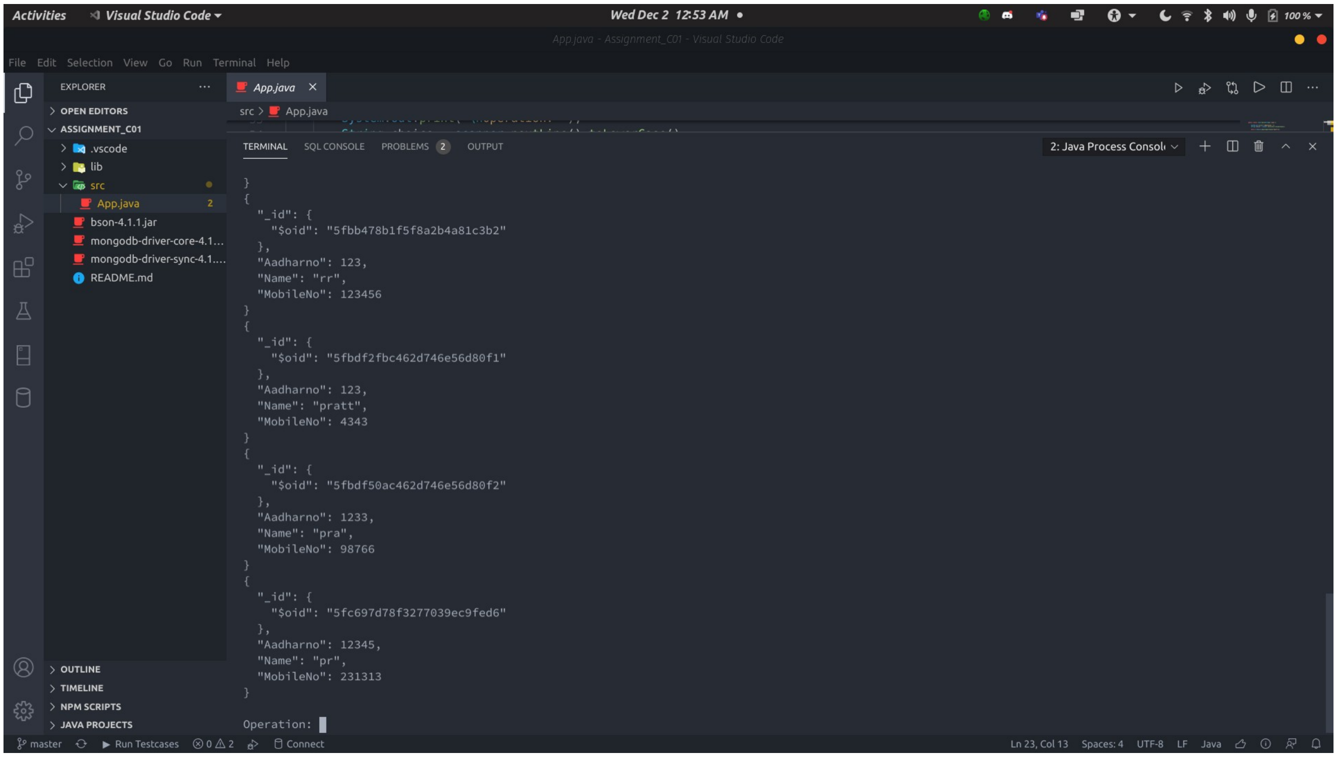
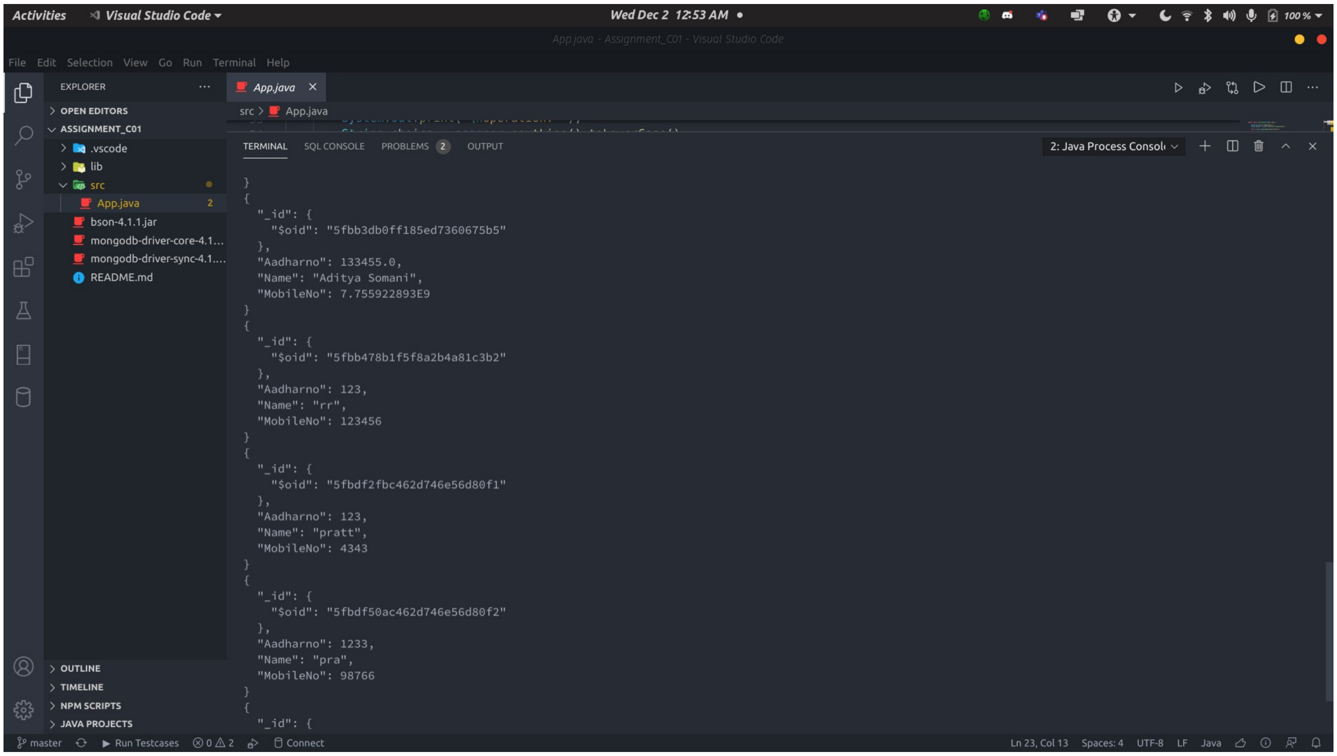
```java
import com.mongodb.client.*;
import com.mongodb.client.model.Updates;
import com.mongodb.client.result.DeleteResult;
import com.mongodb.client.result.InsertOneResult;
import com.mongodb.client.result.UpdateResult;
import org.bson.Document;
import org.bson.json.JsonWriterSettings;

import java.util.Arrays;
import java.util.Scanner;

import static com.mongodb.client.model.Filters.and;
import static com.mongodb.client.model.Filters.eq;

public class App {
    private static final MongoClient mongoClient;
    private static final MongoDatabase db;
    private static final MongoCollection<Document> collection;
    private static final JsonWriterSettings settings;
    private static final Scanner scanner;


    static {
        mongoClient = MongoClients.create("mongodb://localhost:27017");
        db = mongoClient.getDatabase("C01");
        collection = db.getCollection("Aadhar");
        settings = JsonWriterSettings.builder().indent(true).build();
        scanner = new Scanner(System.in);
    }

    public static void main(String[] args) {

        System.out.print("\nOperation: ");
        String choice = scanner.nextLine().toLowerCase();

        while (!choice.equalsIgnoreCase("exit")){

            switch (choice){
                case "insert":
                    insert();
                break;
                case "update":
                    update();
                break;
                case "delete":
                    delete();
                break;
                case "find":
                    find();
            break;
            default:
