Github Link:

https://github.com/charansankara/Cloudsystemclass

index.js

C: > Users > Charan > Desktop > cloudss > Cloudsystemclass > index.js > ...

```
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
        }
    ];
    console.log(drivers);

async function main(){
    const uri = "mongodb://localhost:27017"
    const client = new MongoClient(uri);

try{

```

```
}
PS C:\Users\Charan\Desktop\cloudss\Cloudsystemclass> node index.js
[
  {
    name: 'John Doe',
    vehicletype: 'Sedan',
    isAvailable: true,
    rating: 4.8
  },
  {
    name: 'Alice Smith',
    vehicletype: 'SUV',
    isAvailable: false,
    rating: 4.5
  }
]
Connected to MongoDB!
Document Inserted!
Query result: {
  _id: new ObjectId('68e73a8f25a5376aad7b8ef5'),
  name: 'Charan',
  age: 25
}
PS C:\Users\Charan\Desktop\cloudss\Cloudsystemclass>
```

Try it

JavaScript Demo: Array.prototype.forEach()

```
1 const array = ["John Doe", "Alice Smith"];
2
3 array.forEach((element) => console.log(element));
4
```

Run

Reset

```
> "John Doe"
> "Alice Smith"
```

Try it

JavaScript Demo: Array.prototype.push()

```
1 const drivers = ["name: John Boy", "vehicletype: Sedan"];
2 const total = drivers.push("isAvailable: true", "rating: 4.8");
3
4
5 console.log(drivers); // ['soccer', 'baseball', 'football', 'swimming']
6 console.log(total); // 4
```

Run

Reset

```
> Array ["name: John Boy", "vehicletype: Sedan", "isAvailable: true", "rating: 4.8"]
> 4
```

Task 3

```
_id: ObjectId('690241edcba8239dd926f9f4')
name : "John Doe"
vehicletype : "Sedan"
isAvailable : true
rating : 4.8
```

```
_id: ObjectId('690241edcba8239dd926f9f5')
name : "Alice Smith"
vehicletype : "SUV"
isAvailable : false
rating : 4.5
```

Task 4

```
Connected to MongoDB!
New driver created with result: [object Object]
New driver created with result: [object Object]
Available drivers [
  {
    _id: new ObjectId('6902c4f4716685ade8a9513c'),
    name: 'John Doe',
    vehicletype: 'Sedan',
    isAvailable: true,
    rating: 4.8
  },
  {
    _id: new ObjectId('6902c4f4716685ade8a9513d'),
    name: 'Alice Smith',
    vehicletype: 'SUV',
    isAvailable: false,
    rating: 4.6
  }
]
Document Inserted!
Query result: {
  _id: new ObjectId('68e73a8f25a5376aad7b8ef5'),
  name: 'Charan',
  age: 25
```

Task 5

```
Connected to MongoDB!
New driver created with result: [object Object]
New driver created with result: [object Object]
Driver updated with results: [object Object]
Available drivers [
  {
    _id: new ObjectId('6902c6c414817eb815cbc81b'),
    name: 'John Doe',
    vehicletype: 'Sedan',
    isAvailable: true,
    rating: 4.8999999999999995
  },
  {
    _id: new ObjectId('6902c6c414817eb815cbc81c'),
    name: 'Alice Smith',
    vehicletype: 'SUV',
    isAvailable: false,
    rating: 4.6
  }
]
Document Inserted!
Query result: {
  _id: new ObjectId('68e73a8f25a5376aad7b8ef5'),
  name: 'Charan',
  age: 25
}
```

Task 6:

```
  }
]
Connected to MongoDB!
New driver created with result: [object Object]
New driver created with result: [object Object]
Driver deleted with results:[object Object]
Driver updated with results: [object Object]
Available drivers [
  {
    _id: new ObjectId('6902c8294f28ac26fc1c0ecc'),
    name: 'John Doe',
    vehicletype: 'Sedan',
    isAvailable: true,
    rating: 4.8999999999999995
  }
]
Document Inserted!
Query result: {
  _id: new ObjectId('68e73a8f25a5376aad7b8ef5'),
  name: 'Charan',
  age: 25
}
```

Lab Questions:

1. Explain what is CRUD operations and how it is relates to the mongo functions in the exercise.
   = CRUD operations stand for Create, Read, Update and Delete. These are operations that are require to interact and manage the data in any Database.

2. Identify all the mongo operators used in the exercise, then explain the usage for each.
   = $inc and $gte

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.
   = Previously with update One only the first document with name: "John Doe" will have its rating incremented by 0.1. But with updateMany all documents with name: "John Doe" in the collection will have their rating incremented by 0.1.

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.
   = Previously with deleteOne only the first document where isAvailable is false will be removed from the collection. But with deleteMany all documents in the collection where isAvailable is false will be permanently removed.