Name:**Dhruv Patidar**

Enrollment No:**0801CS211036**

Report on mini-project (ATM)

Characterstics of MiniProject

Starting Date/Time : 6 Nov ,2022 StartTime:10 AM Morning
End Data/Time : 15 Nov ,2022 EndTime:11 PM
Total Time Required :15 hours
Total Line of Code : 600+ lines
Number of Functions : 13

Objectives of Project :

1> This project is intended to have a better functionality in **ATM** system , as this project also shows the graph for the withdraw and deposit per month and also display them separately

2> This project/software have audio facility in it , it also tells the instruction to the customer , like " Enter name ..." and also tells that what wrong they have entered therefore increasing the communication gap between the machines and human

Function Descriptions :

1> *__main__* :

This is the driver function of the code/program , it simply calls the menu() function.

2> menu():

This function ask , whether you want to Login/SignUp/Exit the system.

3> aadharCheck.cpp:

This function or file checks whether the aadhar number entered is correct or not. 4> passCheck():

This function checks whether the password entered by the user secured or not , or it is easy to crack.

5> checkEmail.cpp:

This function or file checks whether the mail entered is correct or not

6> ver():

This function handles the vercode.txt handling , which is used as a security OTP for the person that is trying to withdraw/deposit in the account.

7> login():

This function allows the user to login in into the account , by checking whether the person with the given username and password exist or not.

8> afterlogin():

This function help the user to do the functionalities like deposit , withdraw , see transaction history , see account details and graphs.

9> choiceMainMenu():

This function helps in making choice in afterlogin function , it actually identify whether the choice entered is valid or not in afterlogin choice system.

10> graphChoiceChecker():

This function is used to perform choice function in graphmenu function.

11> graphMenu():

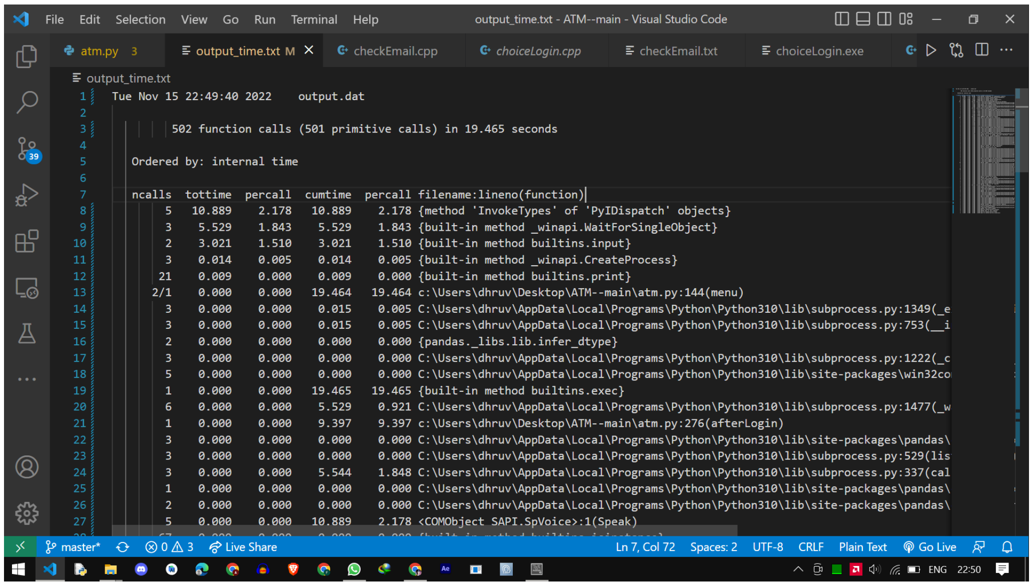
This functions is responsible for showing the graph to the user , that makes easier for the user to understand about his transaction history.

12> choiceAfterLogin.cpp :

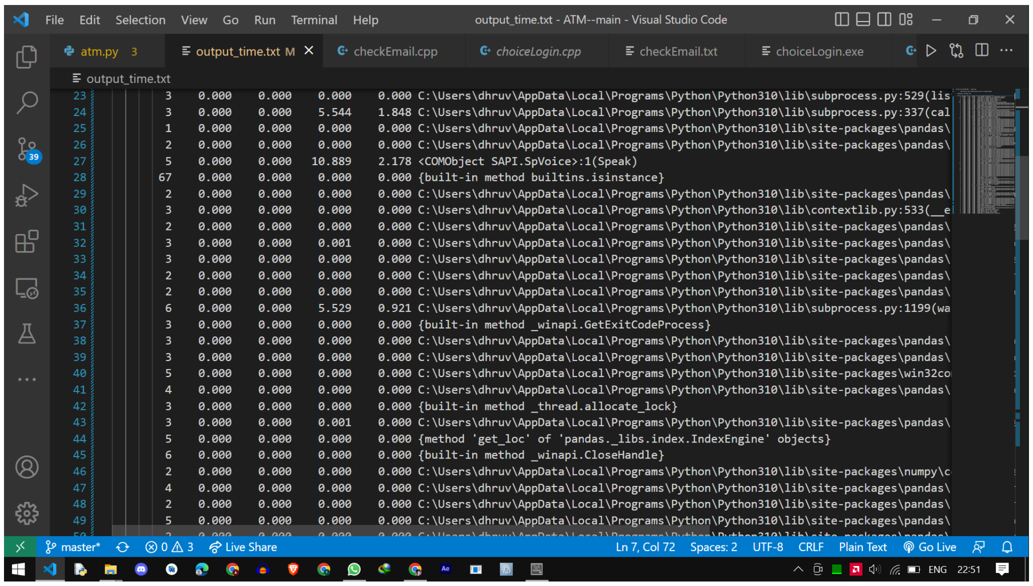
This is the choice selector file for afterLogin function in python 13> choiceLogin.cpp:

This is the choice selector for login menu

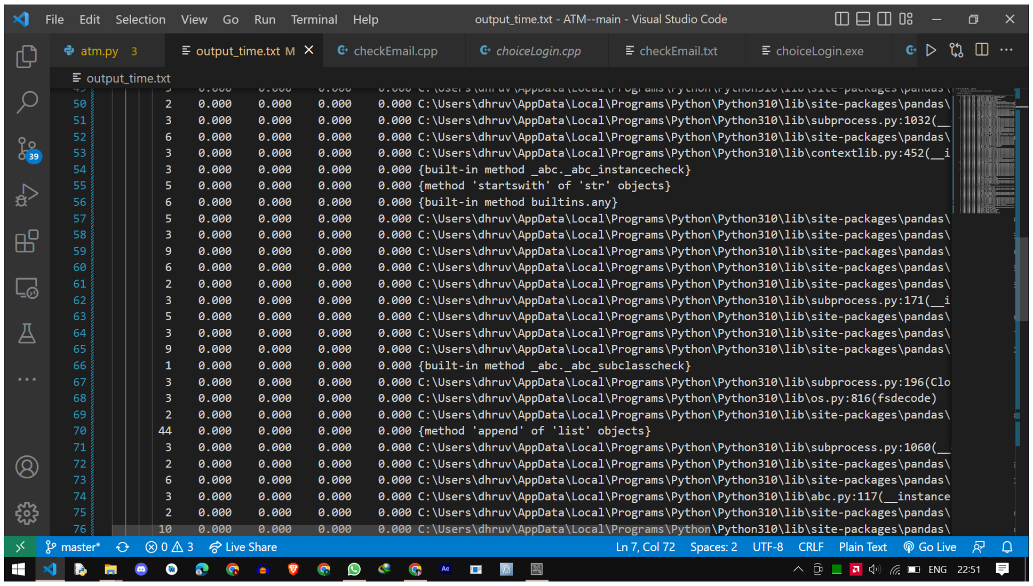
PROFILING



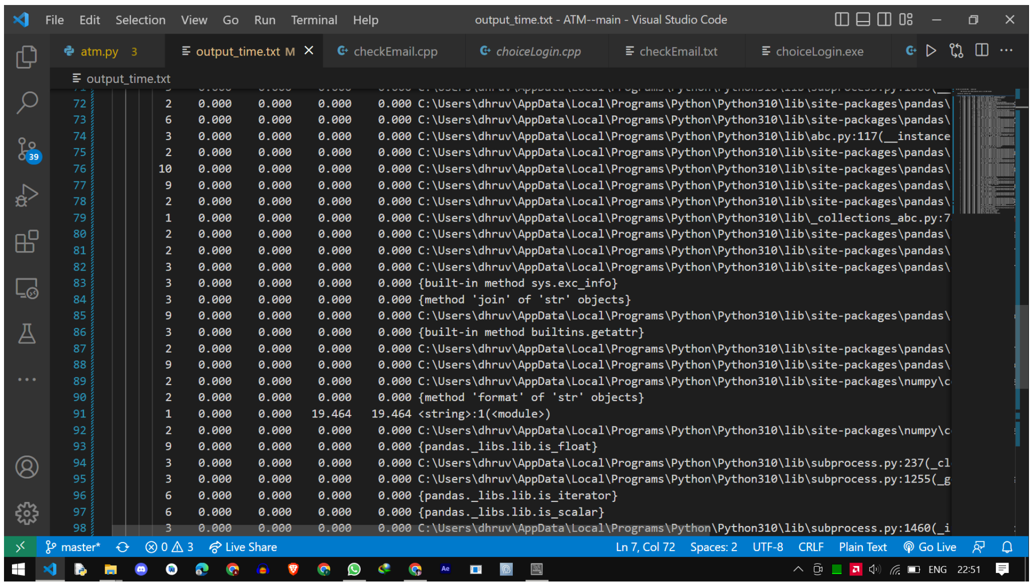
```
Tue Nov 15 22:49:40 2022    output.dat

         502 function calls (501 primitive calls) in 19.465 seconds

   Ordered by: internal time

   ncalls  tottime  percall  cumtime  percall filename:lineno(function)
        5   10.889    2.178   10.889    2.178 {method 'InvokeTypes' of 'PyIDispatch' objects}
        3    5.529    1.843    5.529    1.843 {built-in method _winapi.WaitForSingleObject}
        2    3.021    1.510    3.021    1.510 {built-in method builtins.input}
        3    0.014    0.005    0.014    0.005 {built-in method _winapi.CreateProcess}
       21    0.009    0.000    0.009    0.000 {built-in method builtins.print}
      2/1    0.000    0.000   19.464   19.464 c:\Users\dhruv\Desktop\ATM--main\atm.py:144(menu)
        3    0.000    0.000    0.015    0.005 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:1349(_e
        3    0.000    0.000    0.015    0.005 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:753(__i
        2    0.000    0.000    0.000    0.000 {pandas._libs.lib.infer_dtype}
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:1222(_c
        5    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\win32co
        1    0.000    0.000   19.465   19.465 {built-in method builtins.exec}
        6    0.000    0.000    5.529    0.921 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:1477(_w
        1    0.000    0.000    9.397    9.397 c:\Users\dhruv\Desktop\ATM--main\atm.py:276(afterLogin)
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:529(lis
        3    0.000    0.000    5.544    1.848 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:337(cal
        1    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        5    0.000    0.000   10.889    2.178 <COMObject SAPI.SpVoice>:1(Speak)
```



```
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:529(lis
        3    0.000    0.000    5.544    1.848 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:337(cal
        1    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        5    0.000    0.000   10.889    2.178 <COMObject SAPI.SpVoice>:1(Speak)
       67    0.000    0.000    0.000    0.000 {built-in method builtins.isinstance}
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\contextlib.py:533(__e
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.001    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        6    0.000    0.000    5.529    0.921 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:1199(wa
        3    0.000    0.000    0.000    0.000 {built-in method _winapi.GetExitCodeProcess}
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        5    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\win32co
        4    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.000    0.000 {built-in method _thread.allocate_lock}
        3    0.000    0.000    0.001    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        5    0.000    0.000    0.000    0.000 {method 'get_loc' of 'pandas._libs.index.IndexEngine' objects}
        6    0.000    0.000    0.000    0.000 {built-in method _winapi.CloseHandle}
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\c
        4    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        5    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
```

