Title : Database connectivity - MongoDB :

Problem statement : write a program to implement Mongo DB database connectivity with any front end language to implement Database navigation operations (add, delete, update etc).

Objectives : 1] Understand the concept of connectivity between Python and databases.
2] Understand how Python can invoke CRUD operation.

Outcomes : Student will be able :
1] connect mongodb database with Python.
2] Perform CRUD operations using Python.

Software and Hardware Requirements : 1] Mongodb shell installed with latest version
2] Python 3.8. or greater
3] Laptop | PC with latest OS installed.

References : http://doc.mongodb.org/manual

Concept Related Theory:

MongoDB:

- MongoDB is an open-source document oriented database and a leading NoSQL database.
- MongoDB offers high speed, high availability and high scalibility.
- Main features of mongdb includes ad-hoc queries, indexing, Replication, load balancing, Aggregation, File storage etc. which makes it unique from other noSQL databases.

Connectivity:

Python has a native library for MongoDB. The name of the available library is "PyMongo". To import this execute command:

```
from pymongo import MongoClient
```

- Create a connection:

we can create a client using MongoClient object.

```
client = MongoClient()
```

In order to connect with host and port, we can execute command as:

```
client = MongoClient('localhost', 27017)
```

- Accessing dabase Objects:
To create a database or switch to an existing database we use:

```
mydatabase = client['db_name']
or
mydatabase = client.db_name
```

- Accessing the collection: we can access the collection of database using:

```
mycollection = mydatabase['mycolName']
```

MongoDB store the database in the form of dictionaries.

Insertion operation:
Insert operation adds new document to a collection. we can use following methods to insert to collection in pymongo:

```
record = mycollection.insert_one(record)
```
or for multiple records
```
record = mycollection.insert_many([records])
```

Read operation:
The method like find() is used to get more than one document as result of query.

```
for i in mydb.mytable.find({title:"mongoDB"})
    print(i)
```

Update operation:

The methods like update-one() and update-many() helps to update documents(s) in collection.

Example:

```
result = collection.update_many(
            {"id" : 25},
            {
               "$set" : {"name": "Abhinav",
                        },
              "$currDate" : {"lastModified": True}
            }
          )
print (result).
```

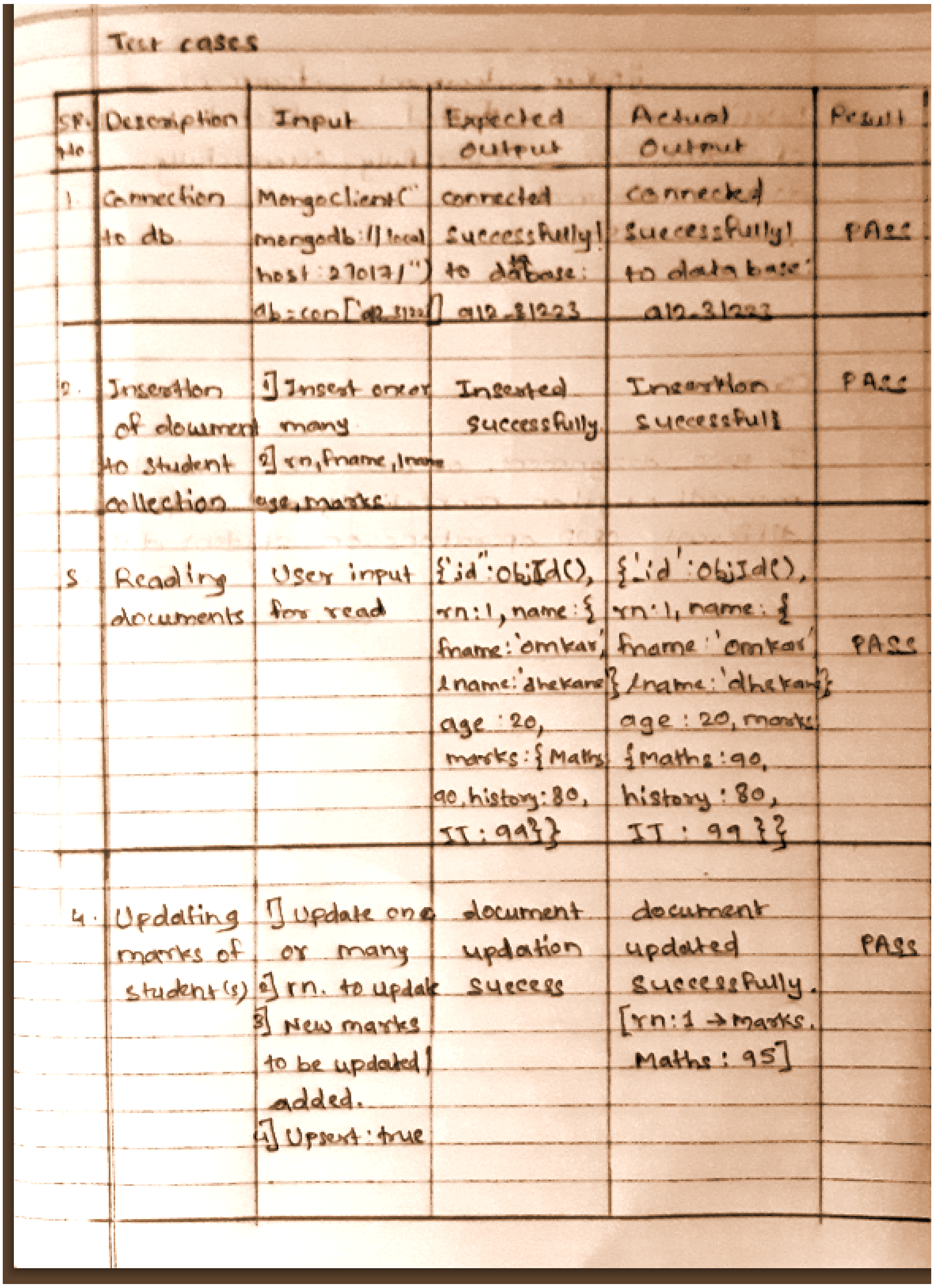
Deletion operation:

To delete documents of collection in a database we can use delete_one() or delete_many() methods on collection object.

```
result = my_collection.delete_many({"name":
                                   "Abhinav"})
```

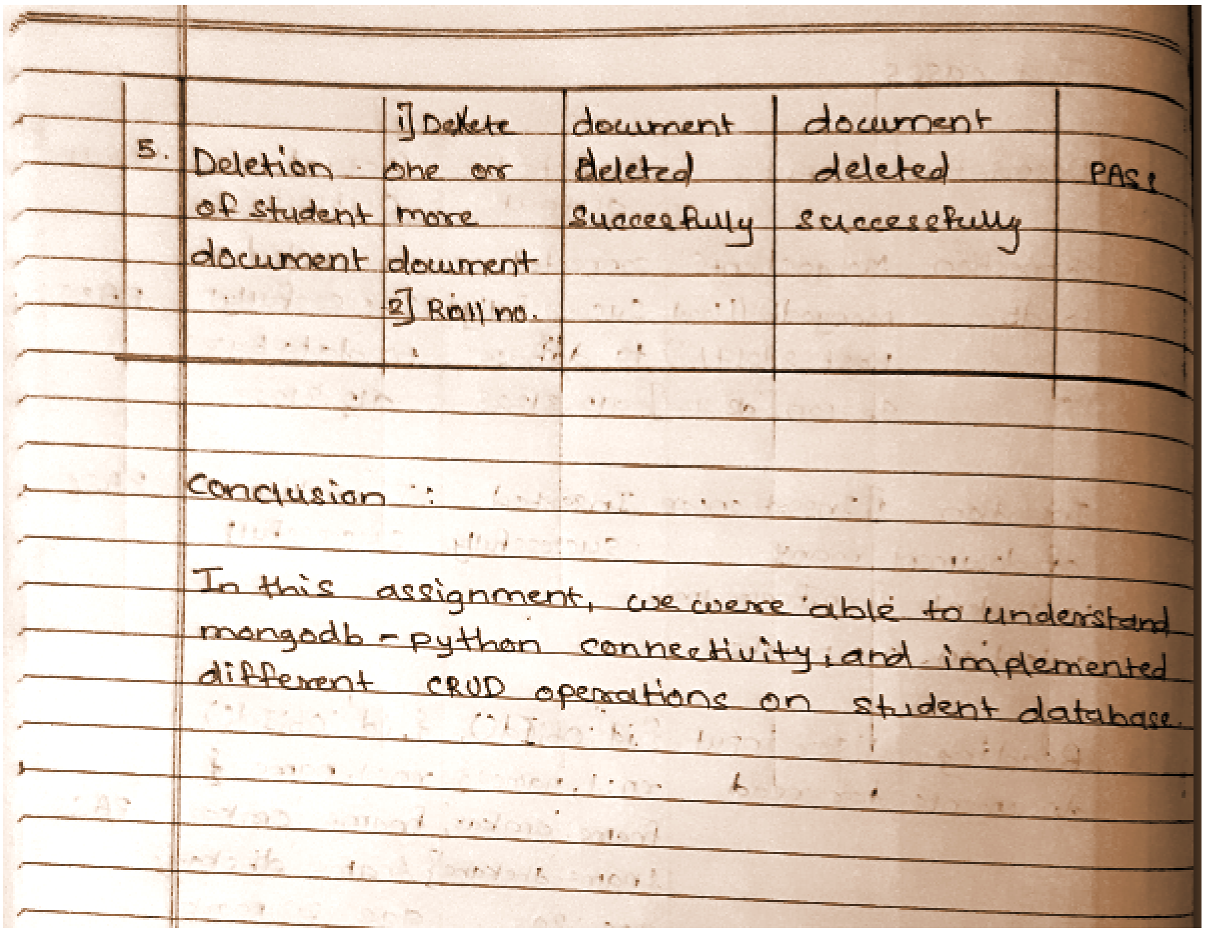
To see no. of documents deleted.

```
print(result.deleted_count)
```



Test cases

Sr. No.	Description	Input	Expected Output	Actual Output	Result
1.	Connection to db.	MongoClient(" mongodb://local host:27017/") db=con['a12_31223']	connected successfully! to database: a12_31223	connected successfully! to database: a12_31223	PASS
2.	Insertion of document to student collection	1] Insert one or many 2] rn, fname, lname age, marks	Inserted successfully.	Insertion successfull	PASS
3	Reading documents	User input for read	{'id':ObjId(), rn:1, name:{ fname:'omkar', lname:'dhekane'} age:20, marks:{Maths: 90, history:80, IT:99}}	{'_id':ObjId(), rn:1, name:{ fname:'omkar', lname:'dhekane'}, age:20, marks: {maths:90, history:80, IT:99}}	PASS
4.	Updating marks of student(s)	1] update one or many 2] rn. to update 3] New marks to be updated / added. 4] Upsert: true	document updation success	document updated successfully. [rn:1 → marks. Maths: 95]	PASS



5.	Deletion of student document	1] Delete one or more document 2] Roll no.	document deleted Successfully	document deleted successfully	PASS

Conclusion :

In this assignment, we were able to understand mongodb - python connectivity, and implemented different CRUD operations on student database.