Build a file-based key-value data store that supports the basic CRD (create, read, and delete) operations. This data store is meant to be used as a local storage for one single process on one laptop. The data store must be exposed as a library to clients that can instantiate a class and work with the data store.

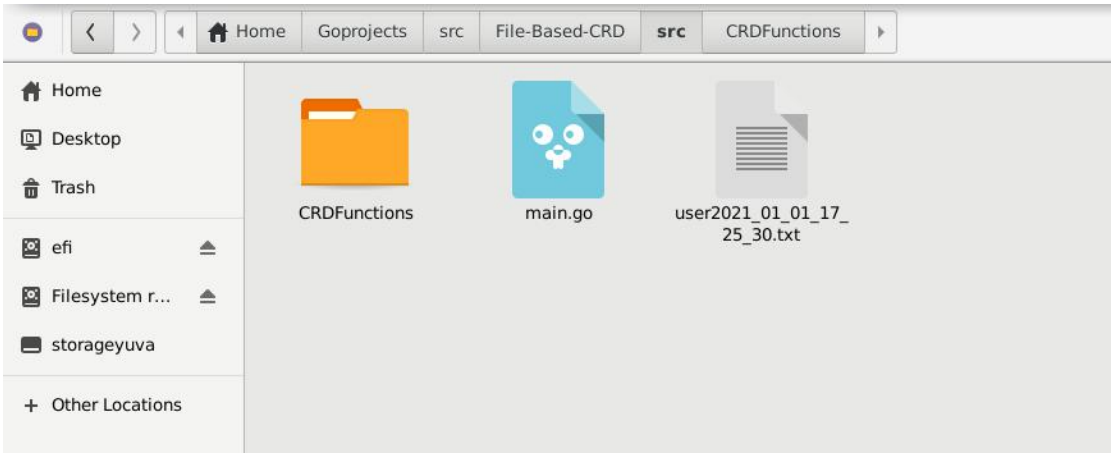
The data store will support the following functional requirements.

1. It can be initialized using an optional file path. If one is not provided, it will reliably" create itself in a reasonable location on the laptop.
2. A new key-value pair can be added to the data store using the Create operation. The key is always a string - capped at 32chars. The value is always a JSON object - capped at 16KB.
3. If Create is invoked for an existing key, an appropriate error must be returned."
4. A Read operation on a key can be performed by providing the key, and receiving the value in response, as a JSON object.
5. A Delete operation can be performed by providing the key.
6. Every key supports setting a Time-To-Live property when it is created. This property is optional. If provided, it will be evaluated as an integer defining the number of seconds the key must be retained in the data store. Once the Time-To-Live for a key has expired, the key will no longer be available for Read or Delete operations."
7. Appropriate error responses must always be returned to a client if it uses the data store inunexpected ways or breaches any limits.

The data store will also support the following non-functional requirements.
1. The size of the file storing data must never exceed 1GB.
2. More than one client process cannot be allowed to use the same file as a data store at any given time.
"3. A client process is allowed to access the data store using multiple threads, if it desires to. The data store must therefore be thread-safe."
"4. The client will bear as little memory costs as possible to use this data store, while deriving maximum performance with respect to response times for accessing the data" store.

1. CREATING A FILE

Whenever the program is initiated by the new user, new file will be generated for each user.



2 . CREATING KEY VALUE PAIRS

After creating a file, the user is allowed to enter key value pair seperated by comma.

