

Lesson 01 Demo 03

Performing CRUD Operations with Mongo Shell

Objective: To perform CRUD operations with Mongo Shell by installing mongosh and performing update operations

Tools Required: VS Code

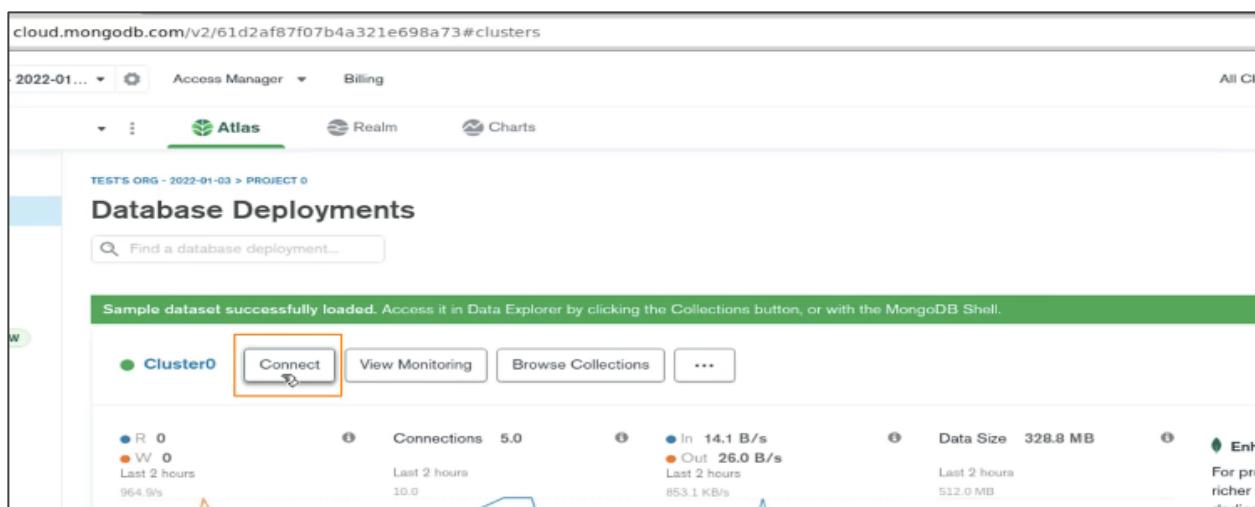
Prerequisites: None

Steps to be followed:

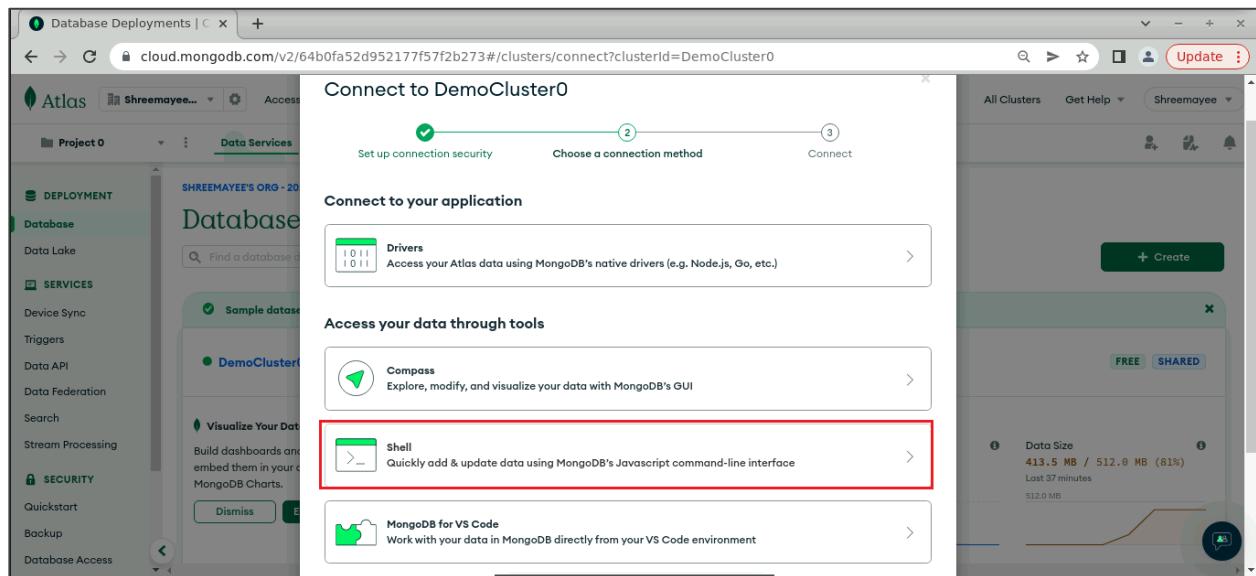
1. Install mongosh
2. Install Mongo Shell
3. Connect to MongoDB
4. Execute commands
5. Create a document
6. Insert more documents
7. Perform update operations

Step 1: Install mongosh

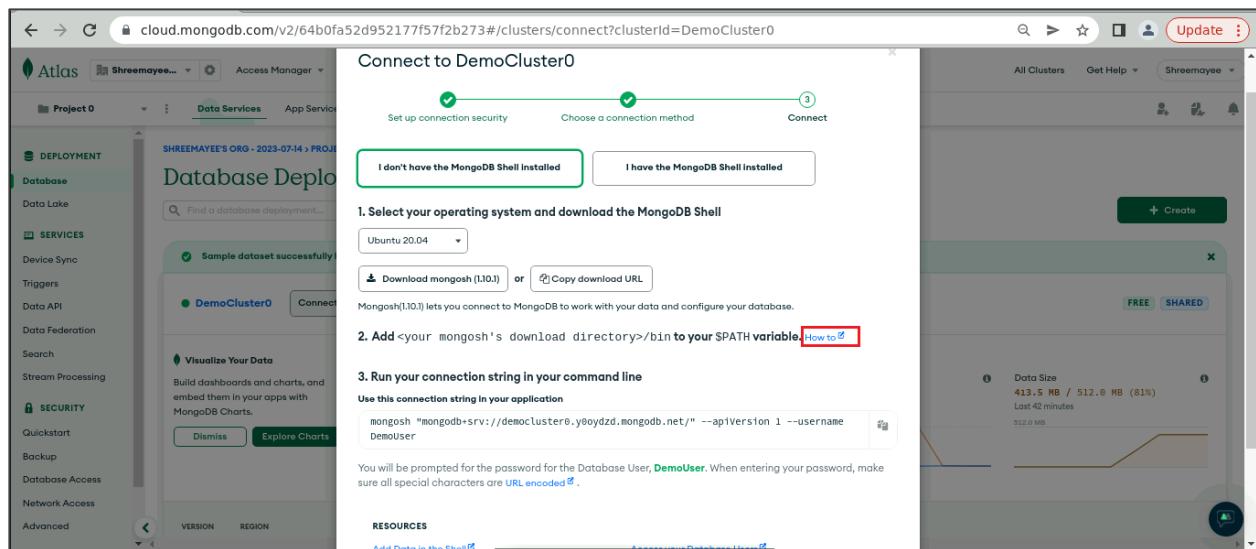
1.1 On the main interface of MongoDB Atlas, click on **Connect**



1.2 Select the **Shell** option to connect with MongoDB Shell



1.3 Click on **How to** and set the path variable for the MongoDB shell



1.4 Follow the instructions to import the public key used by the package management system

1 Import the public key used by the package management system.

From a terminal, issue the following command to import the MongoDB public GPG Key from <https://www.mongodb.org/static/pgp/server-5.0.asc>:

```
 wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
```

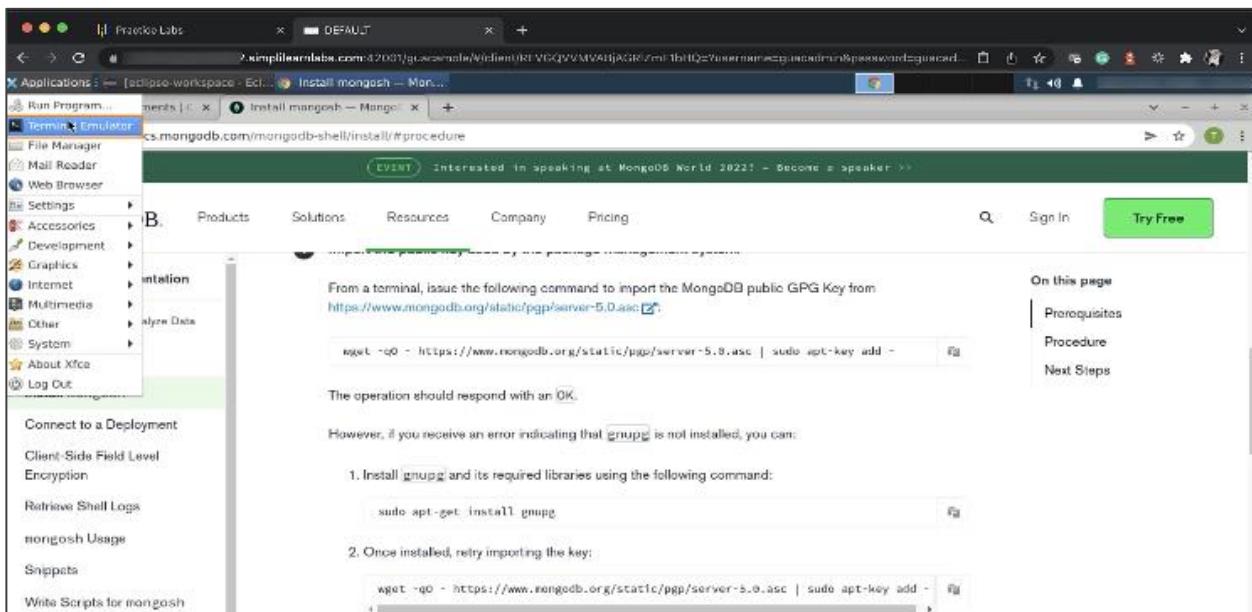
The operation should respond with an **OK**.

However, if you receive an error indicating that `gnupg` is not installed, you can:

1. Install `gnupg` and its required libraries using the following command:

```
sudo apt-get install gnupg
```
2. Once installed, retry importing the key:

1.5 Open a Terminal Emulator



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window contains the following text:

From a terminal, issue the following command to import the MongoDB public GPG Key from <https://www.mongodb.org/static/pgp/server-5.0.asc>:

```
 wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
```

The operation should respond with an **OK**.

However, if you receive an error indicating that `gnupg` is not installed, you can:

1. Install `gnupg` and its required libraries using the following command:

```
sudo apt-get install gnupg
```
2. Once installed, retry importing the key:

1.6 Copy the command below and paste it into the terminal to import the public GPG key:

sudo apt-get install gnupg

1 Import the public key used by the package management system.

From a terminal, issue the following command to import the MongoDB public GPG Key from <https://www.mongodb.org/static/pgp/server-5.0.asc>:

```
 wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
```

The operation should respond with an **OK**.

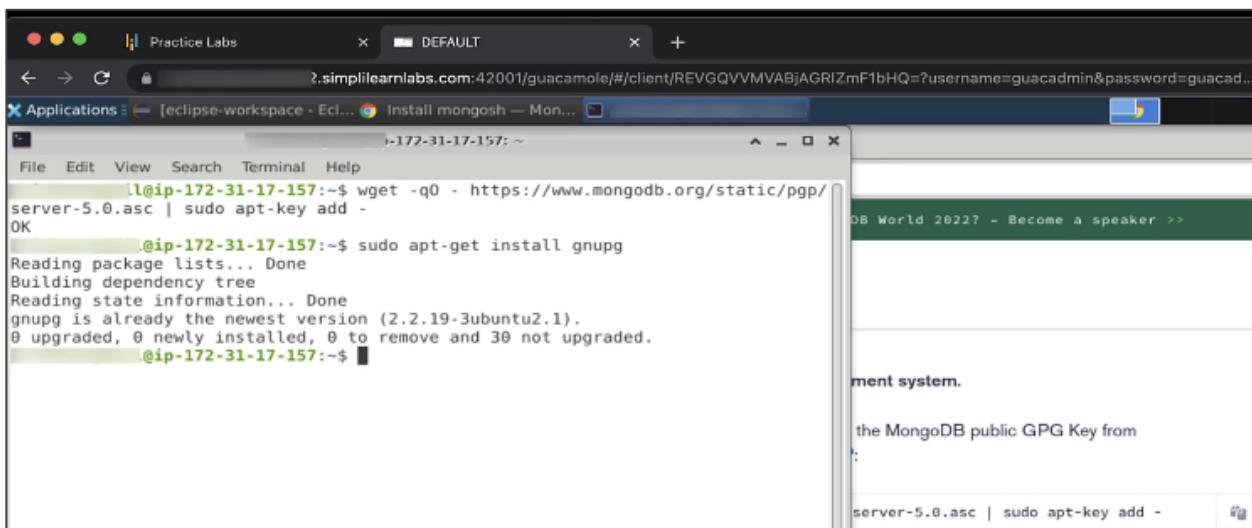
However, if you receive an error indicating that **gnupg** is not installed, you can:

1. Install **gnupg** and its required libraries using the following command:

```
sudo apt-get install gnupg
```
2. Once installed, retry importing the key:

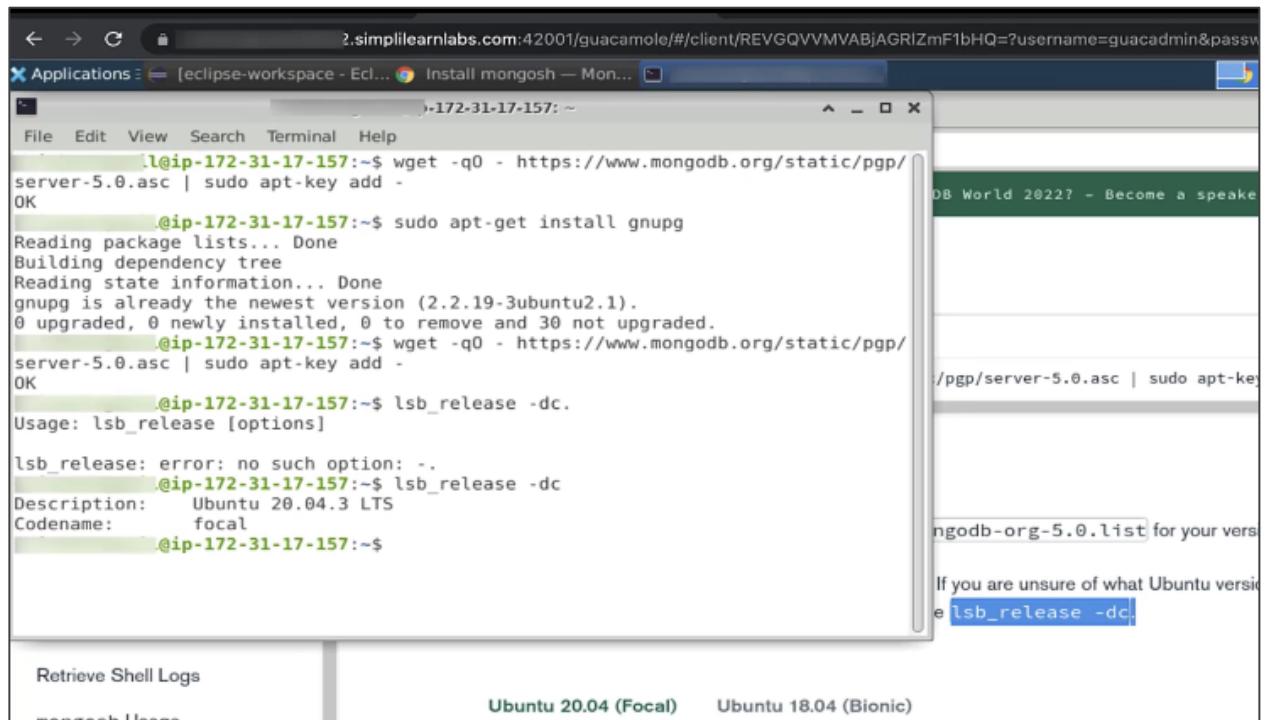
```
 wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
```

1.7 Paste the command in the terminal and press **Enter**



1.8 Execute the release command to check the version

`lsb_release -dc`



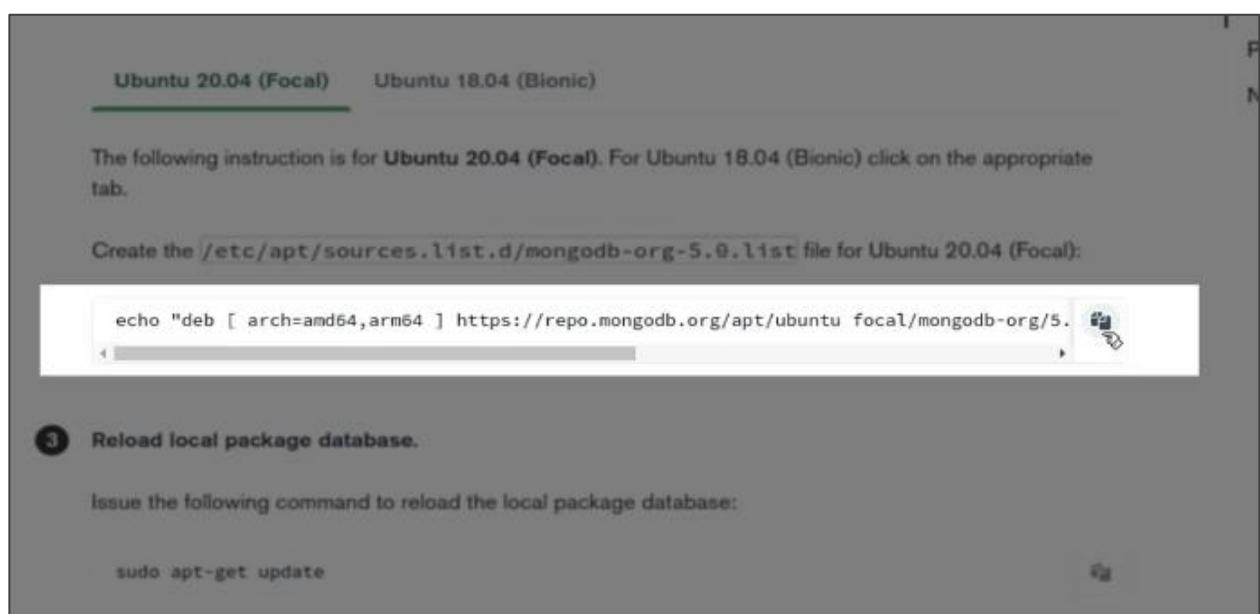
```

File Edit View Search Terminal Help
@ip-172-31-17-157:~$ wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
OK
@ip-172-31-17-157:~$ sudo apt-get install gnupg
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (2.2.19-3ubuntu2.1).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
@ip-172-31-17-157:~$ wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -
OK
@ip-172-31-17-157:~$ lsb_release -dc.
Usage: lsb_release [options]

lsb_release: error: no such option: -.
@ip-172-31-17-157:~$ lsb_release -dc
Description: Ubuntu 20.04.3 LTS
Codename: focal
@ip-172-31-17-157:~$
```

The terminal window shows the execution of the `lsb_release -dc` command. It first attempts to add a GPG key from the MongoDB PGP key server. Then it tries to run `lsb_release -dc.` which is invalid syntax. Finally, it runs `lsb_release -dc` correctly, displaying the Ubuntu 20.04 LTS information.