                System.out.println("Invalid Action");
        }

        System.out.print("\nOperation: ");
        choice = scanner.nextLine().toLowerCase();

    }
}

private static void insert(){
    String n;
    int a,m;
    System.out.print("Aadhar no.: ");
    a = Integer.parseInt(scanner.nextLine());
    System.out.print("Name: ");
    n = scanner.nextLine();
    System.out.print("mobno: ");
    m = Integer.parseInt(scanner.nextLine());

    Document article = new Document();
    article.append("Aadharno",a);
    article.append("Name",n);
    article.append("MobileNo",m);

    InsertOneResult result = collection.insertOne(article);
    System.out.println("result: "+result);

}

private static void update(){
    System.out.print("Aadharno: ");
    int a = Integer.parseInt(scanner.nextLine());

    System.out.print("Name: ");
    String n = scanner.nextLine();

    System.out.print("MobileNo: ");
    int m = Integer.parseInt(scanner.nextLine());

    UpdateResult result = collection.updateOne(
            and(eq("Aadharno",a),eq("Name",n)),
            Updates.set("MobileNo",m)
    );
    System.out.println("result: "+result);

}
```

```
    private static void delete(){
        System.out.print("Aadharno: ");
        int a = Integer.parseInt(scanner.nextLine());

        System.out.print("Name: ");
        String n = scanner.nextLine();

        DeleteResult result = collection.deleteMany(
                and(eq("Aadharno",a),eq("Name",a))
        );
        System.out.println("result: "+result);
    }

    private static void find(){
        for (Document document : collection.find()) {
            System.out.println(document.toJson(settings));
        }
    }
}
```