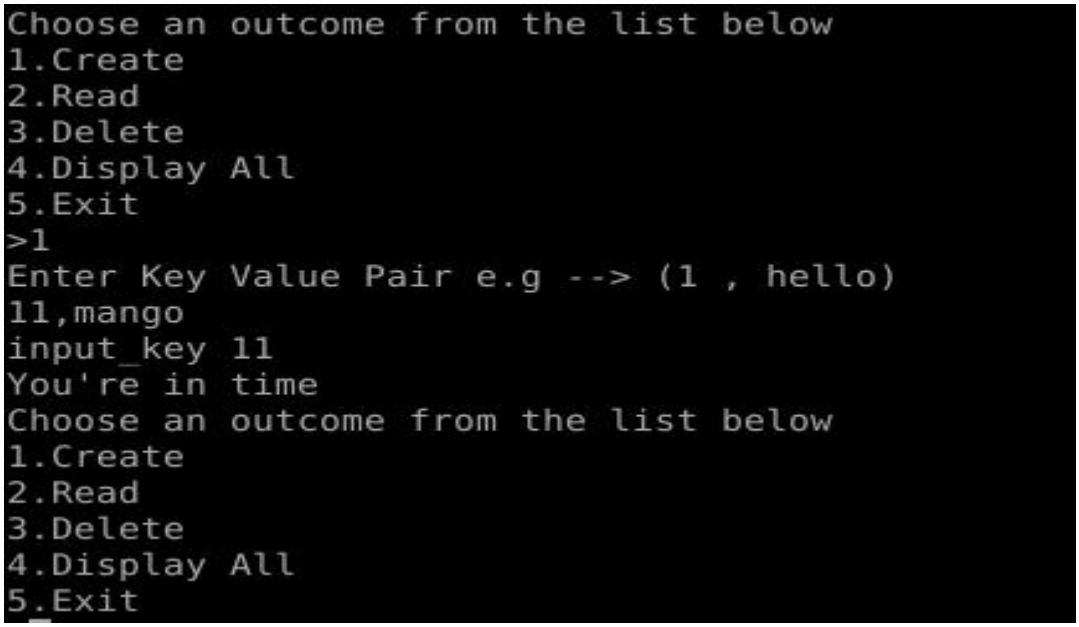
Now, the key value will be stored in the data storage(i.e file)

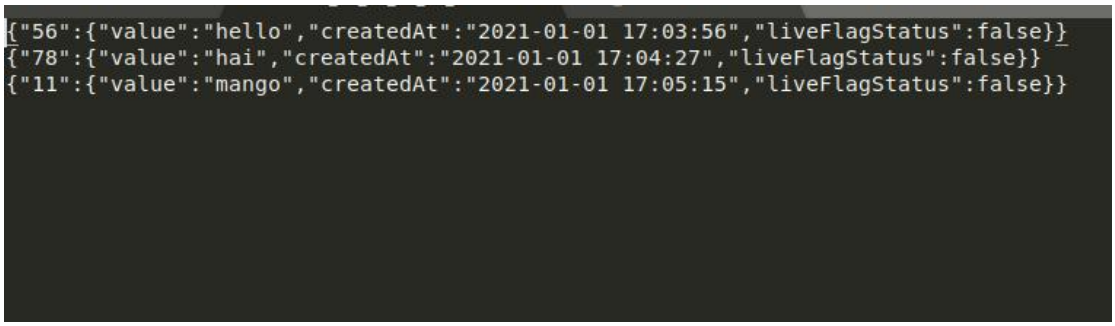
The current data storage file name is displayed above in the below picture

```
/root/Goprojects/src/File-Based-CRD/src/user2021_01_01_17_02_36.txt
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
>1
Enter Key Value Pair e.g --> (1 , hello)
56,hello
input_key 56
You're in time
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
>
```

```
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
>1
Enter Key Value Pair e.g --> (1 , hello)
78,hai
input_key 78
You're in time
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
>
```

The inserted key is stored as key value pair of JSON object is shown below,





Note::

Constrains

Each user is limited for certain time interval to continue the create operation.(eg 20000milliseconds in my case (20 sec))

Whenever each data is inserted in to file, we can see **liveFlagStatus** value is assigned initially as **false** state

So, In the above pic, we can see **liveFlagStatus** assigned to each data key is set as false.This means, the user has entered their key value with in time limit

So, In the terminal, we can see message **“You’re in time”** clearly mentioned the above constraints

Example for Expired TimetoLive Key

```
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
>1
Enter Key Value Pair e.g --> (1 , hello)
12,apple
input_key 12
Time Limit exceeded
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
```

main.go | user2021_01_01_17_02_36.txt | crd_functions.go

```
{"56":{"value":"hello","createdAt":"2021-01-01 17:03:56","liveFlagStatus":false}}
{"78":{"value":"hai","createdAt":"2021-01-01 17:04:27","liveFlagStatus":false}}
{"11":{"value":"mango","createdAt":"2021-01-01 17:05:15","liveFlagStatus":false}}
{"12":{"value":"apple","createdAt":"2021-01-01 17:05:51","liveFlagStatus":true}}
```

