

**Name: Om Panchwate**

**Roll No.: 31455**

**Class: TE-IV**

**Course: DBMSL**

**Batch: M4**

---

### **ASSIGNMENT NO. 12**

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

**Code:-**

```
import com.mongodb.MongoClient;
import com.mongodb.MongoClientURI;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;

import java.util.Scanner;

public class MongoDBCruExample {

    public static void main(String[] args) {
        String connectionString = "mongodb://localhost:27017";

        try (MongoClient client = new MongoClient(new
MongoClientURI(connectionString))) {
            MongoDatabase database = client.getDatabase("test");
            MongoCollection<Document> collection = database.getCollection("mycollection");

            Scanner scanner = new Scanner(System.in);
            boolean exit = false;

            while (!exit) {
                System.out.println("1. Create");
                System.out.println("2. Read");
                System.out.println("3. Update");
                System.out.println("4. Delete");
                System.out.println("5. Exit");
                System.out.println("Choose an operation:");

                int choice = scanner.nextInt();
                scanner.nextLine();
```

```

switch (choice) {
    case 1:
        System.out.println("Enter name:");
        String name = scanner.nextLine();
        System.out.println("Enter age:");
        int age = scanner.nextInt();
        System.out.println("Enter city:");
        String city = scanner.next();
        createDocument(collection, name, age, city);
        break;
    case 2:
        System.out.println("Enter name to retrieve:");
        String retrieveName = scanner.nextLine();
        readDocument(collection, retrieveName);
        break;
    case 3:
        System.out.println("Enter name to update:");
        String updateName = scanner.nextLine();
        System.out.println("Enter new age:");
        int newAge = scanner.nextInt();
        updateDocument(collection, updateName, newAge);
        break;
    case 4:
        System.out.println("Enter name to delete:");
        String deleteName = scanner.nextLine();
        deleteDocument(collection, deleteName);
        break;
    case 5:
        exit = true;
        break;
    default:
        System.out.println("Invalid choice. Please select a valid operation.");
        break;
}
}

} catch (Exception e) {
    e.printStackTrace();
}
}

```

```

private static void createDocument(MongoCollection<Document> collection, String
name, int age, String city) {

```

```

        Document document = new Document("name", name)
            .append("age", age)
            .append("city", city);
        collection.insertOne(document);
        System.out.println("Document created");
    }

    private static void readDocument(MongoCollection<Document> collection, String
name) {
        Document retrievedDocument = collection.find(new Document("name",
name)).first();
        System.out.println("Retrieved document: " + retrievedDocument.toJson());
    }

    private static void updateDocument(MongoCollection<Document> collection, String
name, int newAge) {
        collection.updateOne(new Document("name", name), new Document("$set", new
Document("age", newAge)));
        System.out.println("Document updated");
    }

    private static void deleteDocument(MongoCollection<Document> collection, String
name) {
        collection.deleteOne(new Document("name", name));
        System.out.println("Document deleted");
    }
}

```

## Output:-

### 1. Create and insert data in the collection

```

1. Create
2. Read
3. Update
4. Delete
5. Exit
Choose an operation:
1
Enter name: Om
Enter age: 20
Enter city: New York
Document created

```

## **2. Read value from collection**

1. Create

2. Read

3. Update

4. Delete

5. Exit

Choose an operation:

2

Enter name to retrieve: Om

Retrieved document: {"\_id": {"\$oid": "5f58f15056f98cfc5e33b6a5"}, "name": "Om", "age": 20, "city": "New York"}

## **3. Update value in collection**

1. Create

2. Read

3. Update

4. Delete

5. Exit

Choose an operation:

3

Enter name to update: Om

Enter new age: 19

Document updated

## **4. Delete value from collection**

1. Create

2. Read

3. Update

4. Delete

5. Exit

Choose an operation:

4

Enter name to delete: Om

Document deleted

## **5. Exit**

1. Create

2. Read

3. Update

4. Delete

5. Exit

Choose an operation:

5

The program exits.

---