**Group: -** 1 from T4 Batch
0077 – Aakash Joshi
2039 – Akanksha Lokhande

**Title**-: Database Connectivity(MongoDB)

**Date of Completion-:**

**Objectives-:**

- To develop Database programming skills.
- To develop basic Database administration skills.
- To develop skills to handle NoSQL database.
- To learn, understand and execute process of software application development.

**Outcomes-:**

- Design schema in appropriate normal form considering actual requirements.
- Implement SQL queries for given requirements , using different SQL concepts..
- Implement NoSQL queries using MongoDB.

**Problem Statement-:**

Write a program to implement MongoDB database connectivity with any front end language to implement Database navigation operations (add, delete, edit etc.)

**Software and Hardware requirement-:**

- 64-bit Open source Linux or its derivative.

- Mango-DB

## Theory-:

connect()

Description

**connect(url, user, password)**

Creates a connection to a MongoDB instance and returns the reference to the database. However, in most cases, use the Mongo() object and its getDB() method instead.

| Parameter | Type | Description |
|---|---|---|
| url | string | Specifies the connection string. You can specify either:<br>• <hostname>:<port>/<database><br>• <hostname>/<database><br>• <database> |
| user | string | Optional. Specifies an existing username with access privileges for this database. If user is specified, you must include the password parameter as well. |
| password | string | Optional unless the user parameter is specified. Specifies the password for the user. |

## Example

The following example instantiates a new connection to the MongoDB instance running on the localhost interface and returns a reference to myDatabase:

db = connect(**"localhost:27017/myDatabase"**)

## Code-: (Implementation in JAVA)

```
import java.net.UnknownHostException;
 import java.util.Scanner;
import com.mongodb.*;
public class DatabaseConnectivity { private static void choice_input(){
System.out.println("\n1.insert data into database\n2.update database
documents\n3.delete database documents\n4.show database collections\n5.Exit");
}
public static void main(String[] args) {
String key, value; Scanner = new Scanner(System.in);
 int choice; try {
Mongo = new Mongo("localhost", 27017);
DB = mongo.getDB("myDb"); DBCollection collection =
db.getCollection("dummyColl");
do{ choice_input();
System.out.println("Enter your
choice: ");
 choice = scanner.nextInt();switch (choice){ case 1:
```

```
String ch;
do{ System.out.println("Enter key: ");
key = scanner.next(); System.out.println("Enter value: ");
value = scanner.next();document.put(key, value); System.out.println("Do you want to
enter more(y/n)? "); ch = scanner.next();
} while (!ch.equals("n")); collection.insert(document); break;
case 2:
BasicDBObject searchObj = new BasicDBObject();
System.out.println("Enter searched key: ");
key = scanner.next();
System.out.println("Enter searched value: ");
value = scanner.next();
searchObj.put(key, value);
BasicDBObject newObj = new BasicDBObject(); System.out.println("Enter new key:
");
key = scanner.next();
System.out.println("Enter new value: ");
value = scanner.next(); newObj.put(key, value);
collection.update(searchObj, newObj);
break;
case 3:
System.out.println("Enter removable key: ");
key = scanner.next(); System.out.println("Enter removable value: ");
value = scanner.next();
BasicDBObject removableObj = new
BasicDBObject(); removableObj.put(key, value);
collection.remove(removableObj); break; case 4:
DBCursor cursorDoc = collection.find();
while (cursorDoc.hasNext()) {

System.out.println(cursorDoc.next ());
}
Break;
Case5:
System.e xit(0); break;
}
} while(choice != 6);
} catch (UnknownHostException | MongoException e) { e.printStackTrace();
}
}
}
```

**Output-: (Output of Required System)**

```
1.insert data into database 2.update database documents 3.delete database
documents 4.show database collections 5.Exit
Enter your choice: 1 Enter key:2 Enter value:
1. delete database documents 4.show database collections 5.Exit
Enter your choice:
2
Enter searched key:
2
Enter searched value:
harish
Enter new key:
1
Enter new value:
Sam
1. insert data into database 2.update database documents 3.delete database
documents 4.show database collections 5.Exit

Enter your choice:
4{
"_id" : { "$oid" : "5bb453bce4b0283ac9d3205d"} , "1" : "sam"}
1.insert data into database
2. update database documents
```

```
3.delete database documents 4.show database collections
5.Exit Enter your choice:
3
Enter removable key:3
Enter removable value: hari
1. insert data into database 2.update database documents 3.delete database
documents 4.show database collections 5.Exit
Enter your choice:
4
{
"_id" : { "$oid" : "5bb453bce4b0283ac9d3205d"} , "1" : "sam"}
1.insert data into database
2. update database documents
3.delete database documents
4.show database collections
5.Exit
Enter your choice: 5
```

## **CONCLUSION:**

We have successfully studied the implementation and implemented Database navigation operations in MongoDB.