4. Demonstrate Accessing MongoDB from Node.js. for College Database and students Collection with four CRUD operations Insert, Delete , Update and Find.

```
const { MongoClient, ObjectID, ObjectId } = require('mongodb');


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


// Database Name
const dbName = 'College';


// Create a new MongoClient
const client = new MongoClient(url, { useNewUrlParser: true, useUnifiedTopology: true });


// Connect to the MongoDB server
async function connectDB() {
    try {
        await client.connect();
        console.log('Connected to the database');
    } catch (error) {
        console.error('Error connecting to the database:', error);
    }
}
   // Insert operation (Create)
async function insertStudent(student) {
    const db = client.db(dbName);
    try {
```

```
    const result = await db.collection('students').insertOne(student);
    console.log(`Student with id ${result.insertedId} inserted successfully`);
    } catch (err) {
      console.error('Error inserting student:', err);
    }
}

    // Update operation
    async function updateStudent() {
        const db = client.db(dbName);
        try {
        const result = await db.collection('students').updateOne(
                                                                    {_id:new
ObjectId('66667a24ace74f68b2cdcdf6')},{$set:{Dept:"CSE"}});
            console.log(`Student data updated successfully`);
        } catch (err) {
          console.error('Error updating student:', err);
        }
    }

// Find all students
async function findAllStudents() {
        const db = client.db(dbName);
        try {
        const students = await db.collection('students').find({}).toArray();
        console.log('All students:', students);
        } catch (err) {
          console.error('Error finding students:', err);
```

```
        }
    }

// Delete operation
async function deleteStudent(Id) {
    const db = client.db(dbName);
    try {
        const result = await db.collection('students').deleteOne({ _id:new ObjectId(Id) });
    console.log(`Student with id ${Id} deleted successfully`);
    } catch (err) {
      console.error('Error deleting student:', err);
    }
}
  // Perform operations (uncomment as needed for demonstration)
  connectDB()
  .then(async () => {

  // Insert a student
  const exampleStudent = { name: 'Monisha', age: 18, cgpa:6.38, Dept:"CSE"};
  await insertStudent(exampleStudent);

  // Find all students
  await findAllStudents();

  // Update a student
  await updateStudent();
```

```
  // Delete a student
  const studentIdToDelete = '666bea5ca3a164ff0e37ba34'; // Replace with an existing student id
  await deleteStudent(studentIdToDelete);

  // Close the connection
  client.close();
});
```

Output: node App.js

Connected to the database

Student with id 667050dfabccee4a53178857 inserted successfully

All students: [

```
  {
    _id: new ObjectId('6666613b0f47808a70cdcdfa'),
    age: 18,
    Fulltime: false,
    Feespaid: true,
    name: 'Anu',
    cgpa: 8.25
  },
  {
    _id: new ObjectId('66667a24ace74f68b2cdcdf6'),
    name: 'Manasa',
    age: 30,
    cgpa: 8.45,
    Fulltime: false,
    Feespaid: true
```

```
  },
  {
    _id: new ObjectId('66681a00a4f0630ab8cdcdf6'),
    name: 'Kavya',
    age: 30,
    cgpa: 7.5,
    FullTime: true
  },
  {
    _id: new ObjectId('666bea5ca3a164ff0e37ba34'),
    name: 'Jhonny',
    age: 20,
    grade: 'A'
  },
  {
    _id: new ObjectId('66703c8053328e5f062f99e5'),
    name: 'John Doe',
    age: 20,
    grade: 'A'
  },
    _id: new ObjectId('667050dfabccee4a53178857'),
    name: 'Monisha',
    age: 18,
    cgpa: 6.38,
    Dept: 'CSE'
  }
]
```

Student data updated successfully

Student with id 666bea5ca3a164ff0e37ba34 deleted successfully