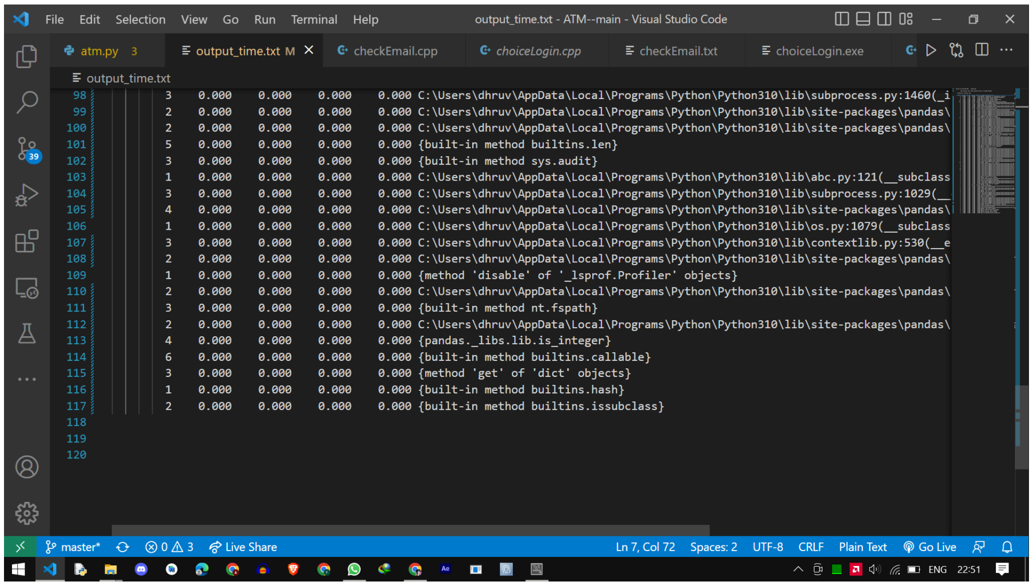


```
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:1032(__
        6    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\contextlib.py:452(__i
        3    0.000    0.000    0.000    0.000 {built-in method _abc._abc_instancecheck}
        5    0.000    0.000    0.000    0.000 {method 'startswith' of 'str' objects}
        6    0.000    0.000    0.000    0.000 {built-in method builtins.any}
        5    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        9    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        6    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:171(__i
        5    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        9    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        1    0.000    0.000    0.000    0.000 {built-in method _abc._abc_subclasscheck}
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:196(Clo
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\os.py:816(fsdecode)
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
       44    0.000    0.000    0.000    0.000 {method 'append' of 'list' objects}
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:1060(__
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        6    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
        3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\abc.py:117(__instance
        2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
       10    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
```



output_time.txt - ATM--main - Visual Studio Code

```
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      6    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\abc.py:117(__instance
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
     10    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      9    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      1    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\_collections_abc.py:7
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      3    0.000    0.000    0.000    0.000 {built-in method sys.exc_info}
      3    0.000    0.000    0.000    0.000 {method 'join' of 'str' objects}
      9    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      3    0.000    0.000    0.000    0.000 {built-in method builtins.getattr}
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      9    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\c
      2    0.000    0.000    0.000    0.000 {method 'format' of 'str' objects}
      1    0.000    0.000   19.464   19.464 <string>:1(<module>)
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\numpy\c
      9    0.000    0.000    0.000    0.000 {pandas._libs.lib.is_float}
      3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:237(_cl
      3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:1255(_g
      6    0.000    0.000    0.000    0.000 {pandas._libs.lib.is_iterator}
      6    0.000    0.000    0.000    0.000 {pandas._libs.lib.is_scalar}
      3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:1460(_i
```

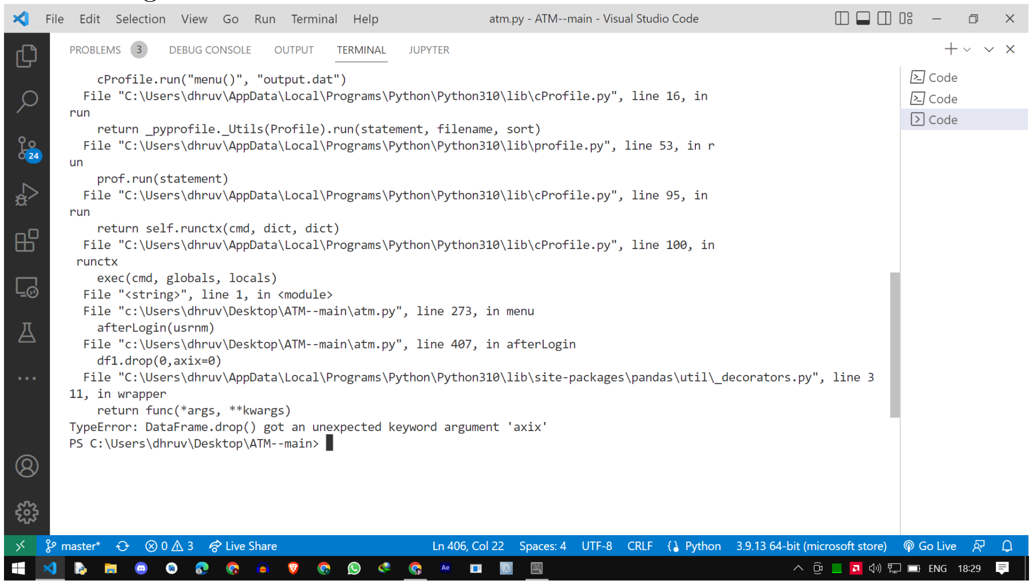


output_time.txt - ATM--main - Visual Studio Code

```
      3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:1460(_i
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      5    0.000    0.000    0.000    0.000 {built-in method builtins.len}
      3    0.000    0.000    0.000    0.000 {built-in method sys.audit}
      1    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\abc.py:121(__subclass
      3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\subprocess.py:1029(__
      4    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      1    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\os.py:1079(__subclass
      3    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\contextlib.py:530(__e
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      1    0.000    0.000    0.000    0.000 {method 'disable' of '_lsprof.Profiler' objects}
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      3    0.000    0.000    0.000    0.000 {built-in method nt.fspath}
      2    0.000    0.000    0.000    0.000 C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\
      4    0.000    0.000    0.000    0.000 {pandas._libs.lib.is_integer}
      6    0.000    0.000    0.000    0.000 {built-in method builtins.callable}
      3    0.000    0.000    0.000    0.000 {method 'get' of 'dict' objects}
      1    0.000    0.000    0.000    0.000 {built-in method builtins.hash}
      2    0.000    0.000    0.000    0.000 {built-in method builtins.issubclass}
```

