

- a. Read the data of a student containing usn, name, sem, year\_of\_admission from node js and store it in the mongodb
- b. For a partial name given in node js, search all the names from mongodb student documents created in Question(a)

### Storing in mongodb script

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

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

// Database Name
const dbName = 'school';

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

async function main() {
  try {
    // Connect the client to the server
    await client.connect();

    console.log("Connected successfully to server");

    // Specify the database to use
    const db = client.db(dbName);

    // Specify the collection to use
    const collection = db.collection('students');

    // Student data to be inserted
    const studentData = {
      usn: '1MS20CS001',
      name: 'John Doe',
      sem: 6,
      year_of_admission: 2020
    };
  }
}
```

```

    // Insert the student data into the collection

    const insertResult = await collection.insertOne(studentData);

    console.log('Inserted document:', insertResult.ops);
  } catch (err) {

    console.error(err);
  } finally {

    // Close the connection

    await client.close();

  }
}

// Run the main function
main().catch(console.error);

```

## Searching script

```

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

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

// Database Name
const dbName = 'school';

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

async function searchStudentsByName(partialName) {
  try {

    // Connect the client to the server

    await client.connect();

    console.log("Connected successfully to server");

    // Specify the database to use

    const db = client.db(dbName);
  }
}

```

```
// Specify the collection to use

const collection = db.collection('students');

// Create a case-insensitive regex for the partial name

const regex = new RegExp(partialName, 'i');

// Search for students with names that match the regex

const results = await collection.find({ name: regex }).toArray();

// Print the search results

console.log('Search results:', results);

} catch (err) {

  console.error('Error occurred:', err);

} finally {

  // Close the connection

  await client.close();

}

}

// Example search

const partialName = 'John'; // Change this to test other partial names

searchStudentsByName(partialName).catch(console.error);
```

## Optional Combined Code for 4a and 4b.

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

// Connection URL

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

// Database Name

const dbName = 'school';

// Create a new MongoClient

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

async function main() {

  try {

    // Connect the client to the server

    await client.connect();

    console.log("Connected successfully to server");

    // Specify the database to use

    const db = client.db(dbName);

    // Specify the collection to use

    const collection = db.collection('students');

    // Student data to be inserted

    const studentData = {

      usn: '1MS20CS001',

      name: 'John Doe',

      sem: 6,

      year_of_admission: 2020

    };

    // Insert the student data into the collection

    const insertResult = await collection.insertOne(studentData);

    console.log('Inserted document:', insertResult.ops);
```

```
// Function to search for names based on a partial match

async function searchStudentsByName(partialName) {

    const regex = new RegExp(partialName, 'i'); // Create a case-insensitive regex

    const results = await collection.find({ name: regex }).toArray();

    return results;

}

// Example search

const partialName = 'ABC'; // Change this to test other partial names

const searchResults = await searchStudentsByName(partialName);

console.log('Search results:', searchResults);

} catch (err) {

    console.error('Error occurred:', err);

} finally {

    // Close the connection

    await client.close();

}

}

// Run the main function

main().catch(console.error);
```