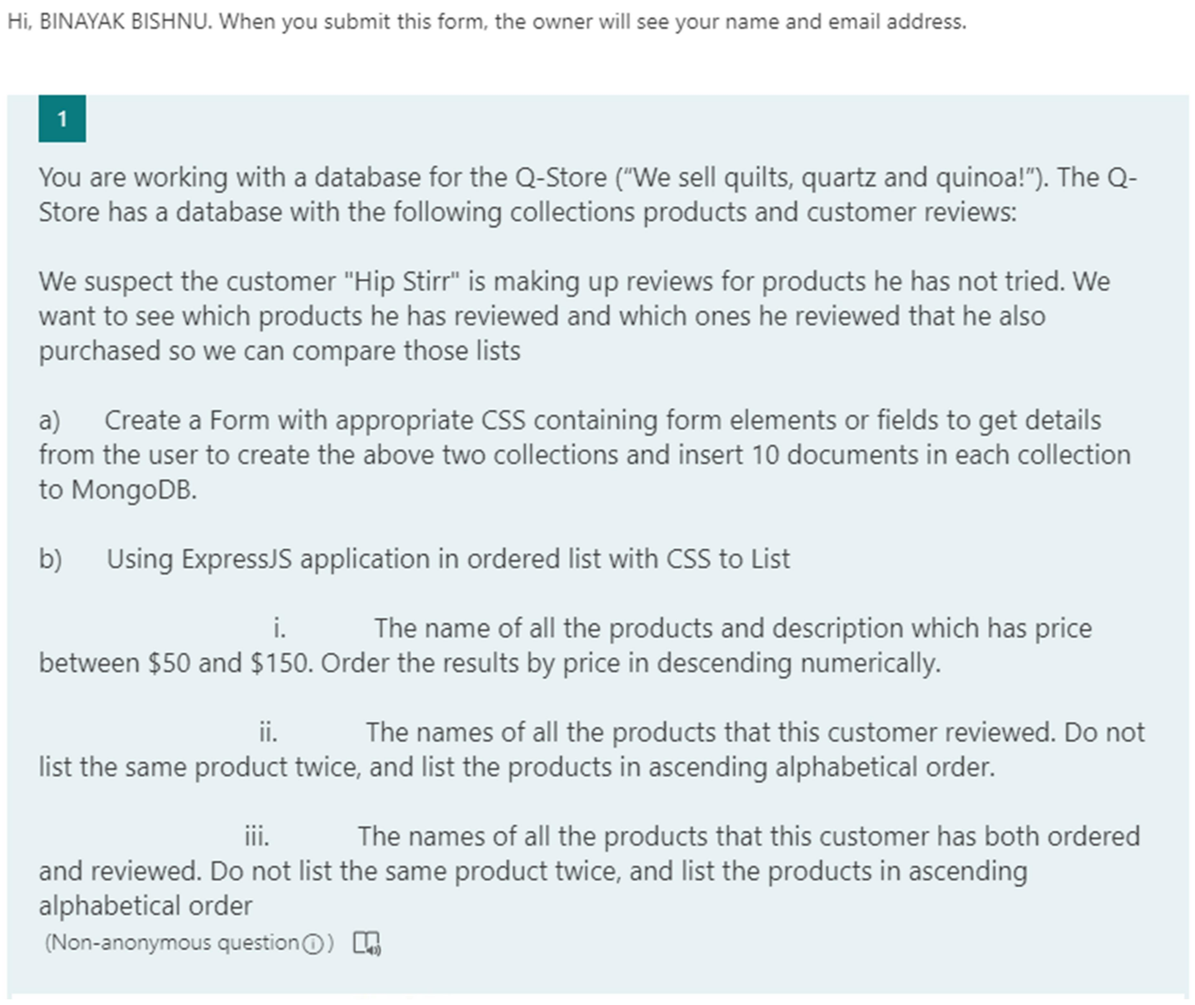
Webtech Lab FAT

Binayak Bishnu

20BIT0155

Question



Hi, BINAYAK BISHNU. When you submit this form, the owner will see your name and email address.