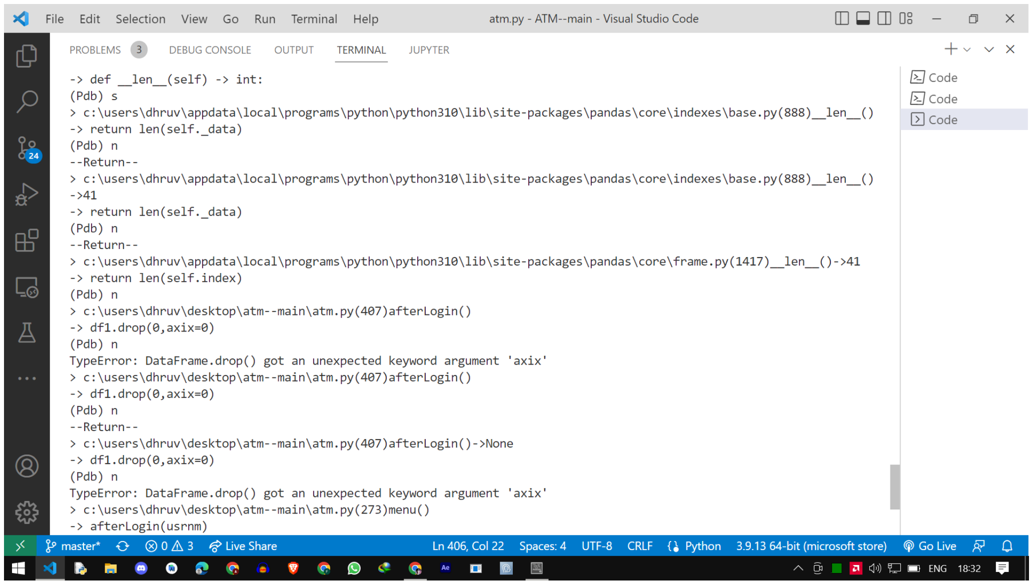
DEBUGGING USED

First Debug – Error–



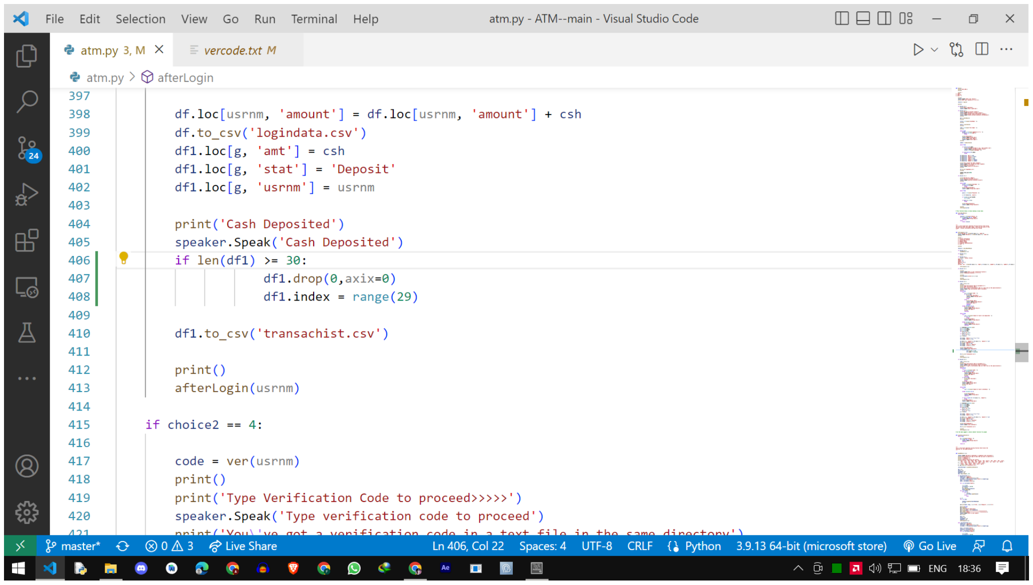
```
    cProfile.run("menu()", "output.dat")
  File "C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\cProfile.py", line 16, in
run
    return _pyprofile._Utils(Profile).run(statement, filename, sort)
  File "C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\profile.py", line 53, in r
un
    prof.run(statement)
  File "C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\cProfile.py", line 95, in
run
    return self.runctx(cmd, dict, dict)
  File "C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\cProfile.py", line 100, in
 runctx
    exec(cmd, globals, locals)
  File "<string>", line 1, in <module>
  File "c:\Users\dhruv\Desktop\ATM--main\atm.py", line 273, in menu
    afterLogin(usrnm)
  File "c:\Users\dhruv\Desktop\ATM--main\atm.py", line 407, in afterLogin
    df1.drop(0,axix=0)
  File "C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\util\_decorators.py", line 3
11, in wrapper
    return func(*args, **kwargs)
TypeError: DataFrame.drop() got an unexpected keyword argument 'axix'
PS C:\Users\dhruv\Desktop\ATM--main>
```

–Caught the error using debugger–(pdb in python)



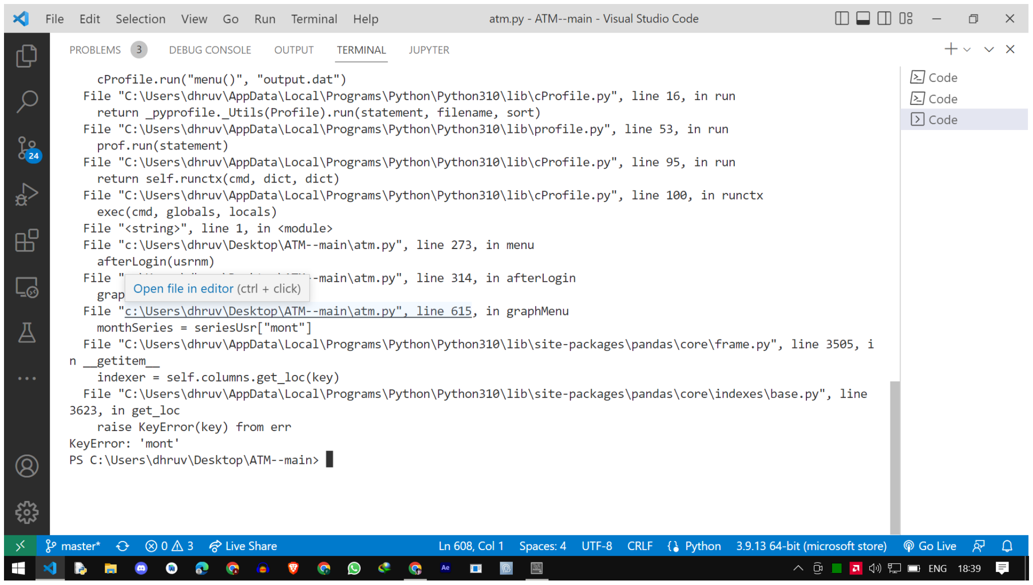
```
-> def __len__(self) -> int:
(Pdb) s
> c:\users\dhruv\appdata\local\programs\python\python310\lib\site-packages\pandas\core\indexes\base.py(888)__len__()
-> return len(self._data)
(Pdb) n
--Return--
> c:\users\dhruv\appdata\local\programs\python\python310\lib\site-packages\pandas\core\indexes\base.py(888)__len__()
->41
-> return len(self._data)
(Pdb) n
--Return--
> c:\users\dhruv\appdata\local\programs\python\python310\lib\site-packages\pandas\core\frame.py(1417)__len__()->41
-> return len(self.index)
(Pdb) n
> c:\users\dhruv\desktop\atm--main\atm.py(407)afterLogin()
-> df1.drop(0,axix=0)
(Pdb) n
TypeError: DataFrame.drop() got an unexpected keyword argument 'axix'
> c:\users\dhruv\desktop\atm--main\atm.py(407)afterLogin()
-> df1.drop(0,axix=0)
(Pdb) n
--Return--
> c:\users\dhruv\desktop\atm--main\atm.py(407)afterLogin()->None
-> df1.drop(0,axix=0)
(Pdb) n
TypeError: DataFrame.drop() got an unexpected keyword argument 'axix'
> c:\users\dhruv\desktop\atm--main\atm.py(273)menu()
-> afterLogin(usrnm)
```

–Corrected the code–



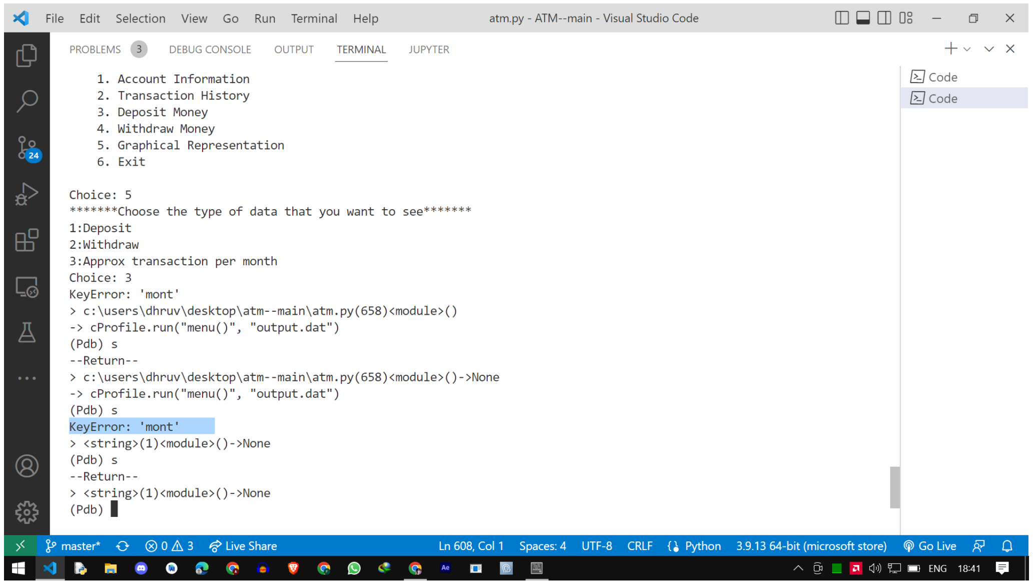
```
        df.loc[usrnm, 'amount'] = df.loc[usrnm, 'amount'] + csh
        df.to_csv('logindata.csv')
        df1.loc[g, 'amt'] = csh
        df1.loc[g, 'stat'] = 'Deposit'
        df1.loc[g, 'usrnm'] = usrnm

        print('Cash Deposited')
        speaker.Speak('Cash Deposited')
        if len(df1) >= 30:
                df1.drop(0,axix=0)
                df1.index = range(29)

        df1.to_csv('transachist.csv')

        print()
        afterLogin(usrnm)

    if choice2 == 4:

        code = ver(usrnm)
        print()
        print('Type Verification Code to proceed>>>>>')
        speaker.Speak('Type verification code to proceed')
```