1.9 Follow the Ubuntu 20.04 instructions. Create a file for Ubuntu 20.04 by copying the command from the screenshot below:



The screenshot shows a step-by-step guide for Ubuntu 20.04 (Focal). It starts with a note that the instruction is for Ubuntu 20.04 (Focal), with a link to Ubuntu 18.04 (Bionic) if needed. It then provides a command to create a sources.list.d file:

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5."
```

Step 3, labeled "Reload local package database.", is shown with the command:

```
sudo apt-get update
```

The interface includes tabs for "Ubuntu 20.04 (Focal)" and "Ubuntu 18.04 (Bionic)".

1.10 Paste it into the terminal

```

Practo Labs                               DEFAULT
Applications : [eclipse-workspace - Ed... Install mongosh - Mon...
File Edit View Search Terminal Help
server-5.0.asc | sudo apt-key add -
OK
      gip-172-31-17-157:~$ sudo apt-get install gnupg
Reading package lists... Done
Building dependency tree
Reading state information... Done
gnupg is already the newest version (2.2.19-3ubuntu2.1).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
      gip-172-31-17-157:~$ wget -qO - https://www.mongodb.org/static/pkp/
server-5.0.asc | sudo apt-key add -
OK
      gip-172-31-17-157:~$ lsb_release -dc.
Usage: lsb_release [options]

lsb_release: error: no such option: -
      gip-172-31-17-157:~$ lsb_release -dc
Description: Ubuntu 20.04.3 LTS
Codename: focal
      gip-172-31-17-157:~$ echo "deb [ arch=amd64,arm64 ] https://repo.mo
ngodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse" | sudo tee /etc/apt/sourc
es.list.d/mongodb-org-5.0.list
deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5
.0 multiverse
      gip-172-31-17-157:~$ 

Retrieve Shell Logs
mongosh Usage
Snippets
Write Scripts for mongosh
Reference
Access mongosh Help
Compatibility Changes with

```

③ Reload local package database.

Issue the following command to reload the local package database:

```
sudo apt-get update
```

1.11 Once the file is created, reload the local package using the command below:

sudo apt-get update

```

Practo Labs                               DEFAULT
Applications : [eclipse-workspace - Ed... Install mongosh - Mon...
File Edit View Search Terminal Help
OK
erishant@gmail|gip-172-31-17-157:~$ lsb_release -dc.
      lsb_release: error: no such option: -
erishant@gmail|gip-172-31-17-157:~$ lsb_release -dc
Description: Ubuntu 20.04.3 LTS
Codename: focal
      gip-172-31-17-157:~$ echo "deb [ arch=amd64,arm64 ] https://repo.mo
ngodb.org/apt/ubuntu focal/mongodb-org/5.0 multiverse" | sudo tee /etc/apt/sourc
es.list.d/mongodb-org-5.0.list
deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/5
.0 multiverse
erishant@gmail|gip-172-31-17-157:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
      'us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
      '/cli.github.com/packages stable InRelease
Hit:6 http://packages.microsoft.com/repos/code stable InRelease
Ign:7 https://pkg.jenkins.io/debian binary/ InRelease
Hit:8 https://pkg.jenkins.io/debian binary/ Release
Hit:9 https://dl.google.com/linux/chrome/deb stable InRelease
      'or headers] [Waiting for headers] [Connecting to repo.zabbix.com (]
      'or headers] [Waiting for headers] [Connecting to repo.zabbix.com (]

Retrieve Shell Logs

```

Step 2: Install Mongo Shell

2.1 Install the Mongo Shell using the command below:

```
sudo apt-get install -y mongodb-mongosh
```

3 Reload local package database.

Issue the following command to reload the local package database:

```
sudo apt-get update
```

4 Install the mongosh package.

To install the latest stable version of `mongosh`, issue the following command:

```
sudo apt-get install -y mongodb-mongosh
```

Next Steps

2.2 Paste the command in the terminal

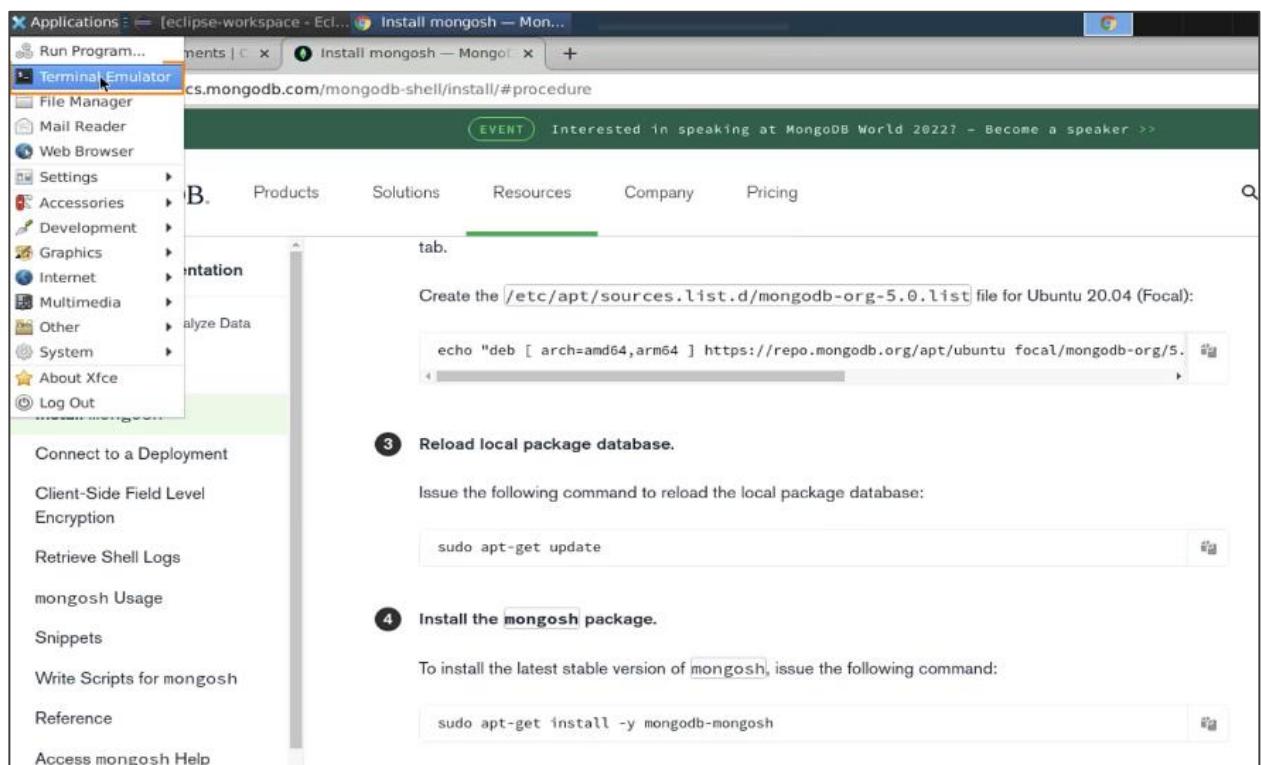
```
gip-172-31-17-157:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 https://cli.github.com/packages stable InRelease
Hit:6 http://packages.microsoft.com/repos/codecvt stable InRelease
Ign:7 https://pkg.jenkins.io/debian binary/ InRelease
Hit:8 https://pkg.jenkins.io/debian binary/ Release
Hit:9 https://dl.google.com/linux/chrome/deb stable InRelease
Ign:10 https://repo.mongodb.org/apt/ubuntu/focal/mongodb-org/5.0 InRelease
Hit:11 https://repo.mongodb.org/apt/ubuntu/focal/mongodb-org/5.0 Release
Hit:12 https://repo.zabbix.com/zabbix/5.8/ubuntu focal InRelease
Hit:13 https://deb.nodesource.com/node_17.x focal InRelease
Reading package lists... Done
gip-172-31-17-157:~$ sudo apt-get install -y mongodb-mongosh
Reading package lists... Done
Building dependency tree
Reading state information... Done
mongodb-mongosh is already the newest version (1.1.7).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
gip-172-31-17-157:~$
```

4 Install the mongosh package.

To install the latest stable version of `mongosh`, issue the following command:

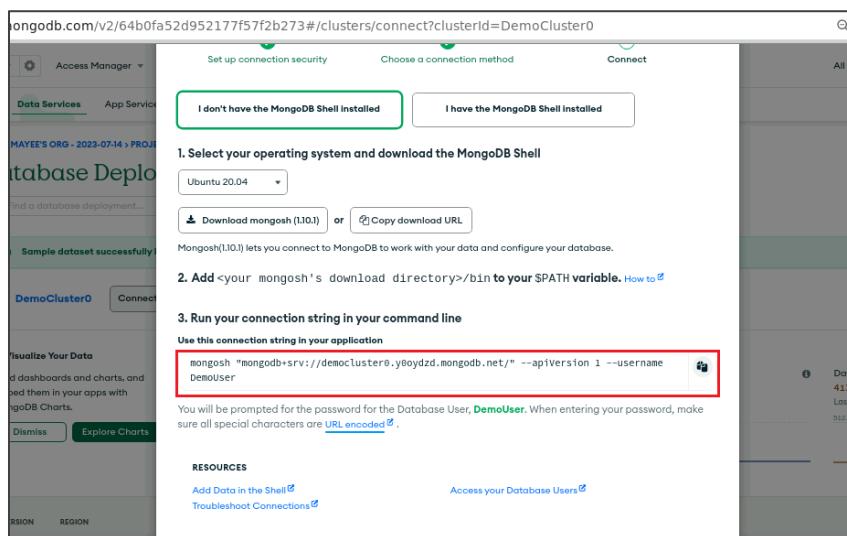
```
sudo apt-get install -y mongodb-mongosh
```

2.3 Close the terminal and reopen it by clicking on Terminal Emulator

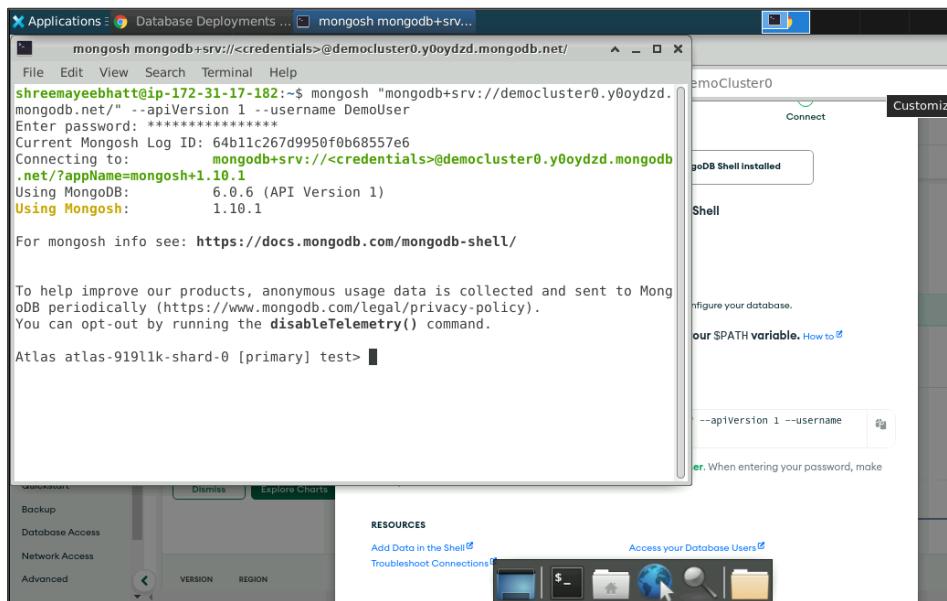


Step 3: Connect to MongoDB

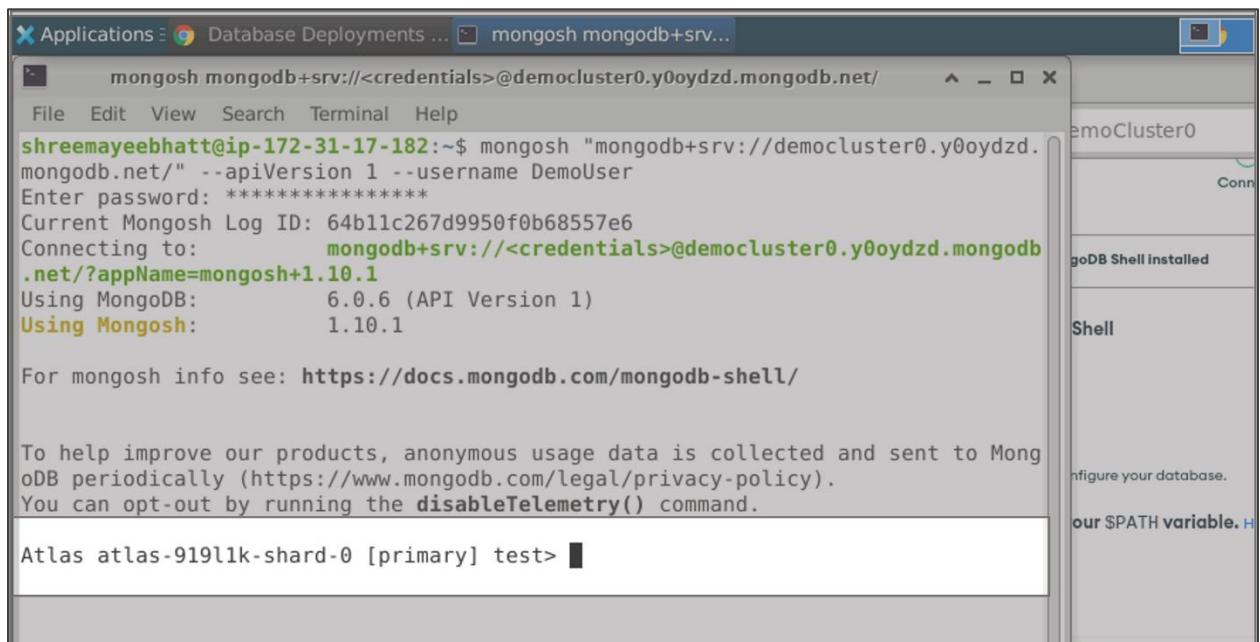
3.1 Connect to MongoDB by using the connection string in the application



3.2 Paste it into the terminal and enter the password provided during account creation



It takes some time to establish the connection; once it is established, you will enter the MongoDB shell.



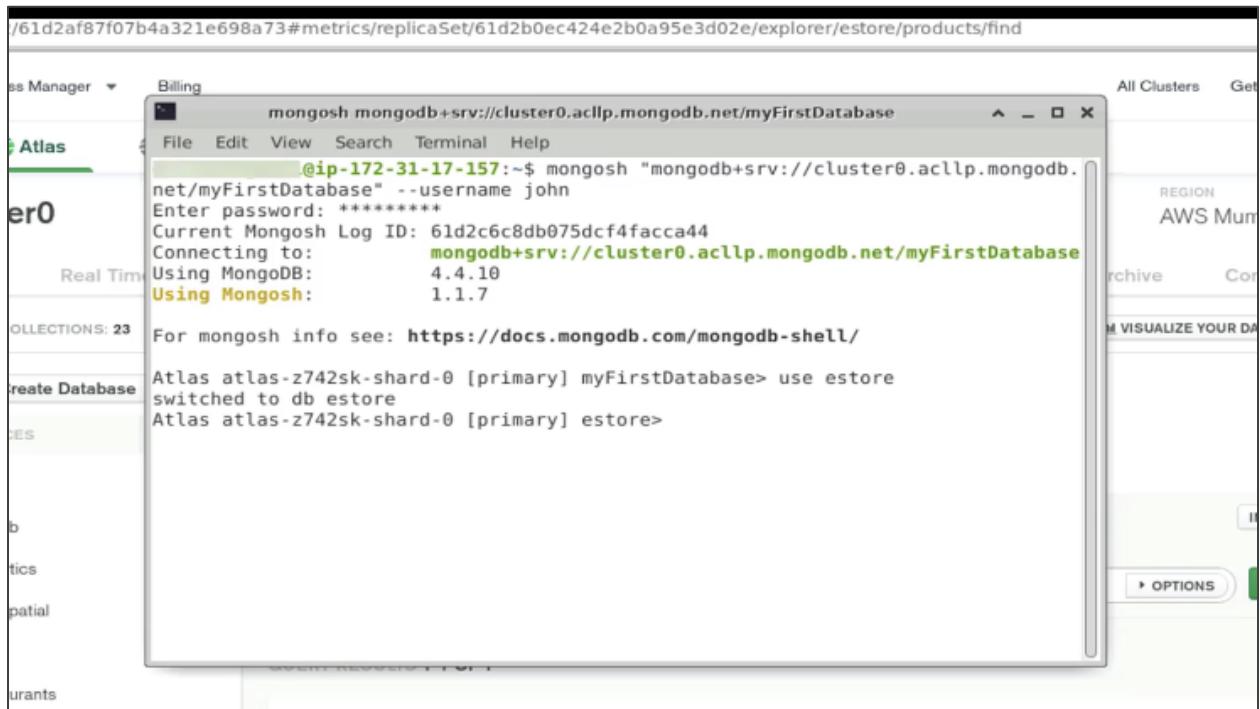
3.3 Select Cluster0

The screenshot shows the MongoDB Cloud interface. On the left, a sidebar menu includes options like Deployment, Databases, Data Lake, Data Services, Triggers, Data API (Preview), Security, Quickstart, Database Access, Network Access, and Advanced. The main area is titled "Database Deployments" and shows "Cluster0". It displays various metrics: R: 0, W: 0 (Last 9 hours), Connections: 1.0 (Last 2 hours), In: 4.4 B/s, Out: 26.0 B/s (Last 9 hours), and Data Size: 328.0 MB (Last 2 hours). Below these metrics are sections for VERSION (4.4.10), REGION (AWS / Mumbai (ap-south-1)), CLUSTER TIER (M0 Sandbox (General)), TYPE (Replica Set - 3 nodes), BACKUPS (Inactive), LINKED REALM APP (None Linked), and ATLAS SEARCH (Create Index). A green "Get Started" button is at the bottom left.

3.4 Navigate to the Collections tab

The screenshot shows the MongoDB Cloud interface with the "Collections" tab selected. The main area displays a message: "Retrieving list of databases and collections...". At the bottom, there is a "System Status: All Good" message and a footer with links to Status, Terms, Privacy, Atlas Blog, and Contact Sales. A "REFRESH" button is located in the top right corner of the main content area.

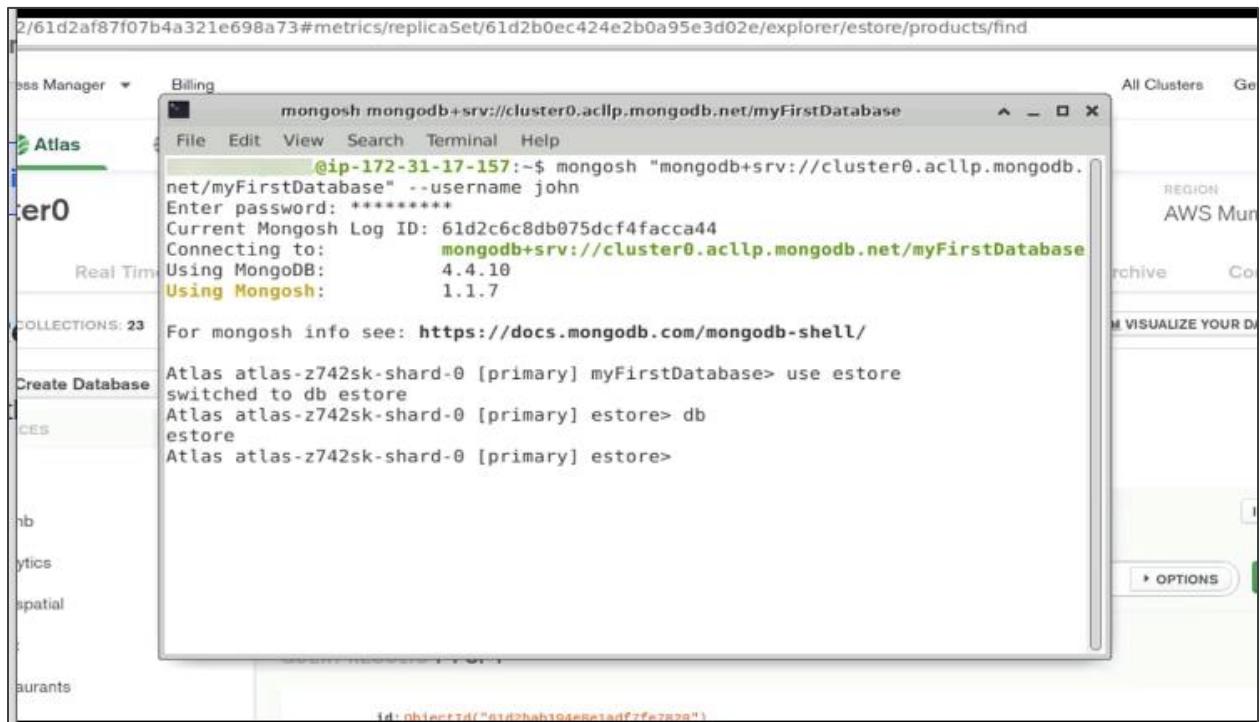
3.5 Enter **use estore** in the terminal window



