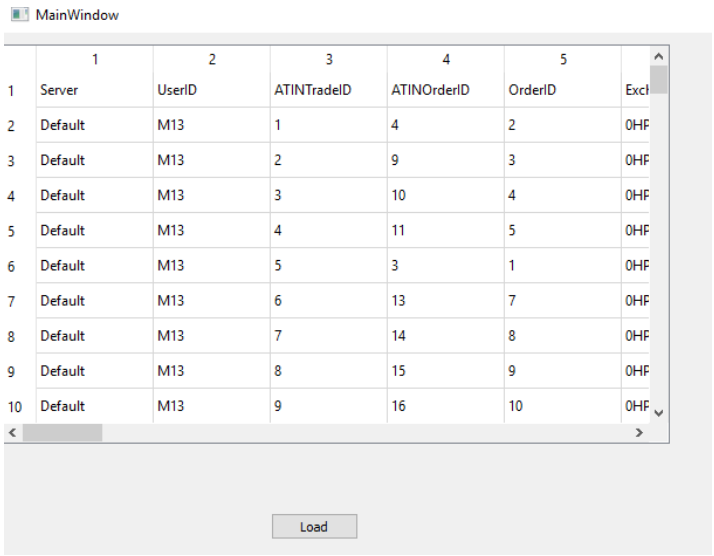




GUI using Page (Tkinter):

GUI using QT:

```
from PyQt5 import QtCore, QtGui, QtWidgets
import sqlite3
class Ui_MainWindow(object):
    def loadData(self):
        connection = sqlite3.connect("e:\TradeBook")
        query = "SELECT * FROM tradebook"
        result = connection.execute(query)
        self.tableWidget.setRowCount(0)
        for row_number,row_data in enumerate(result):
            self.tableWidget.insertRow(row_number)
            for column_number, data in enumerate(row_data):
                self.tableWidget.setItem(row_number,column_number,QtWidgets.QTabl

        connection.close()

    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(797, 600)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.tableWidget = QtWidgets.QTableWidget(self.centralwidget)
        self.tableWidget.setGeometry(QtCore.QRect(0, 10, 571, 341))
        self.tableWidget.setRowCount(106)
        self.tableWidget.setColumnCount(38)
        self.tableWidget.setObjectName("tableWidget")
        self.btn_load = QtWidgets.QPushButton(self.centralwidget)
        self.btn_load.setGeometry(QtCore.QRect(230, 410, 75, 23))
        self.btn_load.setObjectName("btn_load")
        self.btn_load.clicked.connect(self.loadData)

        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 797, 21))
        self.menubar.setObjectName("menubar")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
```



	1	2	3	4	5	
1	Server	UserID	ATINTradeID	ATINOrderID	OrderID	Excl
2	Default	M13	1	4	2	0HP
3	Default	M13	2	9	3	0HP
4	Default	M13	3	10	4	0HP
5	Default	M13	4	11	5	0HP
6	Default	M13	5	3	1	0HP
7	Default	M13	6	13	7	0HP
8	Default	M13	7	14	8	0HP
9	Default	M13	8	15	9	0HP
10	Default	M13	9	16	10	0HP

Pandas:

```
In [1]: import pandas as pd

In [2]: df=pd.read_csv("TradeBook.csv",header=0)

In [3]: df
```

Out[3]:

	Server	UserID	ATINTradeID	ATINOrderID	OrderID	ExchangeOrderNo	ExchangeTradeID	OrderTime	ExchangeOrderTime	ExchangeTradeTime	...
0	Default	M13	1	4	2	0HPYSW004	34292	20190513-10:52:40:1557	20190513-10:49:55.919	20190513-10:53:09.812	...
1	Default	M13	2	9	3	0HPYSW007	34303	20190513-10:52:41:1557	20190513-10:53:09.888	20190513-10:53:09.889	...
2	Default	M13	3	10	4	0HPYSW008	34724	20190513-10:52:59:1557	20190513-10:53:12.071	20190513-10:53:32.831	...
3	Default	M13	4	11	5	0HPYSW009	34745	20190513-10:53:05:1557	20190513-10:53:33.073	20190513-10:53:33.073	...
4	Default	M13	5	3	1	0HPYSW003	35431	20190513-10:53:42:1557	20190513-10:49:55.863	20190513-10:54:11.839	...
5	Default	M13	6	13	7	0HPYSW011	35436	20190513-10:53:44:1557	25024	20190513-10:54:11.935	...
6	Default	M13	7	14	8	0HPYSW012	35521	20190513-10:53:45:1557	20190513-10:54:13.948	20190513-10:54:13.948	...

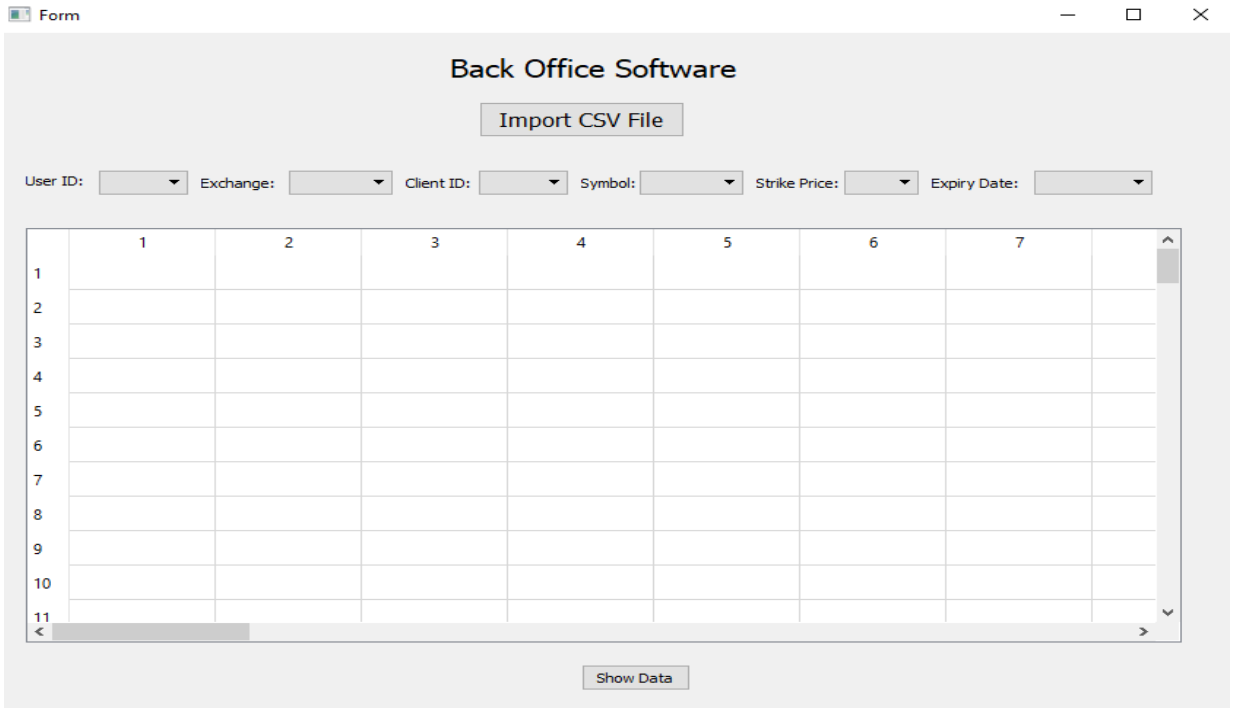
```
In [8]: s_user_id=["A2"]
        Q_userid=df.UserID.isin(s_user_id)
        temp=df[Q_userid]

In [9]: temp
```