We could clearly see the status of **liveFlagStatus** is assigned with **true** and also in terminal message shows **"Time limit exceeded"** for the particular key

Once the key's **liveFlagStatus** is set as **true**, the **particular key cannot be allowed to read or delete.**

Example:

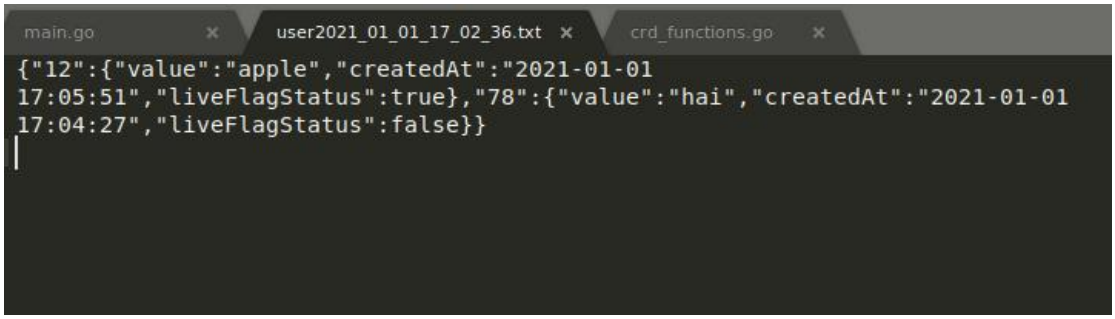
1. When trying to read the expired key

```
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
>2
Enter the Key to read json value:
12
You're not allowed to read the expired key 12
Key not available
```

2. When trying to delete the expired key

```
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
>3
Enter Key to delete: 12
You're not allowed to delete the expired key 12
Key not available
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
>
```

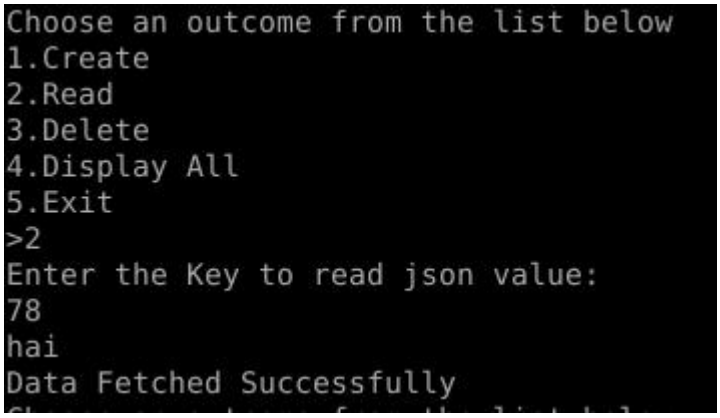
Note:: Even if the key is expired, the data is still available in data store(file) with **LiveFlagStatus** changed to **true**



