Exercise 2: JSON Fundamentals & MongoDB CRUD Operations in JavaScript

Objective: Learn JSON syntax and MongoDB CRUD operations in Node.js.

Lab Overview

Key Topics

1. **JSON Operations**: Syntax, parsing, validation.
2. **MongoDB CRUD in Node.js**: Insert, query, update, delete drivers and rides.

Deliverables

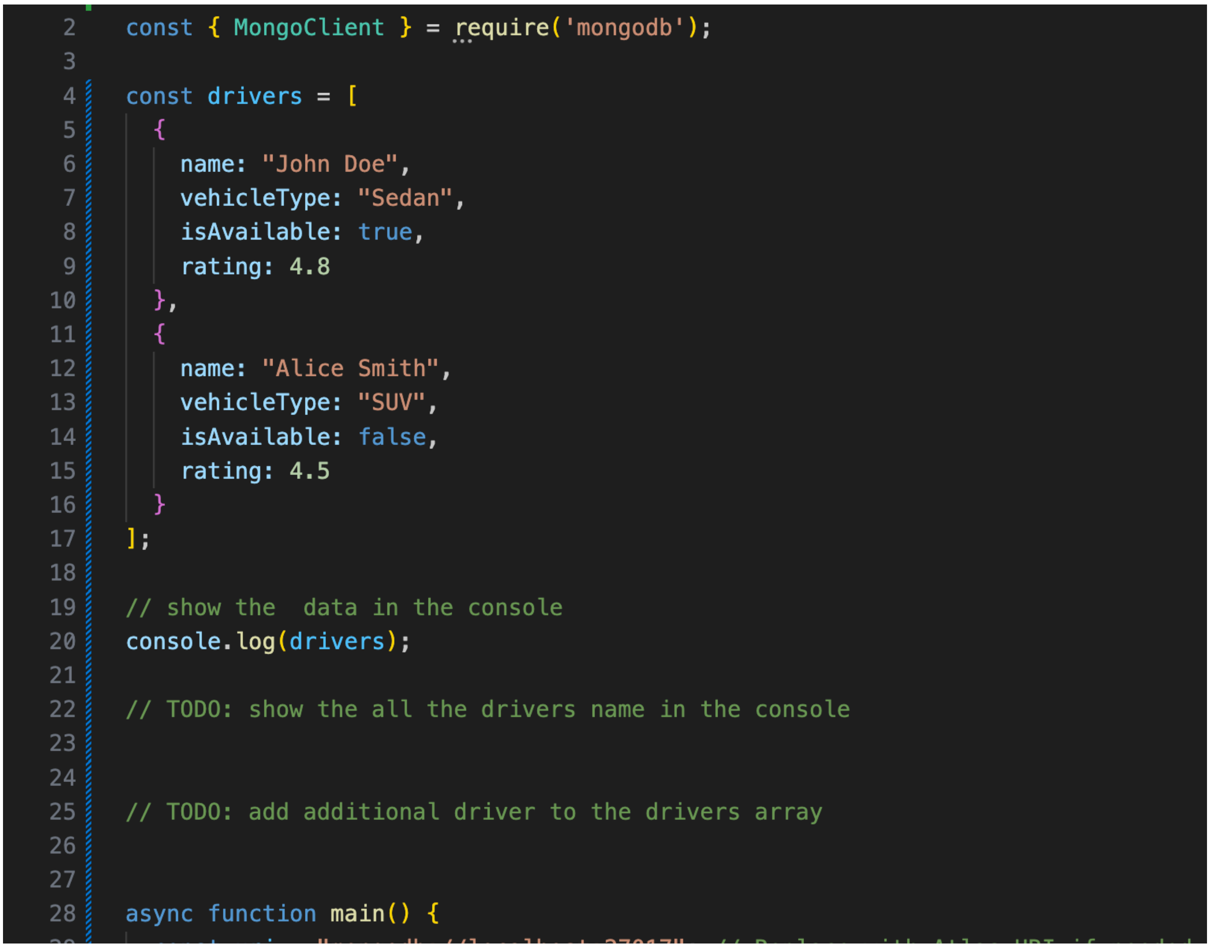
1. JSON file with driver/ride data.
2. Node.js scripts for JSON validation and MongoDB CRUD.
3. Answers to hands-on questions

