```
import sqlite3
from sqlite3 import Error

"""Import This Class
    Create Object and access the data.

    By Vikas Patel @ DTech
    www.villageprogrammer.tech
"""

class AuthenticationDatabase:
    """
    Authentication stuff
    Please do not try to access private methods.
    Only access  if  you are aware what you are doing.
    """

    def __init__(self, dbfile="Databases/config.db"):
        """__init__(Object,dbfile)"""
        self.con = sqlite3.connect(dbfile)
        print("connection:",self.con)

        try:
            self.con = sqlite3.connect(dbfile)
            self.cur = self.con.cursor()
        except Error as e:
            print("Error: " + str(e))
    def __create_table(self):
        q = 'create table Authentication(key text, value text)'
        self.__sql_table(q)

    def __sql_table(self, query):
        self.cur = self.con.cursor()
        self.cur.execute(query)
        self.con.commit()
        self.cur.close()

    """ access the data from outside """

    def authentication_data(self):
        """ access the data from outside """

        query = '''select key,value from Authentication'''
        self.cur = self.con.cursor()
        self.result = self.cur.execute(query).fetchall()
        # self.cur.close()
        return self.__to_dict(self.result)

    def __to_dict(self, result):
        data = dict()
        for row in result:
            # print(row[0], row[1]) # debugging
            data[row[0]] = row[1]
        return data

    def set_authentication(self, key, val):
        q = f'insert into Authentication values("{key}","{val}")'
        self.__sql_table(q)
        # self.cur.close()

    def __remove_data(self):
        q = 'truncate table Authentication'
        self.__sql_table(q)

    def __remove_table(self):
        q = 'drop table Authentication'
        self.__sql_table(q)



class SettingsDatabase:
    def __init__(self, dbfile="Databases/settings.db"):
        """__init__(Object,dbfile)"""
        # self.con = sqlite3.connect("settings.db")
        self.con = sqlite3.connect("settings.db")

        try:
            self.con = sqlite3.connect(dbfile)
            self.cur = self.con.cursor()
        except Error as e:
            print("Error: " + str(e))
    def create_table(self):
        q = 'create table Email(key text, value text)'
        self.__sql_table(q)

    def __sql_table(self, query):
        self.cur = self.con.cursor()
        self.cur.execute(query)
        self.con.commit()
        self.cur.close()
    def sql_table(self, query):
        self.cur = self.con.cursor()
        self.cur.execute(query)
        self.con.commit()
        self.cur.close()


    """ access the data from outside """

    def authentication_data(self):
        """ access the data from outside """
        query = '''select key,value from Email'''
        self.cur = self.con.cursor()
        self.result = self.cur.execute(query).fetchall()
        # self.cur.close()
        return self.__to_dict(self.result)

    def __to_dict(self, result):
        data = dict()
        for row in result:
            # print(row[0], row[1]) # debugging
            data[row[0]] = row[1]
        return data

    def get_mobile(self):
        self.cur = self.con.cursor()
        mob = self.cur.execute('select value from Alert where key="mobile"'
        ).fetchall()
        return mob[0][0]
    def get_email(self):
        self.cur = self.con.cursor()
        em = self.cur.execute('select value from Alert where key="email"').
        fetchall()
        return em[0][0]
    def get_message(self):
        self.cur = self.con.cursor()
        msg = self.cur.execute('select value from Alert where
        key="message"').fetchall()
        return msg[0][0]
    def get_password(self):
        # admin password
        self.cur = self.con.cursor()
        pswd = self.cur.execute('select value from Settings where
        key="adminpass"').fetchall()
        return pswd[0][0]

    def get_sensitivity(self):
        sens = self.cur.execute('select value from Settings where
        key="sensitivity"').fetchall()
        return sens[0][0]

    def getAll(self):
        data = [self.get_mobile(),self.get_email(),self.get_message(),\
                    self.get_sensitivity(),self.get_password()]
        return data

    def set_mobile(self,v):
        q = f'update Alert set value={v} where key="mobile"'
        self.sql_table(q)

    def set_email(self,v):
        q = "update Alert set value="+"\'"+v+"\'"+"where key='email'"
        self.sql_table(q)

    def set_message(self,v):
        v = v.replace("\' "," ")
        q = "update Alert set value=\' "+v+"\' where key='message'"
        self.sql_table(q)

    def set_sensitivity(self,v):
        q = f'update Settings set value={str(v)} where key="sensitivity"'
        self.sql_table(q)






# db = SettingsDatabase()
# db.set_sensitivity("130")
# print(db.getAll())





# q = """create table Authentication(key text, value text)"""
```

```
######### Default Dadb = AuthenticationDatabase()

# db.sql_table(q)
# print(db.authentication_data())
# db.sql_table('insert into Authentication values("Vikas","Patel")')
# db.set_authentication("Vikas","Patel") # insert new authentication data
# db._AuthenticationDatabase__sql_table(q) ## illegal access

# db._AuthenticationDatabase__remove_data()

# db.con.close()ta (Belongs to Vikas Patel) ##########
# q = """insert into Authentication
values('authorization','eRrsSCiZTxW0bHd6wI51J8nqOhpPyX4g7LMAlEuKQ9Bkmf3VtN70
d8aFIDYUQyXAgBucftmnkvKC46PO') ,
# ('Content-type','application/x-www-form-urlencoded'),
# ('Cache-control','no-cache')"""

# q = '''select key,value from Authentication'''


########## DUBUGGING STUFF ###############
# ****************************************

#

# *****************************************
```