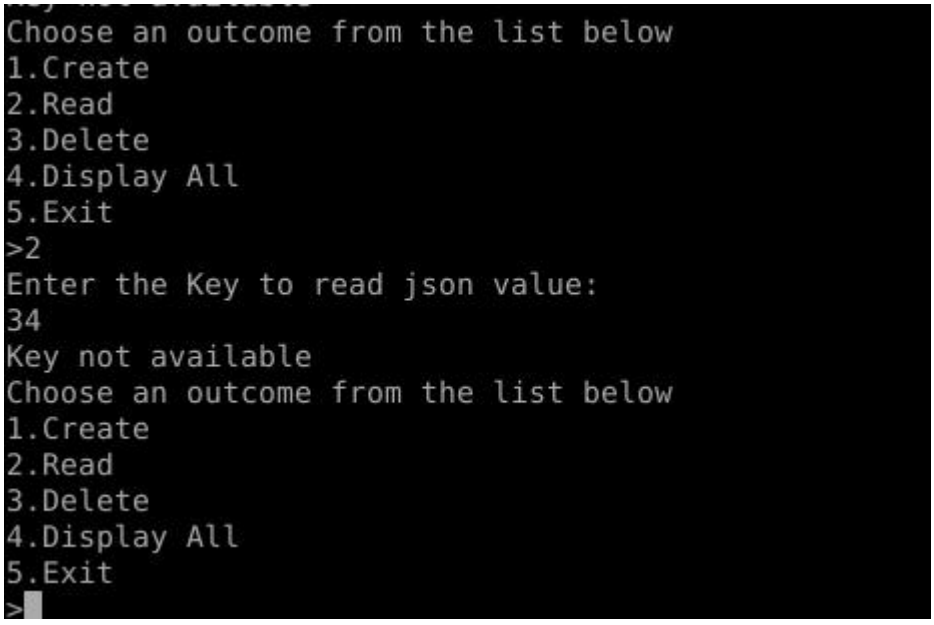
3. Read Value using Key

Fetch json Value when ever the user enter the key

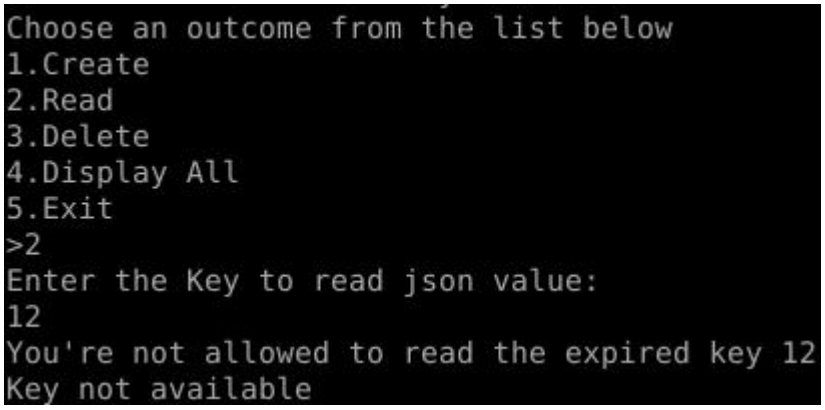
Case 1 : Key Availble, Value can be fetched