Second Debug –Error–



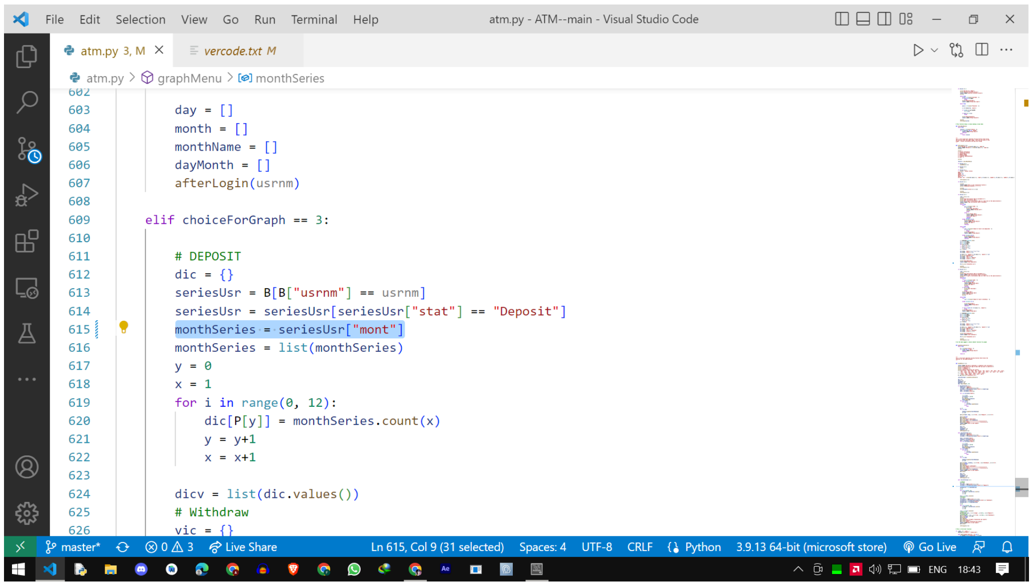
```
    cProfile.run("menu()", "output.dat")
  File "C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\cProfile.py", line 16, in run
    return _pyprofile._Utils(Profile).run(statement, filename, sort)
  File "C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\profile.py", line 53, in run
    prof.run(statement)
  File "C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\cProfile.py", line 95, in run
    return self.runctx(cmd, dict, dict)
  File "C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\cProfile.py", line 100, in runctx
    exec(cmd, globals, locals)
  File "<string>", line 1, in <module>
  File "c:\Users\dhruv\Desktop\ATM--main\atm.py", line 273, in menu
    afterLogin(usrnm)
  File "                      -main\atm.py", line 314, in afterLogin
    grap
  File "c:\Users\dhruv\Desktop\ATM--main\atm.py", line 615, in graphMenu
    monthSeries = seriesUsr["mont"]
  File "C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\frame.py", line 3505, i
n __getitem__
    indexer = self.columns.get_loc(key)
  File "C:\Users\dhruv\AppData\Local\Programs\Python\Python310\lib\site-packages\pandas\core\indexes\base.py", line
3623, in get_loc
    raise KeyError(key) from err
KeyError: 'mont'
PS C:\Users\dhruv\Desktop\ATM--main>
```

–Caught the error using debugger– (pdb in python)



```
    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 5
*******Choose the type of data that you want to see*******
1:Deposit
2:Withdraw
3:Approx transaction per month
Choice: 3
KeyError: 'mont'
> c:\users\dhruv\desktop\atm--main\atm.py(658)<module>()
-> cProfile.run("menu()", "output.dat")
(Pdb) s
--Return--
> c:\users\dhruv\desktop\atm--main\atm.py(658)<module>()->None
-> cProfile.run("menu()", "output.dat")
(Pdb) s
KeyError: 'mont'
> <string>(1)<module>()->None
(Pdb) s
--Return--
> <string>(1)<module>()->None
(Pdb)
```

–Corrected the code–



File Edit Selection View Go Run Terminal Help
atm.py - ATM--main - Visual Studio Code
atm.py 3, M
vercode.txt M
atm.py > graphMenu > monthSeries
master* 0 3 Live Share Ln 615, Col 9 (31 selected) Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (microsoft store) Go Live
ENG 18:43

```
        day = []
        month = []
        monthName = []
        dayMonth = []
        afterLogin(usrnm)

    elif choiceForGraph == 3:

        # DEPOSIT
        dic = {}
        seriesUsr = B[B["usrnm"] == usrnm]
        seriesUsr = seriesUsr[seriesUsr["stat"] == "Deposit"]
        monthSeries = seriesUsr["mont"]
        monthSeries = list(monthSeries)
        y = 0
        x = 1
        for i in range(0, 12):
            dic[P[y]] = monthSeries.count(x)
            y = y+1
            x = x+1

        dicv = list(dic.values())
        # Withdraw
        vic = {}
```

"""

This program/software is a basic ATM(Automated Teller Machine) Software th
can do things like :

1> Login , Signup
2> Deposit , WithDraw , See Graphs(of deposits and withdraw)
3> Account Details , Transaction history

Special Features of this ATM Project :

1> It consist of Audio instructions in it
2> It can also show graph of deposits , withdraw
3> Have security system like OTP , that occurs in the same text file
that is present in this directory

