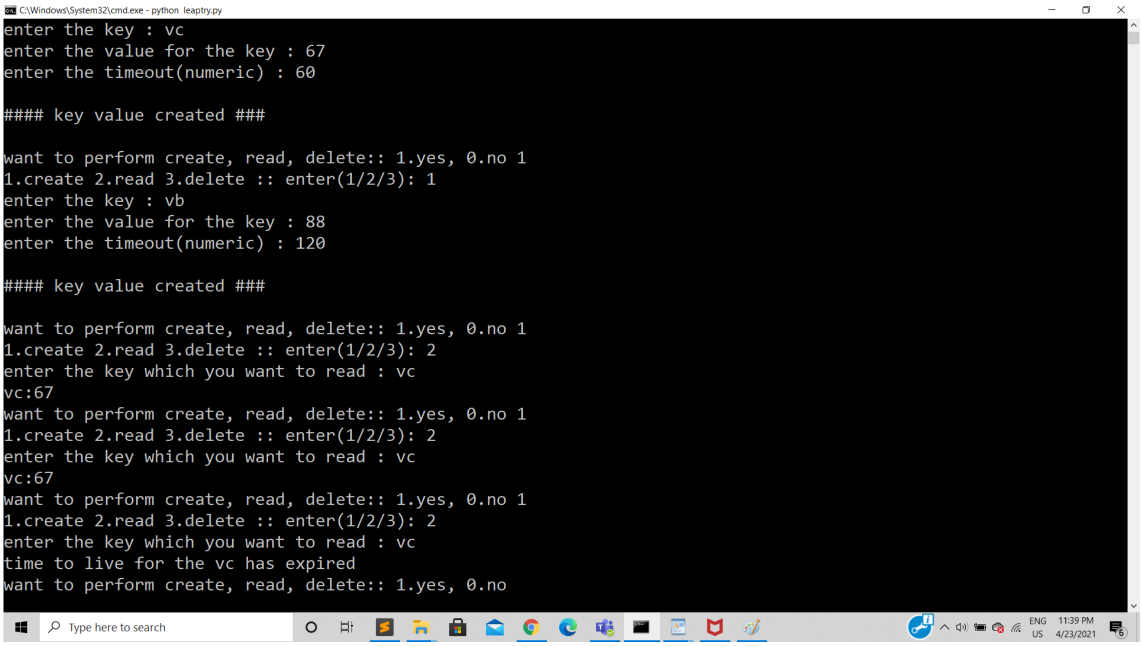
The unit test by leaptry.py file and python shell by leap.py file
And demonstration of thread safe and multi threading at the last

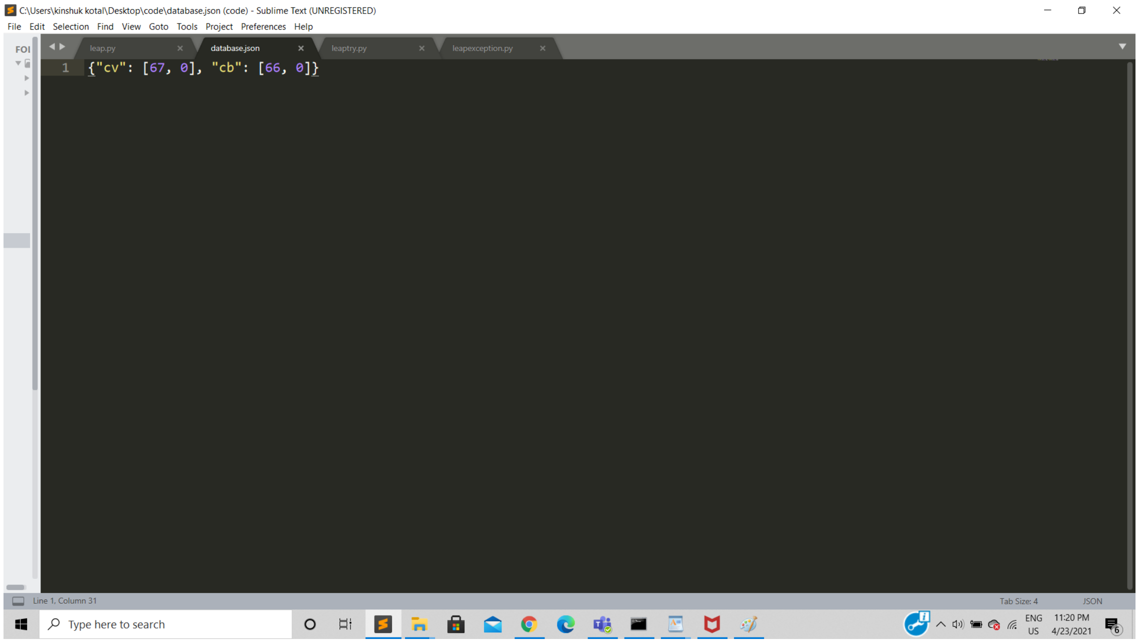
1. When we run the leaptry.py file then it asks to

