TE31413

Assignment no : 12

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

1) Database :

```
connectiondb> db.cutoff.find()

connectiondb>
```

2)

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

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

public class Assignment12 {

    private MongoClient mongoClient;
    private MongoDatabase database;
    private MongoCollection<Document> collection;

    public Assignment12(String dbName, String collectionName) {
        mongoClient = new MongoClient("localhost", 27017);
        database = mongoClient.getDatabase(dbName);
        collection = database.getCollection(collectionName);
    }

    // Create (Insert) Documents
    public void createDocuments() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter college name: ");
        String college = scanner.nextLine();
        System.out.print("Enter branch name: ");
        String branch = scanner.nextLine();

        Document doc = new Document("college", college).append("branch",
branch);
        collection.insertOne(doc);
        System.out.println("Inserted document: " + doc.toJson());
    }

    // Read (Find) Documents
    public void readDocuments() {
        System.out.println("Documents in the collection:");
        MongoCursor<Document> cursor = collection.find().iterator();
        while (cursor.hasNext()) {
            System.out.println(cursor.next().toJson());
        }
    }

    // Update Documents
```

```java
public void updateDocument() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter college name to update: ");
    String college = scanner.nextLine();
    System.out.print("Enter new branch name: ");
    String newBranch = scanner.nextLine();

    collection.updateOne(new Document("college", college),
            new Document("$set", new Document("branch", newBranch)));
    System.out.println("Updated document where college is " + college);
}

// Delete Documents
public void deleteDocuments() {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter college name to delete: ");
    String college = scanner.nextLine();

    collection.deleteOne(new Document("college", college));
    System.out.println("Deleted one document where college is " + college);
}

public void close() {
    mongoClient.close();
    System.out.println("Mongo connection closed.");
}

public static void main(String[] args) {
  Assignment12 example = new Assignment12("connectiondb", "cutoff");
    Scanner scanner = new Scanner(System.in);
    int choice;

    do {
        System.out.println("\n--- MongoDB Menu ---");
        System.out.println("1. Create Document");
        System.out.println("2. Read Documents");
        System.out.println("3. Update Document");
        System.out.println("4. Delete Document");
        System.out.println("5. Exit");
        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();
        scanner.nextLine();

        switch (choice) {
            case 1:
                example.createDocuments();
                break;
            case 2:
                example.readDocuments();
                break;
            case 3:
                example.updateDocument();
                break;
            case 4:
                example.deleteDocuments();
                break;
            case 5:
                example.close();
                break;
            default:
                System.out.println("Invalid choice. Please try again.");
        }
```

```
        } while (choice != 5);
    }
}
```

```
Assignment12 [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java  (30-Sept-2024, 10:25:15 am) [pid: 8464]
Sep 30, 2024 10:25:15 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='300

--- MongoDB Menu ---
1. Create Document
2. Read Documents
3. Update Document
4. Delete Document
5. Exit
Enter your choice: Sep 30, 2024 10:25:15 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1}] to localhost:27017
Sep 30, 2024 10:25:15 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription{address=localhost:27017, type=STANDALONE, 
1
Enter college name: PICT
Enter branch name: Computer Engineering
Sep 30, 2024 10:25:36 AM com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2}] to localhost:27017
Inserted document: { "_id" : { "$oid" : "66fa2f4849b4a021101c417e" }, "college" : "PICT", "branch" : "Computer Engineering" }

--- MongoDB Menu ---
1. Create Document
2. Read Documents
3. Update Document
4. Delete Document
5. Exit
Enter your choice: 1
Enter college name: Cummins
Enter branch name: IT
Inserted document: { "_id" : { "$oid" : "66fa2f9e49b4a021101c417f" }, "college" : "Cummins", "branch" : "IT" }
```

```
connectiondb> db.cutoff.find()
[
  {
    _id: ObjectId('66fa2f4849b4a021101c417e'),
    college: 'PICT',
    branch: 'Computer Engineering'
  },
  {
    _id: ObjectId('66fa2f9e49b4a021101c417f'),
    college: 'Cummins',
    branch: 'IT'
  }
]
connectiondb>
```

```
Assignment12 [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java  (30-Sept-2024, 10:25:15 am) [pid: 8464]
--- MongoDB Menu ---
1. Create Document
2. Read Documents
3. Update Document
4. Delete Document
5. Exit
Enter your choice: 2
Documents in the collection:
{ "_id" : { "$oid" : "66fa2f4849b4a021101c417e" }, "college" : "PICT", "branch" : "Computer Engineering" }
{ "_id" : { "$oid" : "66fa2f9e49b4a021101c417f" }, "college" : "Cummins", "branch" : "IT" }

--- MongoDB Menu ---
1. Create Document
2. Read Documents
3. Update Document
4. Delete Document
5. Exit
Enter your choice: 3
Enter college name to update: pict
Enter new branch name: CS
Updated document where college is pict

--- MongoDB Menu ---
1. Create Document
2. Read Documents
3. Update Document
4. Delete Document
5. Exit
Enter your choice: 2
Documents in the collection:
{ "_id" : { "$oid" : "66fa2f4849b4a021101c417e" }, "college" : "PICT", "branch" : "Computer Engineering" }
{ "_id" : { "$oid" : "66fa2f9e49b4a021101c417f" }, "college" : "Cummins", "branch" : "IT" }
```

```
--- MongoDB Menu ---
1. Create Document
2. Read Documents
3. Update Document
4. Delete Document
5. Exit
Enter your choice: 4
Enter college name to delete: cummins
Deleted one document where college is cummins

    MongoDB Menu
```