```
/61d2af87f07b4a321e698a73#metrics/replicaSet/61d2b0ec424e2b0a95e3d02e/explorer/estore/products/find

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
@ip-172-31-17-157:~$ mongosh "mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase"
Enter password: *****
Current Mongosh Log ID: 61d2c6c8db075dcf4facca44
Connecting to: mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
Using MongoDB: 4.4.10
Using Mongosh: 1.1.7
For mongosh info see: https://docs.mongodb.com/mongodb-shell/
Atlas atlas-z742sk-shard-0 [primary] myFirstDatabase> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore>
```

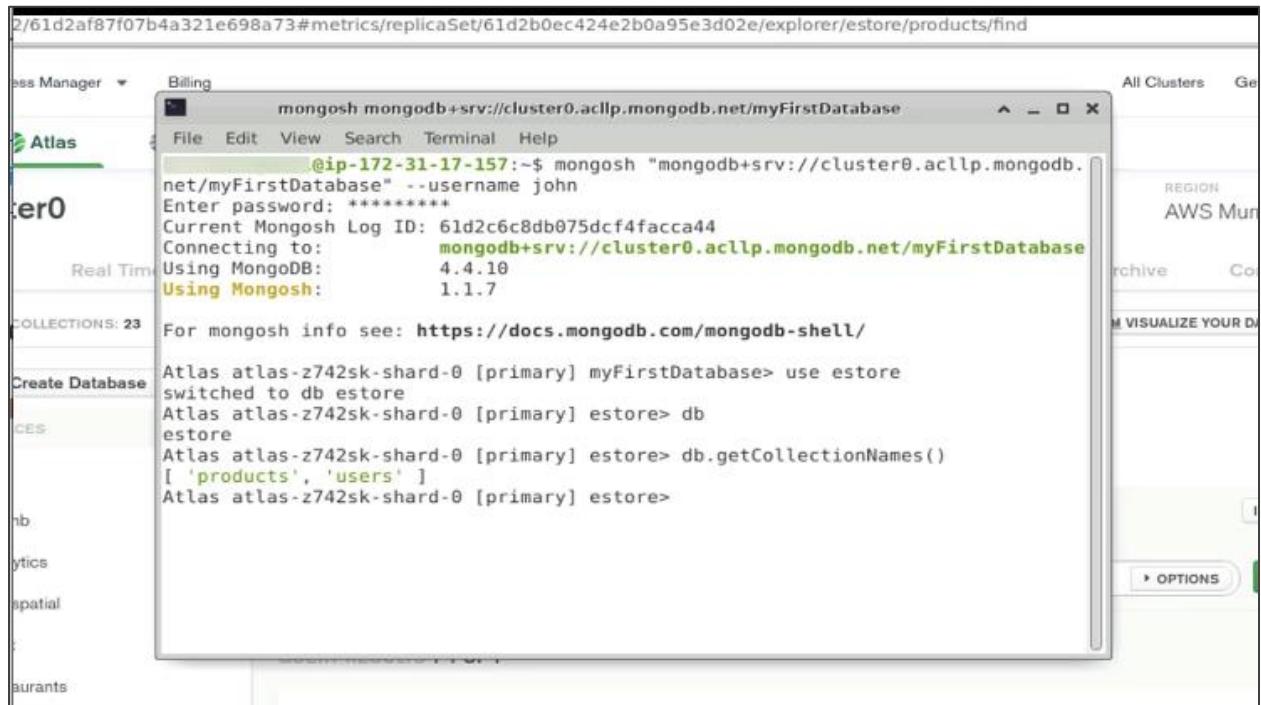
3.6 Enter the **db** command to understand which database you are working on



```
/61d2af87f07b4a321e698a73#metrics/replicaSet/61d2b0ec424e2b0a95e3d02e/explorer/estore/products/find

mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
@ip-172-31-17-157:~$ mongosh "mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase"
Enter password: *****
Current Mongosh Log ID: 61d2c6c8db075dcf4facca44
Connecting to: mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
Using MongoDB: 4.4.10
Using Mongosh: 1.1.7
For mongosh info see: https://docs.mongodb.com/mongodb-shell/
Atlas atlas-z742sk-shard-0 [primary] myFirstDatabase> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore> db
estore
Atlas atlas-z742sk-shard-0 [primary] estore>
```

3.7 Get the collection name using the `db.getCollectionNames()` command



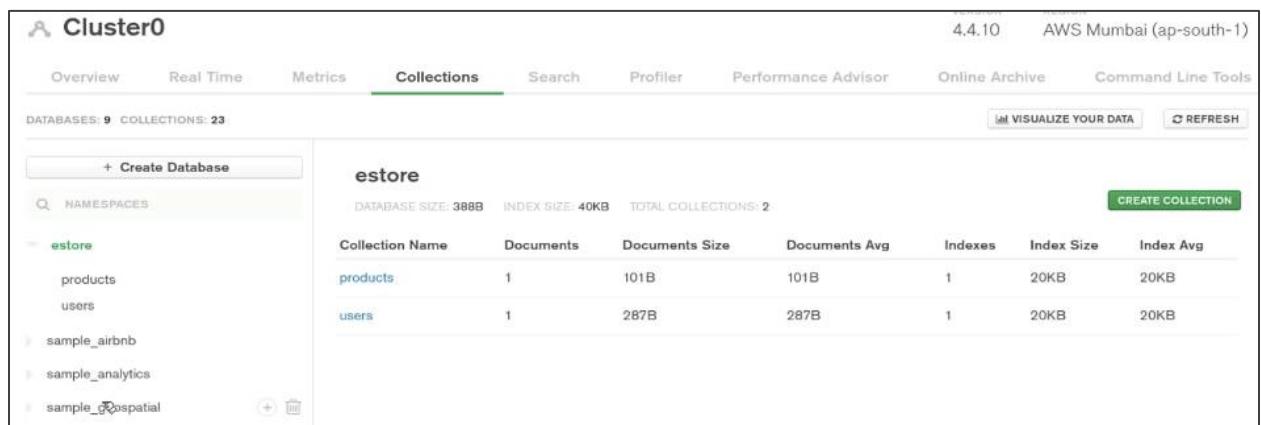
```

2/61d2af87f07b4a321e698a73#metrics/replicaSet/61d2b0ec424e2b0a95e3d02e/explorer/estore/products/find

Billing
mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
@ip-172-31-17-157:~$ mongosh "mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase"
Enter password: *****
Current Mongosh Log ID: 61d2c6c8db075dcf4facca44
Connecting to: mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
Using MongoDB: 4.4.10
Using Mongosh: 1.1.7
For mongosh info see: https://docs.mongodb.com/mongodb-shell/
Atlas atlas-z742sk-shard-0 [primary] myFirstDatabase> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore> db
estore
Atlas atlas-z742sk-shard-0 [primary] estore> db.getCollectionNames()
[ 'products', 'users' ]
Atlas atlas-z742sk-shard-0 [primary] estore>

```

3.8 Open the `estore` collection. You will see the products and users listed.



Cluster0

4.4.10 AWS Mumbai (ap-south-1)

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Command Line Tools

DATABASES: 9 COLLECTIONS: 23

+ Create Database

NAMESPACES

- estore
 - products
 - users
- sample_airbnb
- sample_analytics
- sample_gospatial

estore

DATABASE SIZE: 388B INDEX SIZE: 40KB TOTAL COLLECTIONS: 2

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
products	1	101B	101B	1	20KB	20KB
users	1	287B	287B	1	20KB	20KB

3.9 Select `Sample_restaurants` from the collections

The screenshot shows the MongoDB Atlas interface. On the left, there's a sidebar with a tree view of databases and collections. Under the 'estore' database, the 'products' and 'users' collections are listed. Below them, under 'sample_restaurants', are 'neighborhoods' and 'restaurants'. The 'restaurants' collection is highlighted with a red box. On the right, there's a detailed view of the 'estore' database with statistics: Database Size: 388B, Index Size: 40KB, Total Collections: 2. A table below shows the collection details:

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size
products	1	101B	101B	1	20KB
users	1	287B	287B	1	20KB

3.10 Type `sample_restaurants` in the terminal to access the `sample_restaurants` database

The screenshot shows a terminal window within the MongoDB Atlas interface. The command entered is:

```
mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
```

The terminal output shows the connection process:

```
@ip-172-31-17-157:~$ mongosh "mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase" --username john
Enter password: *****
Current Mongosh Log ID: 61d2c6c8db075dcf4facca44
Connecting to: mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
Using MongoDB: 4.4.10
Using Mongosh: 1.1.7

For mongosh info see: https://docs.mongodb.com/mongodb-shell/
```

Then, the user switches to the `sample_restaurants` database:

```
Atlas atlas-z742sk-shard-0 [primary] myFirstDatabase> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore> db
estore
Atlas atlas-z742sk-shard-0 [primary] estore> db.getCollectionNames()
[ 'products', 'users' ]
Atlas atlas-z742sk-shard-0 [primary] estore> use sample_restaurants
switched to db sample_restaurants
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants> █
```

Step 4: Execute commands

4.1 Execute the command below to get the collection names:

```
Db.getCollectionNames()
```

The screenshot shows the MongoDB shell interface. The command `db.getCollectionNames()` is run in the `sample_restaurants` database, and the output is highlighted with a red box. The output shows two collections: `'neighborhoods'` and `'restaurants'`.

```
mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
@ip-172-31-17-157:~$ mongosh "mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase"
Enter password: *****
Current Mongosh Log ID: 61d2c6c8db075dcf4facca44
Connecting to: mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
Using MongoDB: 4.4.10
Using Mongosh: 1.1.7

For mongosh info see: https://docs.mongodb.com/mongodb-shell/
Atlas atlas-z742sk-shard-0 [primary] myFirstDatabase> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore> db
estore
Atlas atlas-z742sk-shard-0 [primary] estore> db.getCollectionNames()
[ 'products', 'users' ]
Atlas atlas-z742sk-shard-0 [primary] estore> use sample_restaurants
switched to db sample_restaurants
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants> db
sample_restaurants
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants> db.getCollectionNames()

[ 'neighborhoods', 'restaurants' ]
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants>
```

4.2 Execute the `db.restaurants.find()` command to determine the documents inside the collection

The screenshot shows the MongoDB shell interface. The command `db.restaurants.find()` is run in the `sample_restaurants` database, and the output is shown. It displays several documents, each representing a restaurant with fields like `name`, `score`, `date`, `grade`, and `restaurant_id`.