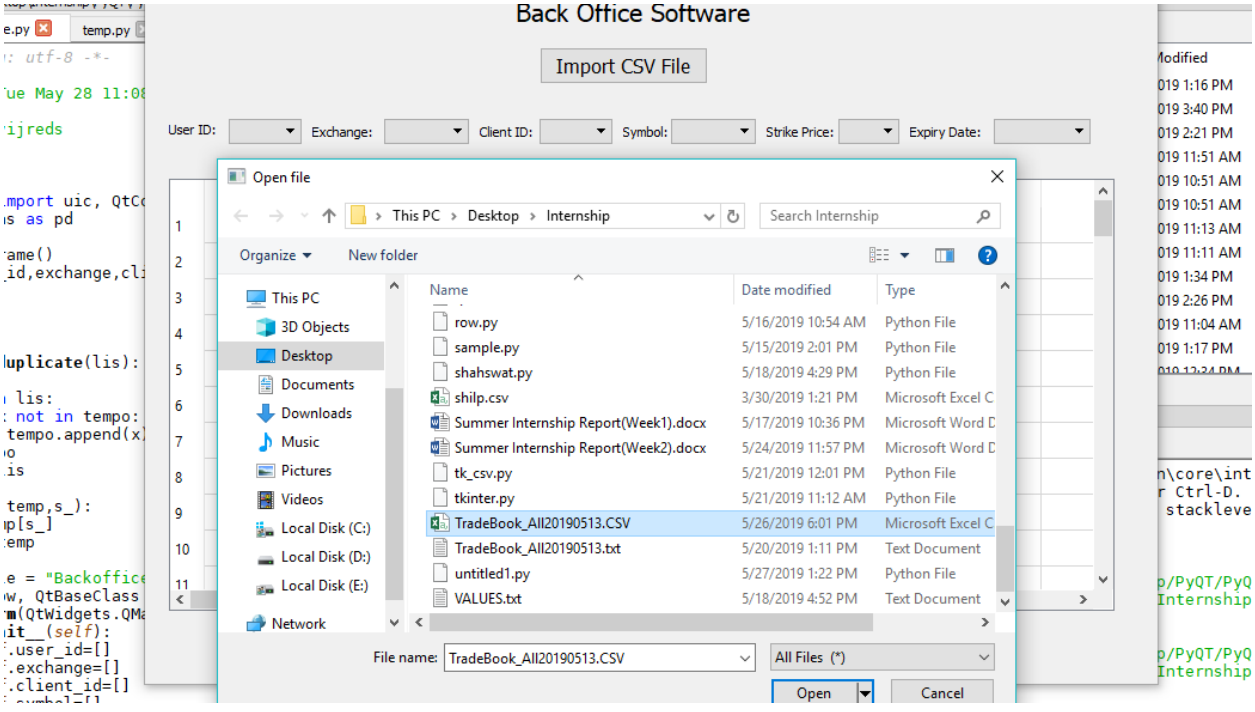
Out[9]:

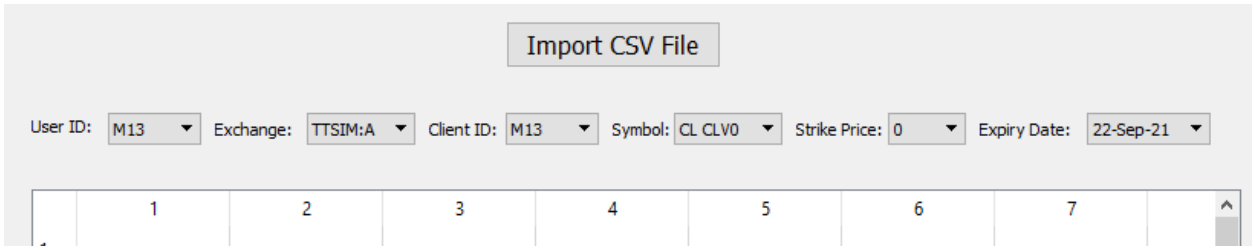
	Server	UserID	ATINTradeID	ATINOrderID	OrderID	ExchangeOrderNo	ExchangeTradeID	OrderTime	ExchangeOrderTime	ExchangeTradeTime	...	Stra
11	Default	A2	12	2	2	0HPYSW002	40062	20190513-10:57:30:1557	20190513-10:49:46.898	20190513-10:58:01.935	...	
12	Default	A2	13	21	3	0HPYSW019	40067	20190513-10:57:34:1557	20190513-10:58:02.052	20190513-10:58:02.052	...	
54	Default	A2	105	1	1	0HPYSW001	93369	20190513-11:23:45:1557	20190513-10:49:46.880	20190513-11:24:13.451	...	
55	Default	A2	108	118	4	0HPYSW116	93394	20190513-11:23:45:1557	26825	20190513-11:24:13.632	...	
62	Default	A2	147	122	5	0HPYSW120	95239	20190513-11:24:49:1557	20190513-11:24:15.661	20190513-11:25:20.538	...	
63	Default	A2	149	159	7	0HPYSW157	95250	20190513-11:24:52:1557	20190513-11:25:20.655	20190513-11:25:20.658	...	
80	Default	A2	547	123	6	0HPYSW121	204250	20190513-12:43:59:1557	20190513-11:24:15.670	20190513-12:44:27.862	...	
81	Default	A2	548	559	79	0HPYSW556	204253	20190513-12:44:00:1557	20190513-12:44:27.938	20190513-12:44:27.938	...	
84	Default	A2	567	584	94	0HPYSW581	209644	20190513-12:49:20:1557	20190513-12:49:48.561	20190513-12:50:03.073	...	

