PROGRAM 4

4a Read the data of a student containing usn, name, sem, year_of_admission from node js and store it in the mongodb

1. **Set up Node.js:** Create a Node.js project and install the necessary dependencies, such as mongodb package, using npm.
2. **Write a Node.js script**: Create a Node.js script to read student data from a source (e.g., a file or user input) and store it in MongoDB.

```
const { MongoClient } = require('mongodb');
const readlineSync = require('readline-sync');

// MongoDB connection URL and database name
const url = 'mongodb://localhost:27017';
const dbName = 'schoolDB';

// Function to get student data from the user
function getStudentData() {
  return {
    usn: readlineSync.question('Enter USN: '),
    name: readlineSync.question('Enter Name: '),
    sem: readlineSync.question('Enter Semester: '),
    year_of_admission: readlineSync.question('Enter Year of Admission: ')
  };
}

// Function to insert student data into MongoDB
async function insertStudentData(studentData) {
  const client = new MongoClient(url, { useNewUrlParser: true,
useUnifiedTopology: true });
  try {
    await client.connect();
    const db = client.db(dbName);
```

```
        const result = await db.collection('students').insertOne(studentData);
        console.log('Student data inserted with _id:', result.insertedId);
    } catch (err) {
        console.error('Error inserting data:', err);
    } finally {
        await client.close();
    }
}

// Main function to run the script
(async function() {
    const studentData = getStudentData();
    await insertStudentData(studentData);
})

();
```

Open another terminal

use studentsDB

db.students.find()

```
schoolDB> db.students.find()
[
  {
    _id: ObjectId('6667ea52b1b532f0d4ca81af'),
    usn: '1bi12',
    name: 'Abhi',
    sem: '4',
    year_of_admission: '2022'
  },
  {
    _id: ObjectId('6667eb70ca3e7741855c42e3'),
    usn: '1bi39',
    name: 'RAM',
    sem: '4',
    year_of_admission: '2022'
  }
]
schoolDB>
```

Program 4b.

4b. For a partial name given in node js, search all the names from mongodb student documents created in Question(a)

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

// Create readline interface for user input
const rl = readline.createInterface({
    input: process.stdin,
    output: process.stdout
});

// Connection URI
const uri = 'mongodb://localhost:27017';

async function main() {
    const client = new MongoClient(uri);

    try {
        // Connect to MongoDB
        await client.connect();

        // Ask user for partial name
        rl.question('Enter partial name to search: ', async (partialName) => {
            // Select the database
            const db = client.db('schoolDB'); // Replace 'studentsDB' with your database name

            // Get the students collection
            const collection = db.collection('students'); // Replace 'students' with your collection
name

            // Search for documents with names containing the partial name
```

```
        const query = { name: { $regex: partialName, $options: 'i' } }; // Case-insensitive
regex
        const students = await collection.find(query).toArray();

        // Print the matching student documents
        console.log(`Matching student documents with partial name "${partialName}":`);
        console.log(students);

        // Close the connection
        await client.close();

        // Close readline interface
        rl.close();
      });
  } catch (error) {
      console.error('Error:', error);
      rl.close();
  }
}

main();
```

EXPLANATAION

Imports: Imports mongodb for MongoDB operations and readline for reading user input from the console.

Readline Interface: Sets up a readline interface to handle user input.

MongoDB Connection URI: Defines the URI to connect to the MongoDB server.

Main Function:

- Connects to MongoDB using the MongoClient.
- Prompts the User to enter a partial name to search.
- Selects the Database and Collection (schoolDB and students).
- Constructs a Query to find documents with names containing the partial name using a case-insensitive regex.
- Fetches and Prints Matching Documents.
- Closes the MongoDB Connection and Readline Interface.

OUTPUT:

```
D:\MERN>node 4b.js
Enter partial name to search: RA
Matching student documents with partial name "RA":
[
  {
    _id: new ObjectId('6667eb70ca3e7741855c42e3'),
    usn: '1bi39',
    name: 'RAM',
    sem: '4',
    year_of_admission: '2022'
  }
]

D:\MERN>node 4b.js
Enter partial name to search: HI
Matching student documents with partial name "HI":
[
  {
    _id: new ObjectId('6667ea52b1b532f0d4ca81af'),
    usn: '1bi12',
    name: 'Abhi',
    sem: '4',
    year_of_admission: '2022'
  }
]

D:\MERN>node 4b.js
Enter partial name to search: I
Matching student documents with partial name "I":
[
  {
    _id: new ObjectId('6667ea52b1b532f0d4ca81af'),
    usn: '1bi12',
    name: 'Abhi',
    sem: '4',
    year_of_admission: '2022'
  }
]
```