Lab Procedures

Part 1: Declare and Validate Driver Data

Task 1: Define Drivers as a JavaScript Variable

1. Update the script `index.js` from previous project with an array of drivers objects:



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

const drivers = [
  {
    name: "John Doe",
    vehicleType: "Sedan",
    isAvailable: true,
    rating: 4.8
  },
  {
    name: "Alice Smith",
    vehicleType: "SUV",
    isAvailable: false,
    rating: 4.5
  }
];

// show the  data in the console
console.log(drivers);

// TODO: show the all the drivers name in the console


// TODO: add additional driver to the drivers array


async function main() {
  const uri = "mongodb://localhost:27017"; // Replace with Atlas URI if needed
```

Task 2: JSON Data Operations

1. Read and display the driver’s name into the console
   https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach

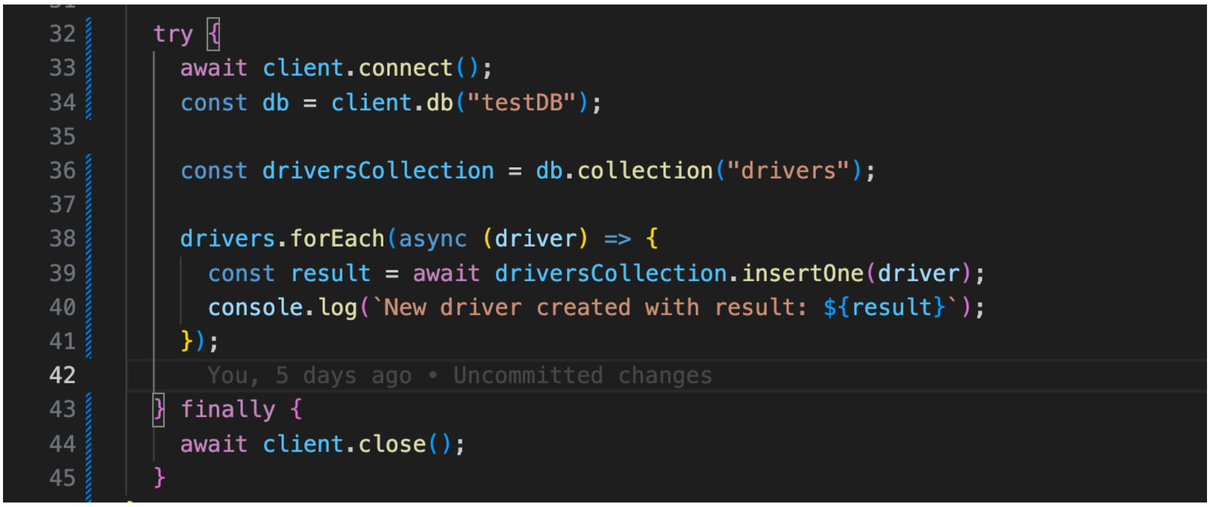
2. Add a new driver directly in the array
   https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/push

---

Part 2: MongoDB CRUD Operations

Task 3: Insert Drivers into MongoDB

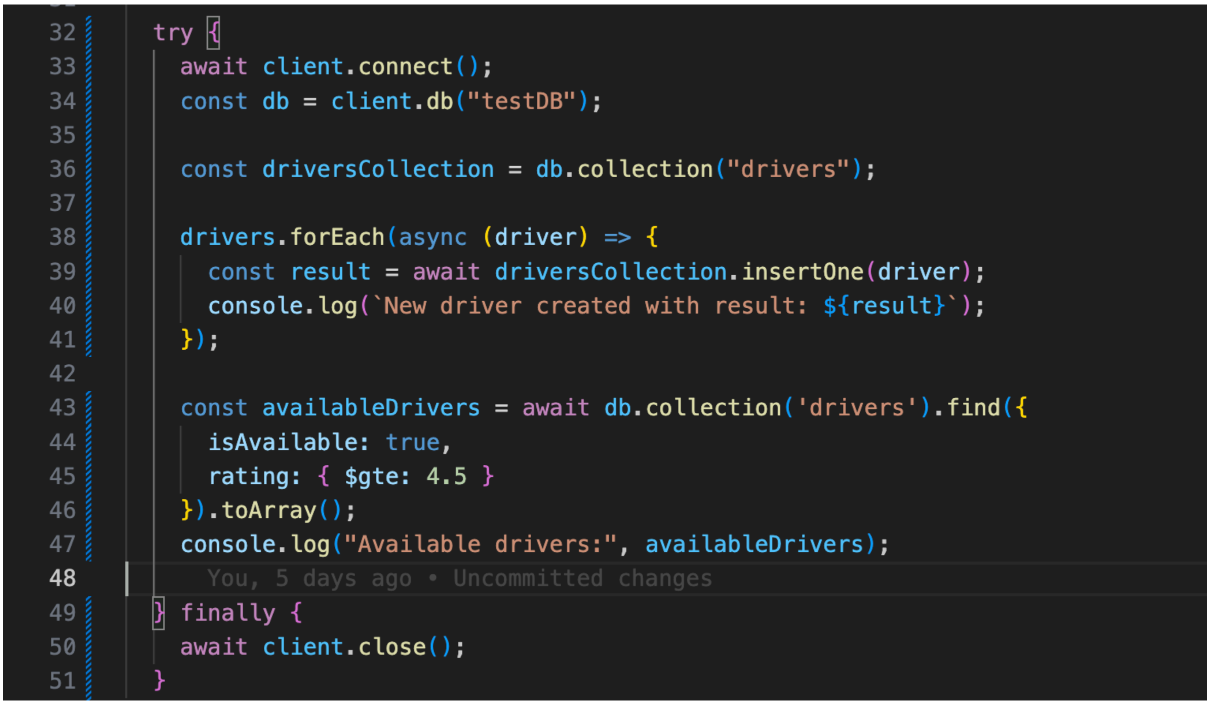
1. Continue editing the index.js to insert the all drivers array into the database
   https://www.mongodb.com/docs/manual/reference/method/db.collection.insertOne/



