**1. Creating a Database and Inserting a Collection**

package connection;

import org.bson.Document;

import com.mongodb.client.\*;

public class CreateDB {

public static void main(String[] args) {

try (MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017")) {

MongoDatabase database = mongoClient.getDatabase("eShopDB");

database.createCollection("products");

MongoCollection<Document> collection = database.getCollection("products");

Document doc = new Document("name", "Laptop").append("price", 999);

collection.insertOne(doc);

System.out.println("Database and collection created successfully.");

System.out.println("Available Databases:");

for (String name : mongoClient.listDatabaseNames()) {

System.out.println("- " + name);

}

}

}

}

**2. Dropping the Database**

package connection;

import com.mongodb.client.\*;

public class DropDB {

public static void main(String[] args) {

try (MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017")) {

MongoDatabase database = mongoClient.getDatabase("eShopDB");

database.drop();

System.out.println("Database 'eShopDB' dropped.");

System.out.println("Remaining Databases:");

for (String name : mongoClient.listDatabaseNames()) {

System.out.println("- " + name);

}

}

}

}

**3. Creating and Displaying Collections**

package connection;

import com.mongodb.client.\*;

public class CreateAndDisplayCollection {

public static void main(String[] args) {

try (MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017")) {

MongoDatabase database = mongoClient.getDatabase("shopDB");

database.createCollection("inventory");

System.out.println("Collections in 'shopDB':");

for (String name : database.listCollectionNames()) {

System.out.println("- " + name);

}

}

}

}

**4. Inserting Multiple Documents**

package connection;

import org.bson.Document;

import com.mongodb.client.\*;

import java.util.\*;

public class InsertDocuments {

public static void main(String[] args) {

try (MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017")) {

MongoDatabase database = mongoClient.getDatabase("shopDB");

MongoCollection<Document> collection = database.getCollection("inventory");

List<Document> docs = new ArrayList<>();

docs.add(new Document("item", "Pen").append("price", 10).append("stock", 100));

docs.add(new Document("item", "Notebook").append("price", 30).append("stock", 50));

docs.add(new Document("item", "Marker").append("price", 15).append("stock", 80));

collection.insertMany(docs);

System.out.println("Documents inserted into 'inventory' collection.");

}

}

}

**5. Retrieving Documents (All and Filtered)**

package connection;

import org.bson.Document;

import com.mongodb.client.\*;

public class RetrieveDocuments {

public static void main(String[] args) {

try (MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017")) {

MongoDatabase database = mongoClient.getDatabase("shopDB");

MongoCollection<Document> collection = database.getCollection("inventory");

System.out.println("=== All Documents ===");

for (Document doc : collection.find()) {

System.out.println(doc.toJson());

}

System.out.println("\n=== Filter: item = 'Pen' ===");

for (Document doc : collection.find(new Document("item", "Pen"))) {

System.out.println(doc.toJson());

}

}

}

}

**6. Inserting and Filtering Based on User Input**

package connection;

import org.bson.Document;

import com.mongodb.client.\*;

import com.mongodb.client.model.Filters;

import java.util.Scanner;

public class UserInputFilter {

public static void main(String[] args) {

try (Scanner sc = new Scanner(System.in);

MongoClient mongoClient = MongoClients.create("mongodb://localhost:27017")) {

MongoDatabase database = mongoClient.getDatabase("shopDB");

MongoCollection<Document> collection = database.getCollection("inventory");

// Insert document via user input

System.out.println("Enter item name:");

String item = sc.nextLine();

System.out.println("Enter price:");

double price = sc.nextDouble();

System.out.println("Enter stock:");

int stock = sc.nextInt();

sc.nextLine(); // Clear newline

Document newDoc = new Document("item", item).append("price", price).append("stock", stock);

collection.insertOne(newDoc);

System.out.println("Document inserted.\n");

// Filter documents

System.out.println("Enter field to filter (e.g., item):");

String field = sc.nextLine();

System.out.println("Enter value:");

String value = sc.nextLine();

System.out.println("\*\*\* Filtered Results \*\*\*");

for (Document doc : collection.find(Filters.eq(field, value))) {

System.out.println(doc.toJson());

}

}

}

}