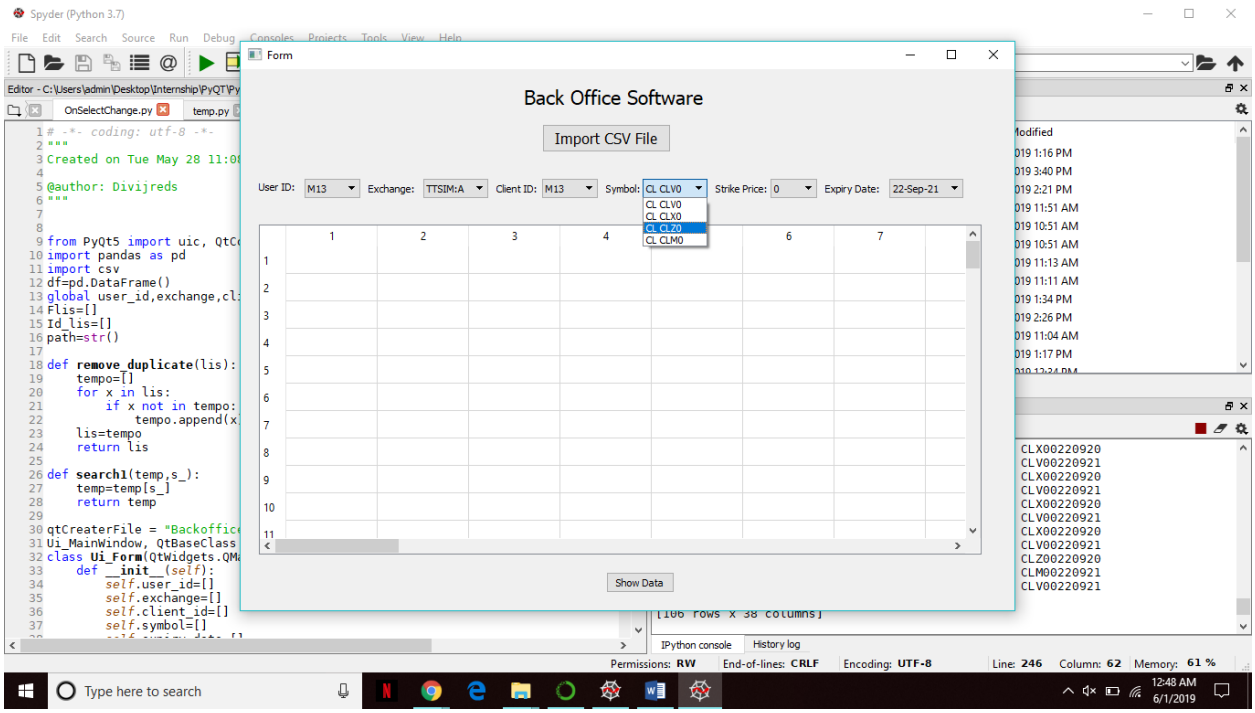
```
In [5]: df["Expiry Date"]

Out[5]: 0     22SEP2021
        1     22SEP2020
        2     22SEP2021
        3     22SEP2020
        4     22SEP2021
        5     22SEP2020
        6     22SEP2021
        7     22SEP2020
        8     22SEP2021
        9     22SEP2020
        10    22SEP2021
        11    22SEP2021
        12    22SEP2020
```

I exactly knew what I wanted in my GUI so I decided to design GUI using PYQT according to my requirements. Having made the GUI, I filtered the data to be displayed on the screen using SQL Queries and Database Sqlite3. I wrote 50 SQL Queries which were of no use as my team lead recommended me not to use SQL Queries and find a more dynamic way to filter the data. So having taken his suggestions I made a search function and filtered the data accordingly as when you select a particular user id you get to see only the data pertaining to that user id. I finalized my GUI by adding exactly what I want on the window using QComboBoxes, QPushButtons, QCheckBoxes, QTableWidget and etc as per my requirements. I also embedded the QCheckBoxes into the QComboBoxes and tried filtering the data on the selected value of QComboBox.