"""

```
# Importing necessary modules

import pandas as pd
from win32com.client import *
import matplotlib.pyplot as plt
import numpy as np
import pygame
import random
import datetime
import re
import cProfile
import pstats
from pstats import SortKey

speaker = Dispatch('SAPI.SpVoice')
df = pd.read_csv('logindata.csv', index_col=0)
df1 = pd.read_csv('transachist.csv', index_col=0)


print('WELCOME TO TATA ATMs')
speaker.Speak('Welcome to tata Automated Teller Machines')


"""
This function is handling the login process of the whole software
1> It takes input of all the required things and even
checks for correct ( password , email and many more )

"""


def login():
    while True:
        try:
            choice1 = int(input('Choice: '))
        except Exception as e:
            print('Wrong Data')
            speaker.Speak('Wrong Choice')
            continue
        if choice1 > 0 and choice1 < 4:
            break
        print('Wrong␣Choice')
        speaker.Speak('Wrong␣Choice')
        print('Enter␣again')
        speaker.Speak('Enter␣Again')
        print()
    return choice1


""" This function checks whether the mail is correct or not
, it checks all the possible mistakes that can be in the gmail
"""


def checkEmail():

    gml = str(input('Gmail␣ID:␣'))
    regex = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'

    if (re.fullmatch(regex, gml)):

        return gml

    else:
        print("Invalid␣Email")
        checkEmail()


"""
This function checks whether the entered password is go to go or
not , as the password should be secure , should not be to easy
"""


def passCheck():
    l, u, p, d = 0, 0, 0, 0
    s = str(input('Password:␣'))
    if (len(s) >= 8):
        for i in s:
            # counting lowercase alphabets
            if (i.islower()):
                l += 1

            # counting uppercase alphabets
            if (i.isupper()):
                u += 1

            # counting digits
            if (i.isdigit()):
                d += 1

            # counting the mentioned special characters
            if (i == '@' or i == '$' or i == '_'):
                p += 1
    if (l >= 1 and u >= 1 and p >= 1 and d >= 1 and l+p+u+d == len(s)):
        return s
    else:
        print("invalid password")
        passCheck()


# As the name suggest , it checks the correctness of aadhar number


def aadharCheck():
    while True:
        try:
            aadhar = int(input('Aadhar No.: '))
        except Exception as e:
            print('Wrong Data')
            speaker.Speak('Wrong Data')
            print('Please Enter Again')
            speaker.Speak('Enter again')
            continue
        if len(str(aadhar)) == 12 and aadhar not in df['aadhar']:
            break

        print('Wrong Data')
        speaker.Speak('Wrong Data')
        print('Please Enter Again')
        speaker.Speak('Enter again')
    return aadhar

# This function help in communicating between CSV file and
# Frontend


def ver(usrnm):
    code = random.randint(1000, 9999)
    f = open('vercode.txt', mode='w')
    f.write(str(code))
    f.close()
    return code


"""
This is the main driving method of the software , that initiate
program havinf Login , Sign up and Exit option
"""


def menu():
    print('TATA ATMs')
    print()

    print('''
1. Login
2. Sign up
3. Exit ''')

    print()
    speaker.Speak('Enter your choice')
    print('Enter your choice(1 or 2 or 3)')

    choice1 = login()

    print()
```

```
if choice1 == 3:
    print('Have a Good Day')
    speaker.Speak('Have a good day')

if choice1 == 2:
    print('Welcome to Account Creator')
    speaker.Speak('Welcome to account creator')
    print('Please provide these informations kindly')
    speaker.Speak('Kindly provide necessary informations')
    print()

    gml = checkEmail()
    print()

    usrnm = str(input('UserName: '))
    print()

    passw = passCheck()
    print()

    name = str(input('Full Name: '))
    print()

    while True:
        gender = str(input('Gender(m or f): '))
        if gender in list('mMfF'):
            break

        print('Wrong data')
        speaker.Speak('Wrong data')
        print('Please enter again')
        speaker.Speak('Enter again')
    print()

    aadhar = aadharCheck()

    while True:

        if usrnm in df.index:
            print('Username already in use, type another one')
```

```
                speaker.Speak('Username␣already␣in␣use')
                usrnm = str(input('UserName:␣'))

            if usrnm not in df.index:
                break

        df.loc[usrnm, 'gml'] = gml
        df.loc[usrnm, 'passw'] = passw
        df.loc[usrnm, 'amount'] = 0
        df.loc[usrnm, 'name'] = name
        df.loc[usrnm, 'gender'] = gender
        df.loc[usrnm, 'aadhar'] = aadhar

        print('Your␣account␣has␣been␣created')
        speaker.Speak('Your␣account␣has␣been␣created')
        print('Enjoy␣our␣services')
        speaker.Speak('Enjoy␣our␣Services')
        print()

        df.to_csv('logindata.csv')
        print()

        pygame.time.wait(2000)
        menu()

    if choice1 == 1:

        print('Welcome␣Sir\\Madam')
        print('Security␣Protocols␣Active')
        speaker.Speak('Security␣Protocols␣active')
        print()

        while True:
            usrnm = str(input('Username:␣'))
            if usrnm in df.index:
                break
            print('Wrong␣Username')
            speaker.Speak('Wrong␣data␣input')

        while True:
            passw = str(input('Password: '))

            a = df.loc[usrnm, 'passw']

            if type(a) == np.float64:
                a = int(a)

            if passw == str(a):
                break

            print('Wrong Password')
            speaker.Speak('Wrong Password')

        print()
        afterLogin(usrnm)


# This function helps in choice making in main menu

def choiceMainMenu():
    while True:

        choice2 = int(input('Choice: '))
        if str(choice2) not in '123456':
            speaker.Speak('Wrong Choice')
            continue
        else:
            return choice2


"""
This is the second most important driving function that drives
the function after the login stage , here you can access to the
Graphs , Account Information and many more things
"""


def afterLogin(usrnm):
    print('Welcome {}'.format(df.loc[usrnm, 'name']))
```

```
speaker.Speak('Welcome {}'.format(df.loc[usrnm, 'name']))
print()

print('''
1. Account Information
2. Transaction History
3. Deposit Money
4. Withdraw Money
5. Graphical Representation
6. Exit''')

print()

choice2 = choiceMainMenu()

if choice2 == 5:
    graphMenu(usrnm)

if choice2 == 6:
    menu()

if choice2 == 1:
    print('''Status: Active
Name: {}
Gender: {}
Aadhar: {}
Balance: Rs.{}
Gmail: {}
At Risk : No'''.format(df.loc[usrnm, 'name'], df.loc[usrnm, 'gender'],

    afterLogin(usrnm)

if choice2 == 2:

    print()
    speaker.Speak('Here is your transaction history')
    print('Transaction History:-----------')

    print()
    print(df1[df1['usrnm'] == usrnm])
```

```
        print()
        afterLogin(usrnm)

    if choice2 == 3:

        code = ver(usrnm)
        print()
        print('Type Verification Code to proceed>>>>>')
        print('Type verification code')
        print('You\'ve got a verification code in a text file in the same 
        speaker.Speak('Type verification code to proceed')
        print()
        while True:
            try:
                cd = int(input('Code: '))
                if cd == code:
                    print('User Verified')
                    speaker.Speak('Welcome user')
                    print()
                    break
                else:
                    print('Wrong code')
                    speaker.Speak('Wrong Choice')
                    print('Type again')
                    continue
                    print()
            except Exception as e:
                print('Wrong code')
                speaker.Speak('Wrong Choice')
                print('Type again')
                print()
                continue

        while True:
            try:
                csh = int(input('Amount of Cash to be deposited: '))
                if csh > 0:
                    break
                print('Wrong Data')
                speaker.Speak('Wrong Choice')

            except Exception as e:
                print('Wrong Data')
                speaker.Speak('Wrong Choice')
                continue

        x = datetime.datetime.now()
        y_x = str(x.year)
        m_x = str(x.month)
        d_x = str(x.day)
        if len(m_x) == 1:
            m_x = '0' + m_x
        if len(d_x) == 1:
            d_x = '0' + d_x
        g = len(df1)

        df1.loc[g, 'date'] = y_x + m_x + d_x
        df1.loc[g, 'month'] = m_x

        df.loc[usrnm, 'amount'] = df.loc[usrnm, 'amount'] + csh
        df.to_csv('logindata.csv')
        df1.loc[g, 'amt'] = csh
        df1.loc[g, 'stat'] = 'Deposit'
        df1.loc[g, 'usrnm'] = usrnm

        print('Cash Deposited')
        speaker.Speak('Cash Deposited')

        df1.to_csv('transachist.csv')

        print()
        afterLogin(usrnm)

    if choice2 == 4:

        code = ver(usrnm)
        print()
        print('Type Verification Code to proceed>>>>>')
        speaker.Speak('Type verification code to proceed')
```

```
print('You\'ve got a verification code in a text file in the same 
print()
while True:
    try:
        cd = int(input('Code: '))
    except Exception as e:
        print('Wrong code')
        speaker.Speak('Wrong Code')
        print('Type again')
        print()
        continue
    if cd == code:
        print('User Verified')
        print()
        break
    print('Wrong code')
    speaker.Speak('Wrong Code')
    print('Type again')
    print()

while True:
    try:
        csh = int(input('Amount of Cash to Withdraw: '))

    except Exception as e:

        print('Wrong Data')
        speaker.Speak('Wrong data')
        continue

    if csh > 0 and csh < df.loc[usrnm, 'amount']:
        break
    print('Wrong Data')
    speaker.Speak('Wrong data')

x = datetime.datetime.now()
y_x = str(x.year)
m_x = str(x.month)
d_x = str(x.day)
if len(m_x) == 1:
```

```
            m_x = '0' + m_x
        if len(d_x) == 1:
            d_x = '0' + d_x
        g = len(df1)

        df1.loc[g, 'date'] = y_x + m_x + d_x
        df1.loc[g, 'month'] = m_x

        df.loc[usrnm, 'amount'] = df.loc[usrnm, 'amount'] + csh
        df.to_csv('logindata.csv')
        df1.loc[g, 'amt'] = csh
        df1.loc[g, 'stat'] = 'Withdraw'
        df1.loc[g, 'usrnm'] = usrnm

        print('Cash Withdrawn')
        speaker.Speak('Cash withdrawn')

        df1.to_csv('transachist.csv')

        print()
        afterLogin(usrnm)

# As the name suggest a choice checker function for graphs


def graphChoiceChecker():
    while True:

        A = int(input('Choice: '))
        if str(A) not in "123":
            speaker.Speak("Wrong Choice")
            continue

        return A


"""
This is the third important driving function that drives the
function for the graph purposes
"""
```

```
def graphMenu(usrnm):

    speaker.Speak('Welcome␣to␣advanced␣a␣i␣graphical␣stat␣calculator')
    print("*******Choose␣the␣type␣of␣data␣that␣you␣want␣to␣see*******")
    print("""1: Deposit""")
    print("""2: Withdraw""")
    print("3: Approx␣transaction␣per␣month")
    L = {"01": "Jan", "02": "Feb", "03": "March", "04": "April", "05": "Ma
         "07": "July", "08": "Aug", "09": "Sept", "10": "Oct", '11': "Nov"
    P = ["Jan", "Feb", "March", "April", "May", "June",
         "July", "Aug", "Sept", "Oct", "Nov", "Dec"]
    B = pd.read_csv("transachist.csv")

    choiceForGraph = graphChoiceChecker()

    day = []
    month = []
    monthName = []
    dayMonth = []
    npp = np.arange(1, 13)

    if choiceForGraph == 1:
        depString = "Deposit"
        seriesUsr = B[B["usrnm"] == usrnm]
        seriesUsr = seriesUsr[seriesUsr["stat"] == depString]
        Depo = seriesUsr["amt"]
        Date = seriesUsr["serno"]+1

        for i in seriesUsr["date"]:

            i = str(i)
            seriesUser3 = i[6:8]
            E = i[4:6]
            day.append(seriesUser3)
            month.append(E)
        for i in month:
            for z in L:
                if i == z:
                    monthName.append(L[z])
                else:
                    pass

        q = 0
        for i in day:

            dayMonth.append(i+monthName[q])
            q = q+1

        plt.plot(Date, Depo, color="red", label="Deposit", marker="o")

        plt.grid(True)
        plt.legend()
        plt.title("Amount␣deposited")
        plt.xticks(Date, labels=dayMonth)
        plt.xlabel("Date:−−−−−−−−−−−−−−−−>>>>>>>>>>>>>")
        plt.ylabel("Amount:−−−−−−−−−−−−−−−−>>>>>>>>>>>>>")
        speaker.Speak('Here␣is␣your␣graph')
        plt.show()

        day = []
        month = []
        monthName = []
        dayMonth = []
        afterLogin(usrnm)

    elif choiceForGraph == 2:
        depString = "Withdraw"
        seriesUsr = B[B["usrnm"] == usrnm]
        seriesUsr = seriesUsr[seriesUsr["stat"] == depString]

        Date = seriesUsr["date"]
        withdraw = seriesUsr["amt"]
        Date = seriesUsr["serno"]+1
        for i in seriesUsr["date"]:

            i = str(i)
            seriesUser3 = i[6:8]
            E = i[4:6]
            day.append(seriesUser3)
            month.append(E)
        for i in month:
            for z in L:
                if i == z:
                    monthName.append(L[z])
                else:
                    pass

        q = 0
        for i in day:

            dayMonth.append(i+monthName[q])
            q = q+1

        plt.plot(Date, withdraw, color="red", label="Withdraw", marker="o"
        plt.grid(True)
        plt.legend()
        plt.title("Amount withdrawed")
        plt.xlabel("Date:------------------------->>>>>>>>>>>>>")
        plt.ylabel("Amount:------------------------->>>>>>>>>>>>>")
        plt.xticks(Date, labels=dayMonth)
        speaker.Speak('Here is your graph')
        plt.show()

        day = []
        month = []
        monthName = []
        dayMonth = []
        afterLogin(usrnm)

    elif choiceForGraph == 3:

        # DEPOSIT
        dic = {}
        seriesUsr = B[B["usrnm"] == usrnm]
        seriesUsr = seriesUsr[seriesUsr["stat"] == "Deposit"]
        monthSeries = seriesUsr["month"]
        monthSeries = list(monthSeries)
        y = 0
    x = 1
    for i in range(0, 12):
        dic[P[y]] = monthSeries.count(x)
        y = y+1
        x = x+1

    dicv = list(dic.values())
    # Withdraw
    vic = {}
    seriesUser3 = B[B["usrnm"] == usrnm]
    seriesUser3 = seriesUser3[seriesUser3["stat"] == "Withdraw"]
    monthSeries3 = seriesUser3["month"]
    monthSeries3 = list(monthSeries3)
    y = 0
    x = 1
    for i in range(0, 12):
        vic[P[y]] = monthSeries3.count(x)
        y = y+1
        x = x+1

    vicv = list(vic.values())
    # Deposit bar
    plt.bar(npp, dicv, color='orange', width=0.2, label="Deposit")
    # withdraw bar
    plt.bar(npp+0.2, vicv, color='red', width=0.2, label="Withdraw")
    plt.xticks(npp, labels=P)
    plt.grid(True)
    plt.legend()
    plt.ylabel("No. of approx transactions per month")
    plt.xlabel("Month")
    plt.title('Approx transaction per month')
    speaker.Speak('Here is your graph')
    plt.show()

    afterLogin(usrnm)


# This is the driver function

if __name__ == "__main__":
cProfile.run("menu()", "output.dat")

with open("output_time.txt", "w") as f:
    p = pstats.Stats("output.dat", stream=f)
    p.sort_stats("time").print_stats()
```

C++ code used in this project

This is the choiceLogin.cpp file

```
#include <iostream>
using namespace std;

int afterLogin()
{
    int choice;
    cin >> choice;
    if (choice > 0 && choice < 4)
    {
        return choice;
    }
    printf("invalid choice\n");
    afterLogin();
    return 0;
}

int main()
{

    return afterLogin();
}
```

