**Objective:** 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.

**Implementation:**

* The code is written in JAVA.
* Its an executable jar.
* User needs to import the jar (kv.jar) in order to invoke its function.
* Once a user imports the jar into his/her code they can call the function by creating an object of the class **Code.**

**Eg. Code obj = new Code();**

* User can use this object (obj) to call create(), read() and deleteByKey() function to perform the desired operation.
* When an user calls the create function it can provide JSONObject or JSONArray as its parameter to create a new key-value pair in the file that is saved in **C** drive with name **my\_db.txt**

**NOTE:** An user needs to pass **\_id** as one of the key as it would act as a **primary key** for validation and other purpose in the code.

**Eg.** JSONArray jrr = **new** JSONArray();

JSONObject jobj = **new** JSONObject();

jobj.put("\_id", "383838");

jobj.put("name","Rahul");

jrr.put(jobj);

obj.create(jobj);

* When user wants to use read function to check the value of any particular key from the file they can call read function by passing key of JSONObject ( \_id with the object to find the particular key) as the argument to get the value associated with it.

**Eg.** JSONObject jj= new JSONObject();

jj.put("name", "Rahul");

jj.put("\_id", "383838");

read(jj);

* when user wants to delete a particular key-value from the file they can call deleteByKey function by passing key JSONObject ( \_id with the object to find the particular key).

**Eg.** JSONObject jj= **new** JSONObject();

jj.put("name", "Tanu");

jj.put("\_id", "383838");

*deleteByKey*(jj);