

```

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 MongoDBDatabase db;
    private static final MongoCollection<Document> collection;
    private static final JsonWriterSettings settings;
    private static final Scanner scanner;

    static {
        mongoClient = MongoClient.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());

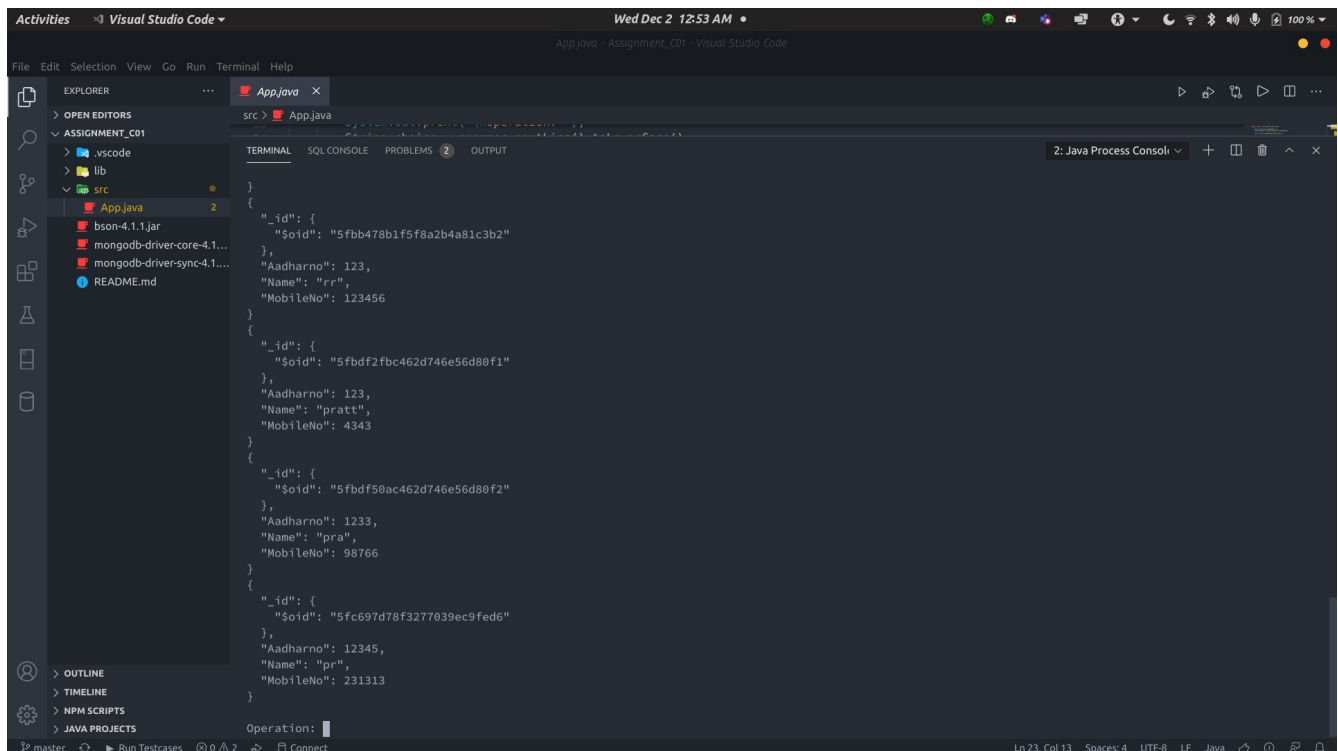
    System.out.print("Name: ");
    String n = scanner.nextLine();

    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": {
    "$oid": "5fbb478b1f5f8a2b4a81c3b2"
  },
  "Aadharno": 123,
  "Name": "rrr",
  "MobileNo": 123456
}
{
  "_id": {
    "$oid": "5fbdf2fbc462d746e56d80f1"
  },
  "Aadharno": 123,
  "Name": "pratt",
  "MobileNo": 4343
}
{
  "_id": {
    "$oid": "5fbdf58ac462d746e56d80f2"
  },
  "Aadharno": 1233,
  "Name": "pra",
  "MobileNo": 98766
}
{
  "_id": {
    "$oid": "5fc697d78f3277039ec9fed6"
  },
  "Aadharno": 12345,
  "Name": "pr",
  "MobileNo": 231313
}