1

You are working with a database for the Q-Store ("We sell quilts, quartz and quinoa!"). The Q-Store has a database with the following collections products and customer reviews:

We suspect the customer "Hip Stirr" is making up reviews for products he has not tried. We want to see which products he has reviewed and which ones he reviewed that he also purchased so we can compare those lists

a) Create a Form with appropriate CSS containing form elements or fields to get details from the user to create the above two collections and insert 10 documents in each collection to MongoDB.

b) Using ExpressJS application in ordered list with CSS to List

i. The name of all the products and description which has price between $50 and $150. Order the results by price in descending numerically.

ii. The names of all the products that this customer reviewed. Do not list the same product twice, and list the products in ascending alphabetical order.

iii. The names of all the products that this customer has both ordered and reviewed. Do not list the same product twice, and list the products in ascending alphabetical order

(Non-anonymous question)

Code

HTML

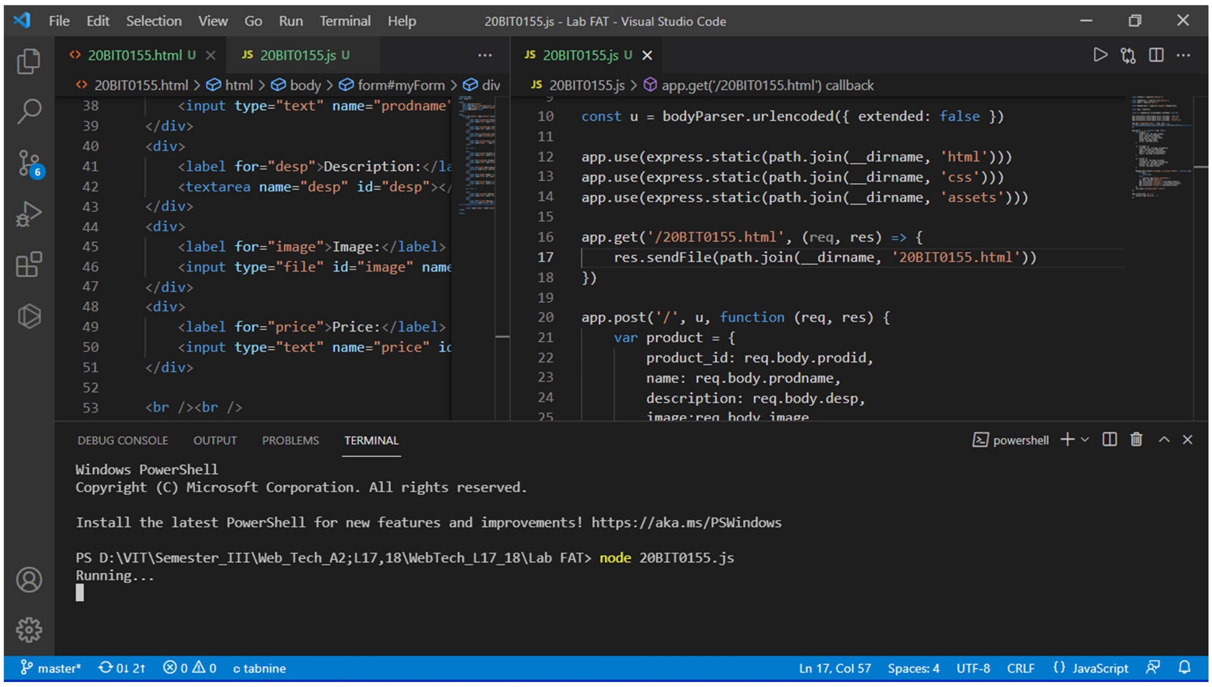
```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>20BIT0155</title>
</head>

<body>
    <form id="myForm" name="myForm" action="/submitted" method="POST">
        <div>
            <label for="customerid">Customer ID:</label>
            <input type="text" name="customerid" id="customerid" />
        </div>
        <div>
            <label for="customername">Customer:</label>
            <input type="text" name="customername" id="customername" />
        </div>
        <div>
            <label for="customeraddr">Customer Address:</label>
            <textarea name="customeraddr" id="customeraddr"></textarea>
        </div>
        <div>
            <label for="customeremail">Customer Email ID:</label>
            <input type="email" name="customeremail" id="customeremail" />
        </div>

        <br /><br />

        <div>
            <label for="prodid">Product ID:</label>
            <input type="text" name="prodid" id="prodid" />
        </div>
        <div>
            <label for="prodname">Product:</label>
            <input type="text" name="prodname" id="prodname" />
        </div>
        <div>
            <label for="desp">Description:</label>
            <textarea name="desp" id="desp"></textarea>
        </div>
        <div>
            <label for="image">Image:</label>
            <input type="file" id="image" name="image" />
        </div>
        <div>
            <label for="price">Price:</label>
            <input type="number" name="price" id="price" />
        </div>

        <br /><br />

        <div>
            <label for="reviewid">Review ID:</label>
            <input type="text" name="reviewid" id="reviewid" />
        </div>
        <div>
            <label for="review">Review:</label>
            <textarea name="review" id="review"></textarea>
        </div>

        <input type="submit" name="submit" value="Submit" id="submit" />
        <br /><br />

    </form>

    <div>
        <h3>Queries</h3>
        <form action="/part_i" method="GET">
            <input type="submit" name="part_i" value="Part i" />
        </form>
        <form action="/part_ii" method="GET">
            <input type="submit" name="part_ii" value="Part ii" />
        </form>
        <form action="/part_iii" method="GET">
            <input type="submit" name="part_iii" value="Part iii" />
        </form>
    </div>
</body>

</html>
```