a. enter the file path 1. Yes 0. That is default in my case database.json file
b. Then it asks whether to perform crd 1.yes 0.no
c. If then then it ask which operation to perform 1.create, 2,read, 3,delete
d. A snapshot of create and read operation



```
enter the key : vc
enter the value for the key : 67
enter the timeout(numeric) : 60

#### key value created ###

want to perform create, read, delete:: 1.yes, 0.no 1
1.create 2.read 3.delete :: enter(1/2/3): 1
enter the key : vb
enter the value for the key : 88
enter the timeout(numeric) : 120

#### key value created ###

want to perform create, read, delete:: 1.yes, 0.no 1
1.create 2.read 3.delete :: enter(1/2/3): 2
enter the key which you want to read : vc
vc:67
want to perform create, read, delete:: 1.yes, 0.no 1
1.create 2.read 3.delete :: enter(1/2/3): 2
enter the key which you want to read : vc
vc:67
want to perform create, read, delete:: 1.yes, 0.no 1
1.create 2.read 3.delete :: enter(1/2/3): 2
enter the key which you want to read : vc
time to live for the vc has expired
want to perform create, read, delete:: 1.yes, 0.no
```

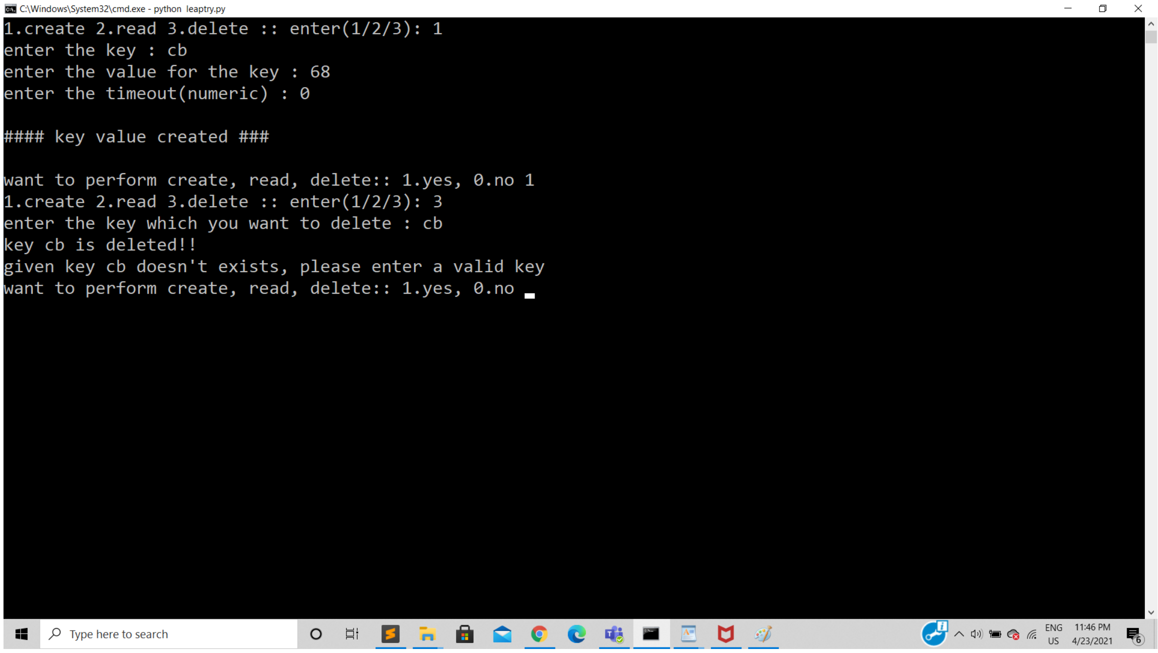
e. When we store first first two key value pair, the snap of database.json file