Case 2 : Key not Availble, Value cannot be fetched

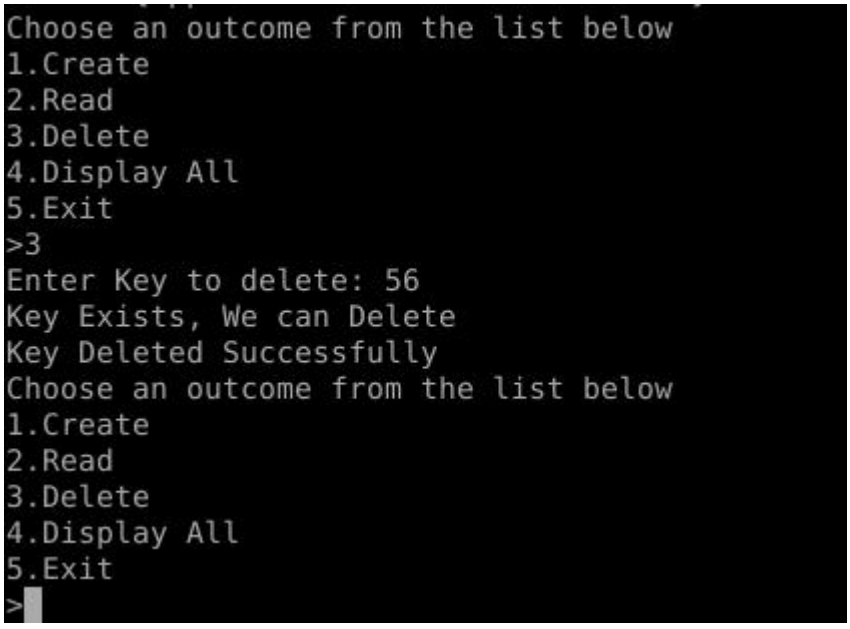


Case 3 : Time to live property for key is explained above



4. Delete the user data from data storage

Case 1 : Key Availble, user data can be deleted from data storage

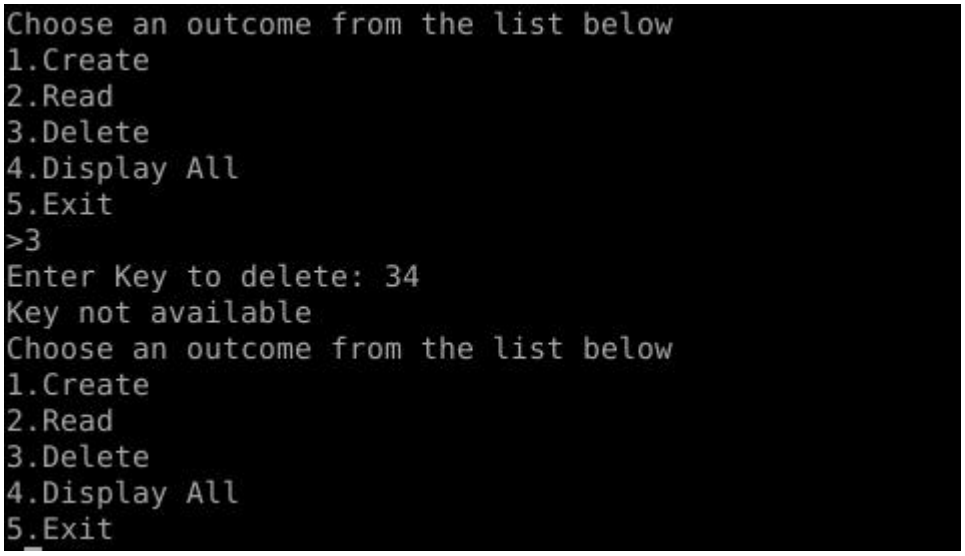


main.go | user2021_01_01_17_02_36.txt | crd_functions.go

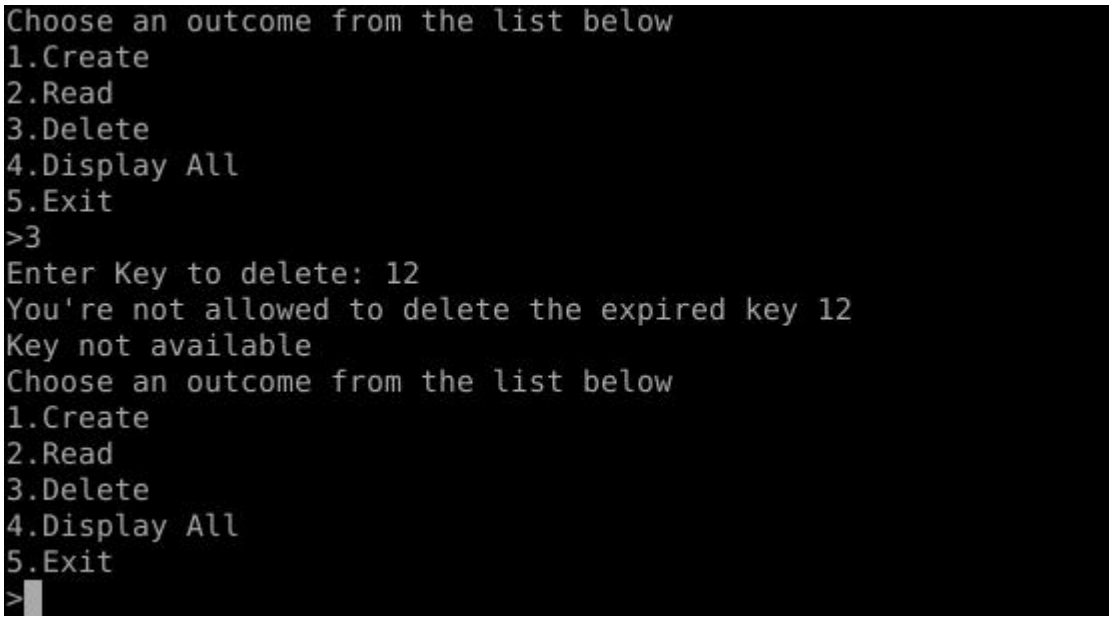
```
{"11":{"value":"mango","createdAt":"2021-01-01
17:05:15","liveFlagStatus":false},"12":{"value":"apple","createdAt":"2021-01-01
17:05:51","liveFlagStatus":true},"78":{"value":"hai","createdAt":"2021-01-01
17:04:27","liveFlagStatus":false}}
```

```
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
>4
78 --> {hai 2021-01-01 17:04:27 false}
11 --> {mango 2021-01-01 17:05:15 false}
12 --> {apple 2021-01-01 17:05:51 true}
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
>3
Enter Key to delete: 11
Key Exists, We can Delete
Key Deleted Successfully
Choose an outcome from the list below
1.Create
2.Read
3.Delete
4.Display All
5.Exit
```