```
mongosh mongodb+srv://cluster0.acllp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
score: 12
},
{
  date: ISODate("2013-01-04T00:00:00.000Z"),
  grade: 'A',
  score: 11
},
{
  date: ISODate("2012-06-07T00:00:00.000Z"),
  grade: 'A',
  score: 6
},
{
  date: ISODate("2012-01-17T00:00:00.000Z"),
  grade: 'A',
  score: 8
}
],
  name: "Bully's Deli",
  restaurant_id: '40361708'
}
]
Type "it" for more
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants>
```

4.3 Enter the **estore** command once again and the database will switch to estore

```
61d2af87f07b4a321e698a73#metrics/replicaSet/61d2b0ec424e2b0a95e3d02e/explorer/estore

mongosh mongodb+srv://cluster0.aclp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
{
  date: ISODate("2013-01-04T00:00:00.000Z"),
  grade: 'A',
  score: 11
},
{
  date: ISODate("2012-06-07T00:00:00.000Z"),
  grade: 'A',
  score: 6
},
{
  date: ISODate("2012-01-17T00:00:00.000Z"),
  grade: 'A',
  score: 8
}
],
name: "Bully'S Deli",
restaurant_id: '40361708'
}
Type "it" for more
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore>
```

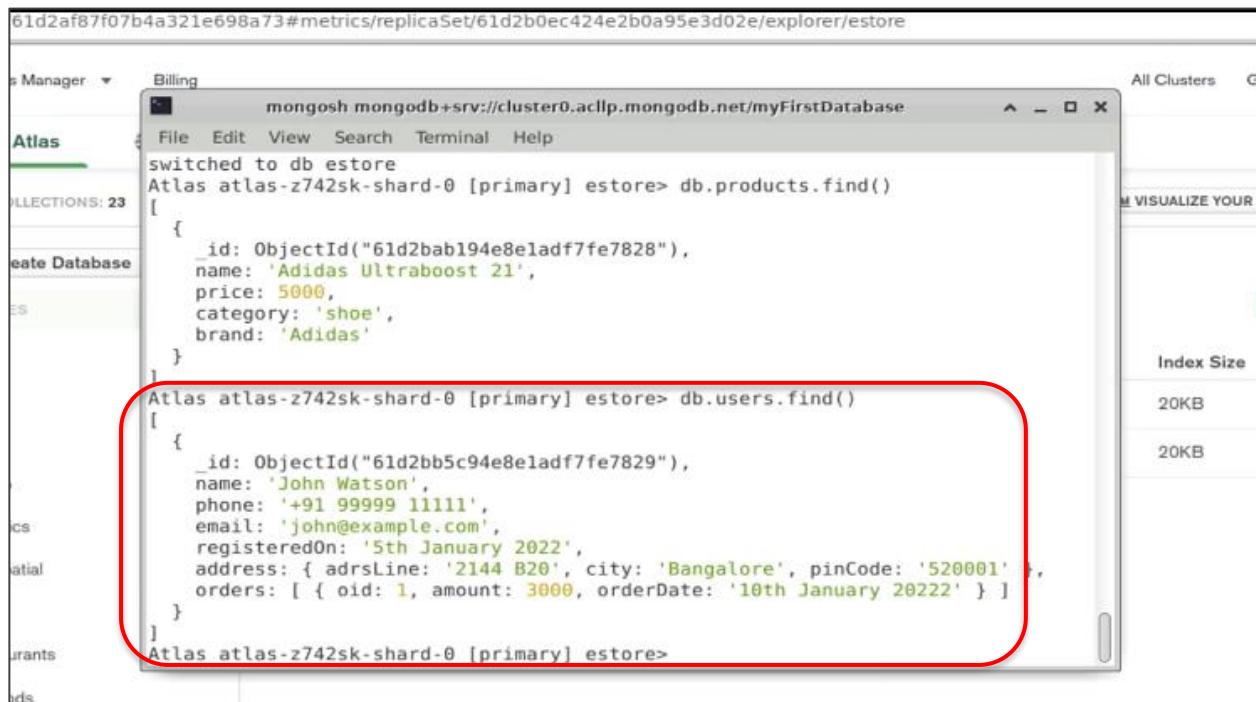
4.4 Use **db.products.find()** to list the products

```
61d2af87f07b4a321e698a73#metrics/replicaSet/61d2b0ec424e2b0a95e3d02e/explorer/estore

mongosh mongodb+srv://cluster0.aclp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
{
  date: ISODate("2012-01-17T00:00:00.000Z"),
  grade: 'A',
  score: 8
}
],
name: "Bully'S Deli",
restaurant_id: '40361708'
}
]
Type "it" for more
Atlas atlas-z742sk-shard-0 [primary] sample_restaurants> use estore
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore> db.products.find()
[
  {
    id: ObjectId("61d2bab194e8eladf7fe7828"),
    name: 'Adidas Ultraboost 21',
    price: 5000,
    category: 'shoe',
    brand: 'Adidas'
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore>
```

You can see that there is only one product listed in the terminal.

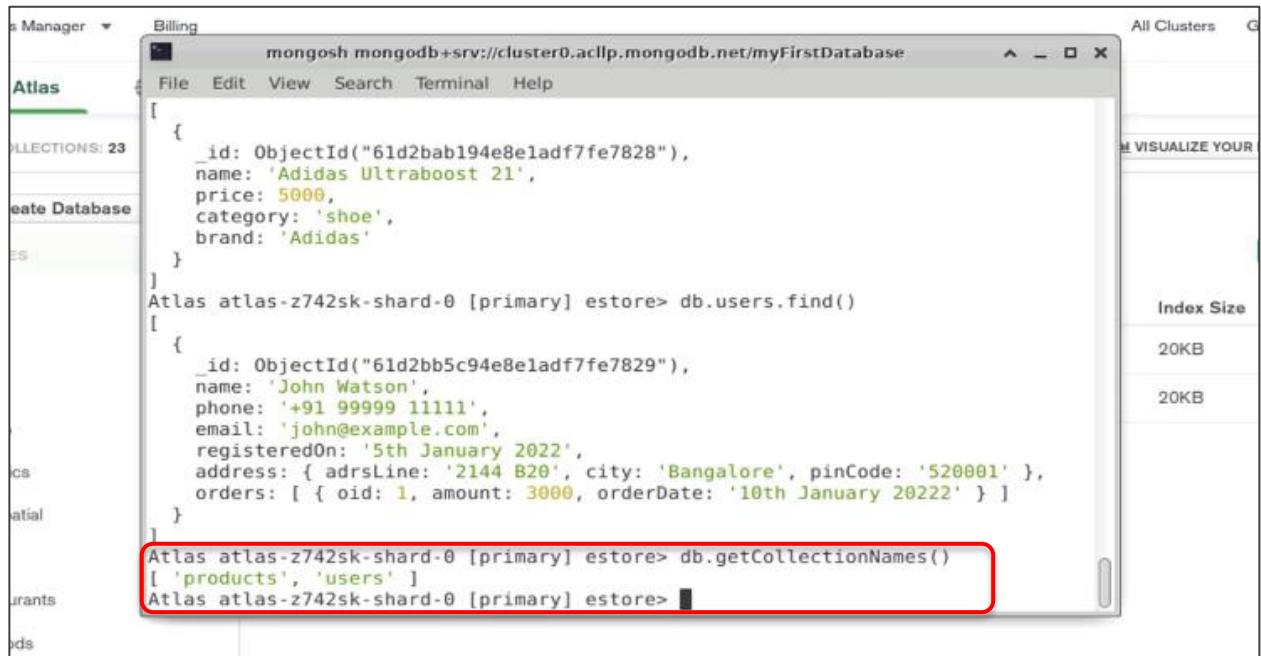
4.5 Similarly, use `db.users.find()` to determine the data



```
61d2af87f07b4a321e698a73#metrics/replicaSet/61d2b0ec424e2b0a95e3d02e/explorer/estore

mongosh mongodb+srv://cluster0.ac1lp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
switched to db estore
Atlas atlas-z742sk-shard-0 [primary] estore> db.products.find()
[
  {
    _id: ObjectId("61d2bab194e8eladf7fe7828"),
    name: 'Adidas Ultraboost 21',
    price: 5000,
    category: 'shoe',
    brand: 'Adidas'
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find()
[
  {
    _id: ObjectId("61d2bb5c94e8eladf7fe7829"),
    name: 'John Watson',
    phone: '+91 99999 11111',
    email: 'john@example.com',
    registeredOn: '5th January 2022',
    address: { adrLine: '2144 B20', city: 'Bangalore', pinCode: '520001' },
    orders: [ { oid: 1, amount: 3000, orderDate: '10th January 2022' } ]
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore>
```

4.6 Get the collection names again using the `db.getCollectionNames()` command

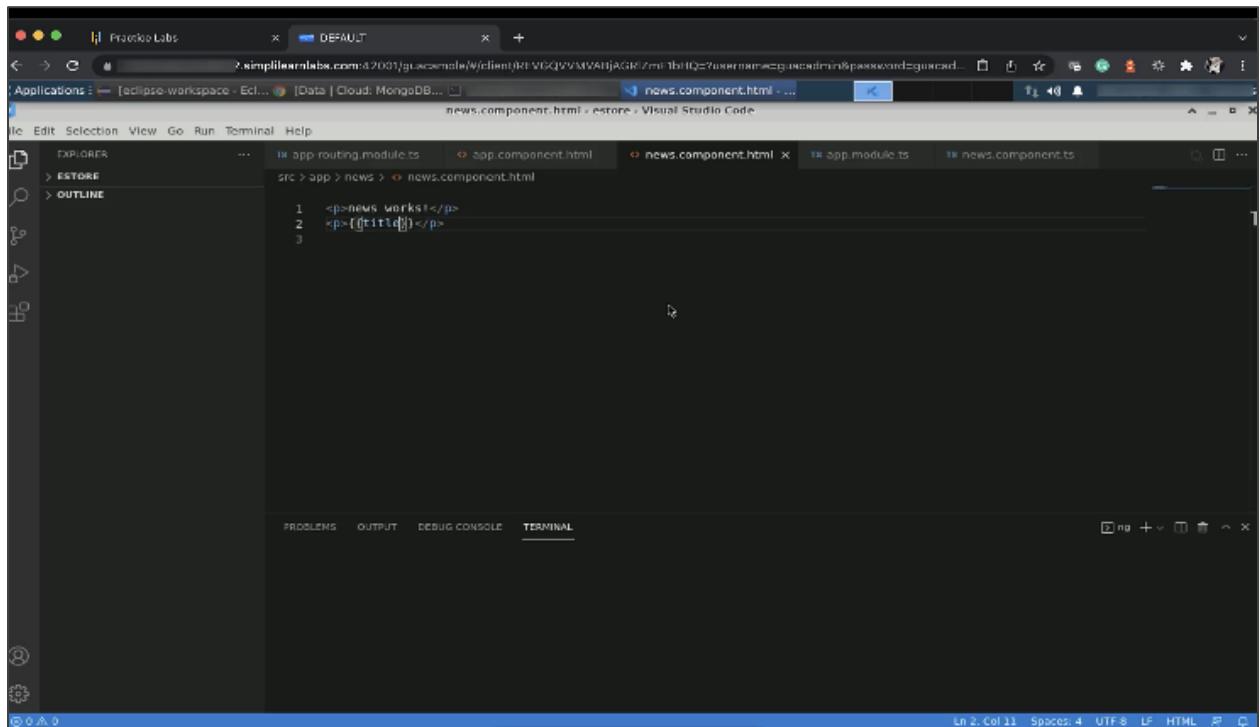


The screenshot shows the MongoDB shell interface within a browser window titled "Billing". The shell window has a menu bar with File, Edit, View, Search, Terminal, and Help. On the left, there's a sidebar with tabs like "Atlas", "Collections: 23", "Create Database", "ES", "CS", "Spatial", "URANTS", and "IDS". The main area of the shell shows some sample documents from the "users" collection. At the bottom of the shell window, the command `db.getCollectionNames()` is highlighted with a red rectangle. The output of this command, which lists the collections "products" and "users", is also highlighted with a red rectangle.

```
[{"_id": ObjectId("61d2bab194e8e1adf7fe7828"), "name": "Adidas Ultraboost 21", "price": 5000, "category": "shoe", "brand": "Adidas"}, {"_id": ObjectId("61d2bb5c94e8e1adf7fe7829"), "name": "John Watson", "phone": "+91 99999 11111", "email": "john@example.com", "registeredOn": "5th January 2022", "address": { "adrLine": "2144 B20", "city": "Bangalore", "pinCode": "520001" }, "orders": [ { "oid": 1, "amount": 3000, "orderDate": "10th January 2022" } ]}]Atlas atlas-z742sk-shard-0 [primary] estore> db.getCollectionNames()
[ 'products', 'users' ]
Atlas atlas-z742sk-shard-0 [primary] estore>
```

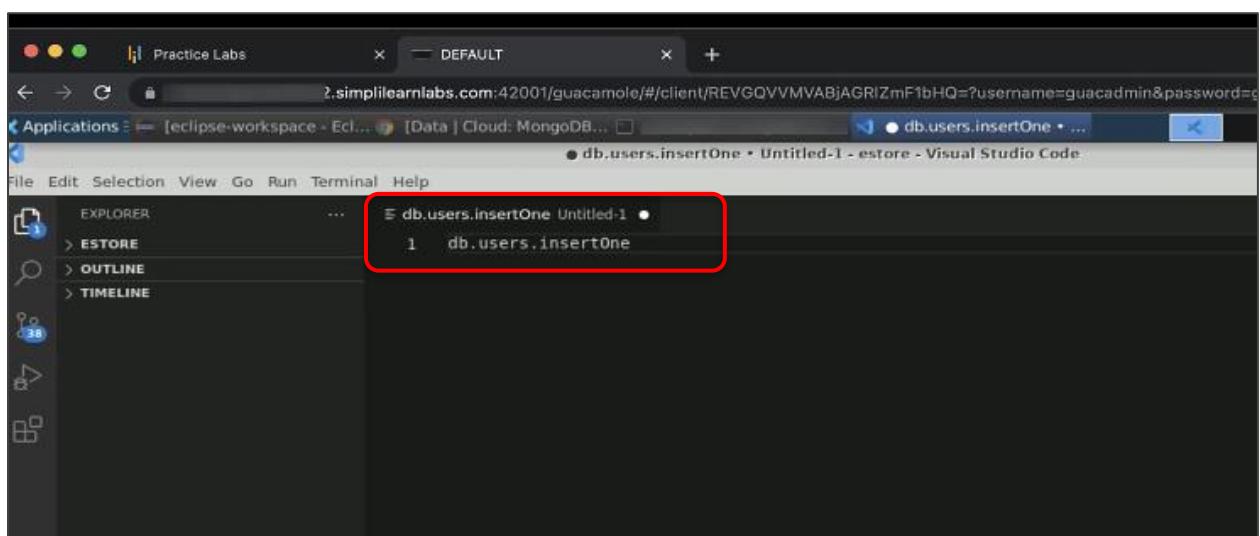
Step 5: Create a document

5.1 Open the Visual Studio Code editor



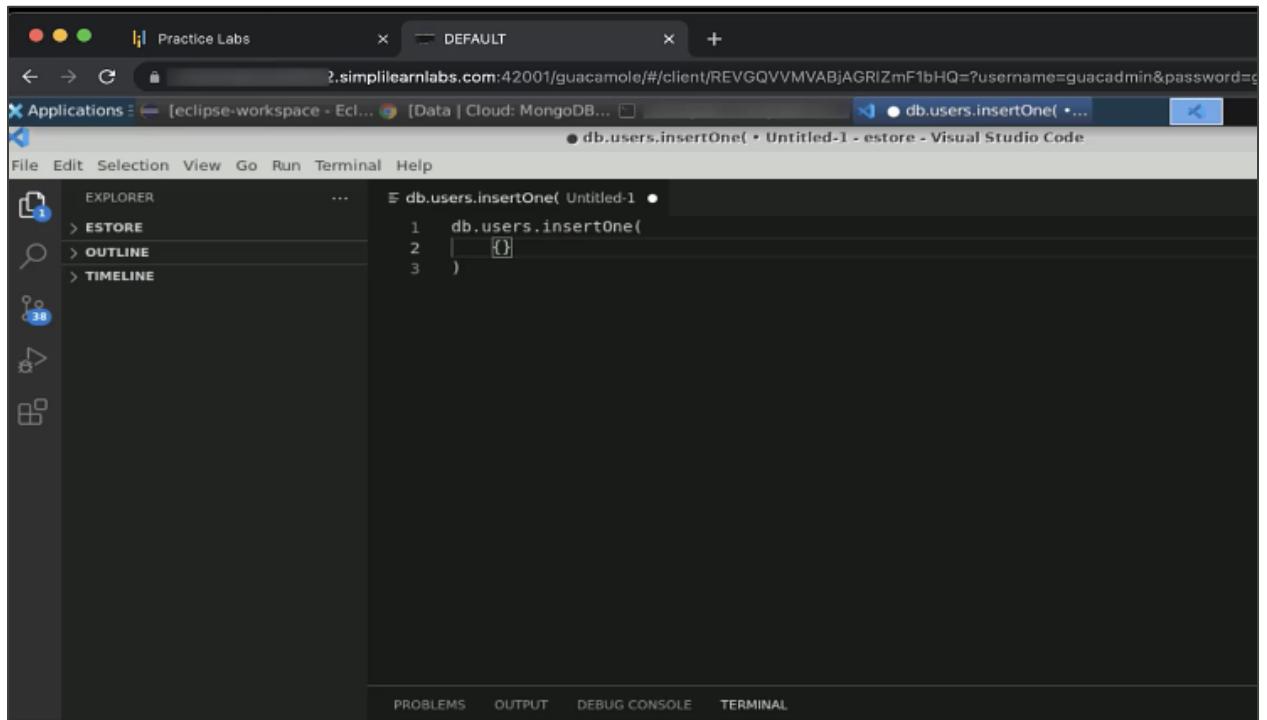
```
<p>news works!</p>
<p>{{title}}</p>
```

5.2 Create a new file and use the **insertOne** method to insert a document inside the database



