

Name: Shubham Rajendra Chemate

Roll Number: 31118

Subject: DBMS Lab – Assignment 4 (Group B) – Database Connectivity with Frontend Language

**Code:**

```
import com.mongodb.client.FindIterable;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.model.Filters;
import com.mongodb.client.model.Updates;
import java.util.Iterator;
import org.bson.Document;
import com.mongodb.MongoClient;
import java.util.concurrent.ThreadLocalRandom;

public class MongoDBConnectivity {

    @SuppressWarnings({ "unused", "resource" })
    public static void main(String args[]) throws Exception {

        System.out.println("Establishing Connection to the database...\n");

        MongoClient mongo = new MongoClient("localhost", 27017);
        MongoDatabase database = mongo.getDatabase("b04_db");

        System.out.println("Conncection Established");

        MongoCollection<Document> collection =
database.getCollection("students");

        String[] nameOfStudents = { "Tony", "Steve", "Stephen", "Loki" };
        String[] subjects = { "OOP", "DSA", "DBMS", "CN", "OS" };

        // insertion
        {
            int roll_num = 1;
            for (int i = 0; i < nameOfStudents.length; i++)
                for (int j = 0; j < subjects.length; j++) {

                    int marks =
ThreadLocalRandom.current().nextInt(70, 100 + 1);

                    Document doc = new Document("_id",
roll_num).append("name", nameOfStudents[i])
                        .append("subject",
subjects[j]).append("marks", marks);

                    collection.insertOne(doc);

                    roll_num++;
                }

            System.out.println(roll_num-1 + " records are successfully
inserted.\n");
```

```

    }

//    read
    {
        System.out.println("Reading records from collection:");
        FindIterable<Document> iterDoc = collection.find();
        Iterator<Document> it = iterDoc.iterator();
        if (it.hasNext())
            while (it.hasNext())
                System.out.println(it.next());
        else
            System.out.println("No records found");
        System.out.println("Record reading done.\n");
    }

//    find details of student with given roll number
    {
        int roll_num = 8;
        System.out.println("Getting details of student having roll
number " + roll_num);
        FindIterable<Document> iterDoc = collection.find(new
Document("_id", roll_num));
        Iterator<Document> it = iterDoc.iterator();
        if (it.hasNext()) {
            while (it.hasNext()) {
                System.out.println(it.next());
            }
        } else {
            System.out.println("Student record not found.");
        }

        System.out.println("Reading done.\n");
    }

//    update
    {
        int roll_num = 2;
        System.out.println("Updating marks of student having roll
number " + roll_num);
        collection.updateOne(Filters.eq("_id", roll_num),
Updates.set("marks", 100));
        FindIterable<Document> iterDoc = collection.find(new
Document("_id", roll_num));
        Iterator<Document> it = iterDoc.iterator();
        if (it.hasNext()) {
            System.out.println("Details of student after update
operation:");
            while (it.hasNext()) {
                System.out.println(it.next());
            }
        } else {
            System.out.println("Student record not found.");
        }

        System.out.println("Updata done.\n");
    }

//    delete
    {

```

```

        int roll_num = 2;
        System.out.println("Deleting record of student having roll
number " + roll_num);
        FindIterable<Document> iterDoc = collection.find(new
Document("_id", roll_num));
        Iterator<Document> it = iterDoc.iterator();
        if (it.hasNext()) {
            collection.deleteOne(Filters.eq("_id", roll_num));
            System.out.println("Student Record Deleted.");
        } else {
            System.out.println("No such record is found.");
        }

        System.out.println("Delete operation done.\n");
    }
}

```

### Ouptut:

```

<terminated> MongoDBConnectivity [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\java
Establishing Connection to the database...

Nov 28, 2021 6:30:55 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SIN
Conncection Established
Nov 28, 2021 6:30:55 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms befo
Nov 28, 2021 6:30:55 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, globalValue:1}] to
NOV 28, 2021 6:37:56 PM com.mongodb.diagnostics
INFO: Opened connection [connectionId{localValue:2, globalValue:2}] to
20 records are successfully inserted.

```

Reading records from collection:

```
Document({_id=1, name=Tony, subject=OOP, marks=76})
Document({_id=2, name=Tony, subject=DSA, marks=97})
Document({_id=3, name=Tony, subject=DBMS, marks=93})
Document({_id=4, name=Tony, subject=CN, marks=88})
Document({_id=5, name=Tony, subject=OS, marks=99})
Document({_id=6, name=Steve, subject=OOP, marks=86})
Document({_id=7, name=Steve, subject=DSA, marks=96})
Document({_id=8, name=Steve, subject=DBMS, marks=77})
Document({_id=9, name=Steve, subject=CN, marks=80})
Document({_id=10, name=Steve, subject=OS, marks=97})
Document({_id=11, name=Stephen, subject=OOP, marks=91})
Document({_id=12, name=Stephen, subject=DSA, marks=79})
Document({_id=13, name=Stephen, subject=DBMS, marks=93})
Document({_id=14, name=Stephen, subject=CN, marks=86})
Document({_id=15, name=Stephen, subject=OS, marks=93})
Document({_id=16, name=Loki, subject=OOP, marks=85})
Document({_id=17, name=Loki, subject=DSA, marks=88})
Document({_id=18, name=Loki, subject=DBMS, marks=75})
Document({_id=19, name=Loki, subject=CN, marks=95})
Document({_id=20, name=Loki, subject=OS, marks=96})
Record reading done.
```

Getting details of student having roll number 8

```
Document({_id=8, name=Steve, subject=DBMS, marks=77})
```

Updating marks of student having roll number 2

Details of student after update operation:

```
Document({_id=2, name=Tony, subject=DSA, marks=100})
```

Update done.

Deleting record of student having roll number 2

Student Record Deleted.

Delete operation done.