C:\Users\kinshuk kotal\Desktop\code\database.json (code) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
leap.py
database.json
leaptry.py
leapexception.py
Line 1, Column 31
Tab Size: 4
JSON
Type here to search
ENG US
11:20 PM
4/23/2021

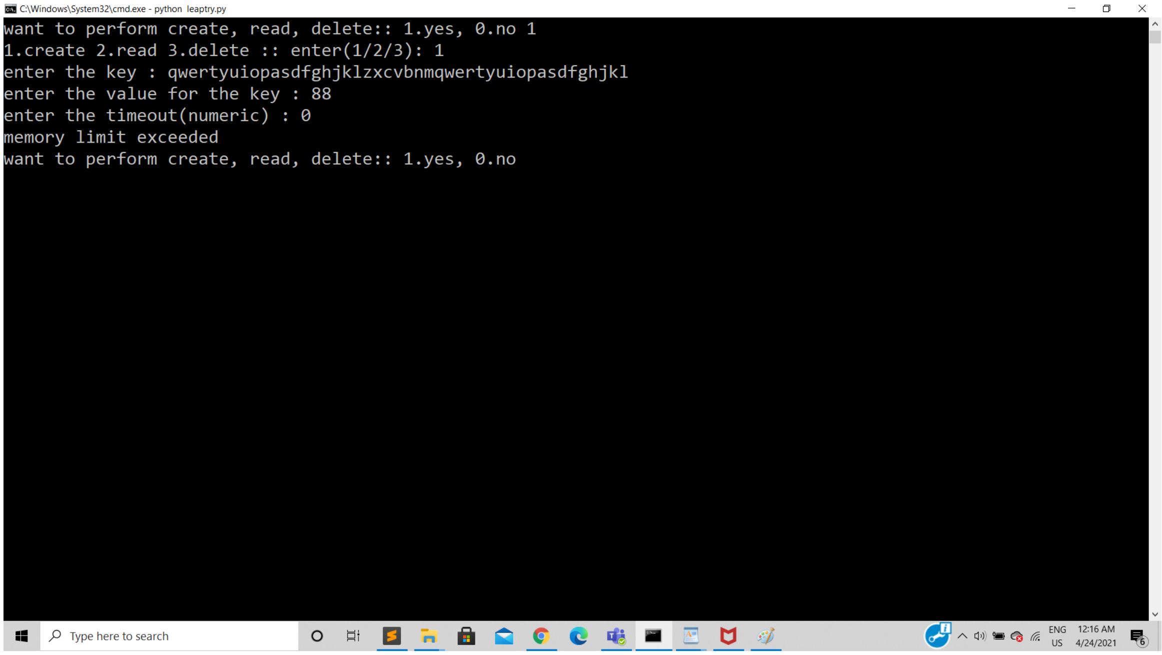
```
{"cv": [67, 0], "cb": [66, 0]}
```

f. When we use delete operation



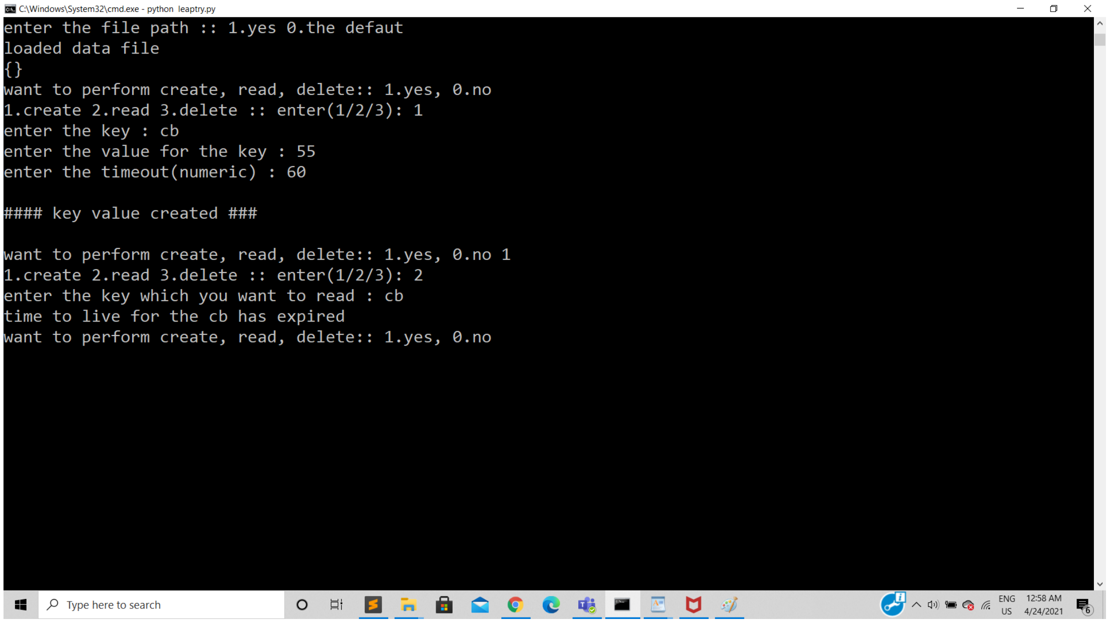
```
1.create 2.read 3.delete :: enter(1/2/3): 1
enter the key : cb
enter the value for the key : 68
enter the timeout(numeric) : 0

#### key value created ###

want to perform create, read, delete:: 1.yes, 0.no 1
1.create 2.read 3.delete :: enter(1/2/3): 3