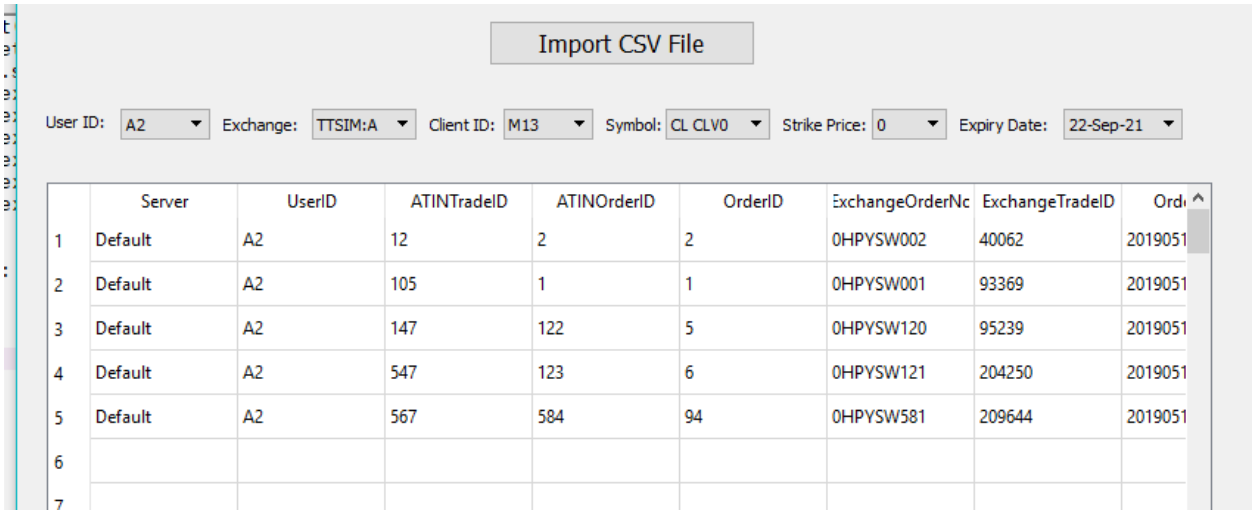






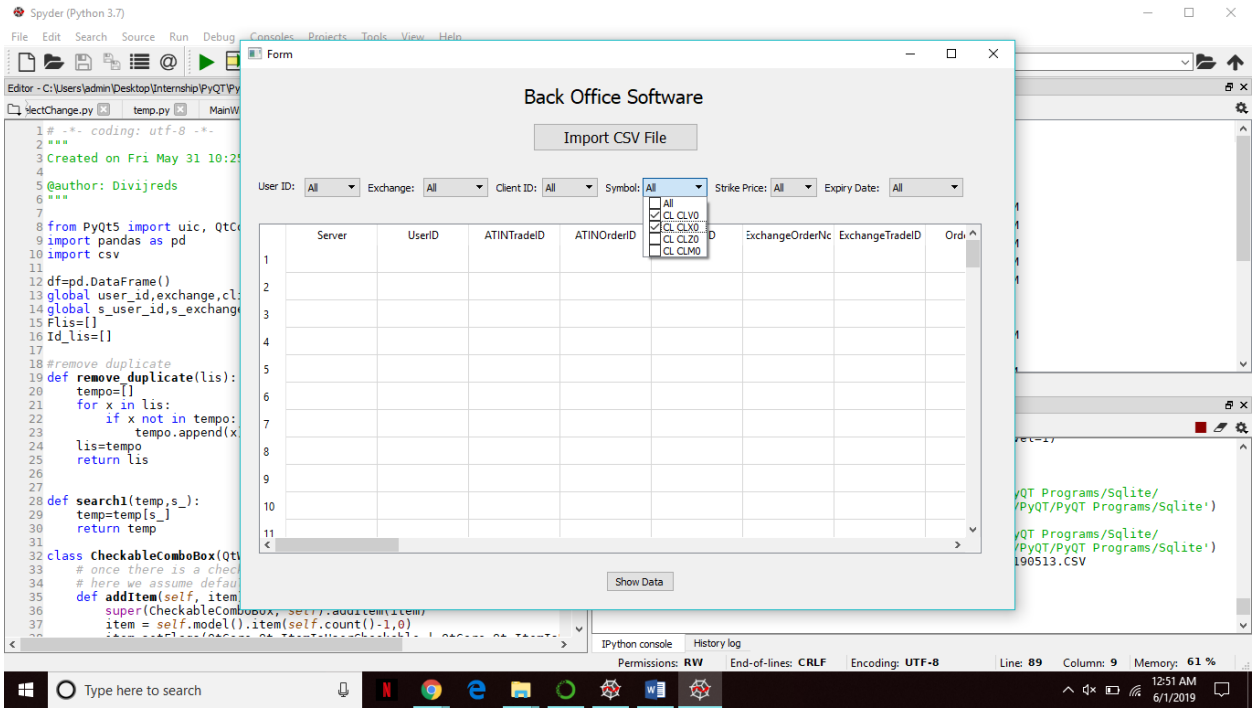


Data getting filtered using Selected Value of particular Combo Box:



	Server	UserID	ATINTradeID	ATINOrderID	OrderID	ExchangeOrderNc	ExchangeTradeID	Ord
1	Default	A2	12	2	2	0HPYSW002	40062	2019051
2	Default	A2	105	1	1	0HPYSW001	93369	2019051
3	Default	A2	147	122	5	0HPYSW120	95239	2019051
4	Default	A2	547	123	6	0HPYSW121	204250	2019051
5	Default	A2	567	584	94	0HPYSW581	209644	2019051
6								
7								

Embedding the Check boxes into Combo boxes:



Some SQL Quries:

```
349
350         elif(self.comboBox_2.currentText() == "M13" and self.comboBox_
351             query = "SELECT * FROM tradebook WHERE UserID = 'M13' AND
352             result = connection.execute(query)
353
354         elif(self.comboBox_2.currentText() == "A2" and self.comboBox_5
355             query = "SELECT * FROM tradebook WHERE UserID = 'A2' AND S
356             result = connection.execute(query)
357
358         elif(self.comboBox_2.currentText() == "A2" and self.comboBox_5
359             query = "SELECT * FROM tradebook WHERE UserID = 'A2' AND S
360             result = connection.execute(query)
361
362         elif(self.comboBox_2.currentText() == "A2" and self.comboBox_5
363             query = "SELECT * FROM tradebook WHERE UserID = 'A2' AND S
364             result = connection.execute(query)
365
366         elif(self.comboBox_2.currentText() == "A2" and self.comboBox_5
367             query = "SELECT * FROM tradebook WHERE UserID = 'A2' AND S
368             result = connection.execute(query)
369
370         elif(self.comboBox_2.currentText() == "A3" and self.comboBox_5
371             query = "SELECT * FROM tradebook WHERE UserID = 'A3' AND S
372             result = connection.execute(query)
373
374         elif(self.comboBox_2.currentText() == "A3" and self.comboBox_5
375             query = "SELECT * FROM tradebook WHERE UserID = 'A3' AND S
376             result = connection.execute(query)
377
378         elif(self.comboBox_2.currentText() == "A3" and self.comboBox_5
379             query = "SELECT * FROM tradebook WHERE UserID = 'A3' AND S
380             result = connection.execute(query)
```

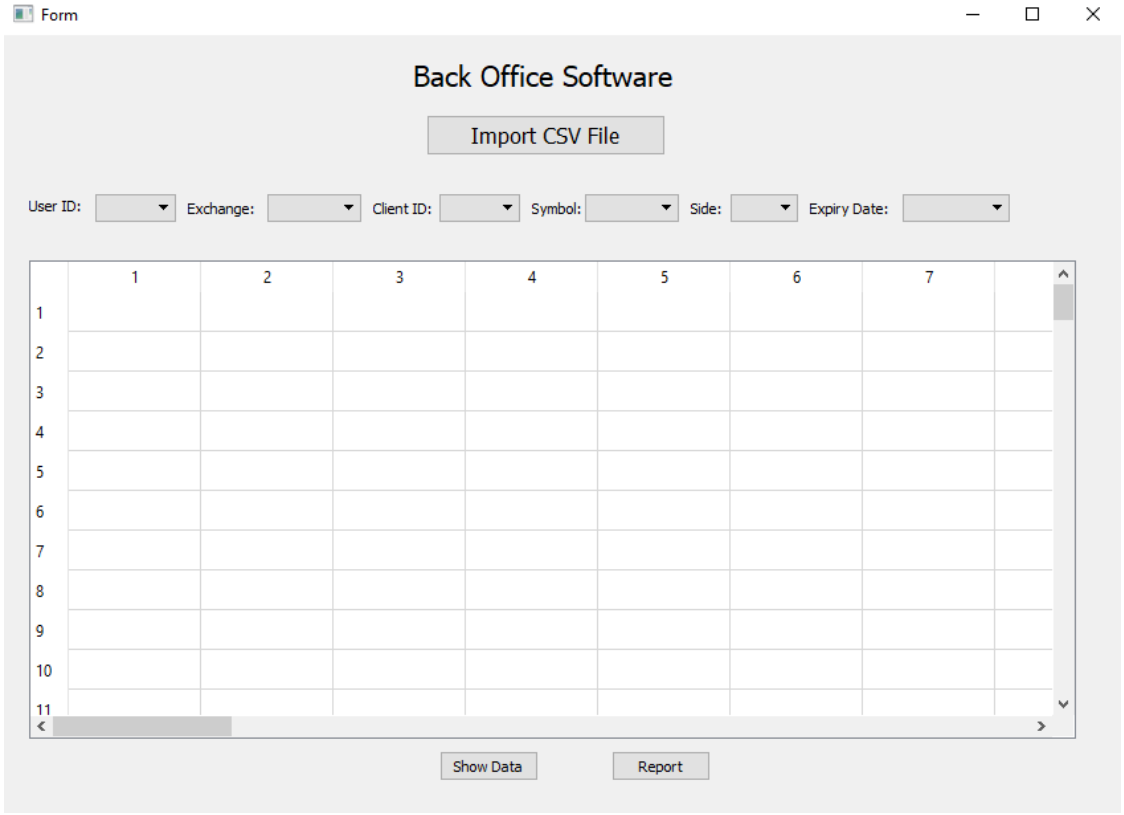
Code to add items into Combo Box from list using add item function:

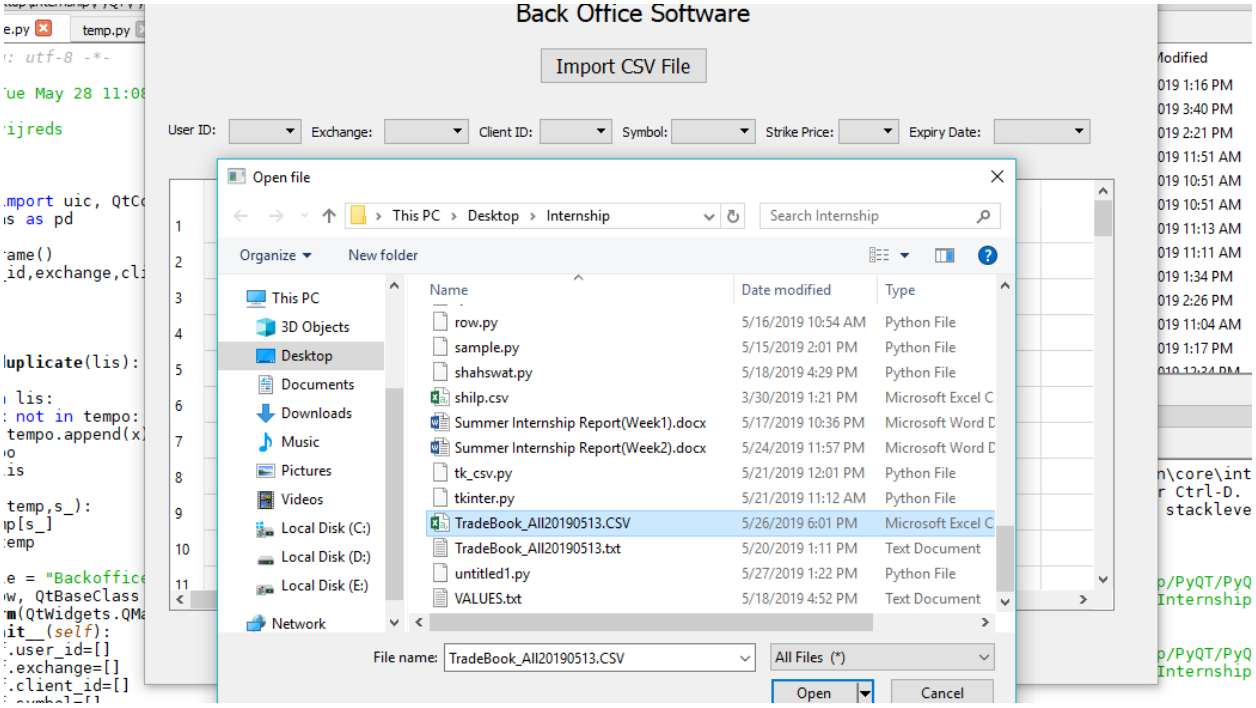
```
self.user_id=remove_duplicate(self.user_id)
self.exchange=remove_duplicate(self.exchange)
self.client_id=remove_duplicate(self.client_id)
self.symbol=remove_duplicate(self.symbol)
self.option_type=remove_duplicate(self.option_type)
self.strike_price=remove_duplicate(self.strike_price)
self.expiry_date=remove_duplicate(self.expiry_date)
self.side=remove_duplicate(self.side)

self.comboBox_2.addItem("All")
for i in self.user_id:
    self.comboBox_2.addItem(i)

self.comboBox_3.addItem("All")
for i in self.exchange:
    self.comboBox_3.addItem(i)

self.comboBox_4.addItem("All")
for i in self.client_id:
    self.comboBox_4.addItem(i)

self.comboBox_5.addItem("All")
for i in self.symbol:
    self.comboBox_5.addItem(i)

self.comboBox_6.addItem("All")
for i in self.strike_price:
    self.comboBox_6.addItem(i)
```

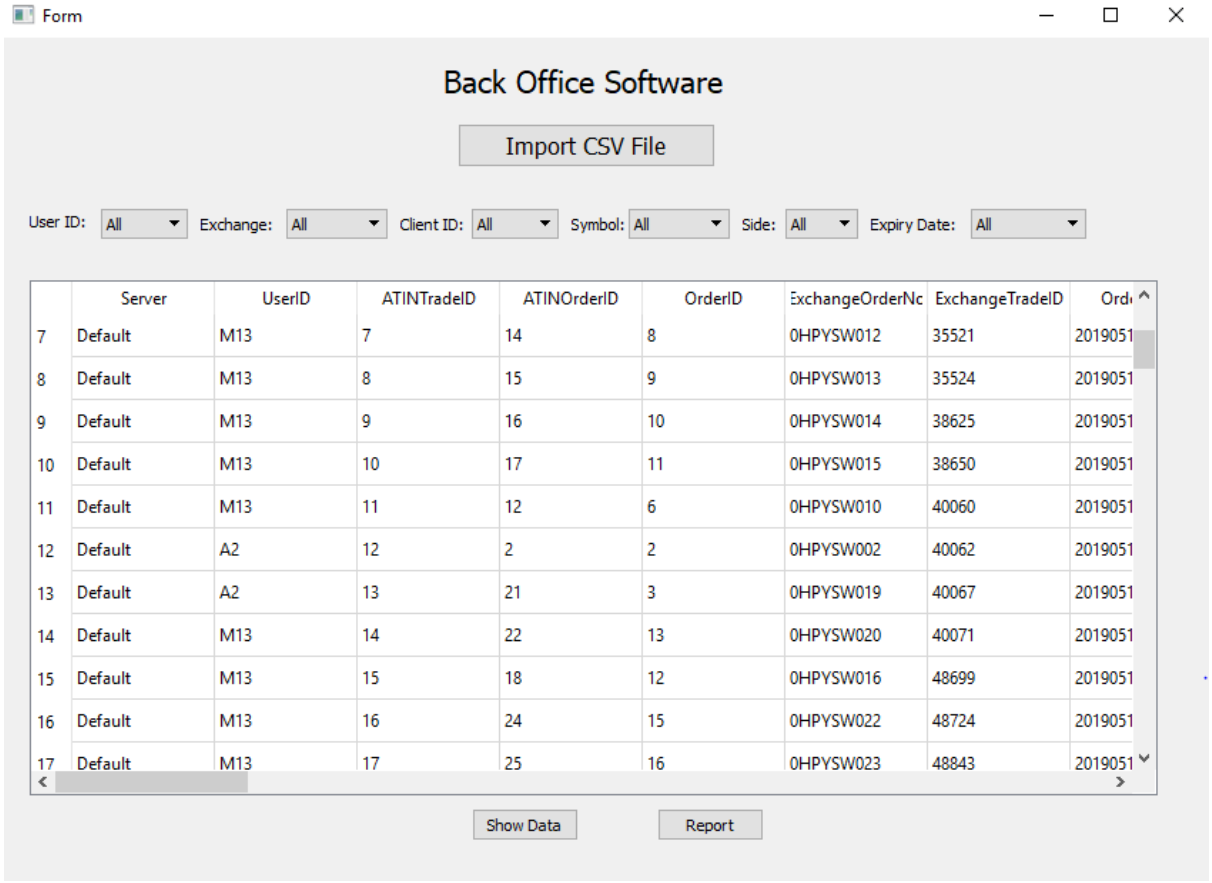
Setup UI Code:

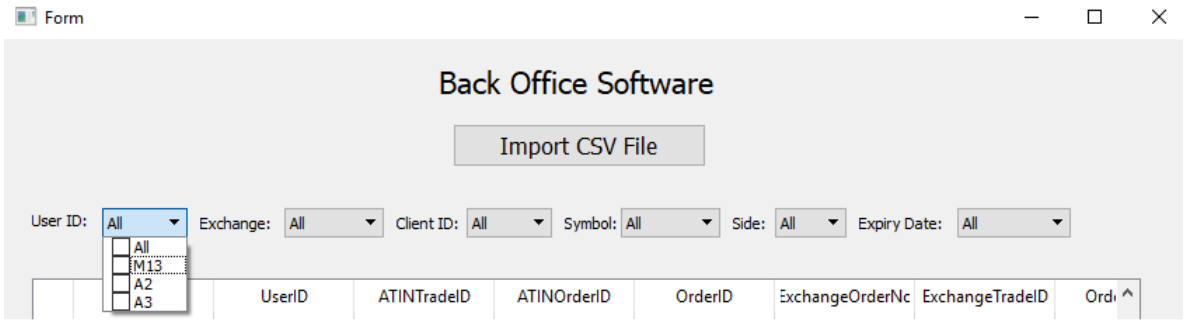
```
def setupUi(self, Form):
    Form.resize(847, 592)
    self.label = QtWidgets.QLabel(Form)
    self.label.setGeometry(QtCore.QRect(310, 20, 201, 21))
    font = QtGui.QFont()
    font.setPointSize(16)
    self.label.setFont(font)
    self.label.setObjectName("label")
    self.pushButton = QtWidgets.QPushButton(Form)
    self.pushButton.setGeometry(QtCore.QRect(400, 550, 75, 23))
    self.pushButton.setObjectName("pushButton")
    self.pushButton.clicked.connect(self.loadData)
    self.pushButton_2 = QtWidgets.QPushButton(Form)
    self.pushButton_2.setGeometry(QtCore.QRect(320, 60, 181, 31))
    self.pushButton_2.clicked.connect(self.getCSV)
    font = QtGui.QFont()
    font.setPointSize(12)
    self.pushButton_2.setFont(font)
    self.pushButton_2.setObjectName("pushButton_2")
    self.label_6 = QtWidgets.QLabel(Form)
    self.label_6.setGeometry(QtCore.QRect(400, 120, 51, 21))
    self.label_6.setObjectName("label_6")
    self.label_5 = QtWidgets.QLabel(Form)
    self.label_5.setGeometry(QtCore.QRect(280, 120, 51, 21))
    self.label_5.setObjectName("label_5")
    self.label_3 = QtWidgets.QLabel(Form)
    self.label_3.setGeometry(QtCore.QRect(20, 120, 51, 16))
    self.label_3.setObjectName("label_3")
    self.label_4 = QtWidgets.QLabel(Form)
    self.label_4.setGeometry(QtCore.QRect(140, 120, 51, 21))
```

