Name: Patel Shivam S.
Enroll no: 21162101019
Sem: 5 Batch: 51 Branch: CBA

Sub: Microservices

Practical 8

AIM: **Implement an application with CRUD operation using MongoDB with NodeJS.**
Sympoms Pvt Ltd company wants to manage their employee records and this task is given to you. They are seeking functionalities:

Practical 8.1: New joiner data should be stored in DB.
Practical 8.2: The admin can see the whole DB,
Practical 8.3: The admin can filter out the DB.
Practical 8.4: The admin can delete the record from DB.
Practical 8.5: Create a database with the name “student”. Create a collection named “cba”. Perform the following tasks:

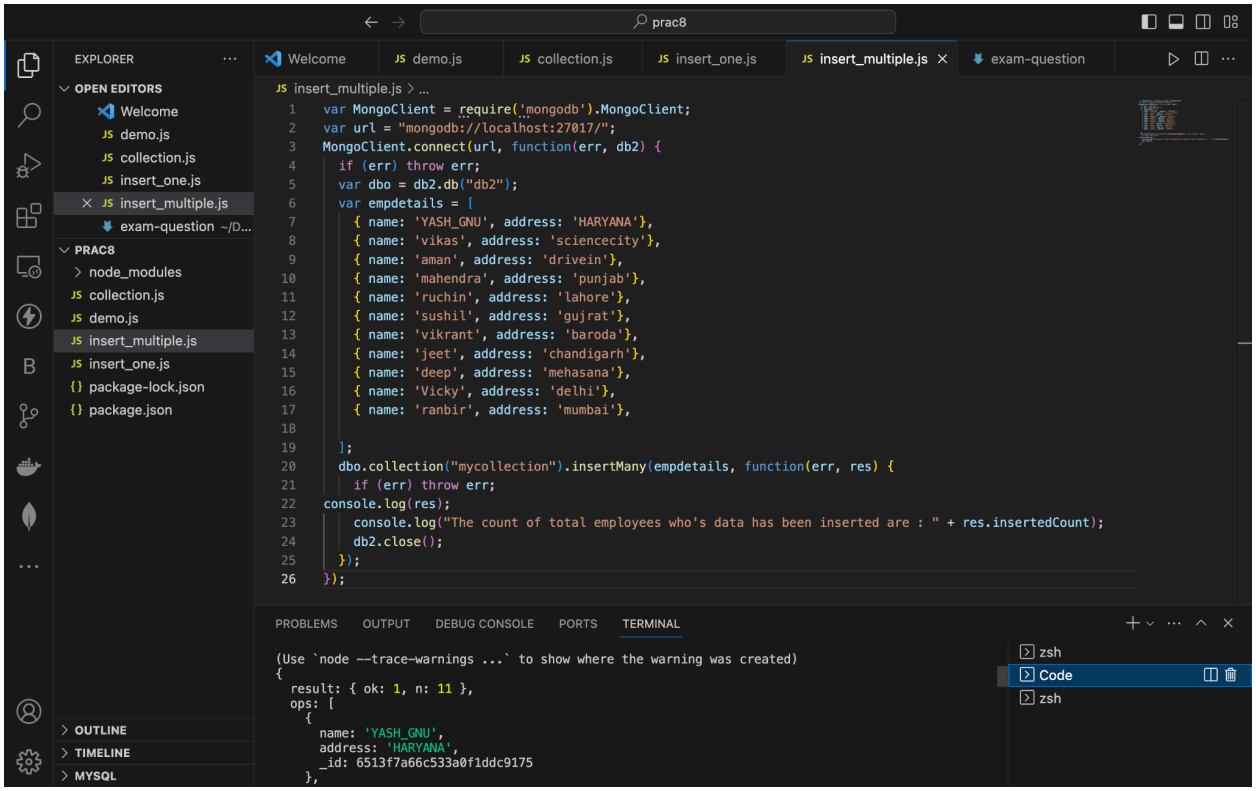
Insert the details of 10 students with the following details: Name, age, address, phone number, and email.

Filter the students whose age is greater than 16.

Delete the student record whose phone number contains the digit ‘6’ in it.

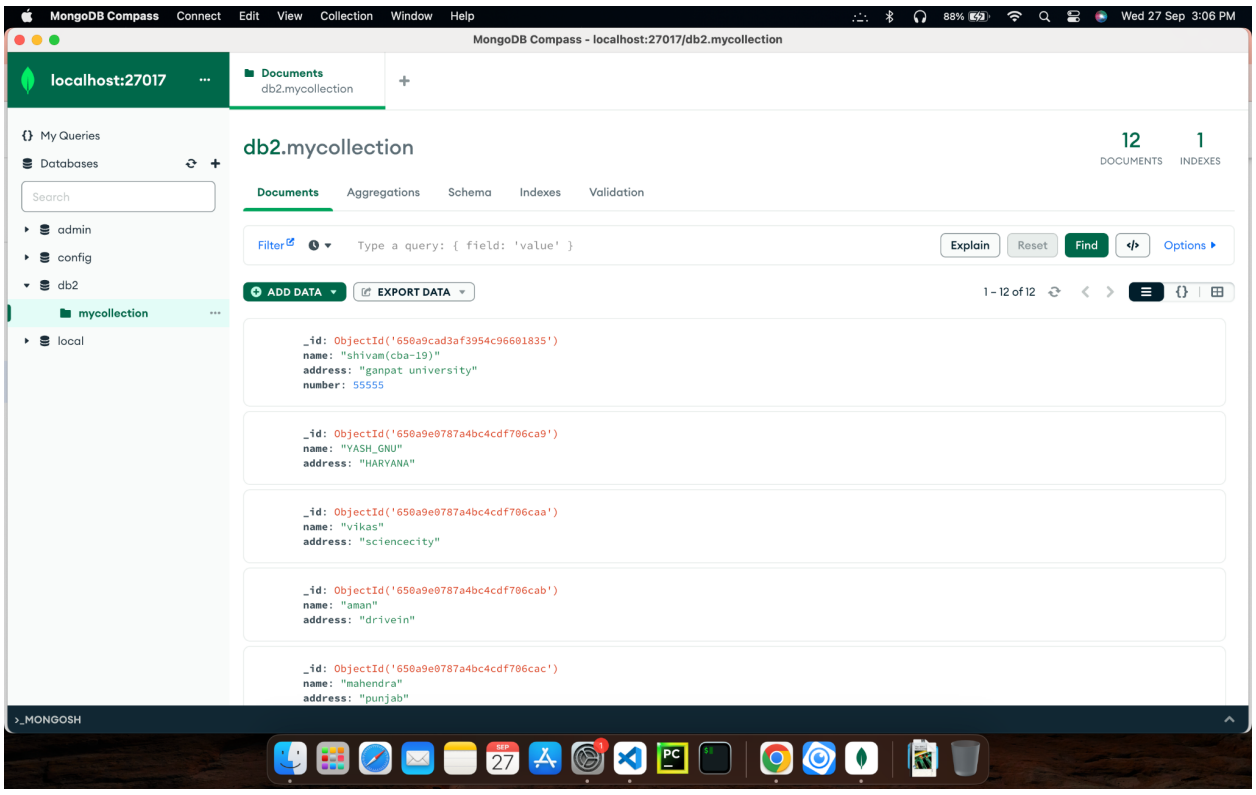
GITHUB LINK:
https://github.com/Shivam3783/microservice_practicals/tree/main/prac8

