PRACTICAL 8

Aim: **Write a nodejs/MongoDbB code to implement CRUD operation**

server.js

```
const express = require('express');
const mongoose = require('mongoose');
// require('dotenv').config();
const path = require('path');
const config=require('config');

const app = express();

const db= config.get('mongoURI');

mongoose
  .connect(db,{useNewUrlParser: true})
  .then(()=> console.log("mongodb connected..."))
  .catch(err=>console.log(err));

const student = require('./models/db.js');

app.get("/db-create_table", (req, res) => {

  try{
    var result=student;
    console.log("Student Table created successfully ",result);
  }
  catch(error)
  {
    console.error(err)
  }
  res.send("Employee table created");
});

app.get("/db-insert",(req,res)=> {
        student.insertMany([
          {
            name:'Jeel',
            stdId:'19ce091',

            mobileno:'6358862545'
        },
        {
          name:'Neel',
          stdId:'19ce092',
     mobileno:'9358822545'
   },
   {
    name:'Vivek',
    stdId:'19ce074',
    mobileno:'9354563553'
  },
  {
   name:'Rahul',
   stdId:'19ce095',
   mobileno:'9826147892'
 }

    ],(err,data)=>{
     if(err!=null){
      console.log(err);
     }
      else
      console.groupCollapsed("DataInserted\n",JSON.stringify(data));

    })
});

app.get("/db-update",(req,res)=>{
 student.updateMany({stdId:'19ce091'},{$set:{name:'Jeel Parmar'}},(err,data)=
>{
  if(!err){
   console.log("Updated Data :/n",data);
  }
  else{
   console.log(err);
  }
 })
});

app.get("/db-delete",(req,res)=>{
student.deleteOne({stdId:'19ce091'},(err,data)=>{
 if(!err){
  console.log("Data deleted :/n",data);
 }
 else{
  console.log(err);
 }
})
});


app.get("/db-display",(req,res)=>{
  student.find({},(err,data)=>{
   if(!err){
    console.log("Fetched datas:\n",data);
   }
   else{
    throw err;
   }
  })
})


app.listen(3000,()=>{
 console.log('Server running on port 3000')
})
```

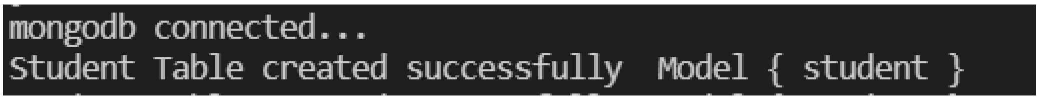
db.js

```
const mongoose = require('mongoose');
const Schema=mongoose.Schema;
const studentschema = new Schema({
  name:{
    type: String,
    required: true
  },
  stdId:{
    type: String,
    required: true
  },

  mobileno:{
    type: String,
    required: true,
    maxlength:10
  }


})
module.exports = Item = mongoose.model('student',studentschema);
```

Output:

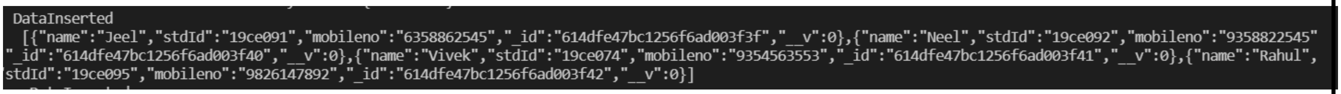
Connect to database:

Table creation:

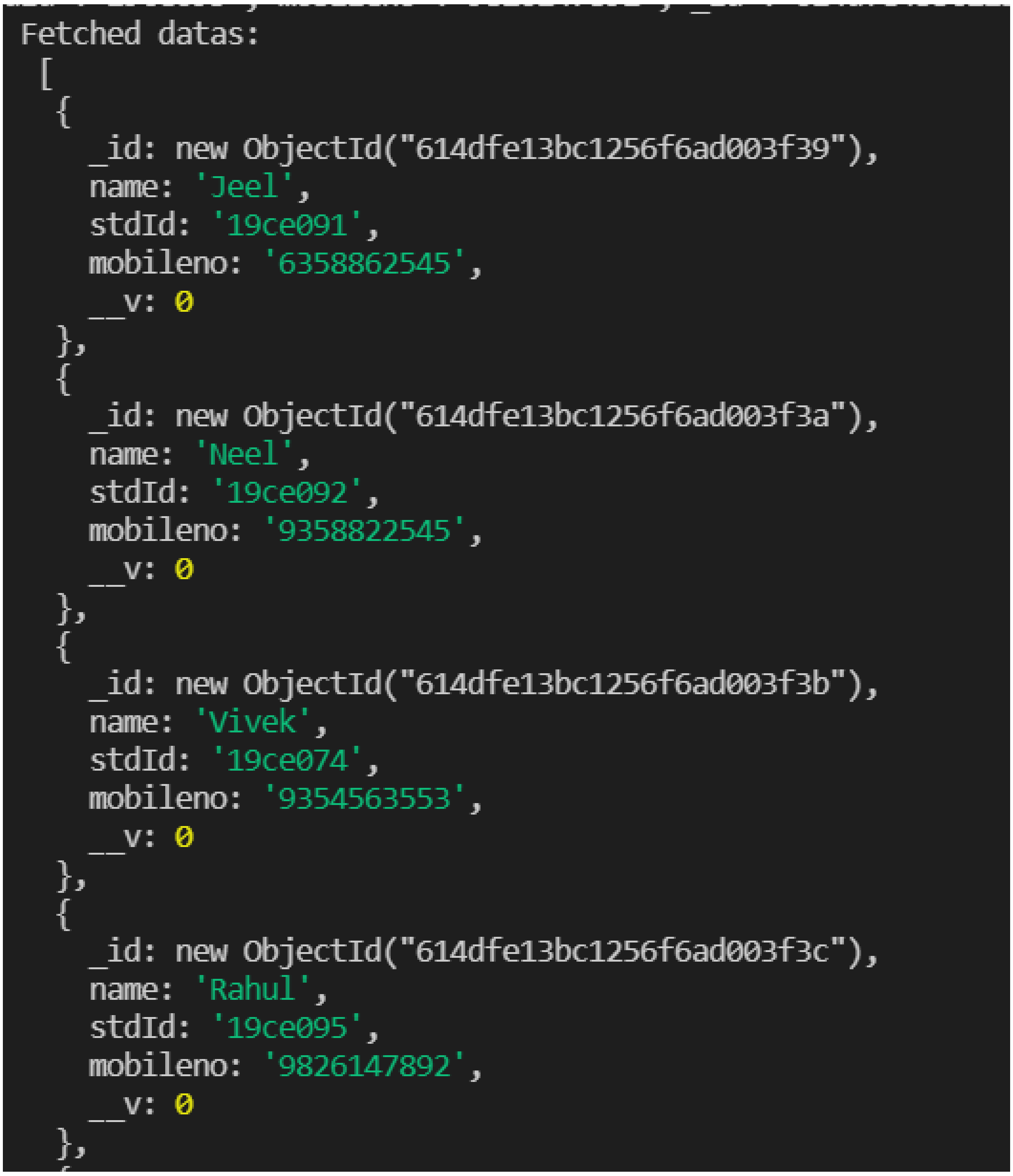


```
mongodb connected...
Student Table created successfully  Model { student }
```