Having completed the designing of the GUI part, I filtered the data to be displayed on the screen using csv but I was not successful in doing it. I now understood that I need to design a separate module to generate the report. So, after doing some research I worked upon adding the report module to my project which will display the data of each user displaying the Gross and Net Profit Symbol Wise.










```
148         self.tableWidget = QtWidgets.QTableWidget(Form)
149         self.tableWidget.setGeometry(QtCore.QRect(30, 90, 611, 281))
150         self.tableWidget.setRowCount(111)
151         self.tableWidget.setObjectName("tableWidget")
152         self.tableWidget.setColumnCount(14)
153         item = QtWidgets.QTableWidgetItem()
154         self.tableWidget.setHorizontalHeaderItem(0, item)
155         item = QtWidgets.QTableWidgetItem()
156         self.tableWidget.setHorizontalHeaderItem(1, item)
157         item = QtWidgets.QTableWidgetItem()
158         self.tableWidget.setHorizontalHeaderItem(2, item)
159         item = QtWidgets.QTableWidgetItem()
160         self.tableWidget.setHorizontalHeaderItem(3, item)
161         item = QtWidgets.QTableWidgetItem()
162         self.tableWidget.setHorizontalHeaderItem(4, item)
163         item = QtWidgets.QTableWidgetItem()
164         self.tableWidget.setHorizontalHeaderItem(5, item)
165         item = QtWidgets.QTableWidgetItem()
166         self.tableWidget.setHorizontalHeaderItem(6, item)
167         item = QtWidgets.QTableWidgetItem()
168         self.tableWidget.setHorizontalHeaderItem(7, item)
169         item = QtWidgets.QTableWidgetItem()
170         self.tableWidget.setHorizontalHeaderItem(8, item)
171         item = QtWidgets.QTableWidgetItem()
172         self.tableWidget.setHorizontalHeaderItem(9, item)
173         item = QtWidgets.QTableWidgetItem()
174         self.tableWidget.setHorizontalHeaderItem(10, item)
175         item = QtWidgets.QTableWidgetItem()
176         self.tableWidget.setHorizontalHeaderItem(11, item)
177         item = QtWidgets.QTableWidgetItem()
178         self.tableWidget.setHorizontalHeaderItem(12, item)
179         item = QtWidgets.QTableWidgetItem()
180         self.tableWidget.setHorizontalHeaderItem(13, item)
181         self.label_2 = QtWidgets.QLabel(Form)
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

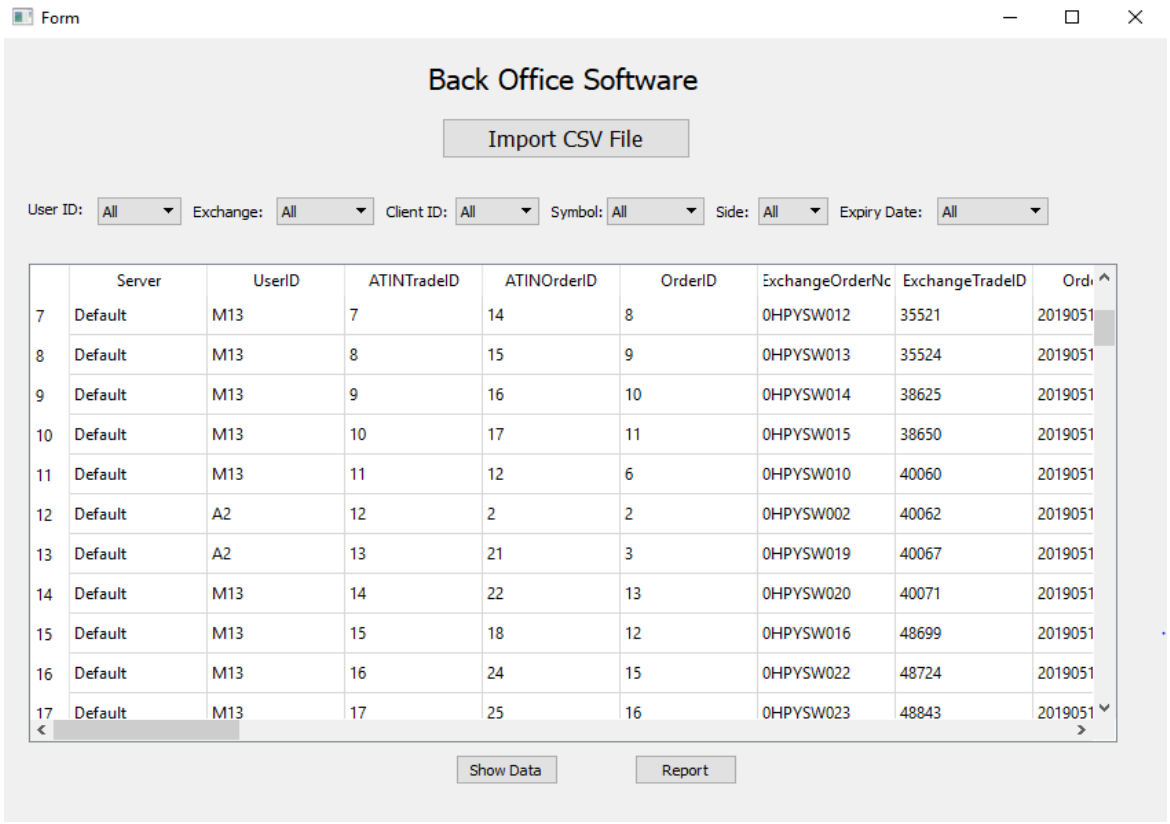
Help | Variable explorer | File explorer | Breakpoints | Profiler

IPython console

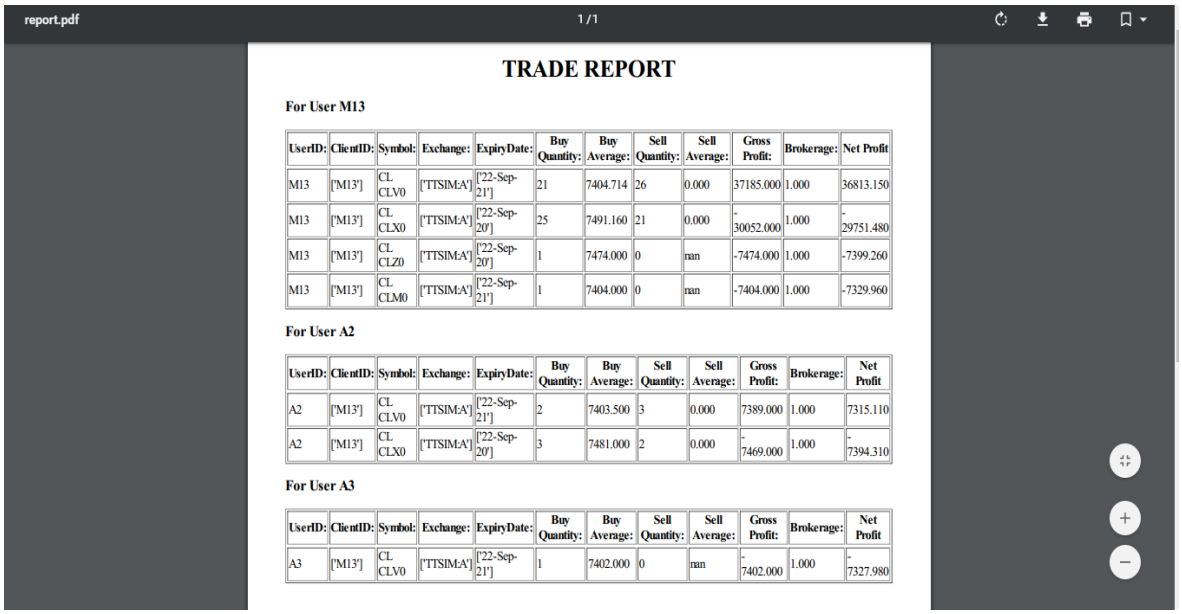
Console 1/A

```
In [1]: runfile('C:/Users/admin/Desktop/Internship/PyQT/PyQT Programs
TradeWindow.py', wdir='C:/Users/admin/Desktop/Internship/PyQT/PyQT P
An exception has occurred, use %tb to see the full traceback.

SystemExit: 0

C:\Users\admin\Anaconda3\lib\site-packages\IPython\core\interactivesh
UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)