enter the key which you want to delete : cb
key cb is deleted!!
given key cb doesn't exists, please enter a valid key
want to perform create, read, delete:: 1.yes, 0.no
```

g. Here we can have created a key whos limit exceeds 32 chars and it throws a message of memory limit exceeded



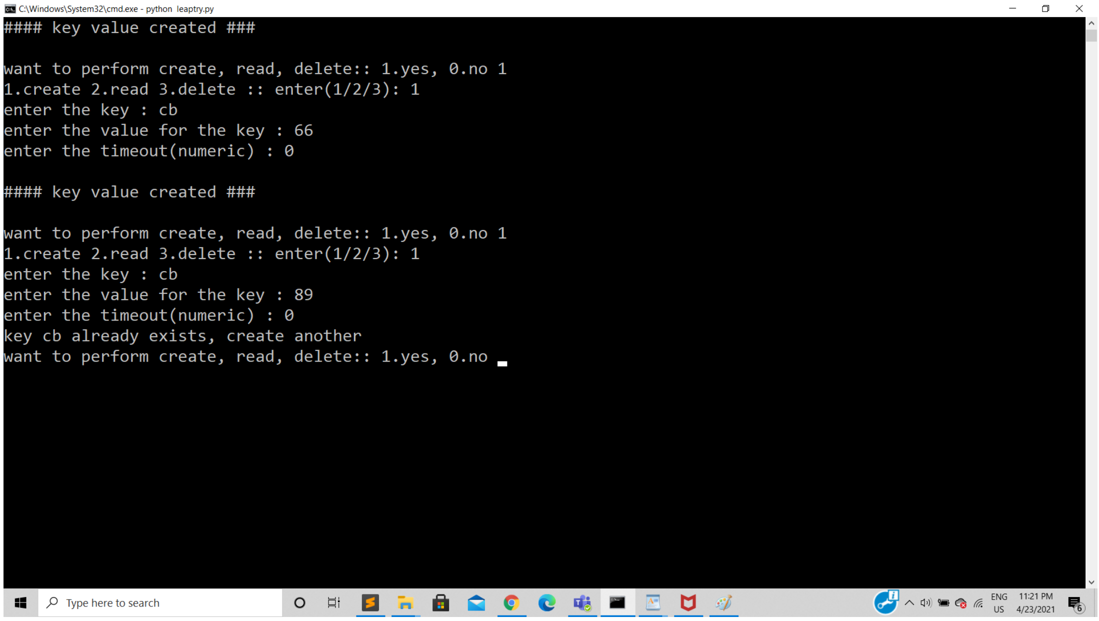
```
want to perform create, read, delete:: 1.yes, 0.no 1
1.create 2.read 3.delete :: enter(1/2/3): 1
enter the key : qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjkl
enter the value for the key : 88
enter the timeout(numeric) : 0
memory limit exceeded
want to perform create, read, delete:: 1.yes, 0.no
```