```
db.users.insertOne
```

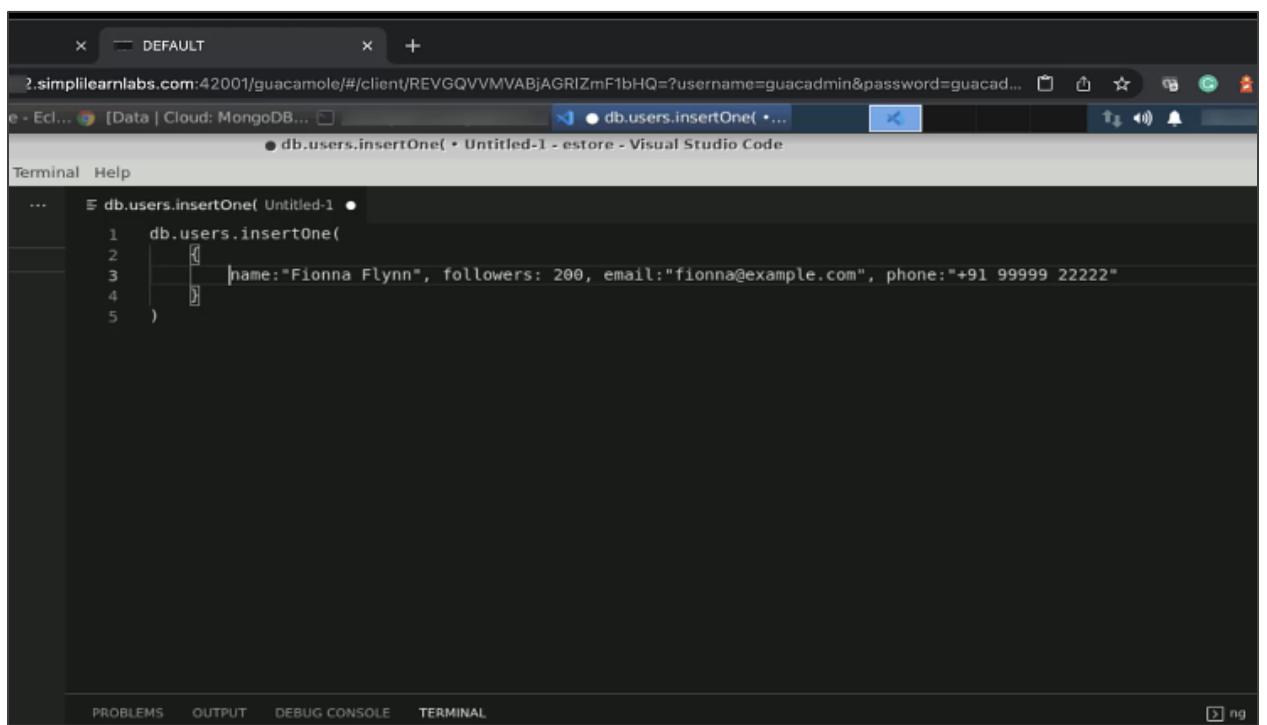
5.3 Create a document as per the JSON standards



A screenshot of a MongoDB client interface. The title bar shows "2.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guacadmin". The main area displays a code editor with the following JSON document:

```
db.users.insertOne( Untitled-1
1   db.users.insertOne(
2     {}
3   )
```

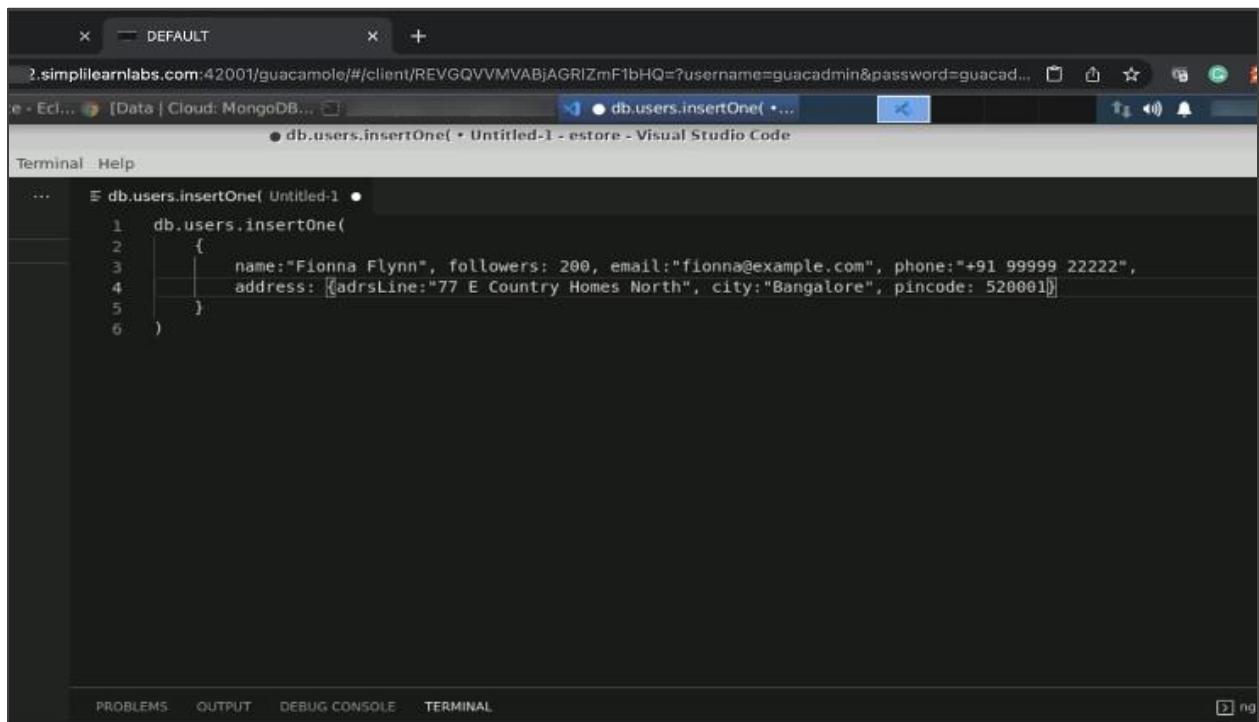
5.4 Give the attributes in key-value pairs, consisting of **name**, **followers**, **email**, and **phone**



A screenshot of a MongoDB client interface. The title bar shows "2.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guacadmin". The main area displays a code editor with the following JSON document:

```
db.users.insertOne( Untitled-1
1   db.users.insertOne(
2     {}
3     |   name:"Fionna Flynn", followers: 200, email:"fionna@example.com", phone:"+91 99999 22222"
4   )
5 )
```

5.5 Add an **address** attribute to represent another object

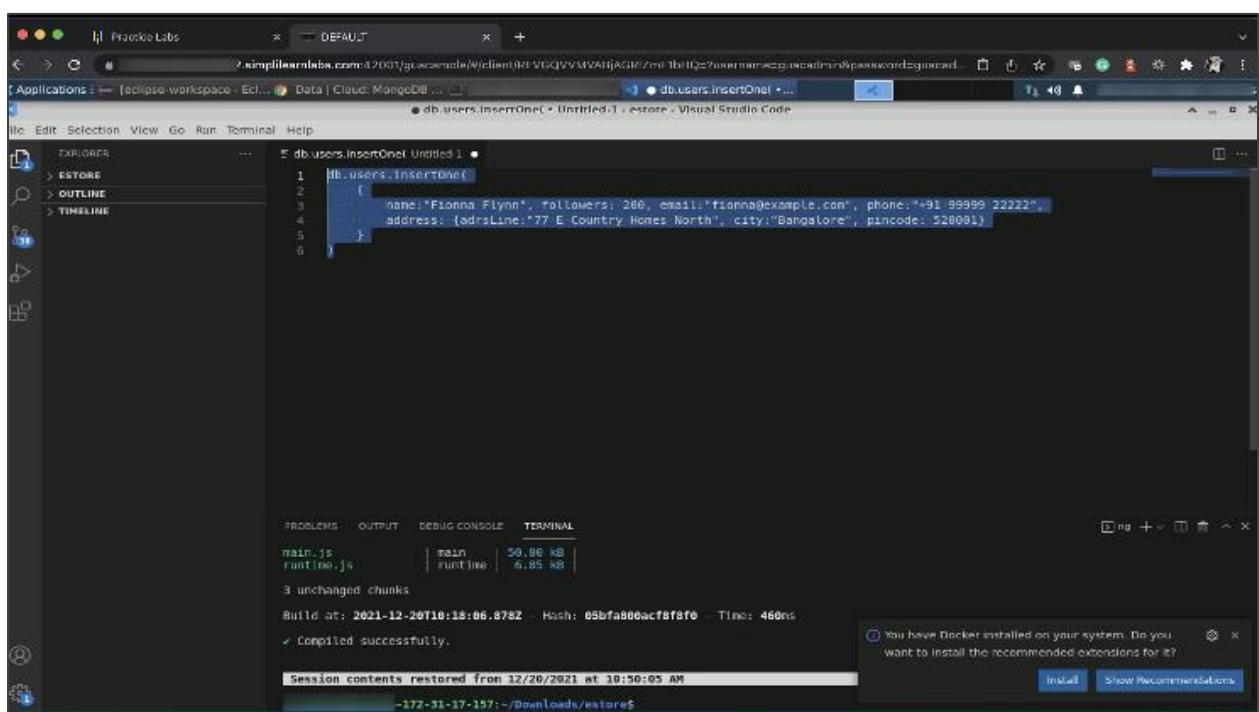


A screenshot of the Visual Studio Code interface. The title bar shows "2.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guacad...". The active tab is "db.users.insertOne(• Untitled-1 - estore - Visual Studio Code)". The code editor contains the following MongoDB insertOne command:

```
1 db.users.insertOne(  
2 {  
3   name:"Fionna Flynn", followers: 200, email:"fionna@example.com", phone:"+91 99999 22222",  
4   address: {adrLine:"77 E Country Homes North", city:"Bangalore", pincode: 520001}  
5 }  
6 )
```

The bottom navigation bar shows "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "TERMINAL".

5.6 Copy the snippet

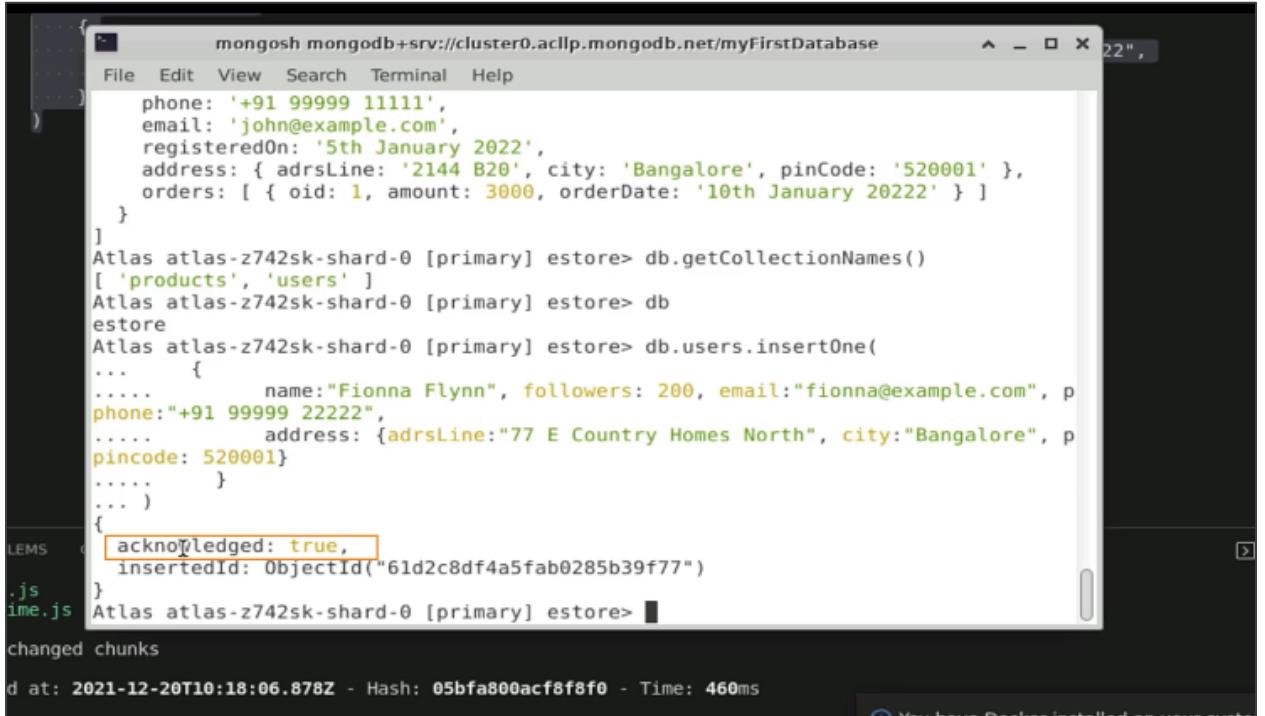


A screenshot of the Visual Studio Code interface, similar to the previous one but with the address object copied. The title bar shows "2.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guacad...". The active tab is "db.users.insertOne(• Untitled-1 - estore - Visual Studio Code)". The code editor contains the same MongoDB insertOne command as before.

The Explorer sidebar on the left shows "ESTORE" and "OUTLINE" under "EXPLORER".

The bottom navigation bar shows "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", and "TERMINAL". The "OUTPUT" tab shows file sizes: "main.js | main | 59.88 KB" and "runtime.js | runtime | 6.85 KB". It also shows "3 unchanged chunks". Below that, it says "Build at: 2021-12-20T08:18:06.878Z Hash: 05bfa886acf8f8f0 Time: 460ms" and "Compiled successfully.". A status bar at the bottom says "Session contents restored from 12/20/2021 at 10:50:05 AM" and "-172-31-27-157--/Downloads/estore\$".

5.7 Paste it into the terminal in **MongoDB Shell** and press Enter

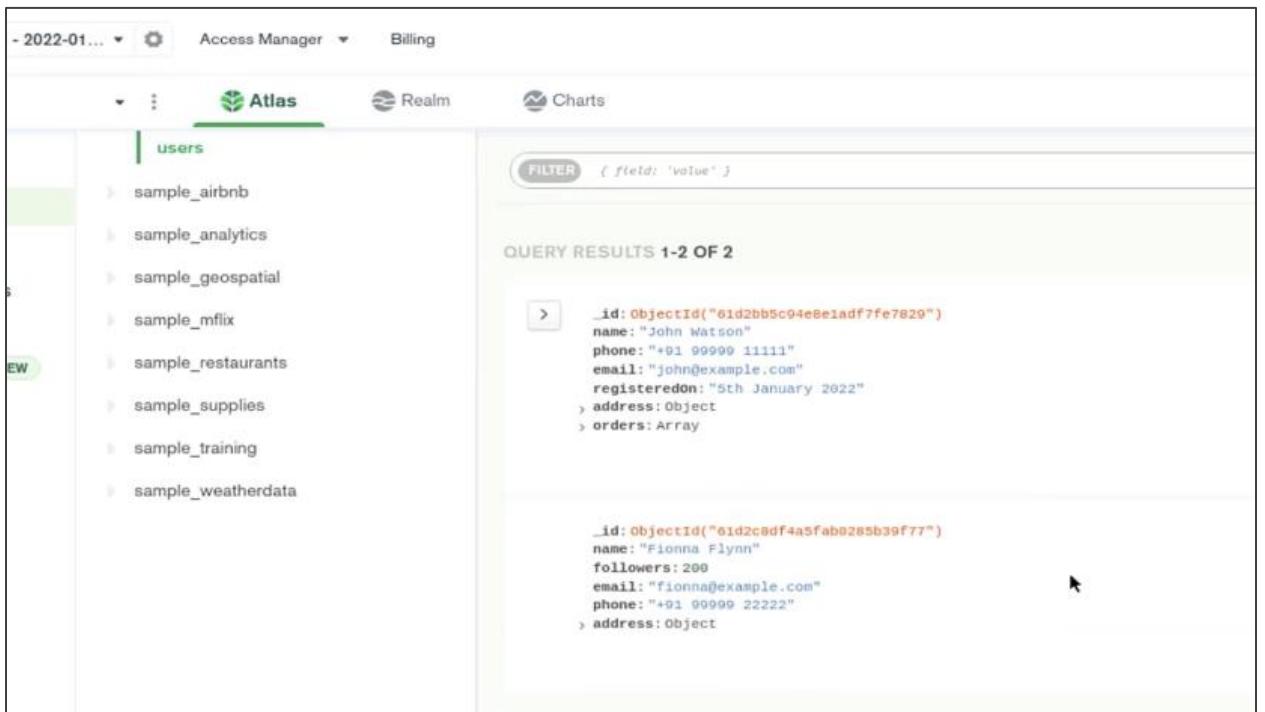


