**Assignment Group: B**

**Assignment NO 13**

**Assignment Title**: 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.)

// Java Code

import com.mongodb.MongoClient;

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 MongoDBCRUD {

private static final String DATABASE\_NAME = "sampleDB";

private static final String COLLECTION\_NAME = "Users";

private static MongoClient mongoClient;

private static MongoDatabase database;

private static MongoCollection<Document> collection;

public static void main(String[] args) {

mongoClient = new MongoClient("localhost", 27017); // Connect to MongoDB

database = mongoClient.getDatabase(DATABASE\_NAME); // Access the database

collection = database.getCollection(COLLECTION\_NAME); // Access the collection

Scanner scanner = new Scanner(System.in);

int choice;

do {

System.out.println("\nMongoDB CRUD Operations");

System.out.println("1. Add User");

System.out.println("2. Edit User");

System.out.println("3. Delete User");

System.out.println("4. Display Users");

System.out.println("0. Exit");

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

scanner.nextLine(); // Consume newline

switch (choice) {

case 1:

// Add User

System.out.print("Enter Name: ");

String name = scanner.nextLine();

System.out.print("Enter Age: ");

int age = scanner.nextInt();

addUser(name, age);

break;

case 2:

// Edit User

System.out.print("Enter User ID to Edit: ");

String idToEdit = scanner.nextLine();

System.out.print("Enter New Name: ");

String newName = scanner.nextLine();

editUser(idToEdit, newName);

break;

case 3:

// Delete User

System.out.print("Enter User ID to Delete: ");

String idToDelete = scanner.nextLine();

deleteUser(idToDelete);

break;

case 4:

// Display Users

displayUsers();

break;

case 0:

System.out.println("Exiting...");

break;

default:

System.out.println("Invalid choice. Please try again.");

}

} while (choice != 0);

mongoClient.close(); // Close the connection

}

private static void addUser(String name, int age) {

Document document = new Document("Name", name)

.append("Age", age);

collection.insertOne(document); // Insert document into collection

System.out.println("User added successfully!");

}

private static void editUser(String id, String newName) {

collection.updateOne(eq("\_id", new org.bson.types.ObjectId(id)), new Document("$set", new Document("Name", newName)));

System.out.println("User updated successfully!");

}

private static void deleteUser(String id) {

collection.deleteOne(eq("\_id", new org.bson.types.ObjectId(id)));

System.out.println("User deleted successfully!");

}

private static void displayUsers() {

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

System.out.println(doc.toJson()); // Print each document in JSON format

}

}

}