h. We then demonstrate the time to live property by creating a key of cb, value=55, timeout=60 sec and after 60 sec invoked read operation for the key cb then it showed the message “ time to live for the cb has expired “



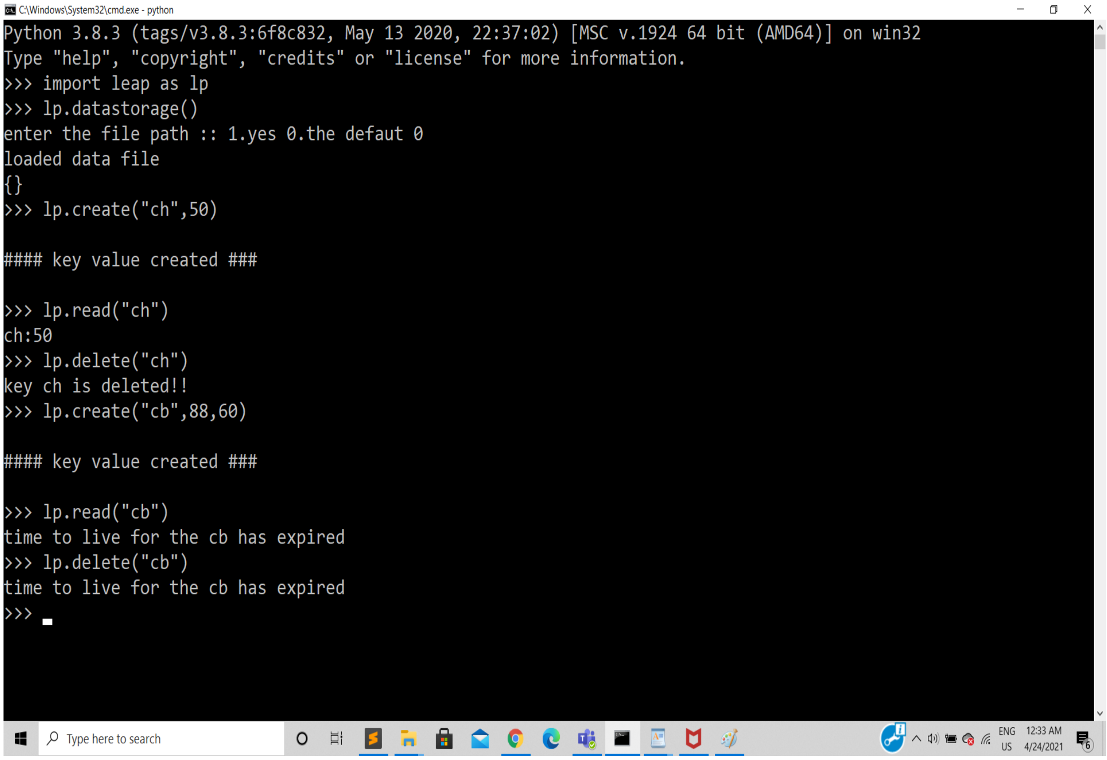
```
enter the file path :: 1.yes 0.the defaut
loaded data file
{}
want to perform create, read, delete:: 1.yes, 0.no
1.create 2.read 3.delete :: enter(1/2/3): 1
enter the key : cb
enter the value for the key : 55
enter the timeout(numeric) : 60

#### key value created ###

want to perform create, read, delete:: 1.yes, 0.no 1
1.create 2.read 3.delete :: enter(1/2/3): 2
enter the key which you want to read : cb
time to live for the cb has expired
want to perform create, read, delete:: 1.yes, 0.no
```

i. We then demonstrate is the key exists then also we try to invoke create operation



```
#### key value created ###

want to perform create, read, delete:: 1.yes, 0.no 1
1.create 2.read 3.delete :: enter(1/2/3): 1
enter the key : cb
enter the value for the key : 66
enter the timeout(numeric) : 0

#### key value created ###

want to perform create, read, delete:: 1.yes, 0.no 1
1.create 2.read 3.delete :: enter(1/2/3): 1
enter the key : cb
enter the value for the key : 89
enter the timeout(numeric) : 0
key cb already exists, create another
want to perform create, read, delete:: 1.yes, 0.no
```