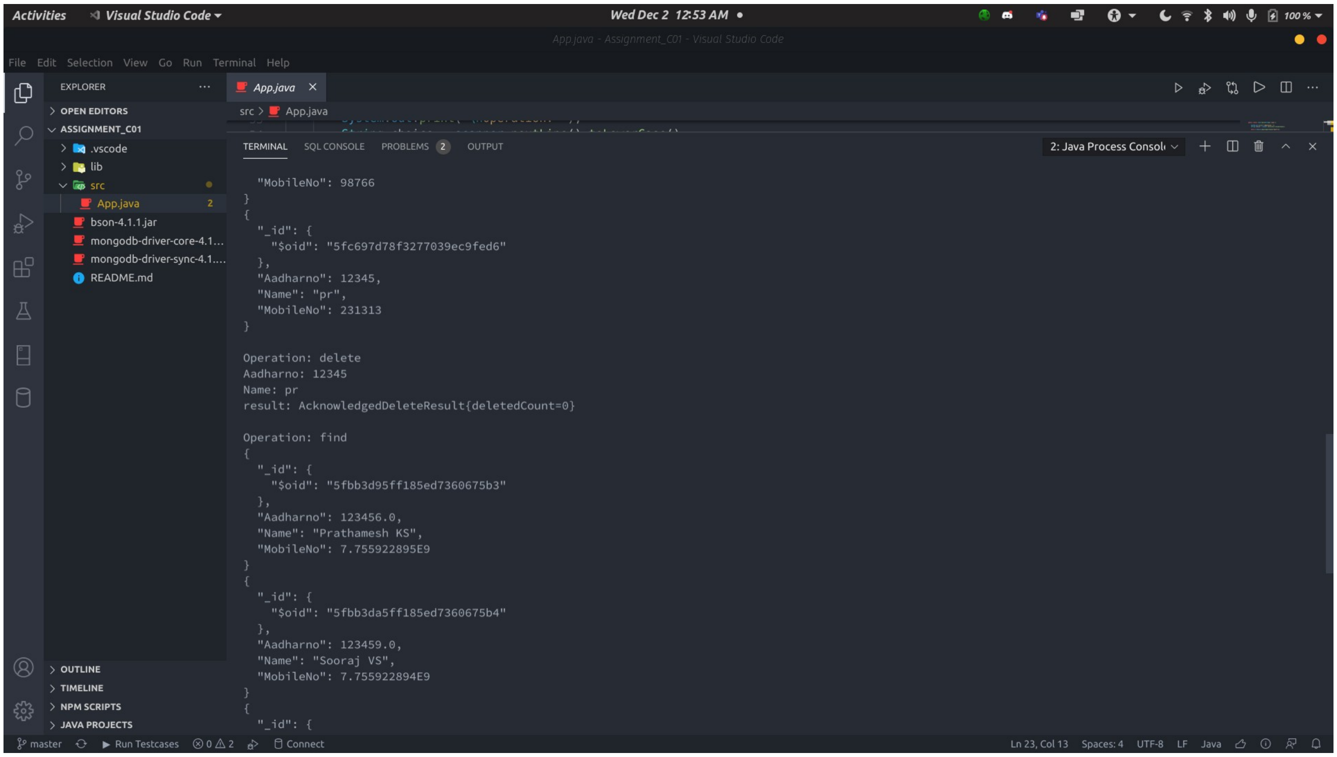
OUTPUT



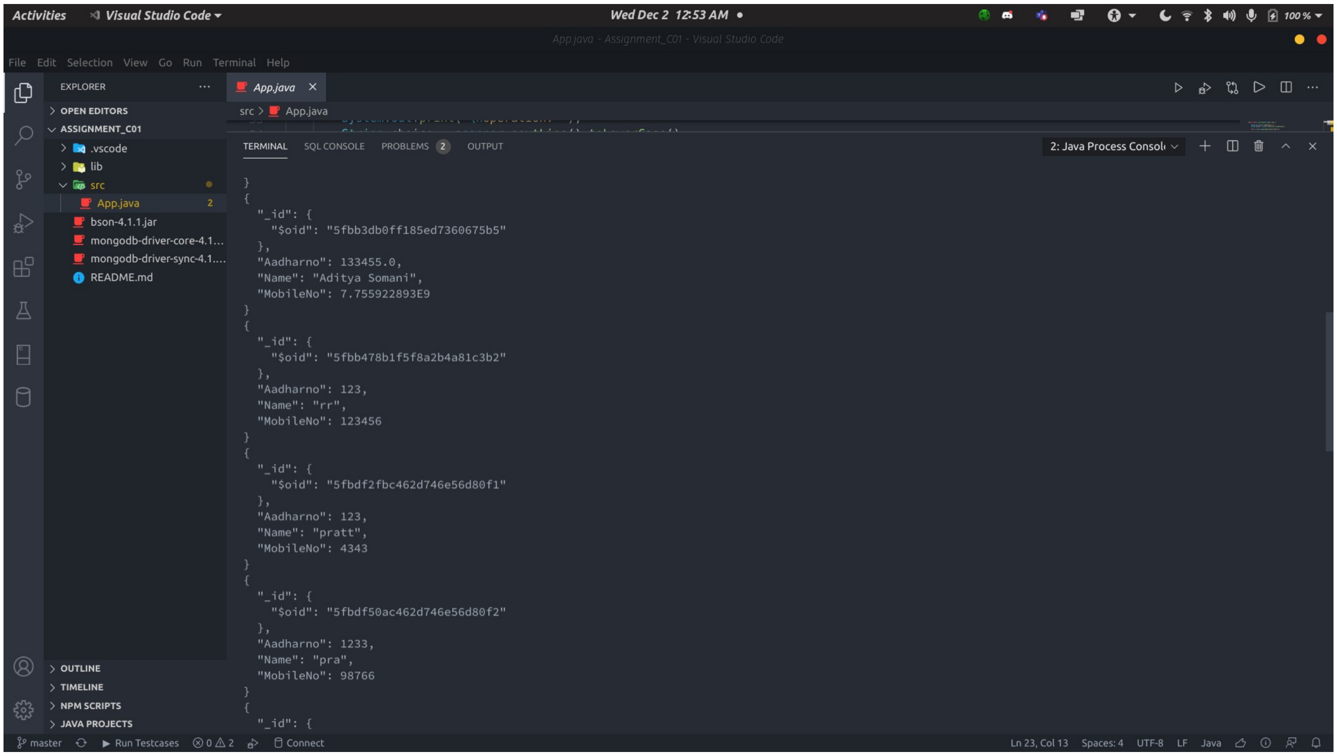
```
}
{
  "_id": {
    "$oid": "5fbb478b1f5f8a2b4a81c3b2"
  },
  "Aadharno": 123,
  "Name": "rr",
  "MobileNo": 123456
}
{
  "_id": {
    "$oid": "5fbdf2fbc462d746e56d80f1"
  },
  "Aadharno": 123,
  "Name": "pratt",
  "MobileNo": 4343
}
{
  "_id": {
    "$oid": "5fbdf50ac462d746e56d80f2"
  },
  "Aadharno": 1233,
  "Name": "pra",
  "MobileNo": 98766
}
{
  "_id": {
    "$oid": "5fc697d78f3277039ec9fed6"
  },
  "Aadharno": 12345,
  "Name": "pr",
  "MobileNo": 231313
}

Operation: 
```