This is the aadharCheck.cpp file

```
#include <iostream>
#include <bits/stdc++.h>
using namespace std;

int main()
```

```
{

    long long int aadharNumber;
    while (true)
    {

        cout << "Enter Your aadhar Number ";
        cin >> aadharNumber;

        if (to_string(aadharNumber).length() == 12)
        {

            fstream file;

            // opening file "Gfg.txt"
            // in out(write) mode
            // ios::out Open for output operations.
            file.open("checkAadhar.txt", ios::out);
            file << to_string(aadharNumber);
            file.close();
            break;
        }
    }

    return 0;
}
```

This is the checkEmail.cpp file

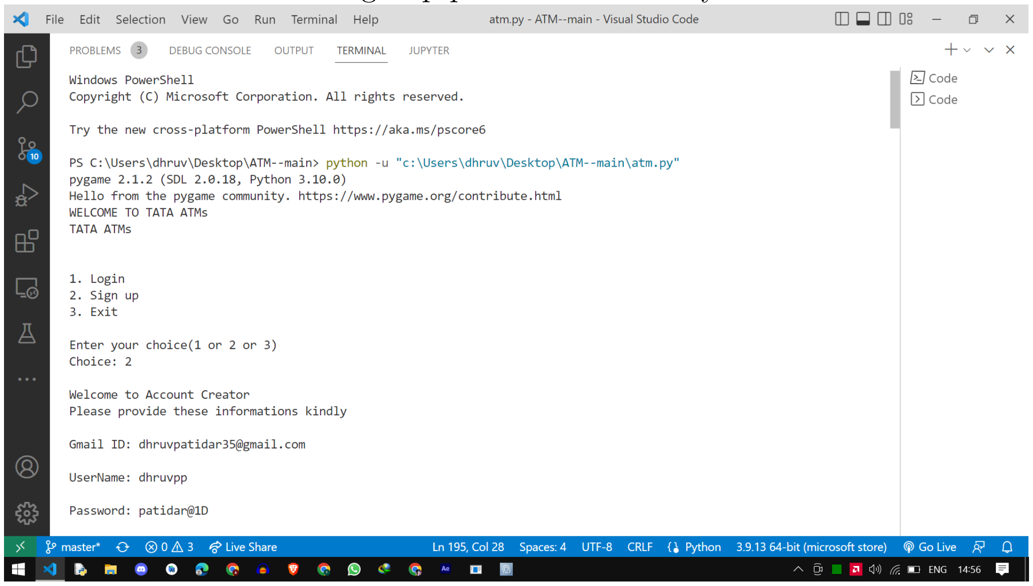
```
#include <iostream>
#include <regex>
#include <bits/stdc++.h>

using namespace std;

int main()
{
    string str;
    const regex pattern("(\\w+)(\\.|_)?(\\w*)@(\\w+)(\\.(\\w+))+");
    while (true)
    {
        cout << "Enter your Email-Id:" << endl;
        cin >> str;

        if (regex_match(str, pattern))
        {

            fstream file;

            file.open("checkEmail.txt", ios::out);
            file << str;
            file.close();

            break;
        }
    }

    return 0;
}
```

This is the choiceAfterLogin file

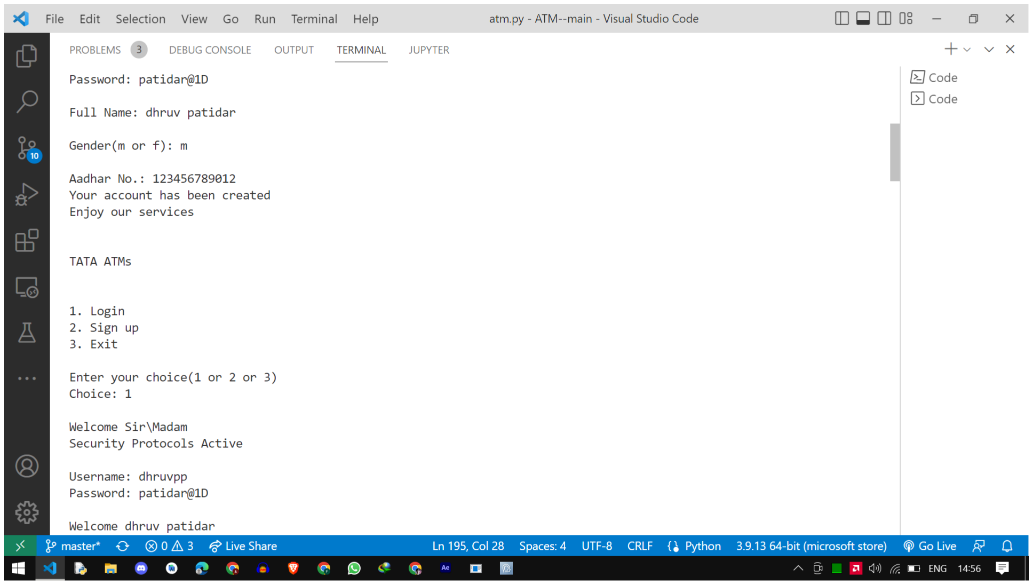
```
#include <iostream>
using namespace std;

int afterLogin()
{

    int choice;
    cout << "Enter the Choice :";
    cin >> choice;
    if (choice > 0 && choice < 7)
    {
        return choice;
    }
    cout << "invalid choice\n";

    afterLogin();
    return 0;
}

int main()
{

    return afterLogin();
}
```

OUTPUT OF THE CODE

This is the sign up part of the ATM system

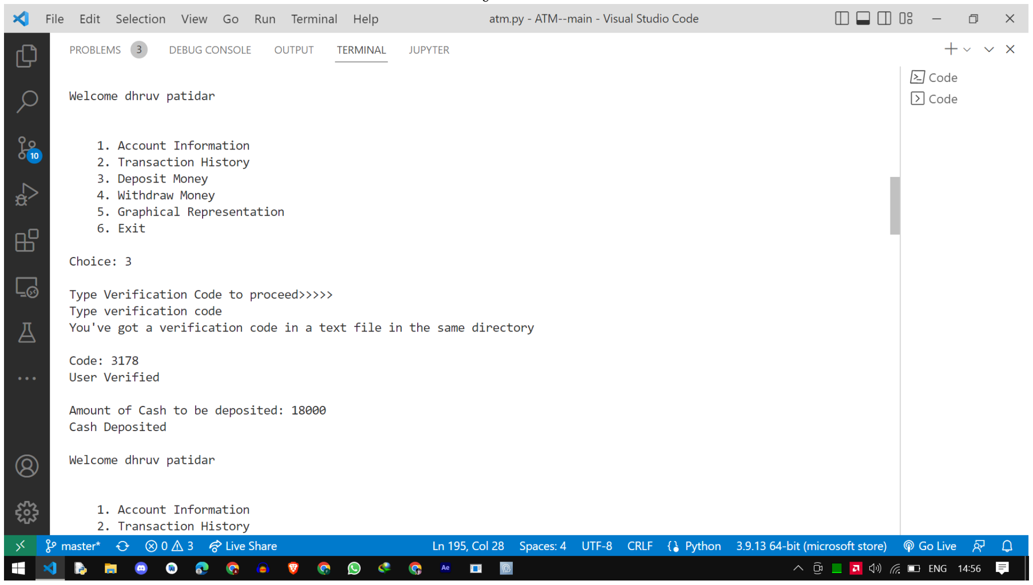

File Edit Selection View Go Run Terminal Help
atm.py - ATM--main - Visual Studio Code
PROBLEMS 3 DEBUG CONSOLE OUTPUT TERMINAL JUPYTER
Code
Code
master* 0 3 Live Share Ln 195, Col 28 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (microsoft store) Go Live
ENG 14:56

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\dhruv\Desktop\ATM--main> python -u "c:\Users\dhruv\Desktop\ATM--main\atm.py"
pygame 2.1.2 (SDL 2.0.18, Python 3.10.0)
Hello from the pygame community. https://www.pygame.org/contribute.html
WELCOME TO TATA ATMs
TATA ATMs