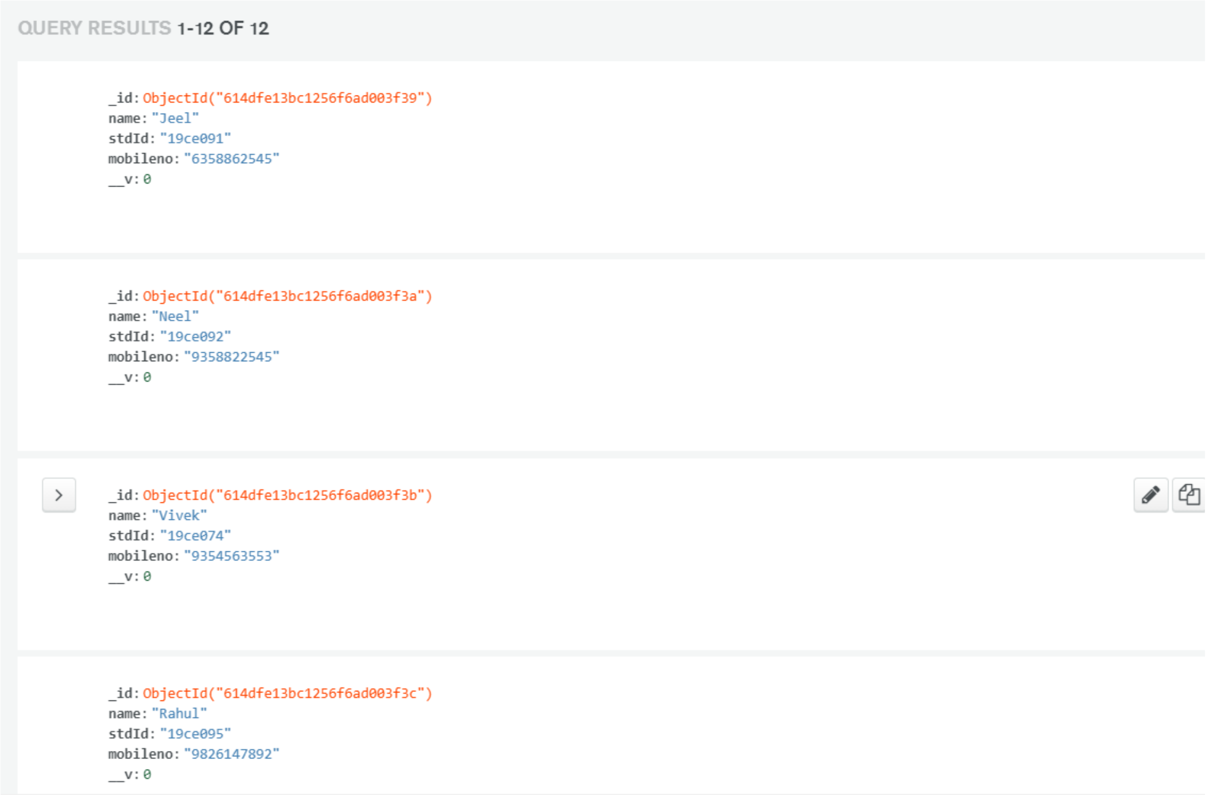
Insert Data



```
 DataInserted
  [{"name":"Jeel","stdId":"19ce091","mobileno":"6358862545","_id":"614dfe47bc1256f6ad003f3f","__v":0},{"name":"Neel","stdId":"19ce092","mobileno":"9358822545"
"_id":"614dfe47bc1256f6ad003f40","__v":0},{"name":"Vivek","stdId":"19ce074","mobileno":"9354563553","_id":"614dfe47bc1256f6ad003f41","__v":0},{"name":"Rahul",
stdId":"19ce095","mobileno":"9826147892","_id":"614dfe47bc1256f6ad003f42","__v":0}]
```



```
Fetched datas:
 [
  {
    _id: new ObjectId("614dfe13bc1256f6ad003f39"),
    name: 'Jeel',
    stdId: '19ce091',
    mobileno: '6358862545',
    __v: 0
  },
  {
    _id: new ObjectId("614dfe13bc1256f6ad003f3a"),
    name: 'Neel',
    stdId: '19ce092',
    mobileno: '9358822545',
    __v: 0
  },
  {
    _id: new ObjectId("614dfe13bc1256f6ad003f3b"),
    name: 'Vivek',
    stdId: '19ce074',
    mobileno: '9354563553',
    __v: 0
  },
  {
    _id: new ObjectId("614dfe13bc1256f6ad003f3c"),
    name: 'Rahul',
    stdId: '19ce095',
    mobileno: '9826147892',
    __v: 0
  },
  {
```



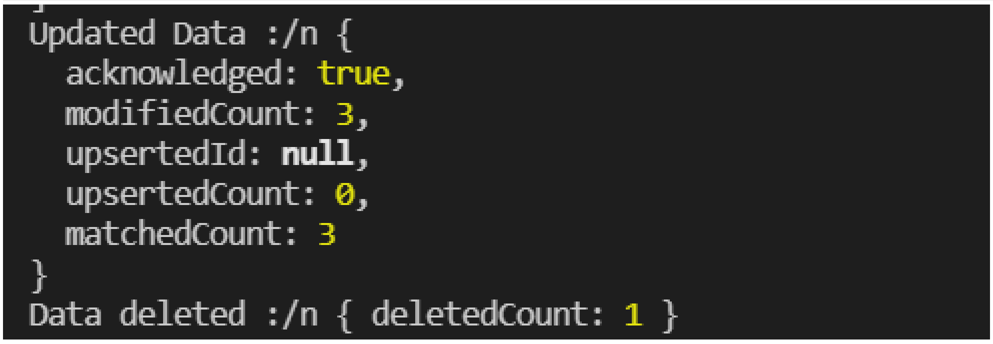
QUERY RESULTS 1-12 OF 12

```
_id: ObjectId("614dfe13bc1256f6ad003f39")
name: "Jeel"
stdId: "19ce091"
mobileno: "6358862545"
__v: 0
```

```
_id: ObjectId("614dfe13bc1256f6ad003f3a")
name: "Neel"
stdId: "19ce092"
mobileno: "9358822545"
__v: 0
```

```
_id: ObjectId("614dfe13bc1256f6ad003f3b")
name: "Vivek"
stdId: "19ce074"
mobileno: "9354563553"
__v: 0
```

```
_id: ObjectId("614dfe13bc1256f6ad003f3c")
name: "Rahul"
stdId: "19ce095"
mobileno: "9826147892"
__v: 0
```



```
Updated Data :/n {
  acknowledged: true,
  modifiedCount: 3,
  upsertedId: null,
  upsertedCount: 0,
  matchedCount: 3
}
Data deleted :/n { deletedCount: 1 }
```