Practical 8.1: New joiner data should be stored in DB.



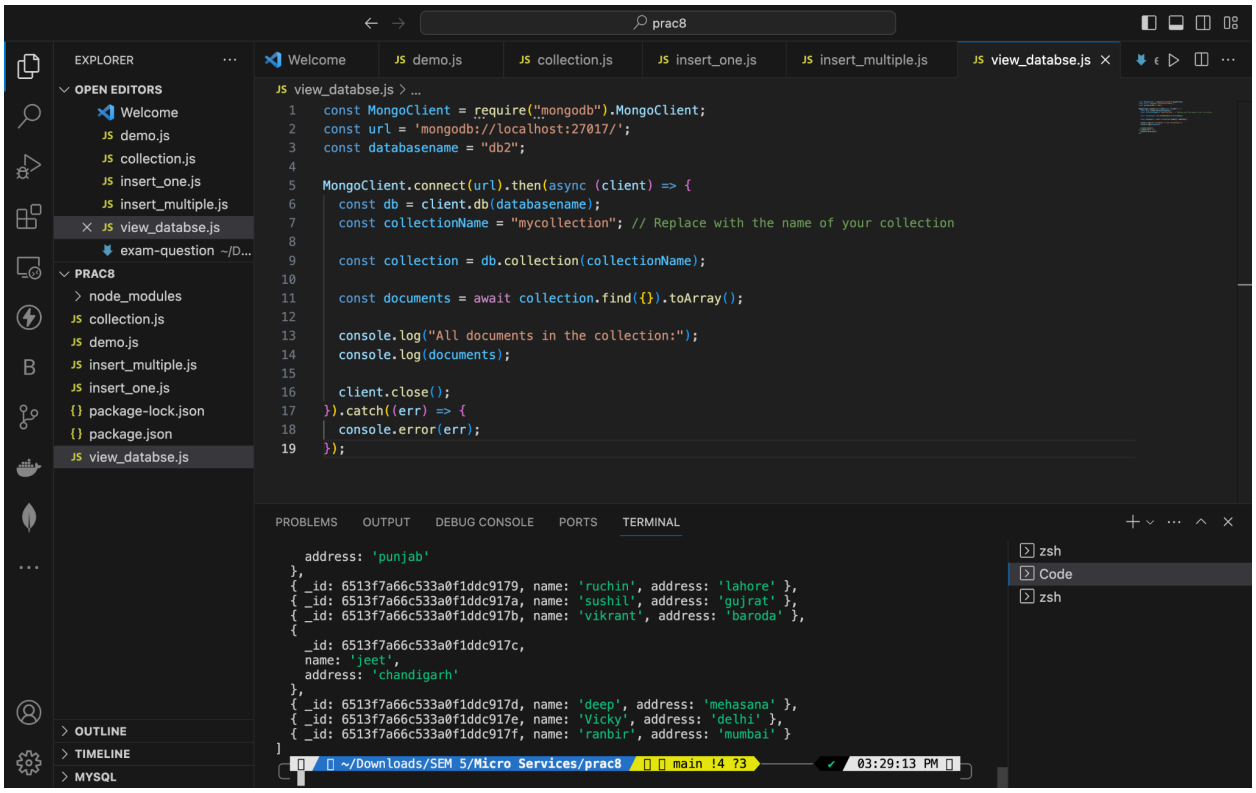
```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";
MongoClient.connect(url, function(err, db2) {
  if (err) throw err;
  var dbo = db2.db("db2");
  var empdetails = [
    { name: 'YASH_GNU', address: 'HARYANA'},
    { name: 'vikas', address: 'sciencecity'},
    { name: 'aman', address: 'drivein'},
    { name: 'mahendra', address: 'punjab'},
    { name: 'ruchin', address: 'lahore'},
    { name: 'sushil', address: 'gujrat'},
    { name: 'vikrant', address: 'baroda'},
    { name: 'jeet', address: 'chandigarh'},
    { name: 'deep', address: 'mehasana'},
    { name: 'Vicky', address: 'delhi'},
    { name: 'ranbir', address: 'mumbai'},

  ];
  dbo.collection("mycollection").insertMany(empdetails, function(err, res) {
    if (err) throw err;
console.log(res);
    console.log("The count of total employees who's data has been inserted are : " + res.insertedCount);
    db2.close();
  });
});
```