1. Login
2. Sign up
3. Exit

Enter your choice(1 or 2 or 3)
Choice: 2

Welcome to Account Creator
Please provide these informations kindly

Gmail ID: dhruvpatidar35@gmail.com

UserName: dhruvpp

Password: patidar@1D
```

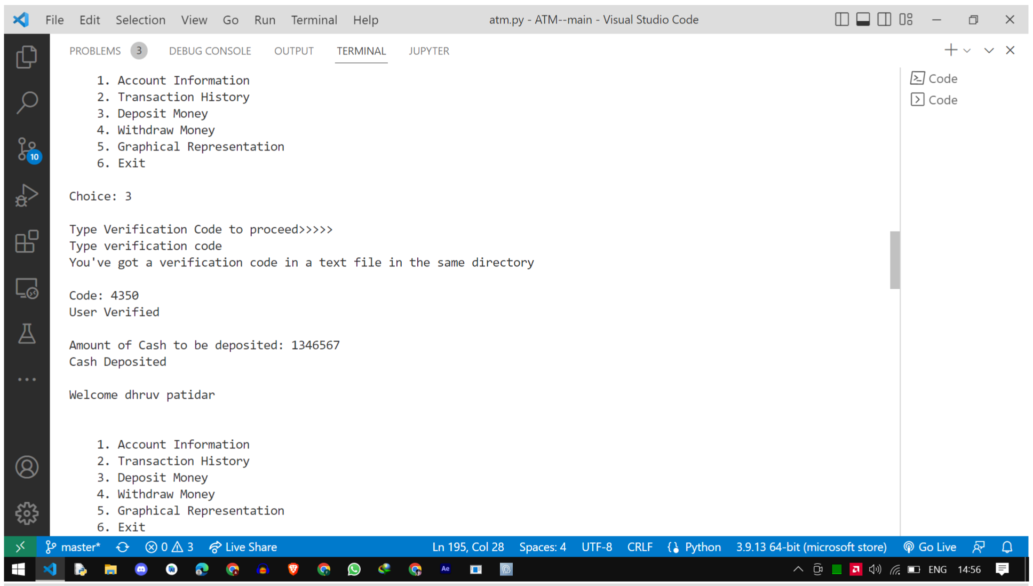


```
Password: patidar@1D

Full Name: dhruv patidar

Gender(m or f): m

Aadhar No.: 123456789012
Your account has been created
Enjoy our services


TATA ATMs


1. Login
2. Sign up
3. Exit

Enter your choice(1 or 2 or 3)
Choice: 1

Welcome Sir\Madam
Security Protocols Active

Username: dhruvpp
Password: patidar@1D

Welcome dhruv patidar
```

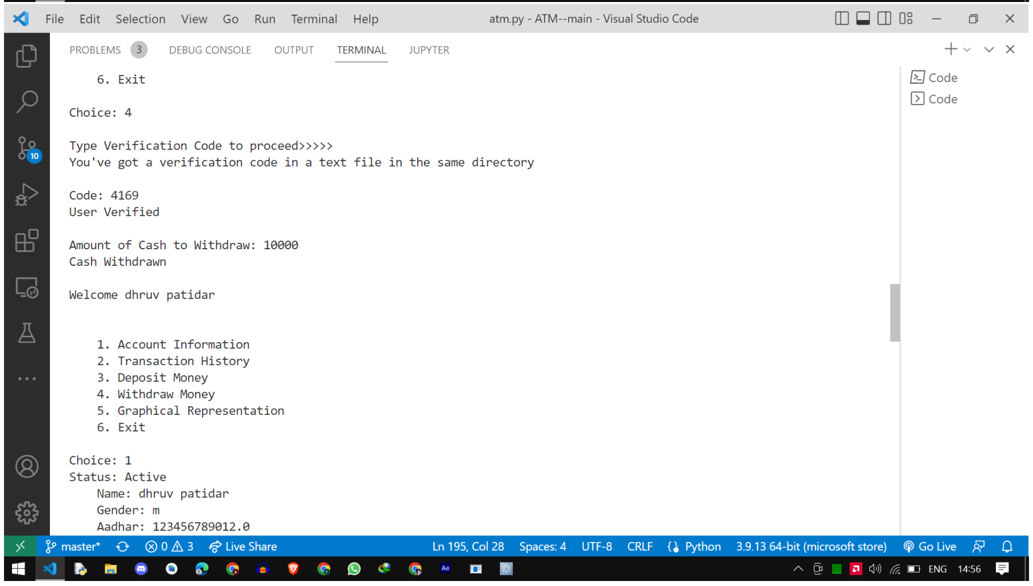
This is the second menu where you can do other functionalities



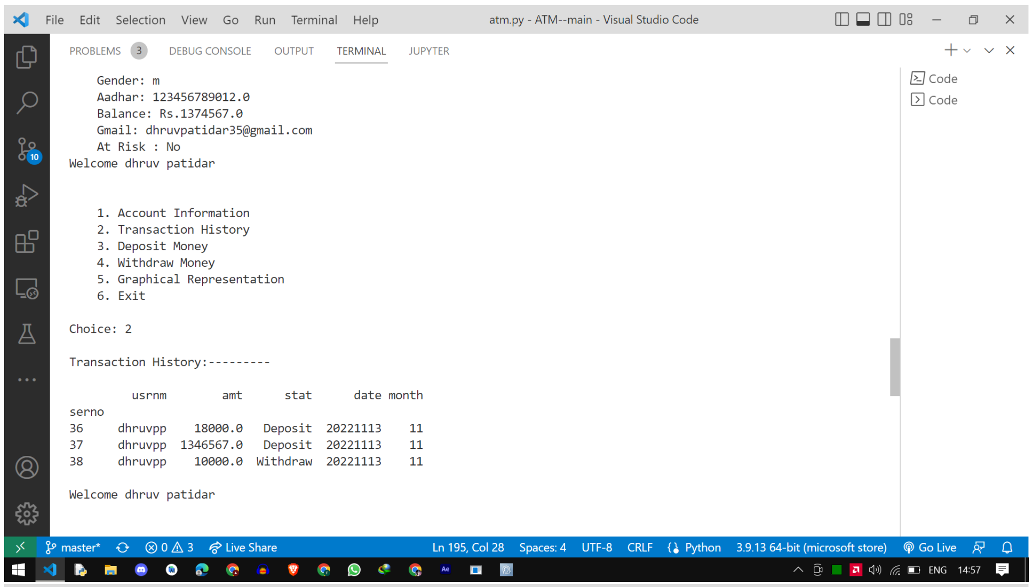
```
Welcome dhruv patidar


    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 3

Type Verification Code to proceed>>>>>
Type verification code
You've got a verification code in a text file in the same directory

Code: 3178
User Verified

Amount of Cash to be deposited: 18000
Cash Deposited

Welcome dhruv patidar


    1. Account Information
    2. Transaction History
```



```
    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 3

Type Verification Code to proceed>>>>>
Type verification code
You've got a verification code in a text file in the same directory

Code: 4350
User Verified

Amount of Cash to be deposited: 1346567
Cash Deposited

Welcome dhruv patidar


    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit
```

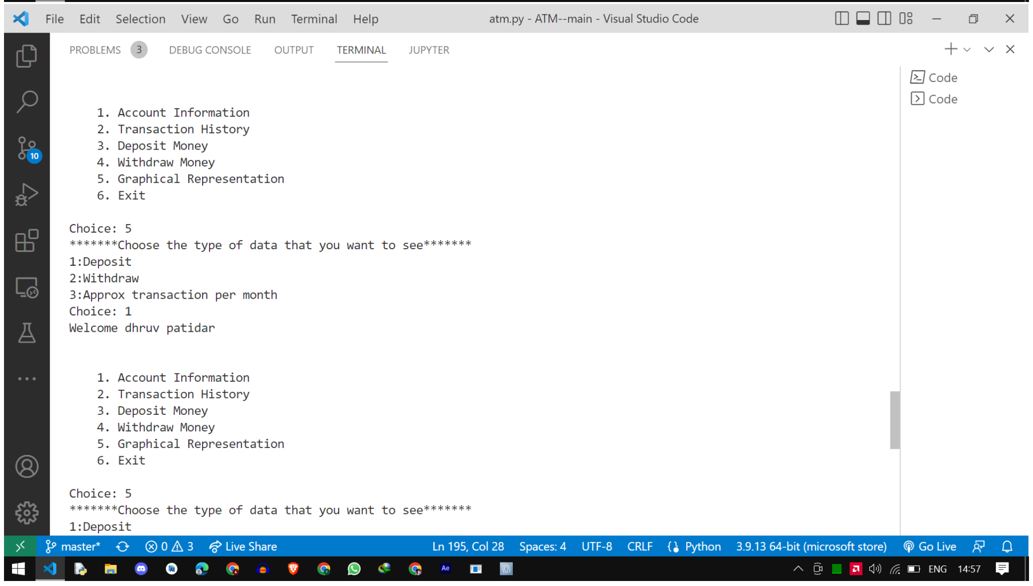


```
    6. Exit

Choice: 4

Type Verification Code to proceed>>>>>
You've got a verification code in a text file in the same directory

Code: 4169
User Verified

Amount of Cash to Withdraw: 10000
Cash Withdrawn

Welcome dhruv patidar


    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 1
Status: Active
    Name: dhruv patidar
    Gender: m
    Aadhar: 123456789012.0
```



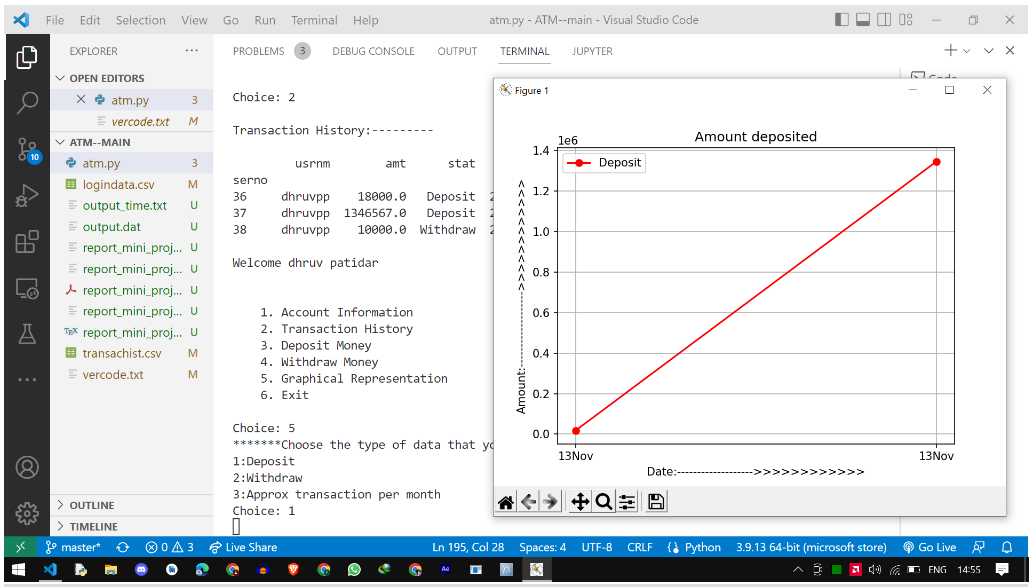
File Edit Selection View Go Run Terminal Help
atm.py - ATM--main - Visual Studio Code
PROBLEMS 3 DEBUG CONSOLE OUTPUT TERMINAL JUPYTER
Code
Code
master* 0 3 Live Share Ln 195, Col 28 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (microsoft store) Go Live
ENG 14:57

```
    Gender: m
    Aadhar: 123456789012.0
    Balance: Rs.1374567.0
    Gmail: dhruvpatidar35@gmail.com
    At Risk : No
Welcome dhruv patidar


    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 2

Transaction History:---------

         usrnm        amt      stat      date month
serno
36     dhruvpp    18000.0   Deposit  20221113    11
37     dhruvpp  1346567.0   Deposit  20221113    11
38     dhruvpp    10000.0  Withdraw  20221113    11