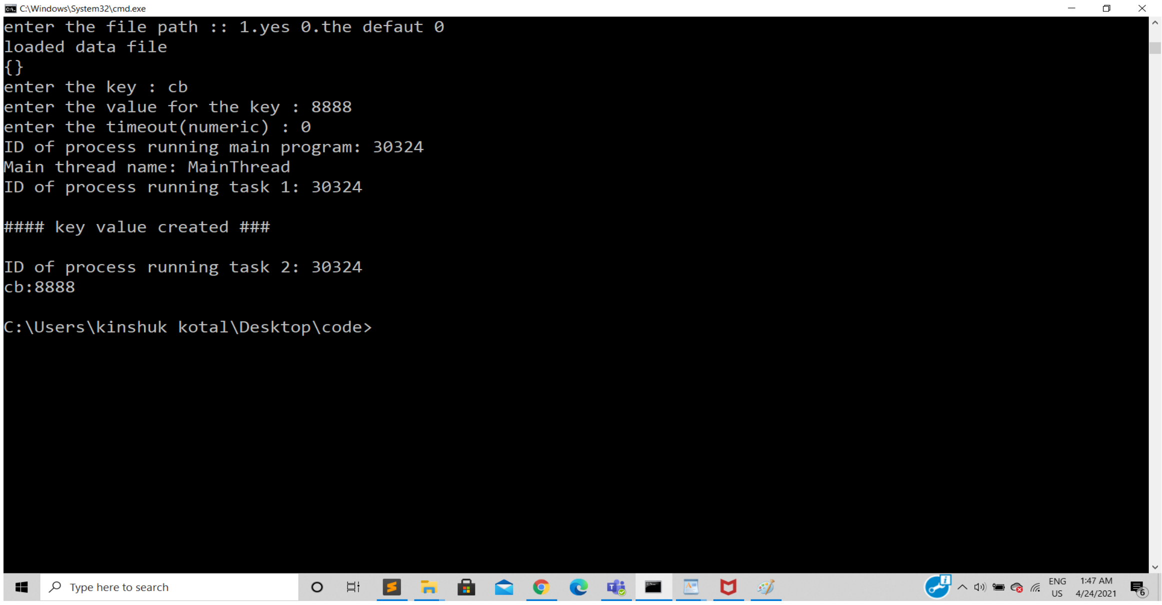
2. We have used python shell to demonstrate the similar test
   We have to run the lead.py file
   a. Run import leap as lp
   b. Then lp.datastorage() must have to run it will either create or load the data storage as per your need
   c. Then can perform the create,read,delete



```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:37:02) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import leap as lp
>>> lp.datastorage()
enter the file path :: 1.yes 0.the defaut 0
loaded data file
{}
>>> lp.create("ch",50)

#### key value created ###

>>> lp.read("ch")
ch:50
>>> lp.delete("ch")
key ch is deleted!!
>>> lp.create("cb",88,60)

#### key value created ###

>>> lp.read("cb")
time to live for the cb has expired
>>> lp.delete("cb")
time to live for the cb has expired
>>>
```

3. Multi threading and thread safe

   a. If we uncomment line number 64 and 99 of the leap.py file and run the leapt.py file We can see the the id of the process running ie in the example create and read are running are same which is 30324
   b. We have run multiple thread concurrently



```
enter the file path :: 1.yes 0.the defaut 0
loaded data file
{}
enter the key : cb
enter the value for the key : 8888
enter the timeout(numeric) : 0
ID of process running main program: 30324
Main thread name: MainThread
ID of process running task 1: 30324

#### key value created ###

ID of process running task 2: 30324
cb:8888

C:\Users\kinshuk kotal\Desktop\code>
```