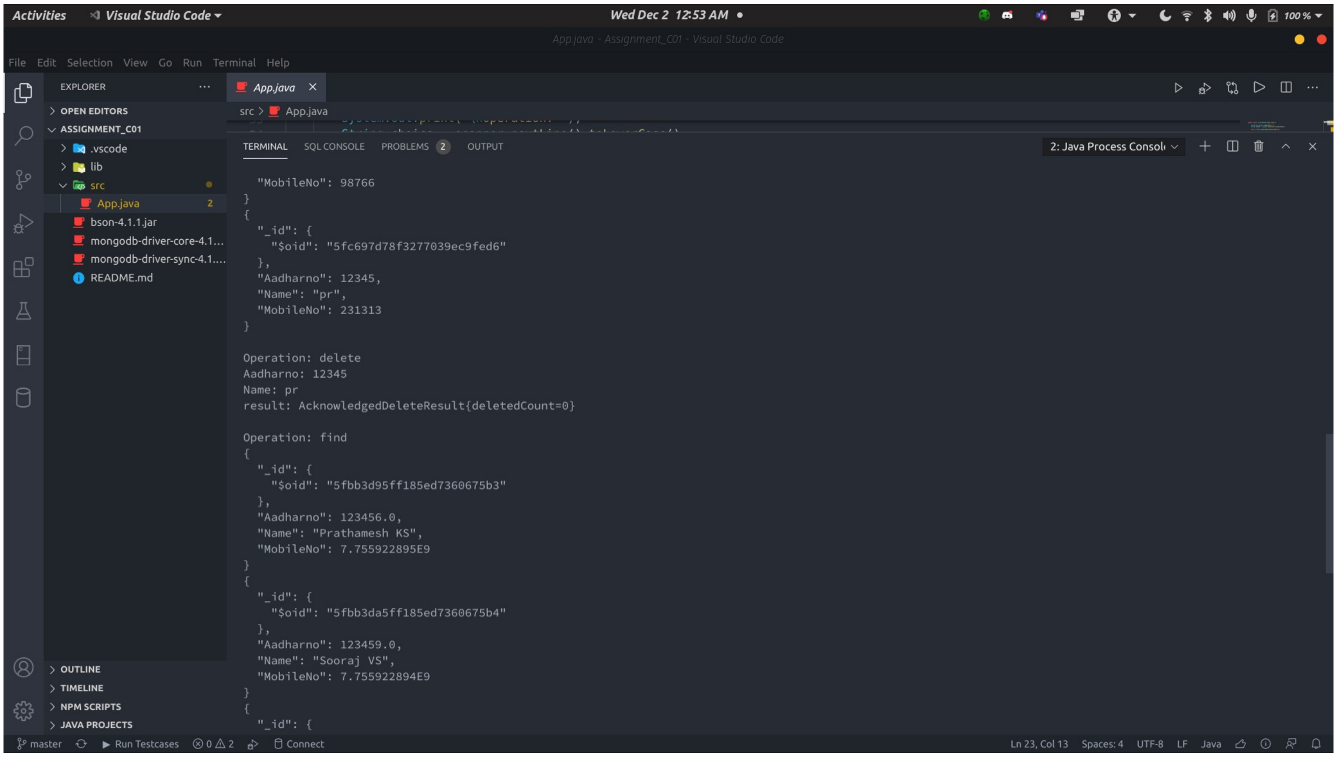
```
}
{
  "_id": {
    "$oid": "5fbb3db0ff185ed7360675b5"
  },
  "Aadharno": 133455.0,
  "Name": "Aditya Somani",
  "MobileNo": 7.755922893E9
}
{
  "_id": {
    "$oid": "5fbb478b1f5f8a2b4a81c3b2"
  },
  "Aadharno": 123,
  "Name": "rr",
  "MobileNo": 123456
}
{
  "_id": {
    "$oid": "5fbdf2fbc462d746e56d80f1"
  },
  "Aadharno": 123,
  "Name": "pratt",
  "MobileNo": 4343
}
{
  "_id": {
    "$oid": "5fbdf50ac462d746e56d80f2"
  },
  "Aadharno": 1233,
  "Name": "pra",
  "MobileNo": 98766
}
{
  "_id": {
```



```
  "MobileNo": 98766
}
{
  "_id": {
    "$oid": "5fc697d78f3277039ec9fed6"
  },
  "Aadharno": 12345,
  "Name": "pr",
  "MobileNo": 231313
}

Operation: delete
Aadharno: 12345
Name: pr
result: AcknowledgedDeleteResult{deletedCount=0}

Operation: find
{
  "_id": {
    "$oid": "5fbb3d95ff185ed7360675b3"
  },
  "Aadharno": 123456.0,
  "Name": "Prathamesh KS",
  "MobileNo": 7.755922895E9
}
{
  "_id": {
    "$oid": "5fbb3da5ff185ed7360675b4"
  },
  "Aadharno": 123459.0,
  "Name": "Sooraj VS",
  "MobileNo": 7.755922894E9
}
{
  "_id": {
```



```
}
{
  "_id": {
    "$oid": "5fbb3db0ff185ed7360675b5"
  },
  "Aadharno": 133455.0,
  "Name": "Aditya Somani",
  "MobileNo": 7.755922893E9
}
{
  "_id": {
    "$oid": "5fbb478b1f5f8a2b4a81c3b2"
  },
  "Aadharno": 123,
  "Name": "rr",
  "MobileNo": 123456
}
{
  "_id": {
    "$oid": "5fbdf2fbc462d746e56d80f1"
  },
  "Aadharno": 123,
  "Name": "pratt",
  "MobileNo": 4343
}
{
  "_id": {
    "$oid": "5fbdf50ac462d746e56d80f2"
  },
  "Aadharno": 1233,
  "Name": "pra",
  "MobileNo": 98766
}
{
  "_id": {
```



```
  "MobileNo": 98766
}
{
  "_id": {
    "$oid": "5fc697d78f3277039ec9fed6"
  },
  "Aadharno": 12345,
  "Name": "pr",
  "MobileNo": 231313
}

Operation: delete
Aadharno: 12345
Name: pr
result: AcknowledgedDeleteResult{deletedCount=0}

Operation: find
{
  "_id": {
    "$oid": "5fbb3d95ff185ed7360675b3"
  },
  "Aadharno": 123456.0,
  "Name": "Prathamesh KS",
  "MobileNo": 7.755922895E9
}
{
  "_id": {
    "$oid": "5fbb3da5ff185ed7360675b4"
  },
  "Aadharno": 123459.0,
  "Name": "Sooraj VS",
  "MobileNo": 7.755922894E9
}
{
  "_id": {
```