```

mongosh mongodb+srv://cluster0.aclp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
)
  phone: '+91 99999 11111',
  email: 'john@example.com',
  registeredOn: '5th January 2022',
  address: { adrsLine: '2144 B20', city: 'Bangalore', pinCode: '520001' },
  orders: [ { oid: 1, amount: 3000, orderDate: '10th January 2022' } ]
}
Atlas atlas-z742sk-shard-0 [primary] estore> db.getCollectionNames()
[ 'products', 'users' ]
Atlas atlas-z742sk-shard-0 [primary] estore> db
estore
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.insertOne(
...
....      name:"Fionna Flynn", followers: 200, email:"fionna@example.com", p
phone:"+91 99999 22222",
....      address: { adrsLine:"77 E Country Homes North", city:"Bangalore", p
pincode: 520001}
....  }
...
{
  acknowledged: true,
  insertedId: ObjectId("61d2c8df4a5fab0285b39f77")
}
Atlas atlas-z742sk-shard-0 [primary] estore>
changed chunks
d at: 2021-12-20T10:18:06.878Z - Hash: 05bfa800acf8f8f0 - Time: 460ms

```

5.8 Refresh the **users** collection. You can see that another document was created.



The screenshot shows the MongoDB Atlas interface. On the left, a sidebar lists databases: sample_airbnb, sample_analytics, sample_geospatial, sample_mflix, sample_restaurants, sample_supplies, sample_training, and sample_weatherdata. The "users" database is selected and highlighted in green. On the right, the "QUERY RESULTS 1-2 OF 2" section displays two documents:

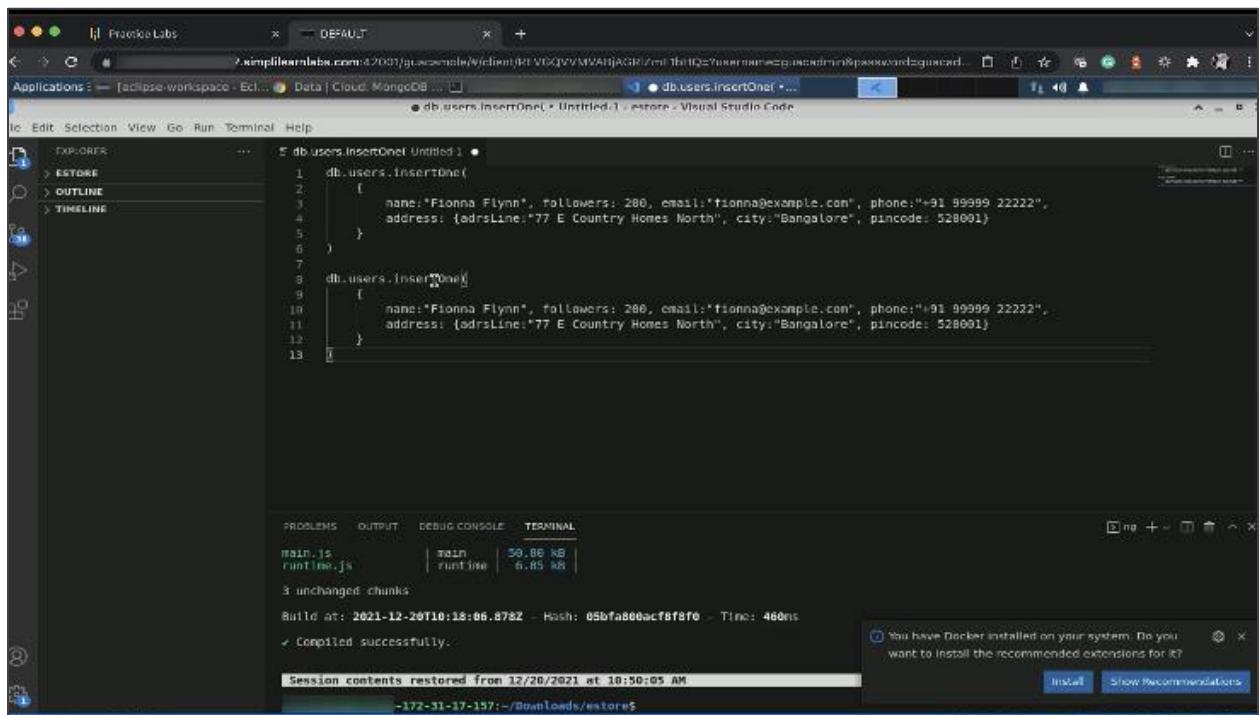
```

_id: ObjectId("61d2bb5c94e8e1adf7fe7829")
name: "John Watson"
phone: "+91 99999 11111"
email: "john@example.com"
registeredOn: "5th January 2022"
address: Object
orders: Array

_id: ObjectId("61d2c8df4a5fab0285b39f77")
name: "Fionna Flynn"
followers: 200
email: "fionna@example.com"
phone: "+91 99999 22222"
address: Object

```

5.9 Paste the same snippet in Visual Studio Code



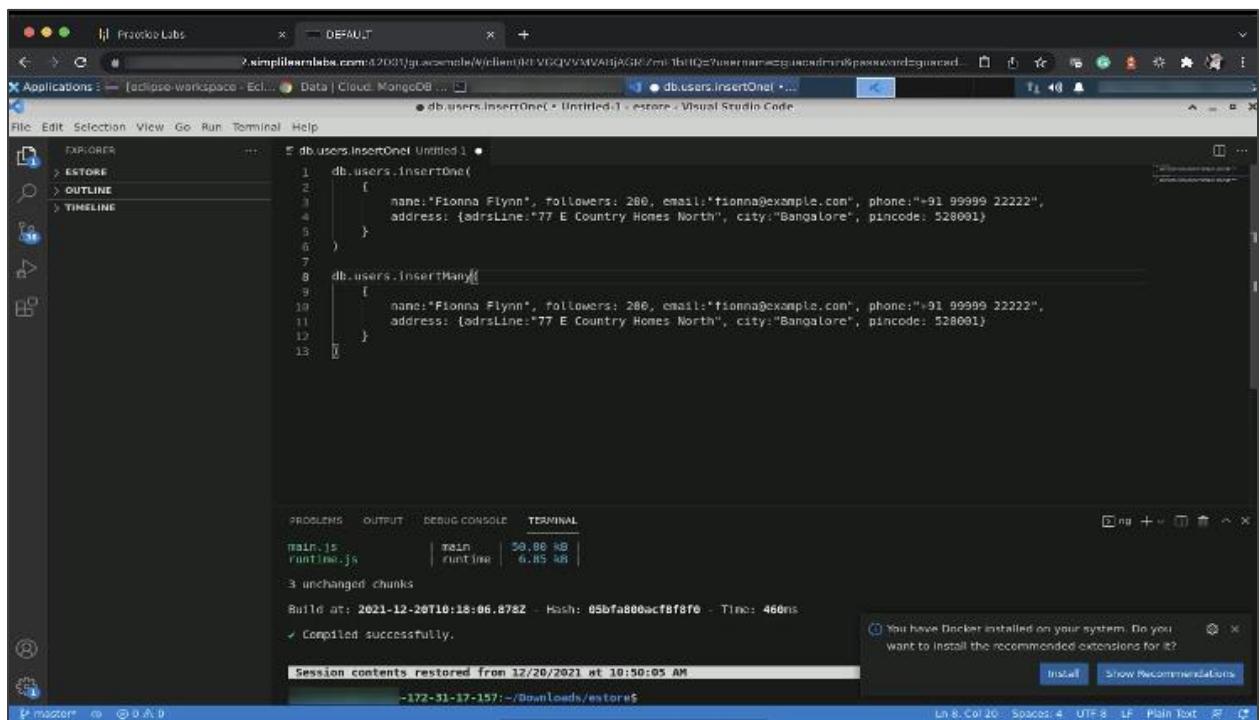
The screenshot shows a Visual Studio Code interface with a dark theme. In the center, there is a terminal window displaying the following MongoDB shell code:

```
db.users.insertOne()
{
  name:"Fionna Flynn", followers: 288, email:"fionna@example.com", phone:"+91 99999 22222",
  address: {adrssLine:"77 E Country Homes North", city:"Bangalore", pincode: 528001}
}

db.users.insertOne()
{
  name:"Fionna Flynn", followers: 288, email:"fionna@example.com", phone:"+91 99999 22222",
  address: {adrssLine:"77 E Country Homes North", city:"Bangalore", pincode: 528001}
}
```

Below the terminal, the Explorer sidebar shows two files: `main.js` and `runtime.js`. The Output panel shows "3 unchanged chunks". The Terminal panel shows the build output: "Build at: 2021-12-20T10:18:06.878Z - Hash: 05bfa800acf8f8f0 - Time: 460ms" and "Compiled successfully.". A tooltip in the bottom right corner suggests installing Docker extensions.

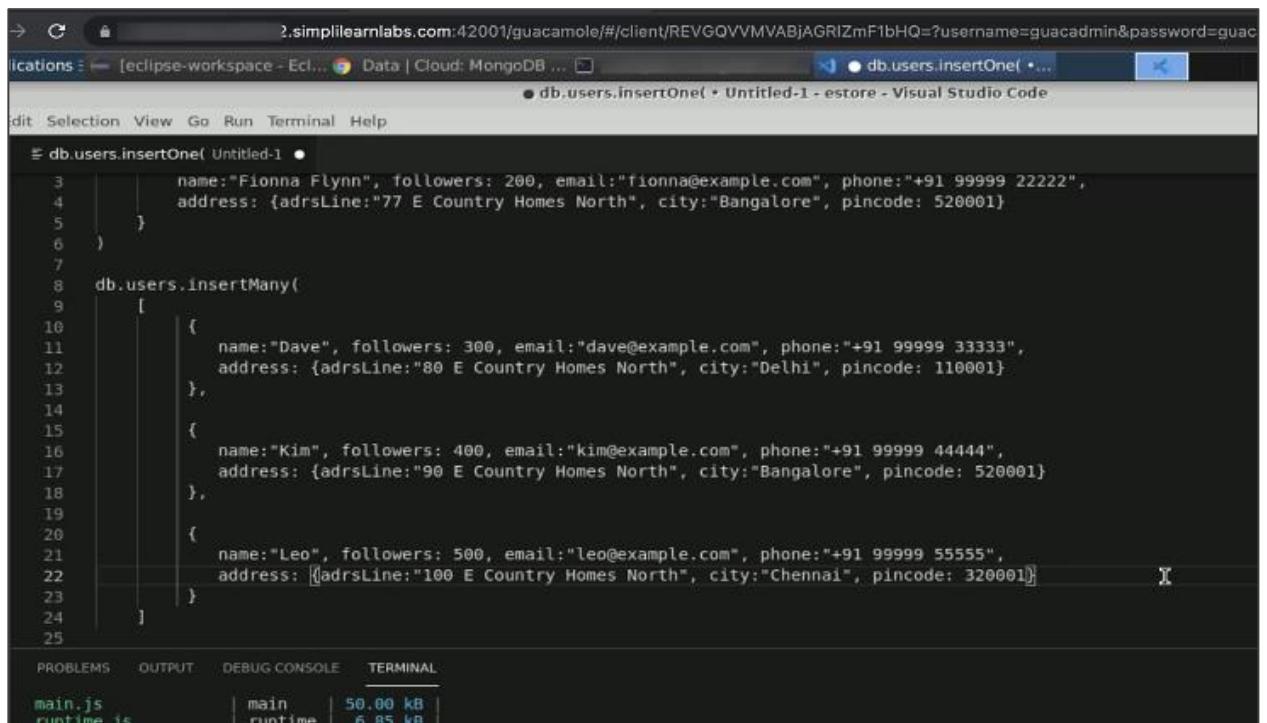
5.10 Use the `db.users.InsertMany()` method to pass an array of documents



This screenshot is identical to the previous one, showing the same MongoDB code in the terminal and the same build output in the terminal and output panels. The Explorer sidebar and status bar are also identical.

Step 6: Insert more documents

6.1 Insert some more documents with different values

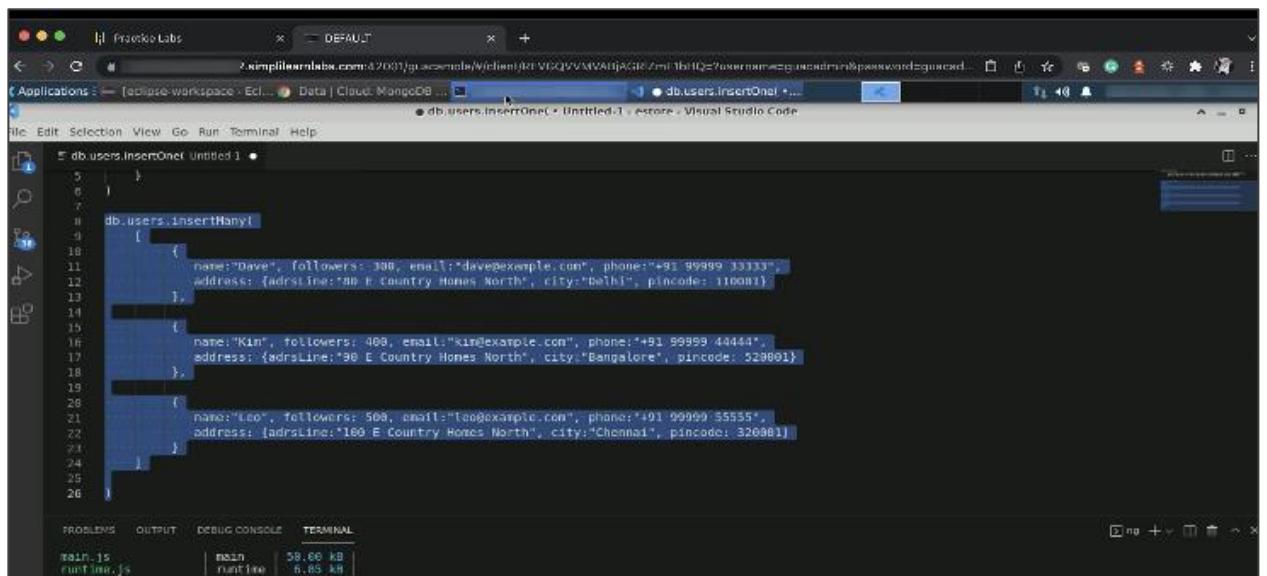


```

2.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guac
Applications: [eclipse-workspace - Eclipse Data | Cloud: MongoDB ...] db.users.insertOne( ... )
dit Selection View Go Run Terminal Help
db.users.insertOne(Untitled-1)
3   {
4     name:"Fionna Flynn", followers: 200, email:"fionna@example.com", phone:"+91 99999 22222",
5     address: {adrsLine:"77 E Country Homes North", city:"Bangalore", pincode: 520001}
6   }
7
8 db.users.insertMany(
9 [
10   {
11     name:"Dave", followers: 300, email:"dave@example.com", phone:"+91 99999 33333",
12     address: {adrsLine:"80 E Country Homes North", city:"Delhi", pincode: 110001}
13   },
14   {
15     name:"Kim", followers: 400, email:"kim@example.com", phone:"+91 99999 44444",
16     address: {adrsLine:"90 E Country Homes North", city:"Bangalore", pincode: 520001}
17   },
18   {
19     name:"Leo", followers: 500, email:"leo@example.com", phone:"+91 99999 55555",
20     address: {adrsLine:"100 E Country Homes North", city:"Chennai", pincode: 320001}
21   }
22 ]
23
24 ]
25
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
main.js | main | 50.00 KB
runtime.js | runtime | 6.85 KB

```

6.2 Copy the code



```

2.simplilearnlabs.com:42001/guacamole/#/client/REVGQVVMVABjAGRIZmF1bHQ=?username=guacadmin&password=guac
Applications: [eclipse-workspace - Eclipse Data | Cloud: MongoDB ...] db.users.insertOne( ... )
dit Selection View Go Run Terminal Help
db.users.insertOne(Untitled-1)
5   }
6   }
7
8 db.users.insertMany(
9 [
10   {
11     name:"Dave", followers: 300, email:"dave@example.com", phone:"+91 99999 33333",
12     address: {adrsLine:"80 E Country Homes North", city:"Delhi", pincode: 110001}
13   },
14   {
15     name:"Kim", followers: 400, email:"kim@example.com", phone:"+91 99999 44444",
16     address: {adrsLine:"90 E Country Homes North", city:"Bangalore", pincode: 520001}
17   },
18   {
19     name:"Leo", followers: 500, email:"leo@example.com", phone:"+91 99999 55555",
20     address: {adrsLine:"100 E Country Homes North", city:"Chennai", pincode: 320001}
21   }
22 ]
23
24 ]
25
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
main.js | main | 50.00 KB
runtime.js | runtime | 6.85 KB

```

6.3 Paste it into the terminal and execute it

```
mongosh mongodb+srv://cluster0.acljp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
...
followers: [
  {
    name: "Kim", followers: 400, email: "kim@example.com", phone: "+91 99999 44444",
    address: {adrLine: "90 E Country Homes North", city: "Bangalore", pincode: 520001},
    ...
  },
  ...
  {
    name: "Leo", followers: 500, email: "leo@example.com", phone: "+91 99999 55555",
    address: {adrLine: "100 E Country Homes North", city: "Chennai", pincode: 320001}
  }
]
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("61d2c99d4a5fab0285b39f78"),
    '1': ObjectId("61d2c99d4a5fab0285b39f79"),
    '2': ObjectId("61d2c99d4a5fab0285b39f7a")
  }
}
Atlas atlas-z742sk-shard-0 [primary] estore>
```

6.4 Refresh the **users** collection page. You can see more documents updated in the database.

The screenshot shows the MongoDB Atlas interface. The left sidebar has sections for Project 0, DEPLOYMENT (selected), DATA SERVICES, SECURITY, and various sample databases. The main area shows the 'Users' collection under the 'estore' database. The 'Find' tab is selected in the top navigation bar. A search bar contains the query `{ field: 'value' }`. Below it, the results are displayed with 1-5 of 5 documents. Each document is shown as an expandable object with fields like `_id`, `name`, `phone`, `email`, `registeredOn`, `address`, and `orders`.