```

Visual Studio Code interface showing a Java application running. The Explorer sidebar shows the project structure with files like `App.java`, `bson-4.1.1.jar`, `mongodb-driver-core-4.1...`, `mongodb-driver-sync-4.1...`, and `README.md`. The main editor displays the `App.java` file with the following content:

```
src > App.java
{
  "_id": {
    "$oid": "5fbb3db0ff185ed7360675b5"
  },
  "Aadharno": 133455.0,
  "Name": "Aditya Somani",
  "MobileNo": 7.755922893E9
}
{
  "_id": {
    "$oid": "5fbb478b1f5f8a2b4a81c3b2"
  },
  "Aadharno": 123,
  "Name": "rrr",
  "MobileNo": 123456
}
{
  "_id": {
    "$oid": "5fbbf2fbc462d746e56d80f1"
  },
  "Aadharno": 123,
  "Name": "pratt",
  "MobileNo": 4343
}
{
  "_id": {
    "$oid": "5fbbf59ac462d746e56d80f2"
  },
  "Aadharno": 1233,
  "Name": "pra",
  "MobileNo": 98766
}
{
  "_id": {
```

The terminal output shows the following JSON documents:

```
{
  "_id": {
    "$oid": "5fbb3db0ff185ed7360675b5"
  },
  "Aadharno": 133455.0,
  "Name": "Aditya Somani",
  "MobileNo": 7.755922893E9
}
{
  "_id": {
    "$oid": "5fbb478b1f5f8a2b4a81c3b2"
  },
  "Aadharno": 123,
  "Name": "rrr",
  "MobileNo": 123456
}
{
  "_id": {
    "$oid": "5fbbf2fbc462d746e56d80f1"
  },
  "Aadharno": 123,
  "Name": "pratt",
  "MobileNo": 4343
}
{
  "_id": {
    "$oid": "5fbbf59ac462d746e56d80f2"
  },
  "Aadharno": 1233,
  "Name": "pra",
  "MobileNo": 98766
}
{
  "_id": {
```

Visual Studio Code interface showing the same Java application. The Explorer sidebar shows the project structure with files like `App.java`, `bson-4.1.1.jar`, `mongodb-driver-core-4.1...`, `mongodb-driver-sync-4.1...`, and `README.md`. The main editor displays the `App.java` file with the following content:

```
src > App.java
{
  "_id": {
    "$oid": "5fbb3db0ff185ed7360675b5"
  },
  "Aadharno": 133455.0,
  "Name": "Aditya Somani",
  "MobileNo": 7.755922893E9
}
{
  "_id": {
    "$oid": "5fbb478b1f5f8a2b4a81c3b2"
  },
  "Aadharno": 123,
  "Name": "rrr",
  "MobileNo": 123456
}
{
  "_id": {
    "$oid": "5fbbf2fbc462d746e56d80f1"
  },
  "Aadharno": 123,
  "Name": "pratt",
  "MobileNo": 4343
}
{
  "_id": {
    "$oid": "5fbbf59ac462d746e56d80f2"
  },
  "Aadharno": 1233,
  "Name": "pra",
  "MobileNo": 98766
}
{
  "_id": {
```

The terminal output shows the following JSON documents:

```
{
  "_id": {
    "$oid": "5fbb3db0ff185ed7360675b5"
  },
  "Aadharno": 133455.0,
  "Name": "Aditya Somani",
  "MobileNo": 7.755922893E9
}
{
  "_id": {
    "$oid": "5fbb478b1f5f8a2b4a81c3b2"
  },
  "Aadharno": 123,
  "Name": "rrr",
  "MobileNo": 123456
}
{
  "_id": {
    "$oid": "5fbbf2fbc462d746e56d80f1"
  },
  "Aadharno": 123,
  "Name": "pratt",
  "MobileNo": 4343
}
{
  "_id": {
    "$oid": "5fbbf59ac462d746e56d80f2"
  },
  "Aadharno": 1233,
  "Name": "pra",
  "MobileNo": 98766
}
{
  "_id": {
```