Assignment C1

- Date of completion: 23/11/20

- Date of Submission: 27/11/20

- Title: MongoDB- Java connectivity

- Problem Statement:
Write a program to implement MongoDB database connectivity with PHP/Python/Java. Implement database navigation operations ('add, delete, edit, etc) using ODBC/JDBC

- Outcomes:
  - Implement database connectivity for MongoDB.
  - Program CURD operations on database.

- S/W & H/W Requirements.
Mongo DB 4.4 server
~~Intelli IDE~~ VS Code
windows 10 / Ubuntu
i5 processor, etc.

- Theory:

MongoDB : Java driver

- MongoDB provides connectivity for Java client application using JAVA driver.
- Using the Java driver is simple just include the driver for mongo.jar in your class path
- The JDBC API can do the following things:
  1) establish a connection with a database or access any tabular data source.
  2) send values MongoDB statements.
  3) Retrieve & process resuts ~~yet~~ received from the database.

- 3 Jar files:
  1) mongodb-driver_core-4.1.1
  2) mongodb-driver-sync-4.1.1
  3) bson-4.1.1

- Connect to database
  1) create a mongo client:
     MongoClient (mongo= new MongClient(" localhost", 27017)
  2) MongoDatabase database= mongo.get-database ("name")

- Create a collection

  Syntax:

```
database. createCollection ("collection-name")
```

- Getting selectiong a connection:

```
MongoClient <Document) collection=
                database. getCollection ("name")
```

- Insert a document:

```
Document doc = new Document ("key1", "value1")
                              . append("key2", "value2")
collection. inserOne (document);
```

- Retrieving all documents

```
FindIterable <Document) interDoc = collection.find()
Iterator it = iteror.iterator()
while (it. hasNext()) {
        system.out. println (it.next());
        i++;
}
```

- Update document

```
collection. updateOne (fillers.eq ("title", 1),
                         update. set ("key2",
                                 "valuee2")
```

- Delete a document

```
collection. deleteOne (Fillers.eq ("title", "MongoDB");
```

- Dropping a collection
  collection.drop()

- TEST CASES:

	Inpaut	Expected o/p	Actual o/p
1)	Enter collection name: songs	collection created	Success
2)	Enter song title: Perfect Enter Album: AA	Document inserted	Success

- Conclusion:
  - Thus we implemented mongoDB database connectivity with JAVA.
  - We implemented CURD operations.