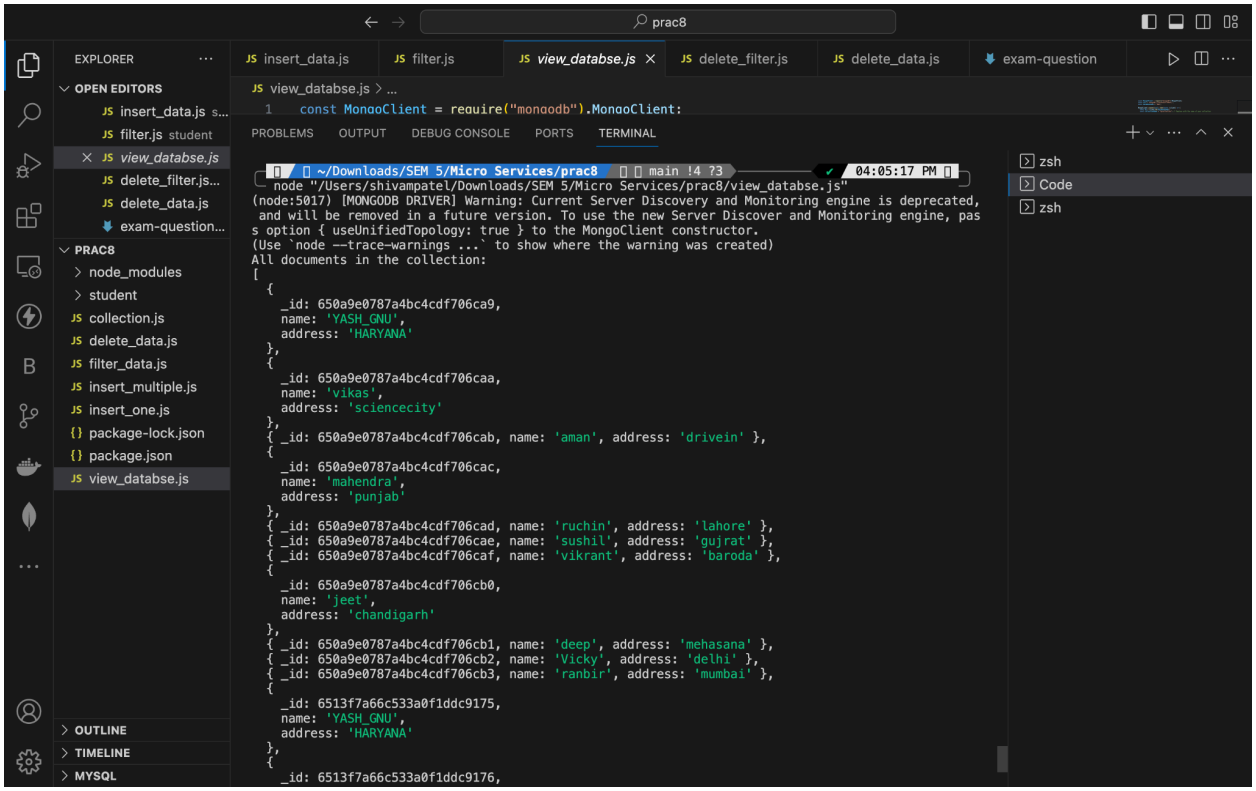
```
(Use `node --trace-warnings ...` to show where the warning was created)
{
  result: { ok: 1, n: 11 },
  ops: [
    {
      name: 'YASH_GNU',
      address: 'HARYANA',
      _id: 6513f7a66c533a0f1ddc9175
    },
```



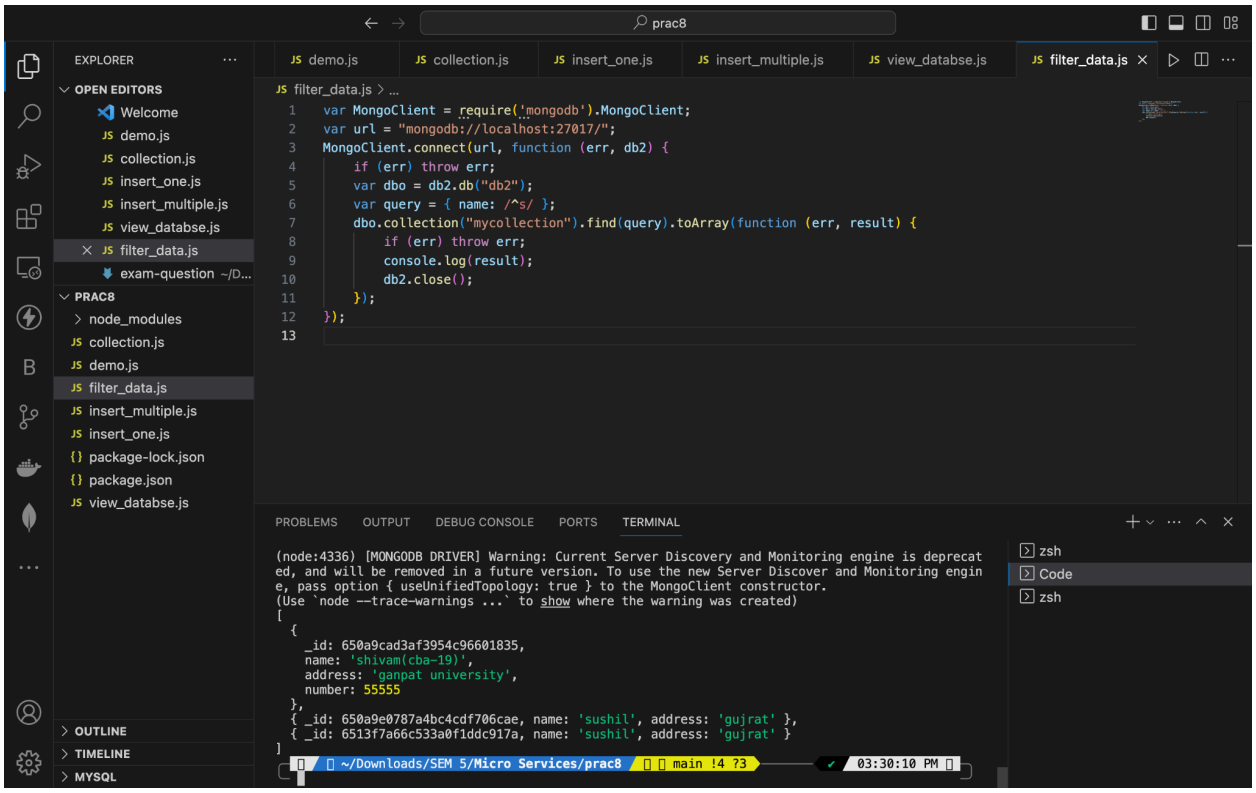
```
_id: ObjectId('650a9cad3af3954c96601835')
name: "shivam(cba-19)"
address: "ganpat university"
number: 55555

_id: ObjectId('650a9e0787a4bc4cdf706ca9')
name: "YASH_GNU"
address: "HARYANA"

_id: ObjectId('650a9e0787a4bc4cdf706caa')
name: "vikas"
address: "sciencecity"

_id: ObjectId('650a9e0787a4bc4cdf706cab')
name: "aman"
address: "drivein"

_id: ObjectId('650a9e0787a4bc4cdf706cac')
name: "mahendra"
address: "punjab"
```