Visual Studio Code interface showing a Java file named `App.java` in the `src` directory. The code defines a `Person` class with attributes `id`, `adharNo`, `name`, and `mobileNo`. The `main` method creates and prints three `Person` objects.

```
src > App.java
import java.util.*;

class Person {
    String id;
    String adharNo;
    String name;
    String mobileNo;
}

public class App {
    public static void main(String[] args) {
        Person p1 = new Person();
        p1.id = "5fbb3db0ff185ed7360675b5";
        p1.adharNo = "133455.0";
        p1.name = "Aditya Somani";
        p1.mobileNo = "7.755922893E9";

        Person p2 = new Person();
        p2.id = "5fbb478b1f5f8a2b4a81c3b2";
        p2.adharNo = "123";
        p2.name = "rrr";
        p2.mobileNo = "123456";

        Person p3 = new Person();
        p3.id = "5fbd2f2fbc462d746e56d80f1";
        p3.adharNo = "123";
        p3.name = "pratt";
        p3.mobileNo = "4343";

        Person p4 = new Person();
        p4.id = "5fbd59ac462d746e56d80f2";
        p4.adharNo = "1233";
        p4.name = "pra";
        p4.mobileNo = "98766";

        System.out.println(p1);
        System.out.println(p2);
        System.out.println(p3);
        System.out.println(p4);
    }
}
```

Visual Studio Code interface showing the same `App.java` file. The code now includes a `delete` operation and a `find` operation. The `main` method prints the results of these operations.

```
src > App.java
import java.util.*;

class Person {
    String id;
    String adharNo;
    String name;
    String mobileNo;
}

public class App {
    public static void main(String[] args) {
        Person p1 = new Person();
        p1.id = "5fbb3db0ff185ed7360675b5";
        p1.adharNo = "133455.0";
        p1.name = "Aditya Somani";
        p1.mobileNo = "7.755922893E9";

        Person p2 = new Person();
        p2.id = "5fbb478b1f5f8a2b4a81c3b2";
        p2.adharNo = "123";
        p2.name = "rrr";
        p2.mobileNo = "123456";

        Person p3 = new Person();
        p3.id = "5fbd2f2fbc462d746e56d80f1";
        p3.adharNo = "123";
        p3.name = "pratt";
        p3.mobileNo = "4343";

        Person p4 = new Person();
        p4.id = "5fbd59ac462d746e56d80f2";
        p4.adharNo = "1233";
        p4.name = "pra";
        p4.mobileNo = "98766";

        System.out.println(p1);
        System.out.println(p2);
        System.out.println(p3);
        System.out.println(p4);

        Operation delete = new Operation();
        delete.delete("12345", "pr");
        System.out.println(delete.getResult());

        Operation find = new Operation();
        find.find("5fbb3d95ff185ed7360675b3");
        System.out.println(find.getResult());
    }
}
```

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 CRUD operations on database.
- SIW & HIW 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 results ret 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 MongoClient("localhost", 27017)
```
 - 2) MongoDB database database=


```
mongo.getDatabase("name")
```


- Create a collection

Syntax:

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

- Getting selecting a connection:

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

- Insert a document:

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

- Retrieving all documents

```
FindIterable <Document> iterDoc = collection.find()  
Iterator it = iterDoc.iterator()  
while (it.hasNext()) {  
    System.out.println(it.next());  
    i++;  
}
```

- Update document

```
collection.updateOne(filters.eq("title", 1),  
                    update.set("key2",  
                                "value2"))
```

- Delete a document

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


- Dropping a collection
collection.drop()

• TEST CASES:

	Input	Expected op	Actual op
1)	Enter collection name: songs	collection created	Success
2)	Enter song title: Perfect Enter Album: AA	Document inserted	Success

• Conclusion:

- This we implemented MongoDB database connectivity with JAVA.
- We implemented CRUD operations.