Welcome dhruv patidar
```

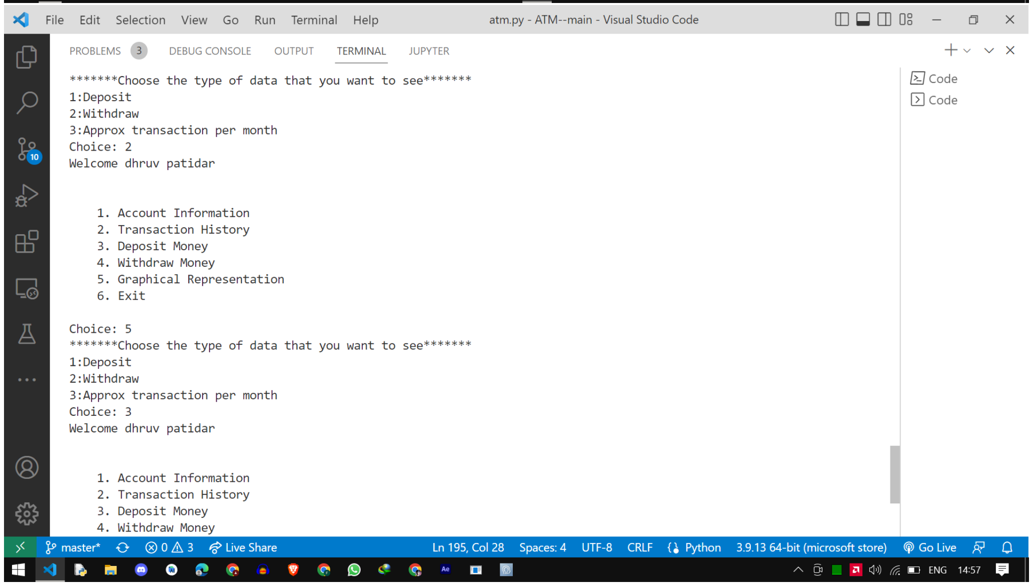


File Edit Selection View Go Run Terminal Help
atm.py - ATM--main - Visual Studio Code
PROBLEMS 3 DEBUG CONSOLE OUTPUT TERMINAL JUPYTER
Code
Code
master* 0 3 Live Share Ln 195, Col 28 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (microsoft store) Go Live
ENG 14:57

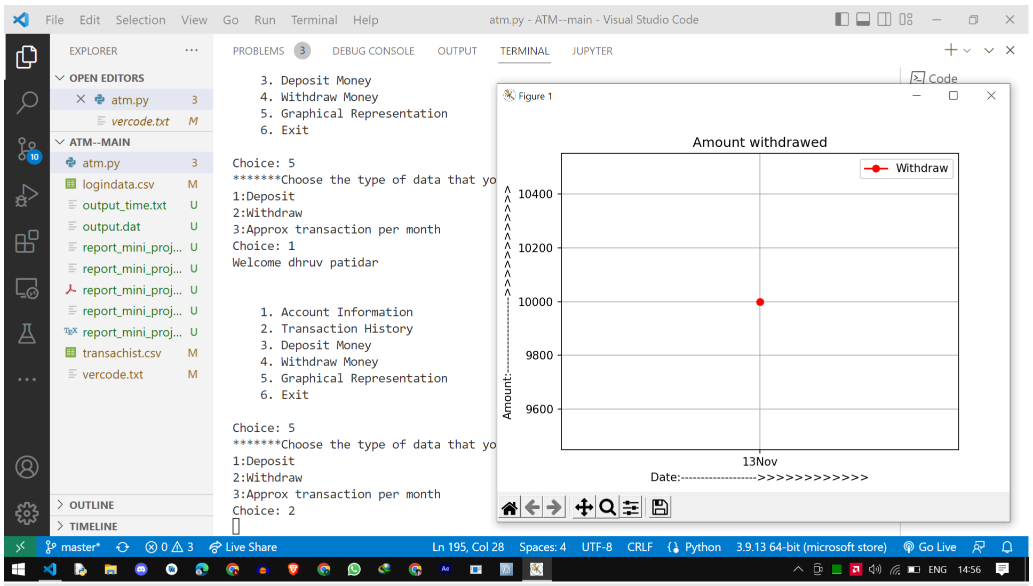
```
    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 5
*******Choose the type of data that you want to see*******
1:Deposit
2:Withdraw
3:Approx transaction per month
Choice: 1
Welcome dhruv patidar


    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 5
*******Choose the type of data that you want to see*******
1:Deposit
```



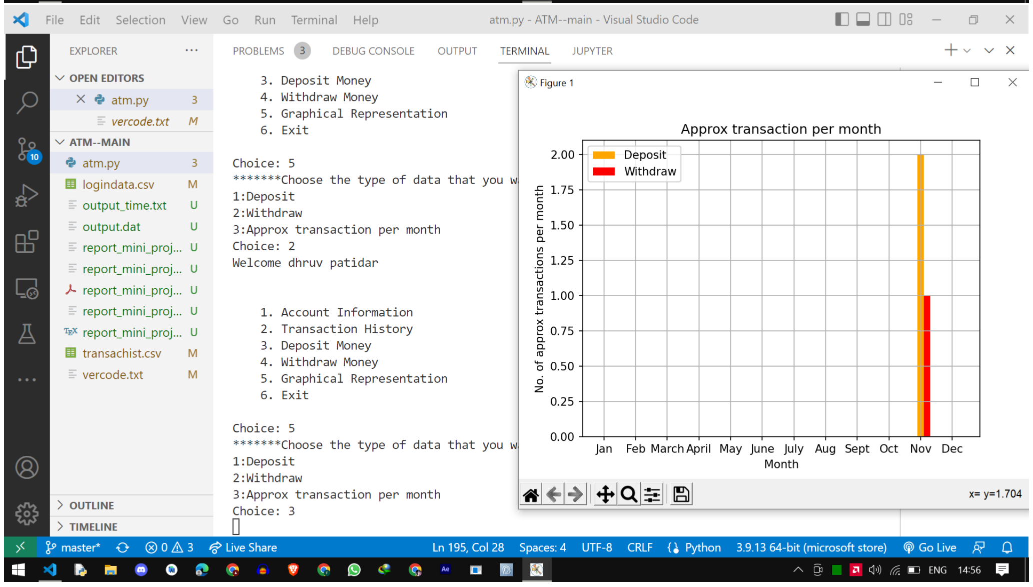
```
Choice: 2

Transaction History:---------

         usrnm        amt      stat
serno
36      dhruvpp    18000.0   Deposit
37      dhruvpp  1346567.0   Deposit
38      dhruvpp    10000.0  Withdraw

Welcome dhruv patidar


    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 5
*******Choose the type of data that yo
1:Deposit
2:Withdraw
3:Approx transaction per month
Choice: 1
```



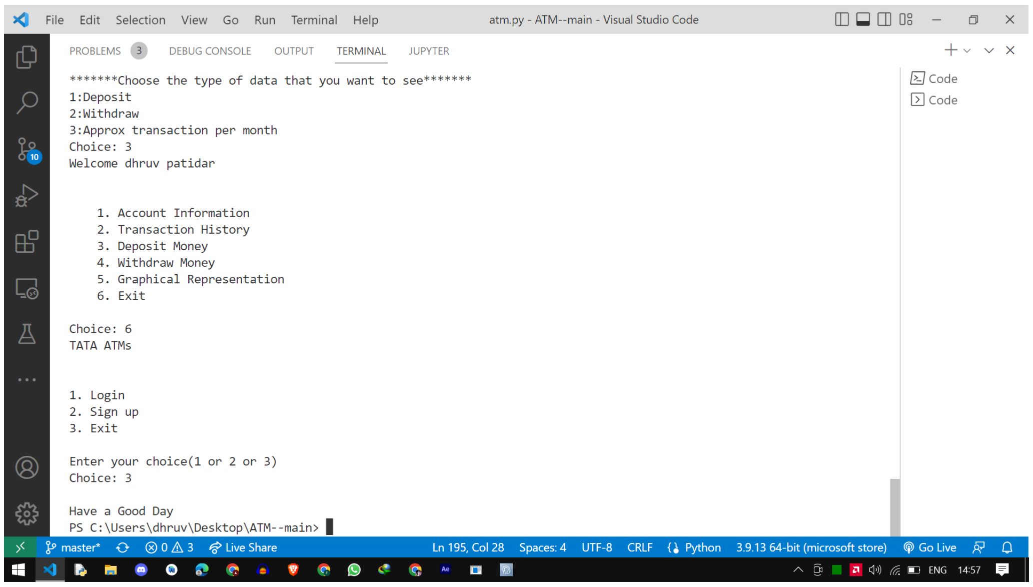
```
*******Choose the type of data that you want to see*******
1:Deposit
2:Withdraw
3:Approx transaction per month
Choice: 2
Welcome dhruv patidar


    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 5
*******Choose the type of data that you want to see*******
1:Deposit
2:Withdraw
3:Approx transaction per month
Choice: 3
Welcome dhruv patidar


    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
```


atm.py - ATM--main - Visual Studio Code
Figure 1
Amount withdrawed
Withdraw
10400
10200
10000
9800
9600
Amount:---------------->>>>>>>>
13Nov
Date:------------------->>>>>>>>>>>>
Ln 195, Col 28
Spaces: 4
UTF-8
CRLF
Python
3.9.13 64-bit (microsoft store)

```
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 5
*******Choose the type of data that yo
1:Deposit
2:Withdraw
3:Approx transaction per month
Choice: 1
Welcome dhruv patidar


    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 5
*******Choose the type of data that yo
1:Deposit
2:Withdraw
3:Approx transaction per month
Choice: 2
```



```
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 5
*******Choose the type of data that you w
1:Deposit
2:Withdraw
3:Approx transaction per month
Choice: 2
Welcome dhruv patidar


    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 5
*******Choose the type of data that you w
1:Deposit
2:Withdraw
3:Approx transaction per month
Choice: 3
```


File Edit Selection View Go Run Terminal Help
atm.py - ATM--main - Visual Studio Code
PROBLEMS 3 DEBUG CONSOLE OUTPUT TERMINAL JUPYTER
Code
Code
master* 0 3 Live Share Ln 195, Col 28 Spaces: 4 UTF-8 CRLF Python 3.9.13 64-bit (microsoft store) Go Live
ENG 14:57

```
*******Choose the type of data that you want to see*******
1:Deposit
2:Withdraw
3:Approx transaction per month
Choice: 3
Welcome dhruv patidar


    1. Account Information
    2. Transaction History
    3. Deposit Money
    4. Withdraw Money
    5. Graphical Representation
    6. Exit

Choice: 6
TATA ATMs


1. Login
2. Sign up
3. Exit

Enter your choice(1 or 2 or 3)
Choice: 3

Have a Good Day
PS C:\Users\dhruv\Desktop\ATM--main>
```