Practical 8.2: The admin can see the whole DB

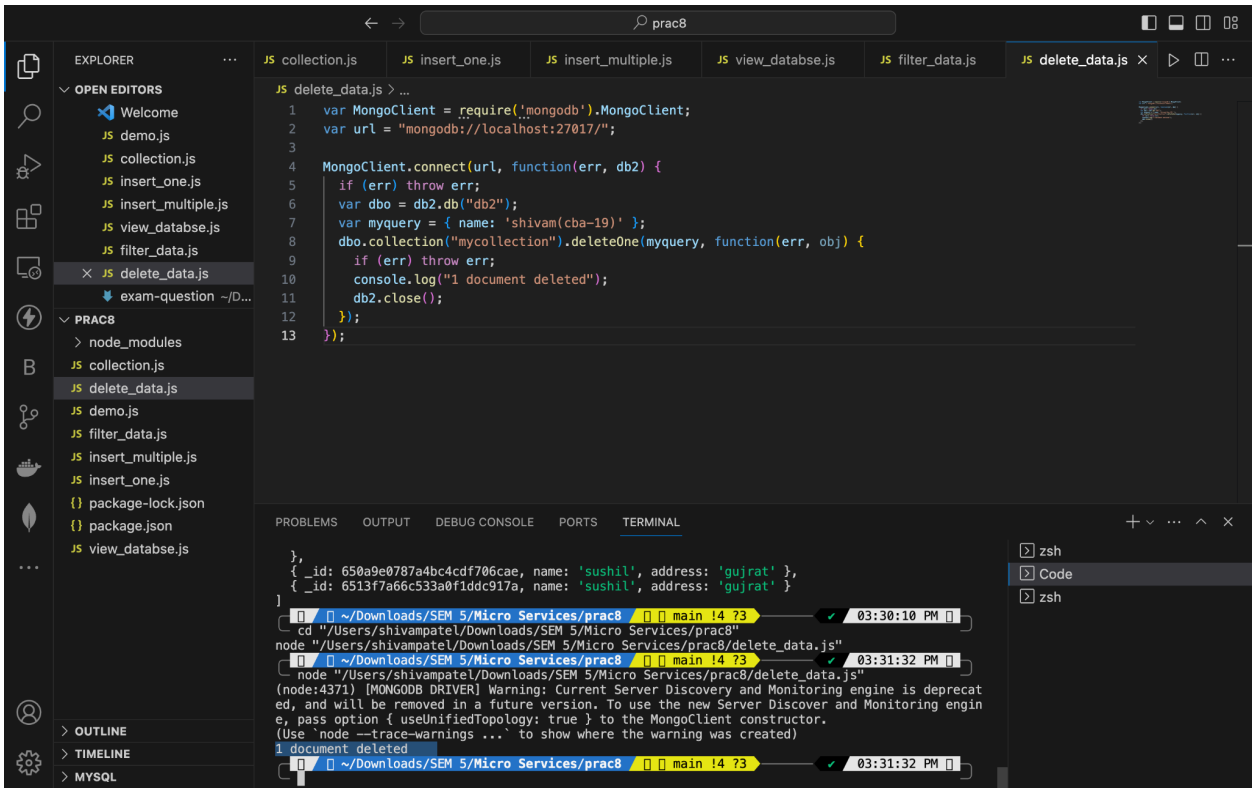




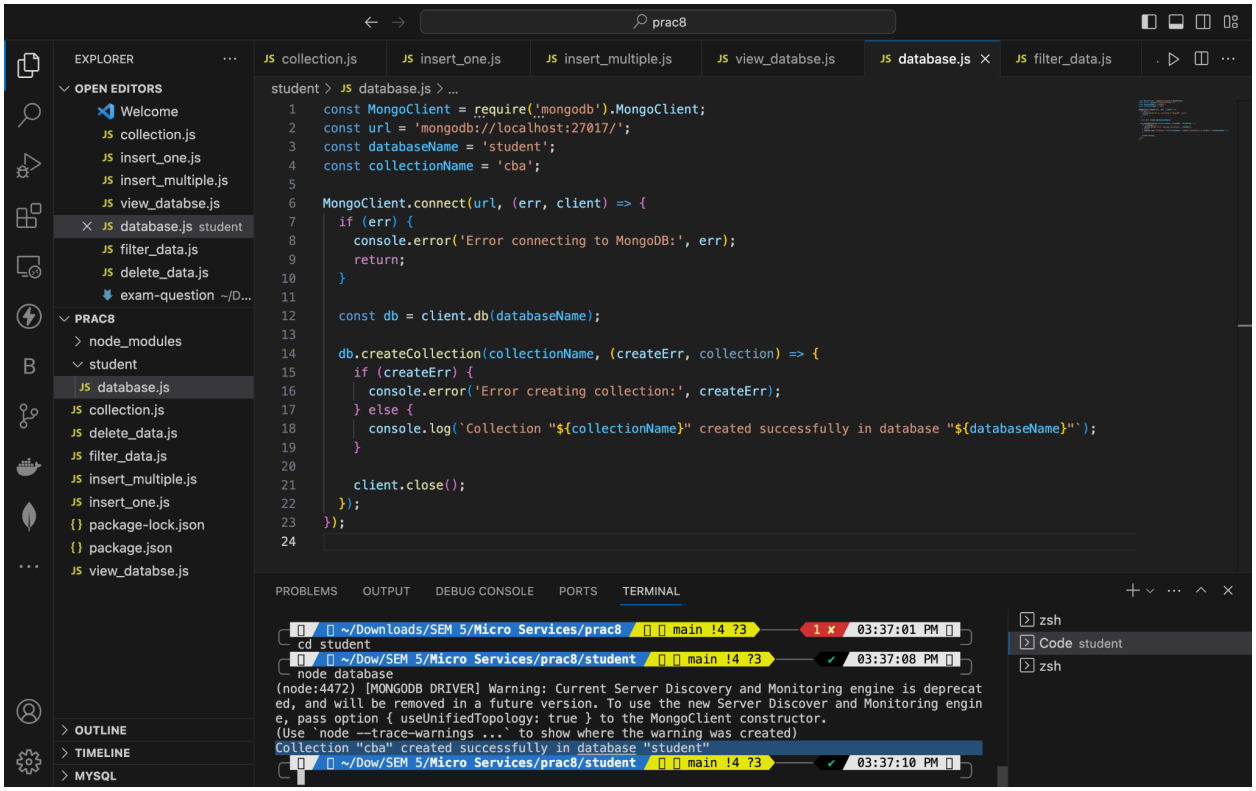
Practical 8.3: The admin can filter out the DB.