```
  try {
    await client.connect();
    const db = client.db("testDB");

    const driversCollection = db.collection("drivers");

    drivers.forEach(async (driver) => {
      const result = await driversCollection.insertOne(driver);
      console.log(`New driver created with result: ${result}`);
    });
      You, 5 days ago • Uncommitted changes
  } finally {
    await client.close();
  }
```

Task 4: Query and Update Drivers

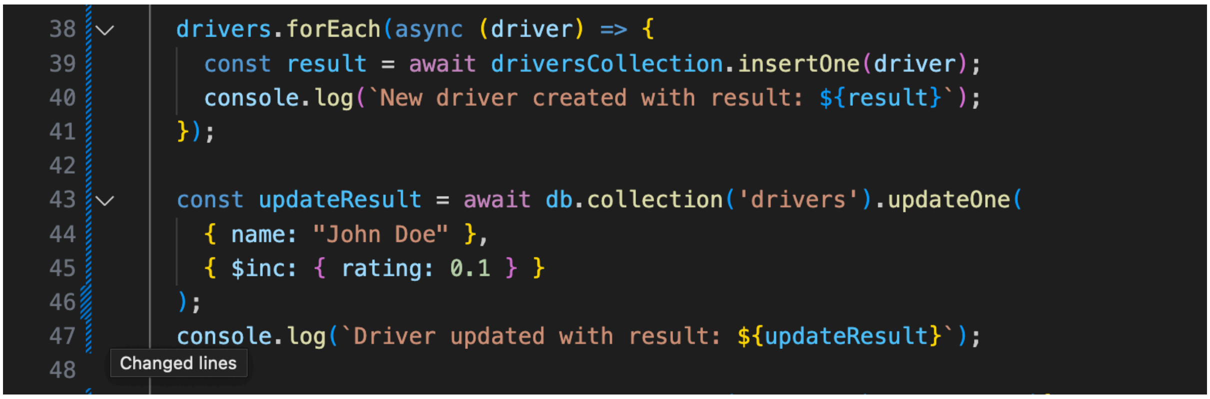
1. Then, add the codes to find all **available drivers with rating ≥ 4.5**:



```
  try {
    await client.connect();
    const db = client.db("testDB");

    const driversCollection = db.collection("drivers");

    drivers.forEach(async (driver) => {
      const result = await driversCollection.insertOne(driver);
      console.log(`New driver created with result: ${result}`);
    });

    const availableDrivers = await db.collection('drivers').find({
      isAvailable: true,
      rating: { $gte: 4.5 }
    }).toArray();
    console.log("Available drivers:", availableDrivers);
      You, 5 days ago • Uncommitted changes
  } finally {
    await client.close();
  }
```

Task 5: Update

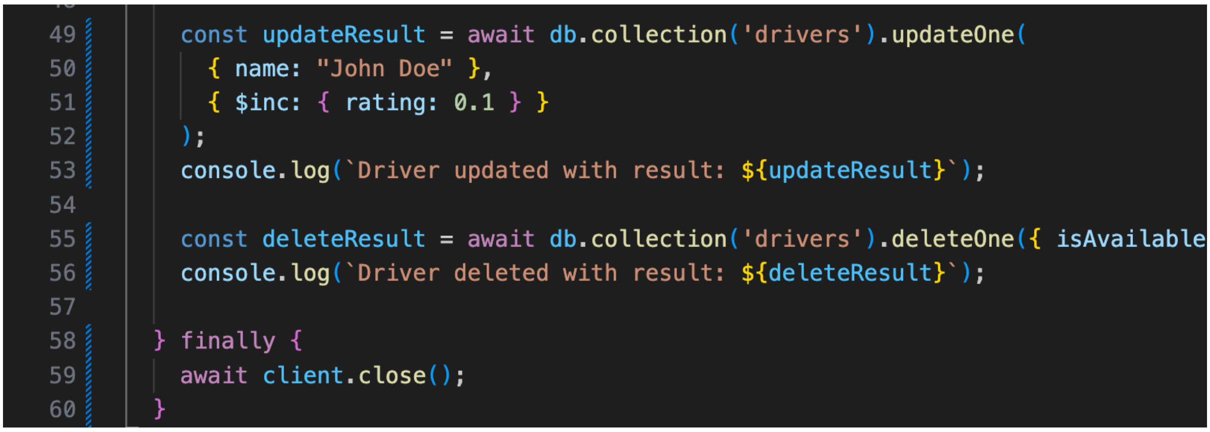
1. Then, **increase John Doe's rating by 0.1**: https://www.mongodb.com/docs/manual/reference/method/db.collection.updateOne/



```
    drivers.forEach(async (driver) => {
      const result = await driversCollection.insertOne(driver);
      console.log(`New driver created with result: ${result}`);
    });

    const updateResult = await db.collection('drivers').updateOne(
      { name: "John Doe" },
      { $inc: { rating: 0.1 } }
    );
    console.log(`Driver updated with result: ${updateResult}`);
```

Task 6: Delete

1. Lastly, edit the program to remove **unavailable drivers**: https://www.mongodb.com/docs/manual/reference/method/db.collection.deleteOne/



```
    const updateResult = await db.collection('drivers').updateOne(
      { name: "John Doe" },
      { $inc: { rating: 0.1 } }
    );
    console.log(`Driver updated with result: ${updateResult}`);

    const deleteResult = await db.collection('drivers').deleteOne({ isAvailable
    console.log(`Driver deleted with result: ${deleteResult}`);

  } finally {
    await client.close();
  }
```

Lab Questions

Answer by completing the tasks above and observing results.

JSON Questions

1. Explain what is CRUD operations and how it is relates to the mongo functions in the exercise.
2. Identify all the mongo operators used in the exercise, then explain the usage for each.
3. Replace the mongo functions in Task 5 to updateMany instead of updateOne, compare the difference based on the result in console and the mongo compass.
4. Replace the mongo functions in Task 6 to deleteMany instead of deleteOne, compare the difference based on the result in console and the mongo compass.

Submission Requirements

1. GitHub repository with:
    - Node.js scripts.
    - Screenshots of MongoDB operations.
2. Lab report with answers to all questions.