```
_id: ObjectId("61d2bb5c94e8e1ad7fe7fe7829")
name: "John Watson"
phone: "+91 99999 11111"
email: "john@example.com"
registeredOn: "5th January 2022"
address: Object
orders: Array

_id: ObjectId("61d2c0df4a5fafab0285b39f77")
name: "Fionna Flynn"
followers: 200
email: "fionna@example.com"
phone: "+91 50999 22222"
address: Object
```

6.5 Now, execute the find function on users using the **db.users.find()** function

```

{
  "city": "Bangalore",
  "pincode": 520001
},
{
  "_id": ObjectId("61d2c99d4a5fab0285b39f78"),
  "name": "Dave",
  "followers": 300,
  "email": "dave@example.com",
  "phone": "+91 99999 33333",
  "address": {
    "adrLine": "80 E Country Homes North",
    "city": "Delhi",
    "pincode": 110001
  }
},
{
  "_id": ObjectId("61d2c99d4a5fab0285b39f79"),
  "name": "Kim",
  "followers": 400,
  "email": "kim@example.com",
  "phone": "+91 99999 44444",
  "address": {
    "adrLine": "90 E Country Homes North",
    "city": "Bangalore",
    "pincode": 560001
  }
}

```

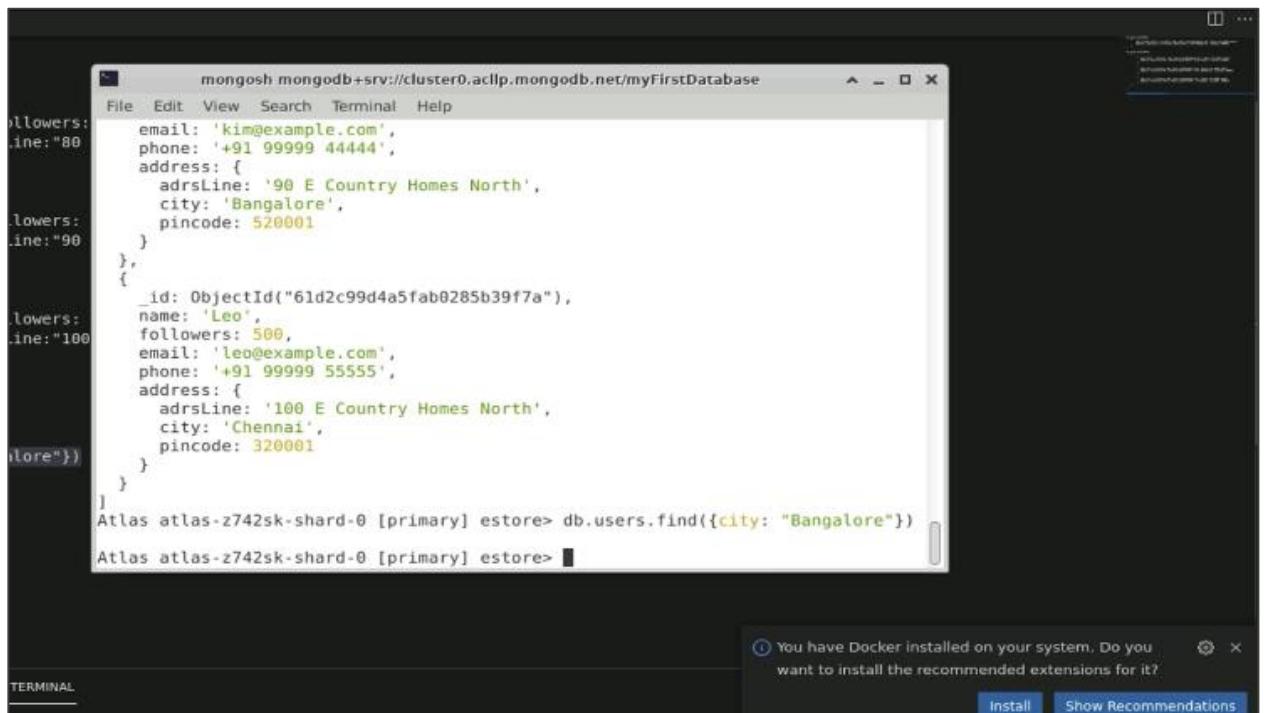
6.6 Use the **db.users.find(city: "Bangalore")** query to find users with the city as Bengaluru

```

db.users.insertOne({
  name: "Dave", followers: 300, email: "dave@example.com", phone: "+91 99999 33333",
  address: {adrLine: "80 E Country Homes North", city: "Delhi", pincode: 110001}
}),
{
  name: "Kim", followers: 400, email: "kim@example.com", phone: "+91 99999 44444",
  address: {adrLine: "90 E Country Homes North", city: "Bangalore", pincode: 560001}
}
]
db.users.find([{"city": "Bangalore"}])

```

6.7 Execute the query in the terminal



```

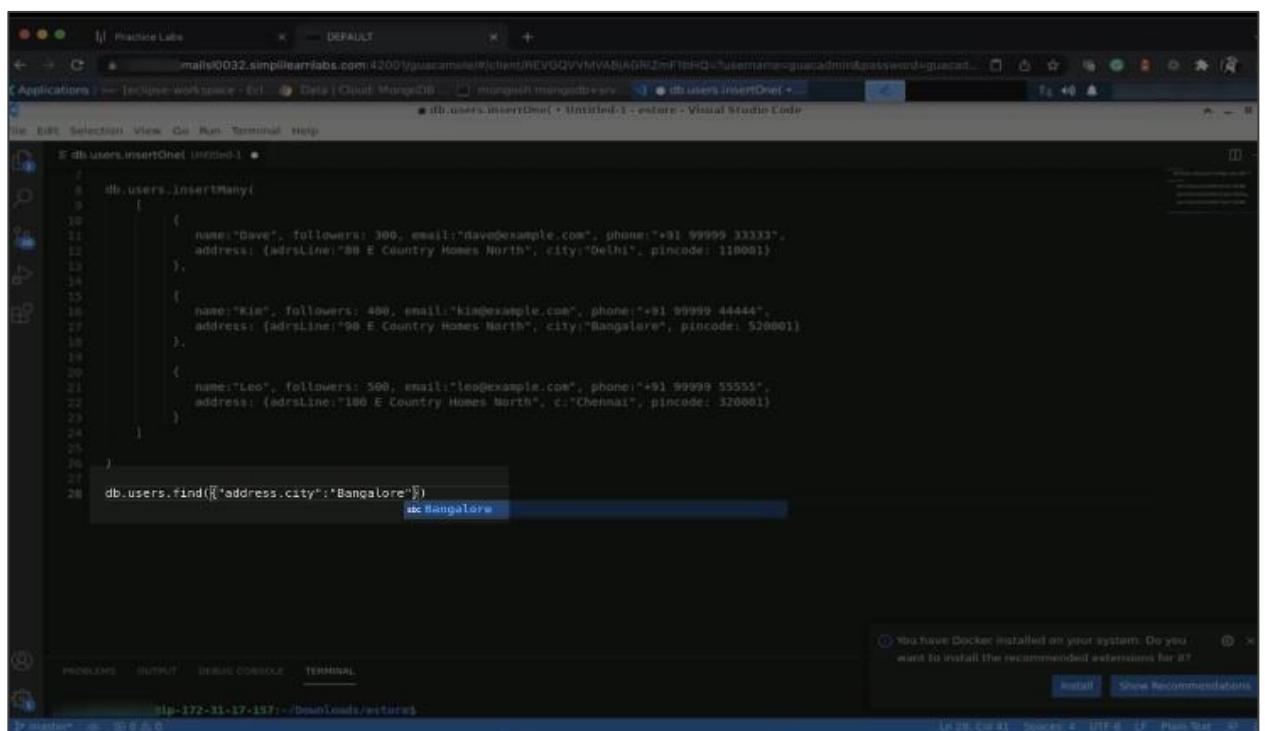
mongosh mongodb+srv://cluster0.aclp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
followers:
line: "80
email: 'kim@example.com',
phone: '+91 99999 44444',
address: {
  adrLine: '90 E Country Homes North',
  city: 'Bangalore',
  pincode: 520001
},
{
  _id: ObjectId("61d2c99d4a5fab0285b39f7a"),
  name: 'Leo',
  followers: 500,
  email: 'leo@example.com',
  phone: '+91 99999 55555',
  address: {
    adrLine: '100 E Country Homes North',
    city: 'Chennai',
    pincode: 320001
  }
}
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({city: "Bangalore"})
Atlas atlas-z742sk-shard-0 [primary] estore>

```

You have Docker installed on your system. Do you want to install the recommended extensions for it?

Install Show Recommendations

6.8 Change the key value to address.city



```

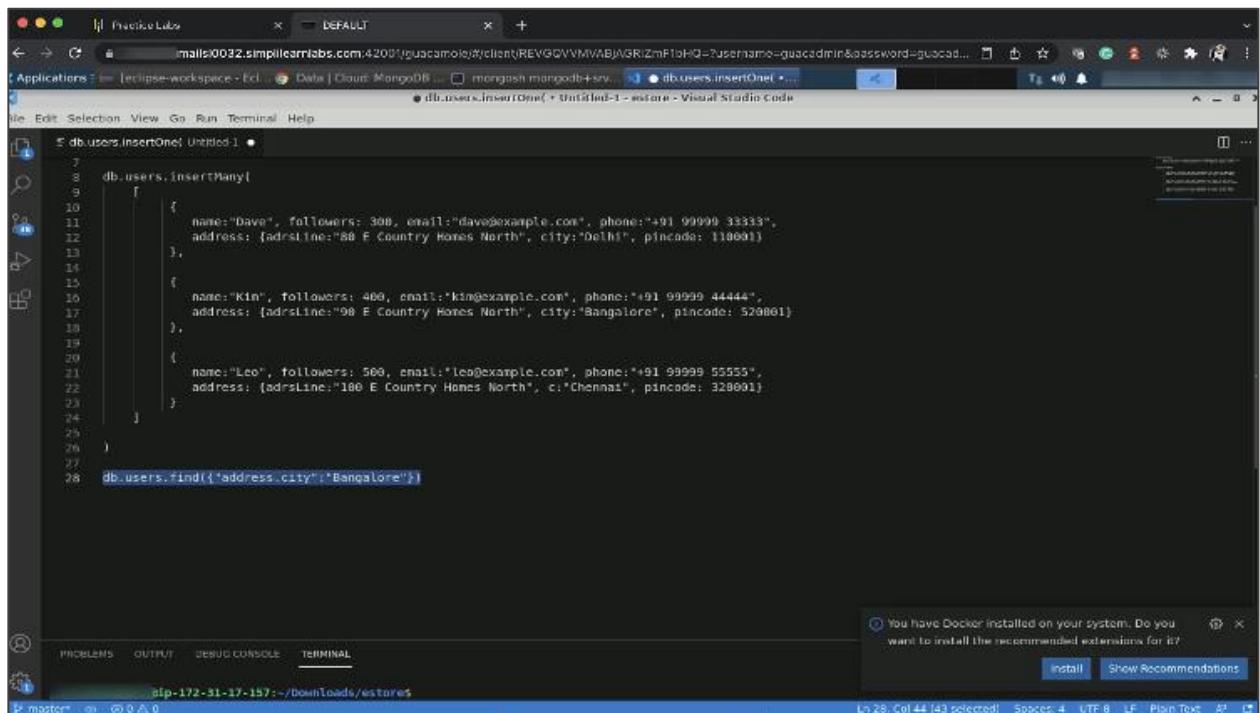
db.users.insertOne( {
  name: "Dave", followers: 300, email: "dave@example.com", phone: "+91 99999 33333",
  address: {adrLine: "80 E Country Homes North", city: "Delhi", pincode: 110001}
},
{
  name: "Kim", followers: 400, email: "kim@example.com", phone: "+91 99999 44444",
  address: {adrLine: "90 E Country Homes North", city: "Bangalore", pincode: 520001}
},
{
  name: "Leo", followers: 500, email: "leo@example.com", phone: "+91 99999 55555",
  address: {adrLine: "100 E Country Homes North", city: "Chennai", pincode: 320001}
}
)
db.users.find({address.city: "Bangalore"})

```

You have Docker installed on your system. Do you want to install the recommended extensions for it?

Install Show Recommendations

6.9 Copy the query

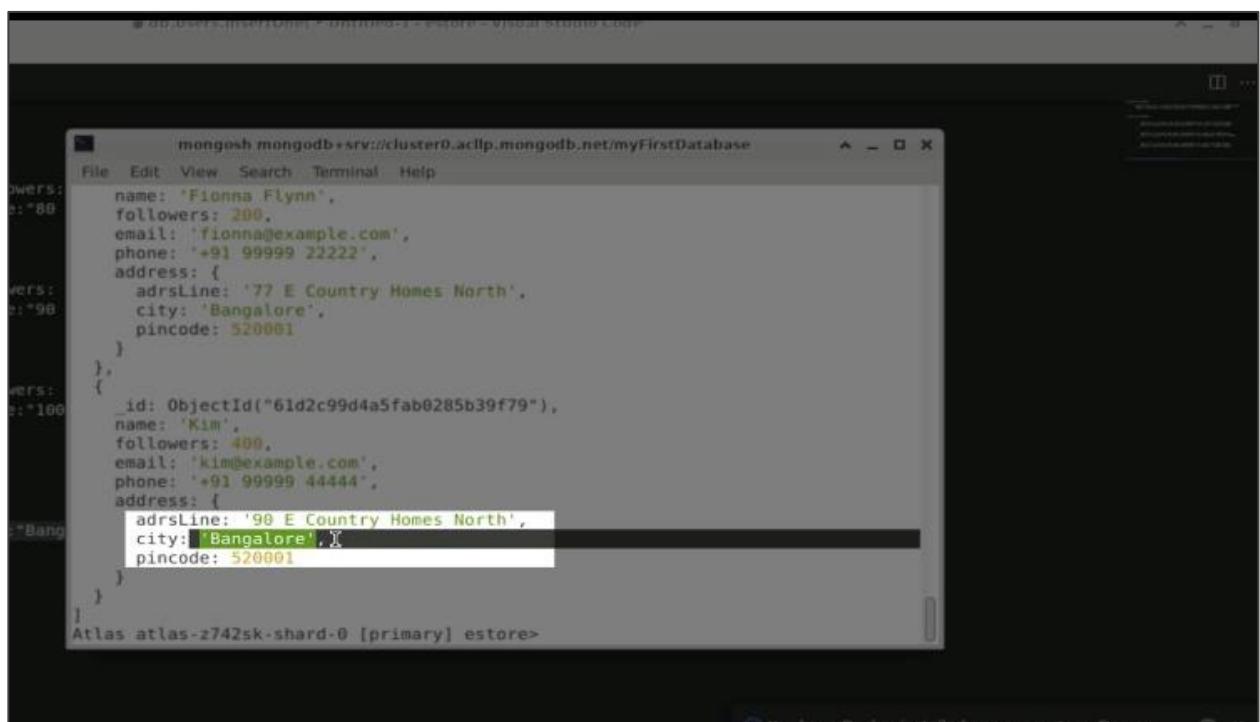


A screenshot of the Visual Studio Code interface. The title bar shows the URL `https://0032.simplilearncode.com:4200/quacamole/client/REVGQVVVMVABWGRZmF1bHQ?username=quacadmin&password=quacadmin`. The left sidebar has a tree view with a file named `db.users.insertOne.js`. The main editor area contains the following MongoDB query:

```
db.users.insertOne( {
  name: "Dave",
  followers: 300,
  email: "dave@example.com",
  phone: "+91 99999 33333",
  address: { addressLine: "88 E Country Homes North", city: "Delhi", pincode: 110001 }
},
{
  name: "Kim",
  followers: 400,
  email: "kim@example.com",
  phone: "+91 99999 44444",
  address: { addressLine: "98 E Country Homes North", city: "Bangalore", pincode: 520001 }
},
{
  name: "Leo",
  followers: 500,
  email: "leo@example.com",
  phone: "+91 99999 55555",
  address: { addressLine: "100 E Country Homes North", city: "Chennai", pincode: 320001 }
}
)
db.users.find({ "address.city": "Bangalore" })
```

The bottom status bar shows the path `slip-172-31-17-187:~/Downloads/estore`, the terminal mode indicator, and the file `db.users.insertOne.js`.

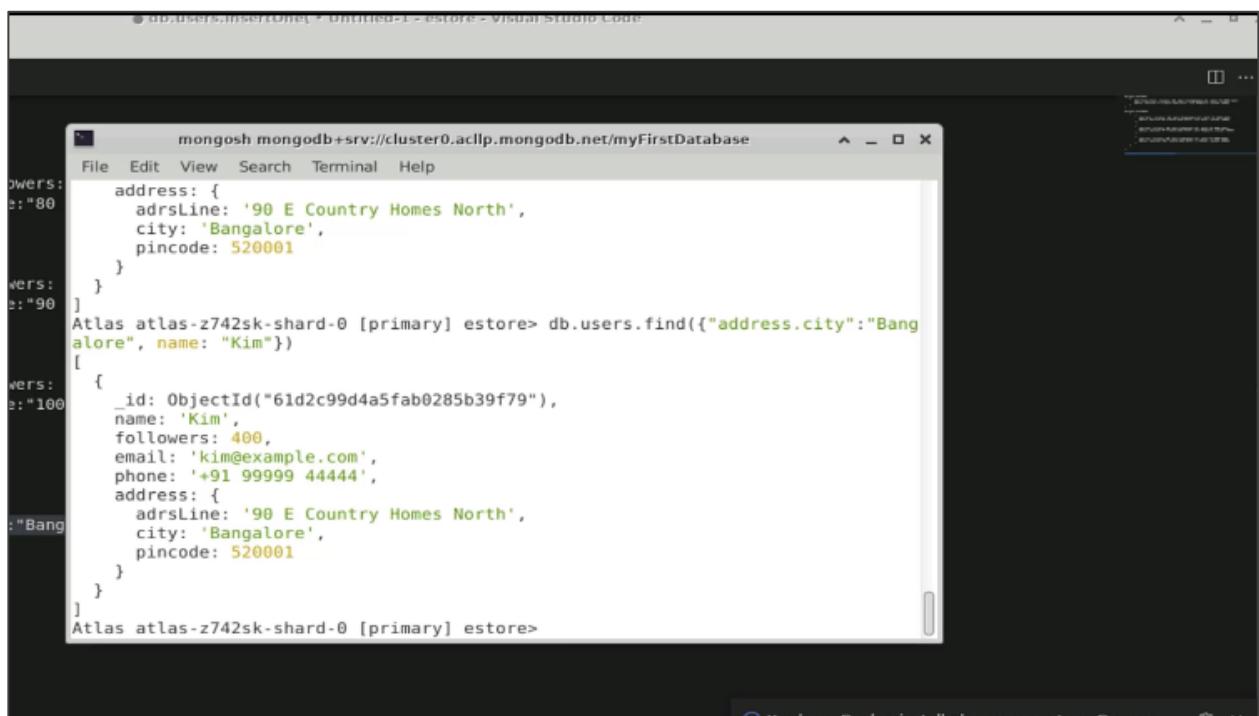
6.10 Paste it into the terminal



A screenshot of a terminal window titled `mongosh mongodb+srv://cluster0.aclp.mongodb.net/myFirstDatabase`. The window displays the MongoDB query from the previous step. The cursor is positioned at the end of the query, specifically after the closing brace of the third user object.

```
File Edit View Search Terminal Help
users: [
  {
    name: "Fionna Flynn",
    followers: 200,
    email: "fionna@example.com",
    phone: "+91 99999 22222",
    address: {
      addressLine: "77 E Country Homes North",
      city: "Bangalore",
      pincode: 520001
    }
  },
  {
    _id: ObjectId("61d2c99d4a5fab0285b39f79"),
    name: "Kim",
    followers: 400,
    email: "kim@example.com",
    phone: "+91 99999 44444",
    address: {
      addressLine: "98 E Country Homes North",
      city: "Bangalore",
      pincode: 520001
    }
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore>
```

6.11 Add a find statement to list the results of user Kim in Bengaluru



```

mongosh mongodb+srv://cluster0.aclp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help

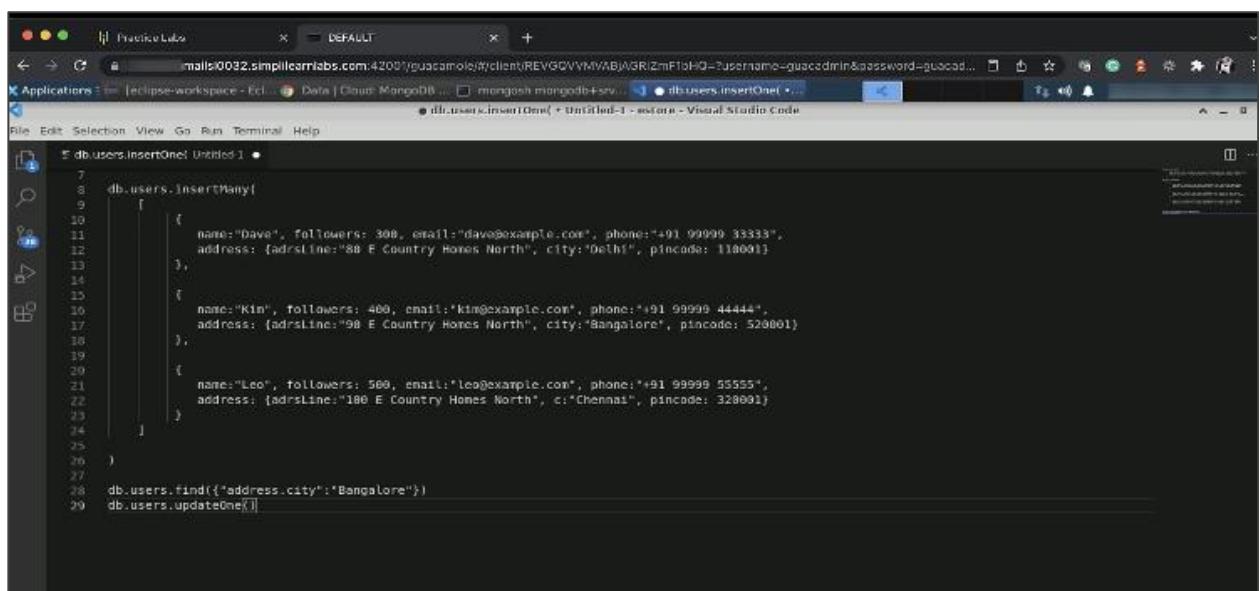
powers: [
  {
    address: {
      adrLine: '90 E Country Homes North',
      city: 'Bangalore',
      pincode: 520001
    }
  }
]
vers: [
  {
    address: {
      adrLine: '90 E Country Homes North',
      city: 'Bangalore',
      pincode: 520001
    }
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({"address.city": "Bangalore", name: "Kim"})
[{
  _id: ObjectId("61d2c99d4a5fab0285b39f79"),
  name: 'Kim',
  followers: 400,
  email: 'kim@example.com',
  phone: '+91 99999 44444',
  address: {
    adrLine: '90 E Country Homes North',
    city: 'Bangalore',
    pincode: 520001
  }
}]
Atlas atlas-z742sk-shard-0 [primary] estore>

```

You have Docker installed on your system. Do you want to use it? [Y/n]

Step 7: Perform update operations

7.1 Perform the update operation using the `updateOne()` function



```

File Edit Selection View Go Run Terminal Help
db.users.insertOne()
7
8 db.users.insertMany([
9
10   {
11     name: "Dave", followers: 300, email: "dave@example.com", phone: "+91 99999 33333",
12     address: {adrLine: "80 E Country Homes North", city: "Delhi", pincode: 110001}
13   },
14   {
15     name: "Kim", followers: 400, email: "kim@example.com", phone: "+91 99999 44444",
16     address: {adrLine: "90 E Country Homes North", city: "Bangalore", pincode: 520001}
17   },
18   {
19     name: "Leo", followers: 500, email: "leo@example.com", phone: "+91 99999 55555",
20     address: {adrLine: "100 E Country Homes North", city: "Chennai", pincode: 328001}
21   }
22 ]
23 )
24
25
26 )
27
28 db.users.find({"address.city": "Bangalore"})
29 db.users.updateOne()

```

7.2 Enter conditions to update, such as email

```
 7
 8 db.users.insertOne( Untitled-1 •
 9 [
10   {
11     name:"Dave", followers: 300, email:"dave@example.com", phone:"+91 99999 33333",
12     address: {adrsLine:"80 E Country Homes North", city:"Delhi", pincode: 110001}
13   },
14   {
15     name:"Kim", followers: 400, email:"kim@example.com", phone:"+91 99999 44444",
16     address: {adrsLine:"90 E Country Homes North", city:"Bangalore", pincode: 520001}
17   },
18   {
19     name:"Leo", followers: 500, email:"leo@example.com", phone:"+91 99999 55555",
20     address: {adrsLine:"100 E Country Homes North", c:"Chennai", pincode: 320001}
21   }
22 ]
23 )
24 )
25
26 )
27 db.users.find({"address.city":"Bangalore"})
28 db.users.updateOne(
29   [email: "leo@example.com"]
30 )
31 )
```

7.3 Set the details to update using the set variable that specifies the followers and pincode (zip code)

```
 7
 8 db.users.insertOne( Untitled-1 •
 9 [
10   {
11     name:"Dave", followers: 300, email:"dave@example.com", phone:"+91 99999 33333",
12     address: {adrsLine:"80 E Country Homes North", city:"Delhi", pincode: 110001}
13   },
14   {
15     name:"Kim", followers: 400, email:"kim@example.com", phone:"+91 99999 44444",
16     address: {adrsLine:"90 E Country Homes North", city:"Bangalore", pincode: 520001}
17   },
18   {
19     name:"Leo", followers: 500, email:"leo@example.com", phone:"+91 99999 55555",
20     address: {adrsLine:"100 E Country Homes North", c:"Chennai", pincode: 320001}
21   }
22 ]
23 )
24 )
25
26 )
27 db.users.find({"address.city":"Bangalore"})
28 db.users.updateOne(
29   [email: "leo@example.com"]
30 )
31 )
```

7.4 Copy the snippet

```

db.users.insertOne( {
  name: "Dave",
  followers: 300,
  email: "dave@example.com",
  phone: "+91 99999 33333",
  address: {adrssLine: "88 E Country Homes North", city: "Delhi", pincode: 110001}
},
{
  name: "Kim",
  followers: 400,
  email: "kim@example.com",
  phone: "+91 99999 44444",
  address: {adrssLine: "98 E Country Homes North", city: "Bangalore", pincode: 520001}
},
{
  name: "Leo",
  followers: 500,
  email: "leo@example.com",
  phone: "+91 99999 55555",
  address: {adrssLine: "100 E Country Homes North", c: "Chennai", pincode: 320001}
}
)

db.users.find({address.city:"Bangalore"})
db.users.updateOne(
  {email: "leo@example.com"},
  {
    $set: {followers: 3700, "address.pincode":520001}
  }
)

```

You have Docker installed on your system. Do you want to install the recommended extensions for it?

7.5 Paste it and press Enter. You can see that the acknowledgment is true.

```

mongosh mongodb+srv://cluster0.aclp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
Followers: 400, email: 'kim@example.com', phone: '+91 99999 44444', address: {adrssLine: '98 E Country Homes North', city: 'Bangalore', pincode: 520001}
Followers: 100
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.updateOne(
...   {email: "leo@example.com"}, ...
...   {
...     $set: {followers: 3700, "address.pincode":520001}
...   }
...
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
)
Atlas atlas-z742sk-shard-0 [primary] estore>

```

You have Docker installed on your system. Do you want to install the recommended extensions for it?

7.6 Write `db.deleteOne()` and add a condition to delete one of the documents

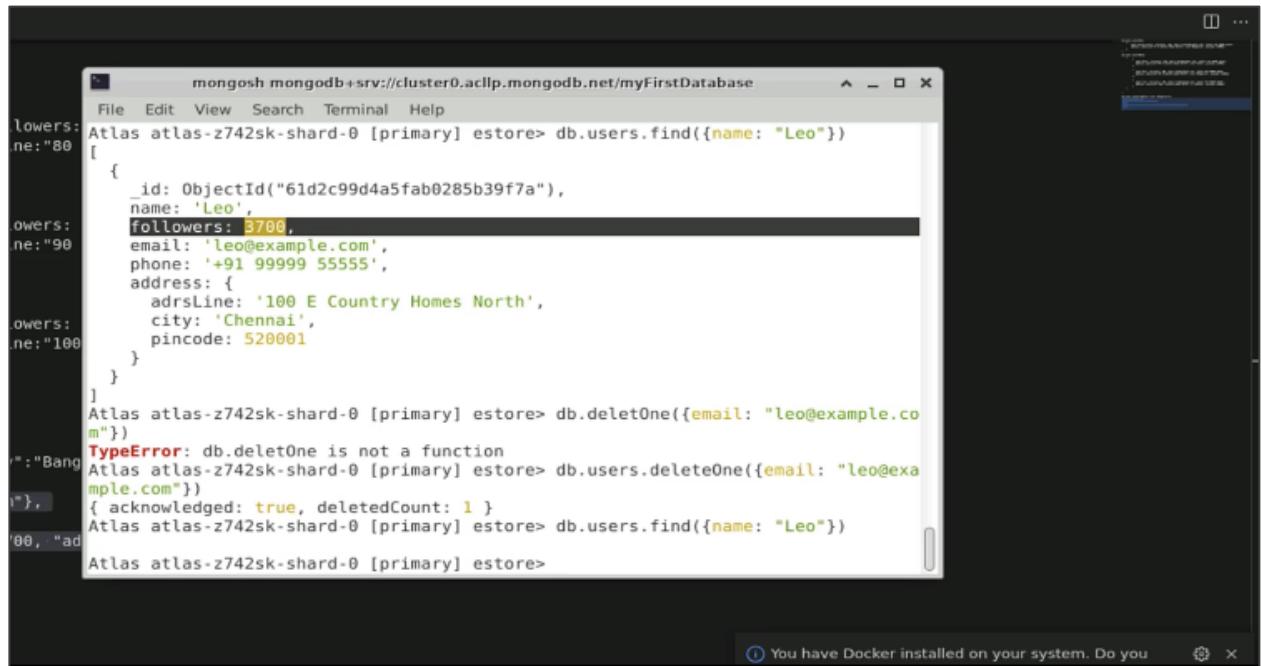
```
mongosh mongodb+srv://cluster0.acIIP.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
> 1 | db.users.find({name: "Leo"})
| 2 |
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({name: "Leo"})
[
  {
    _id: ObjectId("61d2c99d4a5fab0285b39f7a"),
    name: 'Leo',
    followers: 3700,
    email: 'leo@example.com',
    phone: '+91 99999 55555',
    address: {
      addressLine: '100 E Country Homes North',
      city: 'Chennai',
      pincode: 520001
    }
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore> db.deleteOne({email: "leo@example.com"})
TypeError: db.deleteOne is not a function
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.deleteOne({email: "leo@example.com"})
700, "ad"
You have Docker installed on your system. Do you want to use it? [Y/n]
```

7.7 Update the query as `db.users.deleteOne()` and press Enter

```
mongosh mongodb+srv://cluster0.acIIP.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
2 |
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({name: "Leo"})
[
  {
    _id: ObjectId("61d2c99d4a5fab0285b39f7a"),
    name: 'Leo',
    followers: 3700,
    email: 'leo@example.com',
    phone: '+91 99999 55555',
    address: {
      addressLine: '100 E Country Homes North',
      city: 'Chennai',
      pincode: 520001
    }
  }
]
Atlas atlas-z742sk-shard-0 [primary] estore> db.deleteOne({email: "leo@example.com"})
TypeError: db.deleteOne is not a function
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.deleteOne({email: "leo@example.com"})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-z742sk-shard-0 [primary] estore>
You have Docker installed on your system. Do you want to use it? [Y/n]
```

You can see that the acknowledgment is **true** and one of the documents is deleted.

7.8 Execute another command named **Leo**



```
mongosh mongodb+srv://cluster0.aclp.mongodb.net/myFirstDatabase
File Edit View Search Terminal Help
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({name: "Leo"})
[{"_id": ObjectId("61d2c99d4a5fab0285b39f7a"),
  "name": "Leo",
  "followers": 3700,
  "email": "leo@example.com",
  "phone": "+91 99999 55555",
  "address": {
    "addressLine": "100 E Country Homes North",
    "city": "Chennai",
    "pincode": 520001
  }
}
Atlas atlas-z742sk-shard-0 [primary] estore> db.deleteOne({email: "leo@example.com"})
TypeError: db.deleteOne is not a function
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.deleteOne({email: "leo@example.com"})
{ acknowledged: true, deletedCount: 1 }
Atlas atlas-z742sk-shard-0 [primary] estore> db.users.find({name: "Leo"})
Atlas atlas-z742sk-shard-0 [primary] estore>
```

The terminal window shows the MongoDB shell interface. It first runs a find command to search for a user named 'Leo'. The result is a document with fields like '_id', 'name', 'followers', 'email', 'phone', and 'address'. Then, it attempts to run a 'db.deleteOne' command, which results in a 'TypeError' because 'deleteOne' is not a function. Finally, it runs a 'db.users.deleteOne' command with the same filter, which successfully deletes the document and returns an acknowledgment object with 'acknowledged: true' and 'deletedCount: 1'. A message at the bottom of the terminal window says 'You have Docker installed on your system. Do you'.

No such document is available.

These CRUD operations can be executed and will remain the same for any API. This applies to programming languages as well, for which these methods and functions remain the same.

By following these steps, you have successfully performed CRUD operations with Mongo Shell by installing mongosh and performing update operations.