In [2]:
```

Now getting started on my new module I installed pdfkit & learned how to use it. In addition to this, I worked upon making the format of the report using pdfkit using HTML code. During this time while extracting the data directly from the csv using HTML code to make the report, I faced certain challenges but at last I was successful in building it. I had to solve various errors which I did with the help of Stack Overflow. On the fourth day, my Project was ready displaying the Net & Gross Profit Symbol Wise in the report that gets created in your current folder. Note that the report gets created on clicking Report button. On fifth day and the last day of my internship, I found some bugs in my project & cleared it accordingly.



	Server	UserID	ATINTradeID	ATINOrderID	OrderID	ExchangeOrderNc	ExchangeTradeID	Ord
7	Default	M13	7	14	8	0HPYSW012	35521	2019051
8	Default	M13	8	15	9	0HPYSW013	35524	2019051
9	Default	M13	9	16	10	0HPYSW014	38625	2019051
10	Default	M13	10	17	11	0HPYSW015	38650	2019051
11	Default	M13	11	12	6	0HPYSW010	40060	2019051
12	Default	A2	12	2	2	0HPYSW002	40062	2019051
13	Default	A2	13	21	3	0HPYSW019	40067	2019051
14	Default	M13	14	22	13	0HPYSW020	40071	2019051
15	Default	M13	15	18	12	0HPYSW016	48699	2019051
16	Default	M13	16	24	15	0HPYSW022	48724	2019051
17	Default	M13	17	25	16	0HPYSW023	48843	2019051



For User M13

UserID:	ClientID:	Symbol:	Exchange:	ExpiryDate:	Buy Quantity:	Buy Average:	Sell Quantity:	Sell Average:	Gross Profit:	Brokerage:	Net Profit
M13	['M13']	CL CLV0	['TTSIM-A']	['22-Sep-21']	21	7404.714	26	0.000	37185.000	1.000	36813.150
M13	['M13']	CL CLX0	['TTSIM-A']	['22-Sep-20']	25	7491.160	21	0.000	-30052.000	1.000	-29751.480
M13	['M13']	CL CLZ0	['TTSIM-A']	['22-Sep-20']	1	7474.000	0	nan	-7474.000	1.000	-7399.260
M13	['M13']	CL CLM0	['TTSIM-A']	['22-Sep-21']	1	7404.000	0	nan	-7404.000	1.000	-7329.960

For User A2

UserID:	ClientID:	Symbol:	Exchange:	ExpiryDate:	Buy Quantity:	Buy Average:	Sell Quantity:	Sell Average:	Gross Profit:	Brokerage:	Net Profit
A2	['M13']	CL CLV0	['TTSIM-A']	['22-Sep-21']	2	7403.500	3	0.000	7389.000	1.000	7315.110
A2	['M13']	CL CLX0	['TTSIM-A']	['22-Sep-20']	3	7481.000	2	0.000	-7469.000	1.000	-7394.310

For User A3

UserID:	ClientID:	Symbol:	Exchange:	ExpiryDate:	Buy Quantity:	Buy Average:	Sell Quantity:	Sell Average:	Gross Profit:	Brokerage:	Net Profit
A3	['M13']	CL CLV0	['TTSIM-A']	['22-Sep-21']	1	7402.000	0	nan	-7402.000	1.000	-7327.980

CHAPTER 5: CONCLUSION AND FUTURE WORK

Conclusion:

This software will be very helpful for stock traders as they need not have to spend numerous hours calculating their net position daily. They can simply know their net & gross profit by importing the csv file into this software.

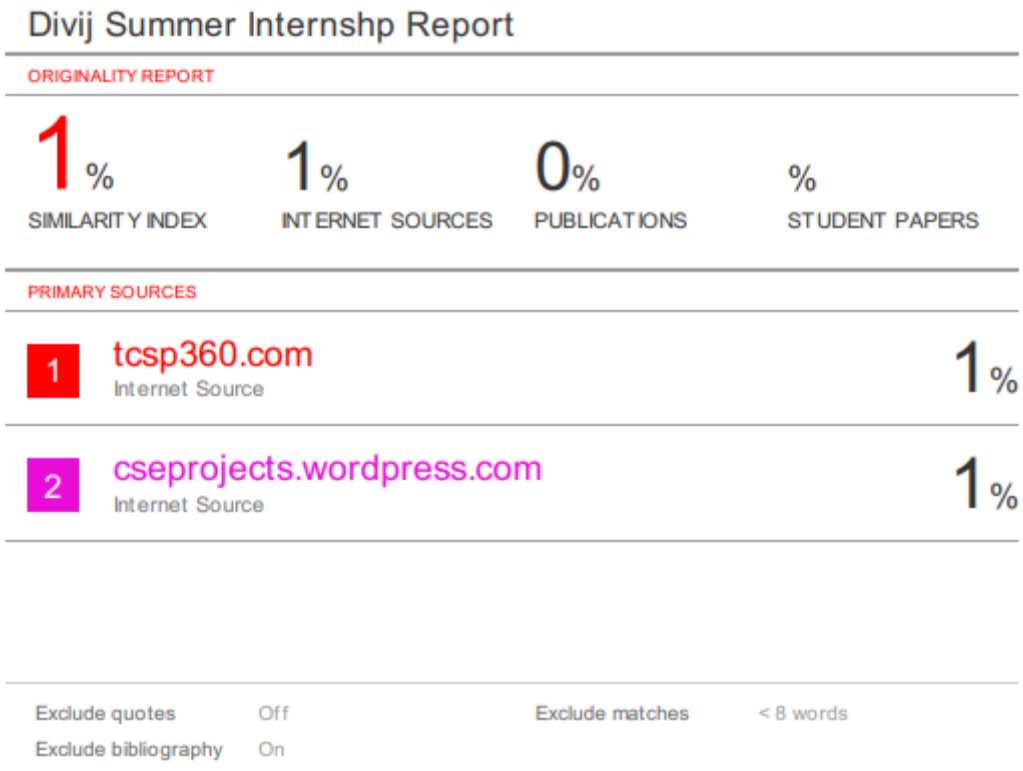
Future work:

Currently this software is compatible to windows operating system only, my future goal is to make it compatible to any given operating system such Linux or mac os.

CHAPTER 6: REFERENCES

1) https://stackoverflow.com/questions/5226091/checkboxes-in-a-combobox-using-pyqt/5291885
2) https://www.tutorialspoint.com/pyqt/pyqt_using_qt_designer.htm
3) https://pypi.org/project/pdfkit/
4) https://www.tutorialspoint.com/python/index.htm
5) https://pypi.org/project/PyQt5/
6) https://www.youtube.com/watch?v=l2OoXj1Z2hM&list=WL&index=268&t=0s
7) https://www.youtube.com/watch?v=g3ITENmadN0&list=WL&index=270&t=0s
8) https://www.youtube.com/watch?v=vmEHCJofslg&list=WL&index=266&t=933s

PLAGIARISM REPORT



Divij Summer Internshp Report

ORIGINALITY REPORT

1%	1%	0%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	tcsp360.com Internet Source	1%
2	cseprojects.wordpress.com Internet Source	1%

Exclude quotes	Off	Exclude matches	< 8 words
Exclude bibliography	On		

COMPLETION CERTIFICATE

multitrade
Technology that enhances trade

Date: 14/06/2019

CERTIFICATE

This is to certify that Mr. Divij Bhutani Student of B.Tech in Computer Science & Engineering from Institute of Computer Technology, Ganpat University has successfully completed his project. To the best of our knowledge this is an original and bona fide work done by him. He has worked upon building a Back-Office Software based on Python at our company Multitrade Softech Pvt Ltd from 13/05/19 to 13/06/19.

During his tenure at this organization, he was found to be sincere and meticulous in his work. We appreciate his enthusiasm & dedication towards the work assigned to him.

Best Regards,

For, Multitrade Softech Pvt. Ltd.

Mahesh Khunt
(H R Manager)