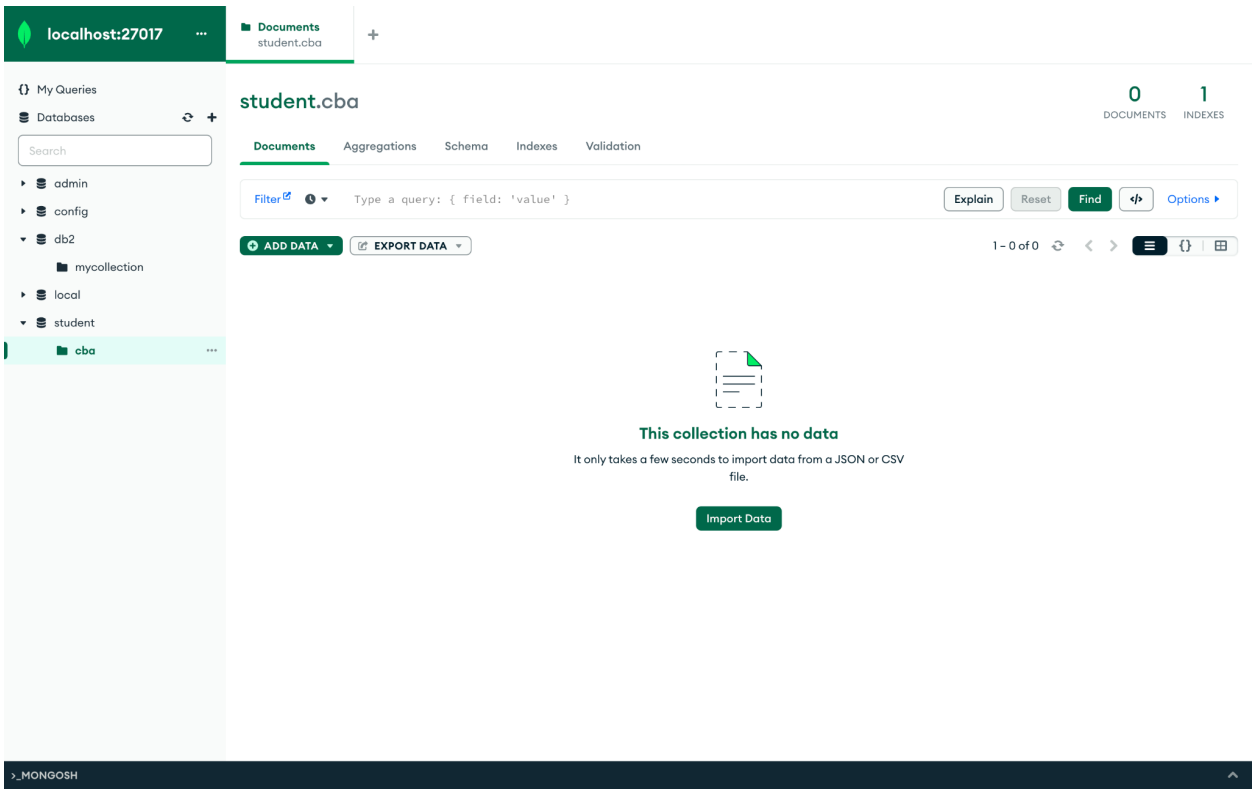


Practical 8.4: The admin can delete the record from DB.