Case 2 : Key not availble, no user data cannot be deleted

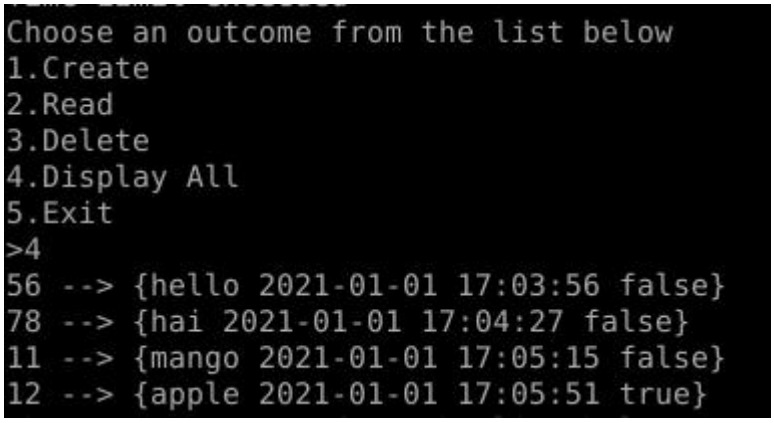


Case 3 : Time to live property for key is explained above.

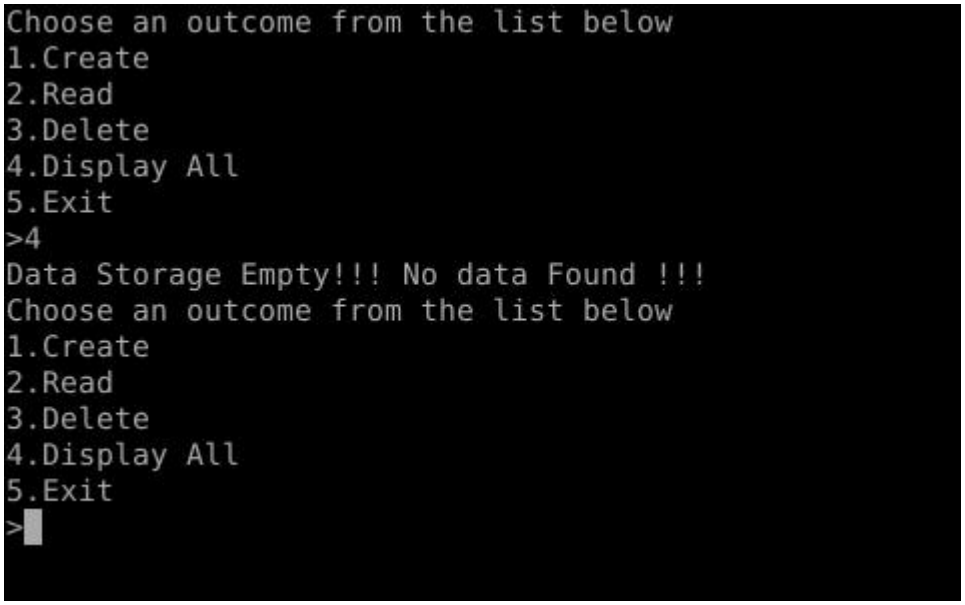


5. Displaying the key values

Case 1 : If data is available, the function will fetch key & values



Case 2 : If data is not available, the function will return empty



6. Exit

Press 5 to exit the process