DBMSL ASSIGNMENT – 12

Roll No.: 31446

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

JAVA PROGRAM

```
package mypackage;

import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
import java.util.Scanner;
import static com.mongodb.client.model.Filters.eq;

public class MongoDBNavigation {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        //Step 1: Connect to yourMongoDB server
        String connectionString =
"mongodb://te31446:te31446@10.10.8.119:27017/?authSource=te31446_db";
        MongoClient mongoClient = MongoClients.create(connectionString);
        System.out.println("Connected to MongoDB successfully!");

        //Step 2: Access database and collection
        MongoDatabase db = mongoClient.getDatabase("te31446_db");
        MongoCollection<Document> collection =
db.getCollection("student");

        //Step 3: Menu-driven operations
        while (true) {
            System.out.println("\n===== MONGO DB NAVIGATION MENU =====");
            System.out.println("1. Add Record");
            System.out.println("2. View All Records");
            System.out.println("3. Update Record");
            System.out.println("4. Delete Record");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();

            switch (choice) {
                case 1 -> addRecord(collection, sc);
                case 2 -> viewRecords(collection);
                case 3 -> updateRecord(collection, sc);
                case 4 -> deleteRecord(collection, sc);
                case 5 -> {
                    mongoClient.close();
                    System.out.println("Disconnected from MongoDB!");
                    return;
                }
                default -> System.out.println("Invalid choice. Try
again.");
            }
        }
    }

    // Add record
    static void addRecord(MongoCollection<Document> collection, Scanner
sc) {
        sc.nextLine(); // clear buffer
```

```
        System.out.print("Enter name: ");
        String name = sc.nextLine();
        System.out.print("Enter age: ");
        int age = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter department: ");
        String dept = sc.nextLine();
        Document doc = new Document("name", name)
                .append("age", age)
                .append("department", dept);
        collection.insertOne(doc);
        System.out.println("Record added successfully!");
    }

    // View records
    static void viewRecords(MongoCollection<Document> collection) {
        System.out.println("\n--- Student Records ---");
        for (Document doc : collection.find()) {
            System.out.println(doc.toJson());
        }
    }

    // Update record
    static void updateRecord(MongoCollection<Document> collection,
Scanner sc) {
        sc.nextLine();
        System.out.print("Enter name of student to update: ");
        String name = sc.nextLine();
        System.out.print("Enter new age: ");
        int age = sc.nextInt();
        sc.nextLine();
        System.out.print("Enter new department: ");
        String dept = sc.nextLine();

        Document updateDoc = new Document("$set",
                new Document("age", age).append("department", dept));
        var result = collection.updateOne(eq("name", name), updateDoc);

        if (result.getMatchedCount() > 0)
            System.out.println("Record updated successfully!");
        else
            System.out.println("No record found with name: " + name);
    }

    // Delete record
    static void deleteRecord(MongoCollection<Document> collection,
Scanner sc) {
        sc.nextLine();
        System.out.print("Enter name of student to delete: ");
        String name = sc.nextLine();
        var result = collection.deleteOne(eq("name", name));

        if (result.getDeletedCount() > 0)
            System.out.println("Record deleted successfully!");
        else
            System.out.println("No record found with name: " + name);
    }
}
```

OUTPUT

```
Oct 27, 2025 10:15:02 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017],
mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000
ms', maxWaitQueueSize=500}
Connected to MongoDB successfully!

===== MONGO DB NAVIGATION MENU =====
1. Add Record
2. View All Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice: Oct 27, 2025 10:15:02 PM
com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:20}] to
localhost:27017
Oct 27, 2025 10:15:02 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description
ServerDescription{address=localhost:27017, type=STANDALONE,
state=CONNECTED, ok=true, version=ServerVersion{versionList=[8, 2, 1]},
minWireVersion=0, maxWireVersion=27, maxDocumentSize=16777216,
logicalSessionTimeoutMinutes=30, roundTripTimeNanos=3590100}
1
Enter name: Vineet Sharma
Enter age: 23
Enter department: CS
Oct 27, 2025 10:15:24 PM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:21}] to
localhost:27017
Record added successfully!

===== MONGO DB NAVIGATION MENU =====
1. Add Record
2. View All Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice: 1
Enter name: Sayali Pawar
Enter age: 20
Enter department: CS
Record added successfully!

===== MONGO DB NAVIGATION MENU =====
1. Add Record
2. View All Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice: 1
Enter name: Yash Patel
Enter age: 23
Enter department: IT
Record added successfully!
```

```
===== MONGO DB NAVIGATION MENU =====
1. Add Record
2. View All Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice: 2

--- Student Records ---
{"_id": {"$oid": "68ffa1a45fd16a230a0a0bb5"}, "name": "Vineet Sharma",
"age": 23, "department": "CS"}
{"_id": {"$oid": "68ffa1b45fd16a230a0a0bb6"}, "name": "Sayali Pawar",
"age": 20, "department": "CS"}
{"_id": {"$oid": "68ffa1c95fd16a230a0a0bb7"}, "name": "Yash Patel",
"age": 23, "department": "IT"}

===== MONGO DB NAVIGATION MENU =====
1. Add Record
2. View All Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice: 3
Enter name of student to update: Yash Patel
Enter new age: 21
Enter new department: CS
Record updated successfully!

===== MONGO DB NAVIGATION MENU =====
1. Add Record
2. View All Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice: 2

--- Student Records ---
{"_id": {"$oid": "68ffa1a45fd16a230a0a0bb5"}, "name": "Vineet Sharma",
"age": 23, "department": "CS"}
{"_id": {"$oid": "68ffa1b45fd16a230a0a0bb6"}, "name": "Sayali Pawar",
"age": 20, "department": "CS"}
{"_id": {"$oid": "68ffa1c95fd16a230a0a0bb7"}, "name": "Yash Patel",
"age": 21, "department": "CS"}

===== MONGO DB NAVIGATION MENU =====
1. Add Record
2. View All Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice: 4
Enter name of student to delete: Sayali Pawar
Record deleted successfully!

===== MONGO DB NAVIGATION MENU =====
1. Add Record
2. View All Records
3. Update Record
```

```
4. Delete Record
5. Exit
Enter your choice: 2

--- Student Records ---
{"_id": {"$oid": "68ffa1a45fd16a230a0a0bb5"}, "name": "Vineet Sharma",
"age": 23, "department": "CS"}
{"_id": {"$oid": "68ffa1c95fd16a230a0a0bb7"}, "name": "Yash Patel",
"age": 21, "department": "CS"}

===== MONGO DB NAVIGATION MENU =====
1. Add Record
2. View All Records
3. Update Record
4. Delete Record
5. Exit
Enter your choice: 5
Disconnected from MongoDB!
```