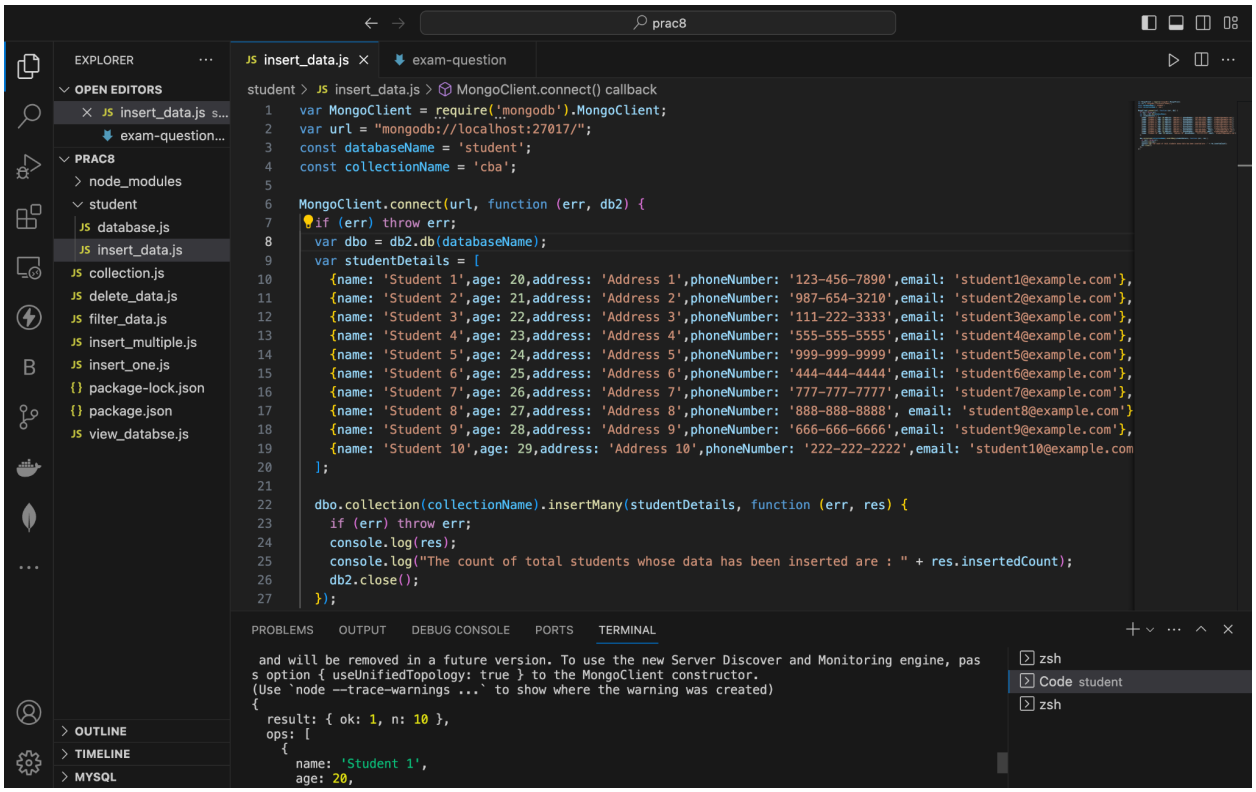


Practical 8.5: Create a database with the name “student”. Create a collection named “cba”. Perform the following tasks:



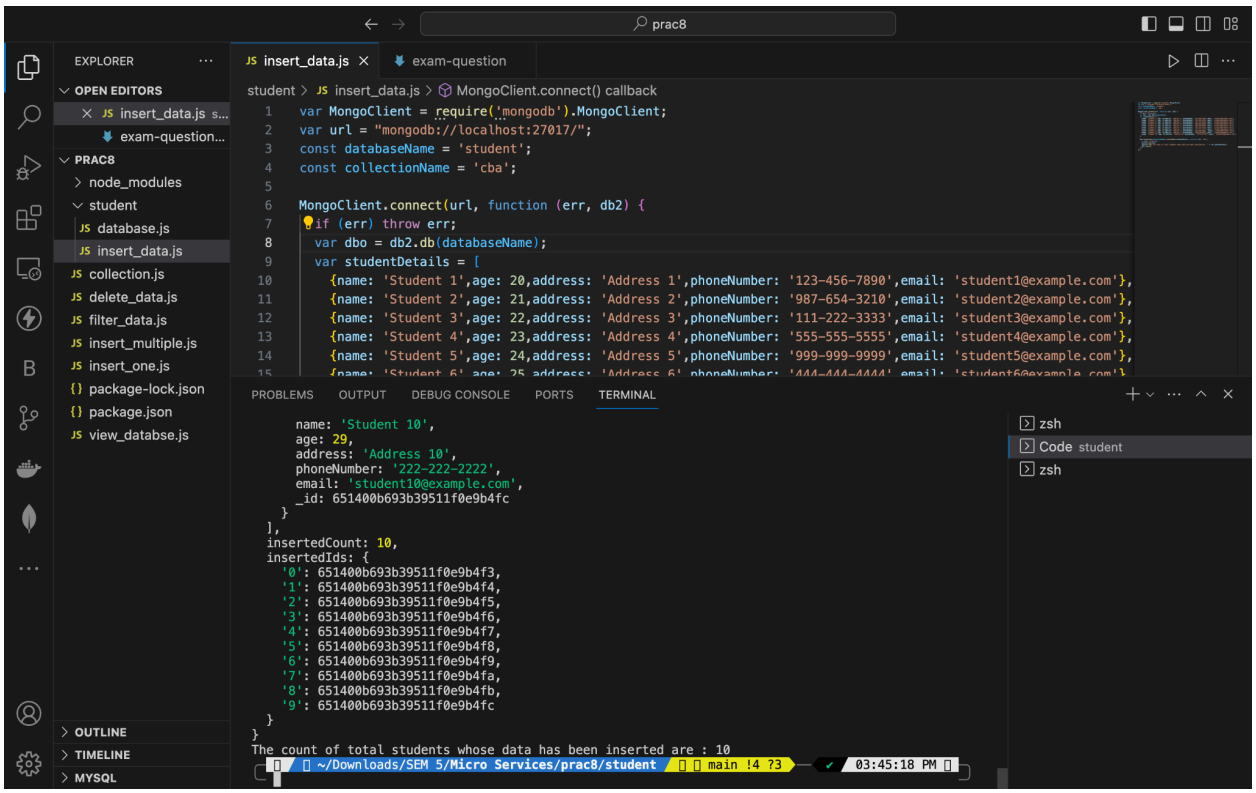


→ Insert the details of 10 students with the following details: Name, age, address, phone number, and email.

