6. **Implement basic commands on databases and collections using MongoDB**

This program demonstrates basic **MongoDB** commands, including creating a database, working with collections, and performing CRUD (Create, Read, Update, Delete) operations.

Install **MongoDB** and ensure the service is running.  
Install the **MongoDB Node.js driver** using:

npm install mongodb

1. **Connects to MongoDB** using MongoClient.
2. **Creates/Selects a database (myDatabase)** and a collection (users).
3. **Performs CRUD operations**:

* Inserts one document.
* Inserts multiple documents.
* Finds a single document.
* Retrieves all documents in the collection.
* Updates a document.
* Deletes a document.

1. **Lists all collections** in the database.
2. **Closes the MongoDB connection**.

const { MongoClient } = require('mongodb');

// MongoDB connection URL

const url = 'mongodb://localhost:27017';

// Create a new MongoDB client

const client = new MongoClient(url);

async function run() {

try {

// Connect to MongoDB

await client.connect();

console.log("Connected to MongoDB");

// Select the database and collection

const database = client.db('myDatabase');

const collection = database.collection('users');

// Insert a document

const insertResult = await collection.insertOne({ name: 'Alice', age: 25, city: 'New York' });

console.log("Inserted document ID:", insertResult.insertedId);

// Insert multiple documents

await collection.insertMany([

{ name: 'Bob', age: 28, city: 'Los Angeles' },

{ name: 'Charlie', age: 32, city: 'Chicago' }

]);

console.log("Multiple documents inserted");

// Find a single document

const user = await collection.findOne({ name: 'Alice' });

console.log('User found:', user);

// Find multiple documents

const users = await collection.find({}).toArray();

console.log('All Users:', users);

// Update a document

await collection.updateOne({ name: 'Alice' }, { $set: { age: 26 } });

console.log("Document updated");

// Delete a document

await collection.deleteOne({ name: 'Charlie' });

console.log("Document deleted");

// List all collections in the database

const collections = await database.listCollections().toArray();

console.log("Collections:", collections.map(col => col.name));

} catch (error) {

console.error("Error:", error);

} finally {

// Close the MongoDB connection

await client.close();

console.log("Connection closed");

}

}

// Execute the function

run().catch(console.dir);

**Output:**

Connected to MongoDB

Inserted document ID: 660c1e7...

Multiple documents inserted

User found: { \_id: ObjectId("660c1e7..."), name: 'Alice', age: 25, city: 'New York' }

All Users: [ { \_id: ObjectId(...), name: 'Alice', age: 25, city: 'New York' }, { \_id: ObjectId(...), name: 'Bob', age: 28, city: 'Los Angeles' }, { \_id: ObjectId(...), name: 'Charlie', age: 32, city: 'Chicago' } ]

Document updated

Document deleted

Collections: [ 'users' ]

Connection closed