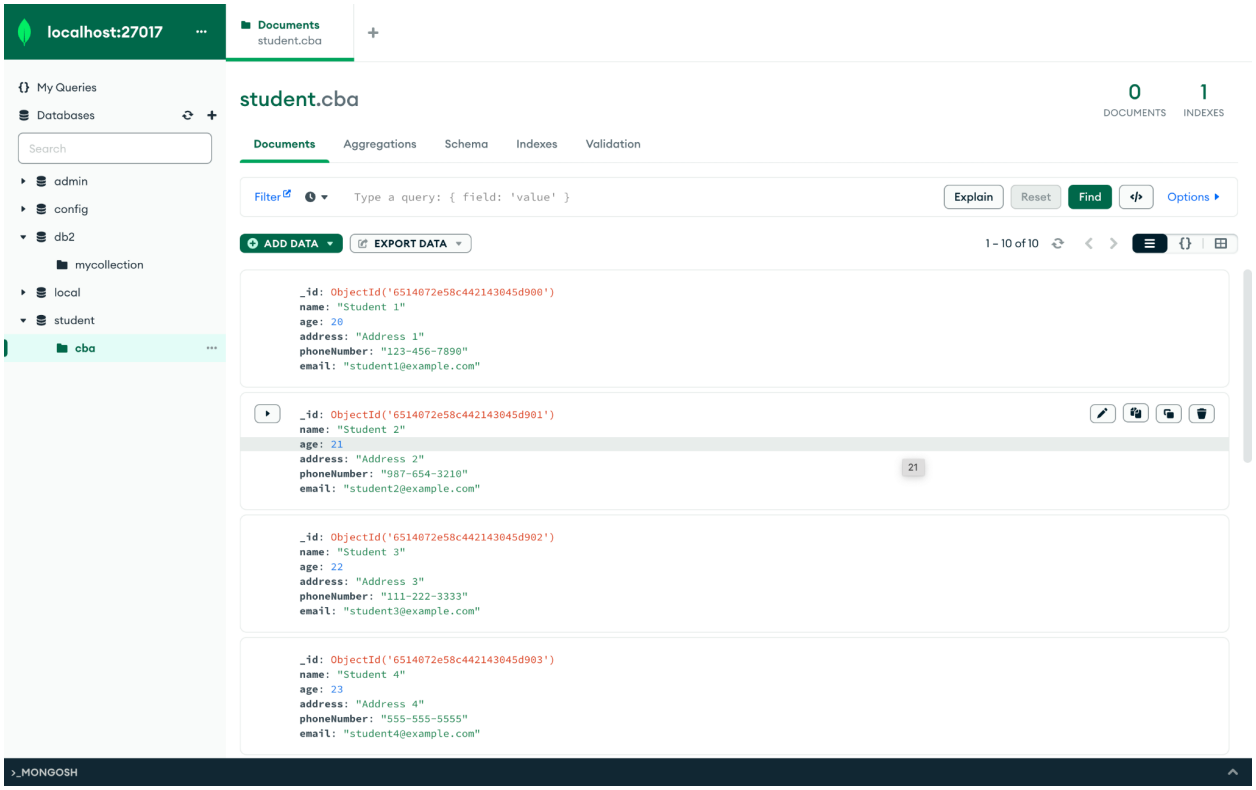


```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";
const databaseName = 'student';
const collectionName = 'cba';

MongoClient.connect(url, function (err, db2) {
if (err) throw err;
  var dbo = db2.db(databaseName);
  var studentDetails = [
    {name: 'Student 1',age: 20,address: 'Address 1',phoneNumber: '123-456-7890',email: 'student1@example.com'},
    {name: 'Student 2',age: 21,address: 'Address 2',phoneNumber: '987-654-3210',email: 'student2@example.com'},
    {name: 'Student 3',age: 22,address: 'Address 3',phoneNumber: '111-222-3333',email: 'student3@example.com'},
    {name: 'Student 4',age: 23,address: 'Address 4',phoneNumber: '555-555-5555',email: 'student4@example.com'},
    {name: 'Student 5',age: 24,address: 'Address 5',phoneNumber: '999-999-9999',email: 'student5@example.com'},
    {name: 'Student 6',age: 25,address: 'Address 6',phoneNumber: '444-444-4444',email: 'student6@example.com'},
    {name: 'Student 7',age: 26,address: 'Address 7',phoneNumber: '777-777-7777',email: 'student7@example.com'},
    {name: 'Student 8',age: 27,address: 'Address 8',phoneNumber: '888-888-8888', email: 'student8@example.com'}
    {name: 'Student 9',age: 28,address: 'Address 9',phoneNumber: '666-666-6666',email: 'student9@example.com'},
    {name: 'Student 10',age: 29,address: 'Address 10',phoneNumber: '222-222-2222',email: 'student10@example.com
  ];

  dbo.collection(collectionName).insertMany(studentDetails, function (err, res) {
    if (err) throw err;
    console.log(res);
    console.log("The count of total students whose data has been inserted are : " + res.insertedCount);
    db2.close();
  });
```

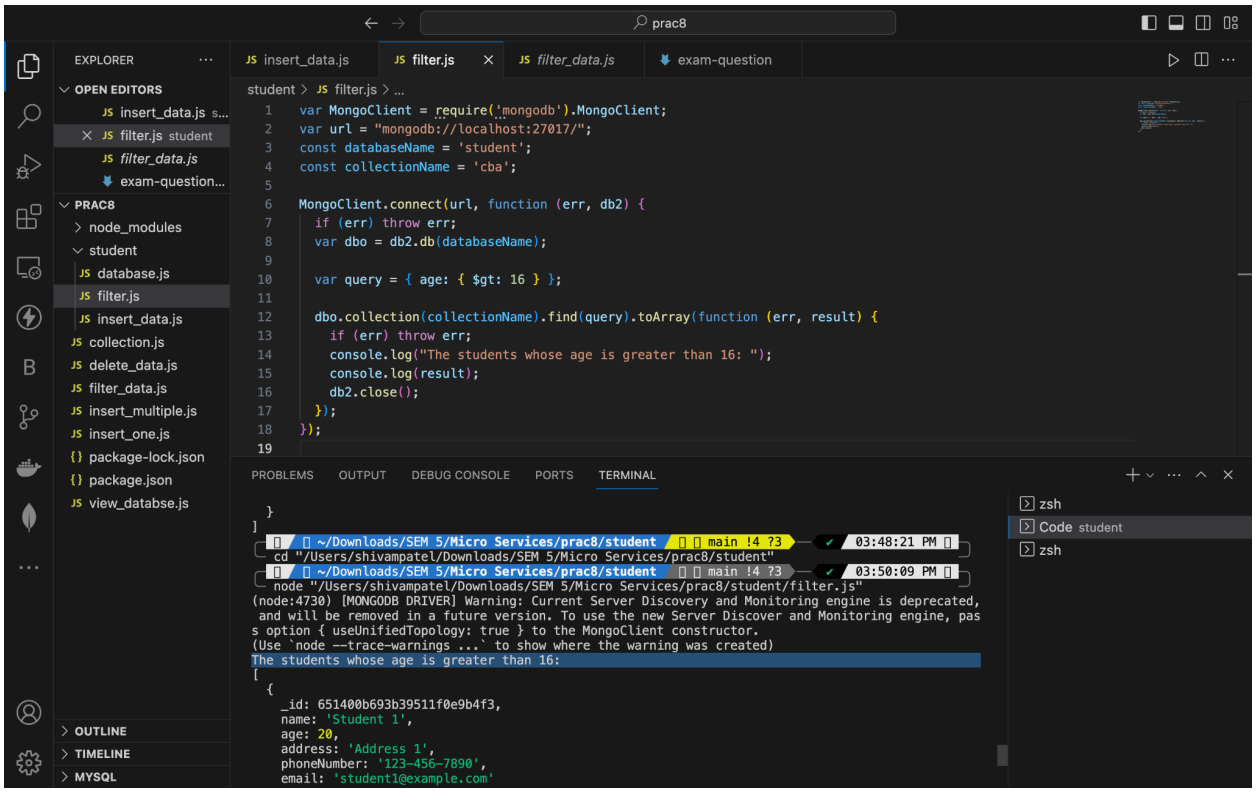
```
 and will be removed in a future version. To use the new Server Discover and Monitoring engine, pas
s option { useUnifiedTopology: true } to the MongoClient constructor.
(Use `node --trace-warnings ...` to show where the warning was created)
{
  result: { ok: 1, n: 10 },
  ops: [
    {
      name: 'Student 1',
      age: 20,
```

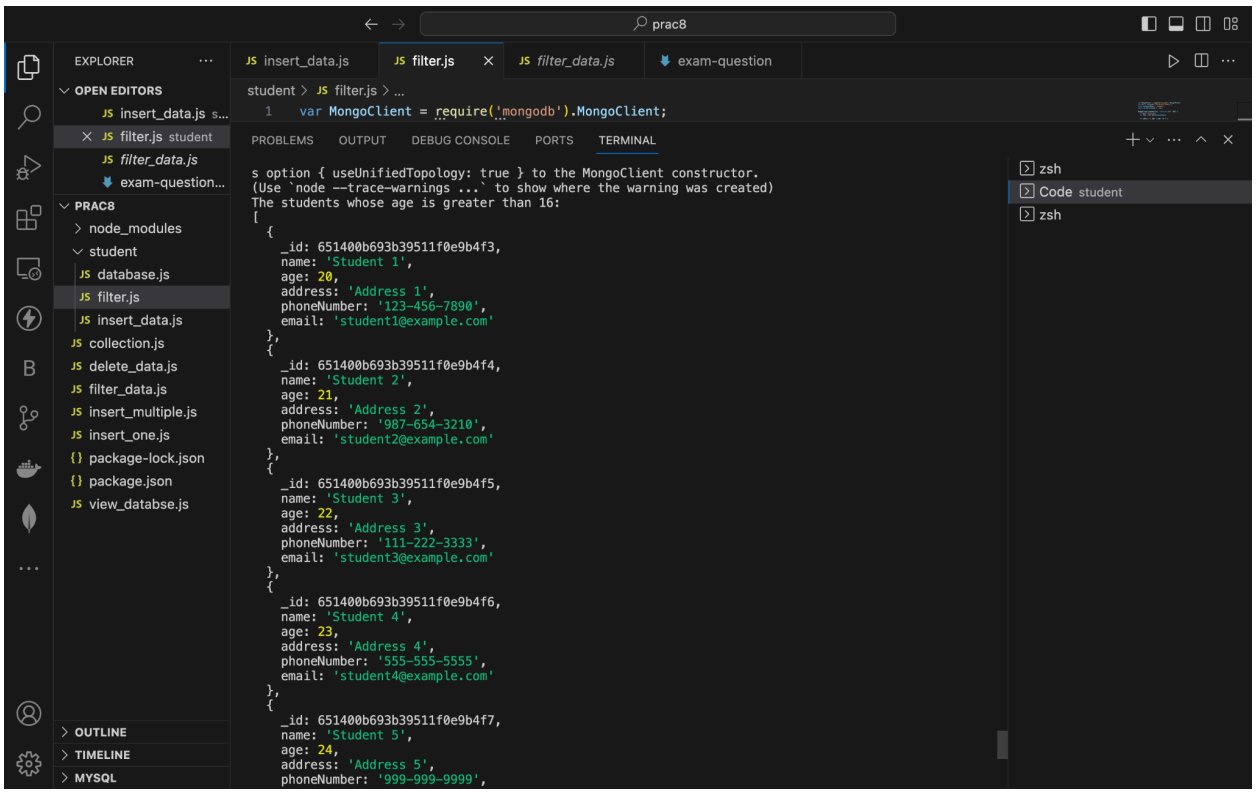


```
      name: 'Student 10',
      age: 29,
      address: 'Address 10',
      phoneNumber: '222-222-2222',
      email: 'student10@example.com',
      _id: 651400b693b39511f0e9b4fc
    }
  ],
  insertedCount: 10,
  insertedIds: {
    '0': 651400b693b39511f0e9b4f3,
    '1': 651400b693b39511f0e9b4f4,
    '2': 651400b693b39511f0e9b4f5,
    '3': 651400b693b39511f0e9b4f6,
    '4': 651400b693b39511f0e9b4f7,
    '5': 651400b693b39511f0e9b4f8,
    '6': 651400b693b39511f0e9b4f9,
    '7': 651400b693b39511f0e9b4fa,
    '8': 651400b693b39511f0e9b4fb,
    '9': 651400b693b39511f0e9b4fc
  }
}
The count of total students whose data has been inserted are : 10
~/Downloads/SEM 5/Micro Services/prac8/student  main !4 ?3  03:45:18 PM
```


localhost:27017
Documents
student.cba
My Queries
Databases
Search
admin
config
db2
mycollection
local
student
cba
student.cba
0
DOCUMENTS
1
INDEXES
Documents
Aggregations
Schema
Indexes
Validation
Filter
Type a query: { field: 'value' }
Explain
Reset
Find
Options
ADD DATA
EXPORT DATA
1 – 10 of 10
_id: ObjectId('6514072e58c442143045d900')
name: "Student 1"
age: 20
address: "Address 1"
phoneNumber: "123-456-7890"
email: "student1@example.com"
_id: ObjectId('6514072e58c442143045d901')
name: "Student 2"
age: 21
address: "Address 2"
phoneNumber: "987-654-3210"
email: "student2@example.com"
21
_id: ObjectId('6514072e58c442143045d902')
name: "Student 3"
age: 22
address: "Address 3"
phoneNumber: "111-222-3333"
email: "student3@example.com"
_id: ObjectId('6514072e58c442143045d903')
name: "Student 4"
age: 23
address: "Address 4"
phoneNumber: "555-555-5555"
email: "student4@example.com"
>_MONGOSH

→ Filter the students whose age is greater than 16.





→ Delete the student record whose phone number contains the digit ‘6’ in it.