Server

```
const express = require('express');

const bodyParser = require('body-parser');
const path = require('path');

const MongoClient = require('mongodb').MongoClient;

const app = express()

const u = bodyParser.urlencoded({ extended: false })

app.use(express.static(path.join(__dirname, 'html')))
app.use(express.static(path.join(__dirname, 'css')))
app.use(express.static(path.join(__dirname, 'assets')))

app.get('/20BIT0155.html', (req, res) => {
    res.sendFile(path.join(__dirname, '20BIT0155.html'))
})

app.post('/submitted', u, function (req, res) {
    var product = {
        product_id: req.body.prodid,
        name: req.body.prodname,
        description: req.body.desp,
        image: req.body.image,
        price: req.body.price,
    }

    var customer = {
        customer_id: req.body.customerid,
        name: req.body.customername,
        address: req.body.customeraddr,
        email: req.body.customeremail,
    }

    var review = {
        review_id: req.body.reviewid,
        customer_id: req.body.customerid,
        product_id: req.body.prodid,
        review: req.body.review,
    }

    MongoClient.connect('mongodb://localhost:27017/', function (err, db) {
        if (err)
            throw err;
        else
            console.log("mongo connected!")
        var dbo = db.db("QStore");
        dbo.collection('products').insertOne(product);
        dbo.collection('customers').insertOne(customer);
        dbo.collection('reviews').insertOne(review);
    })
    res.send("<h1>Submitted!!!</h1>")
})

app.get('/part_i', u, function (req, res) {
    MongoClient.connect('mongodb://localhost:27017/', function (err, db) {
        if (err) throw err;

        var dbo = db.db("QStore");

        var query1 = { price: {$gte:25,$lte:150} };

        dbo.collection("products").find(query1).toArray(function (err, result)
{
            if (err) throw err;

            console.log(result);
            res.send(result);
            db.close();
        });
    });
})

app.get('/part_ii', u, function (req, res) {
    MongoClient.connect('mongodb://localhost:27017/', function (err, db) {
        if (err) throw err;

        var dbo = db.db("QStore");

        var query2 = { customer_id: '1234' };

        dbo.collection("products").find(query2).toArray(function (err, result)
{
            if (err) throw err;

            console.log(result);
            res.send(result);
            db.close();
        });
    });
})

app.get('/part_iii', u, function (req, res) {
    MongoClient.connect('mongodb://localhost:27017/', function (err, db) {
        if (err) throw err;

        var dbo = db.db("QStore");

        var query3 = { customer_id: '1234' };

        dbo.collection("reviews").find(query3).toArray(function (err, result)
{
            if (err) throw err;

            console.log(result);
            res.send(result);
            db.close();
        });
        dbo.collection("products").find(query3).toArray(function (err, result)
{
            if (err) throw err;

            console.log(result);
            res.send(result);
            db.close();
        });
    });
})

app.listen(1234, () => {
    console.log('Running...')
})
```

Outputs

Server running

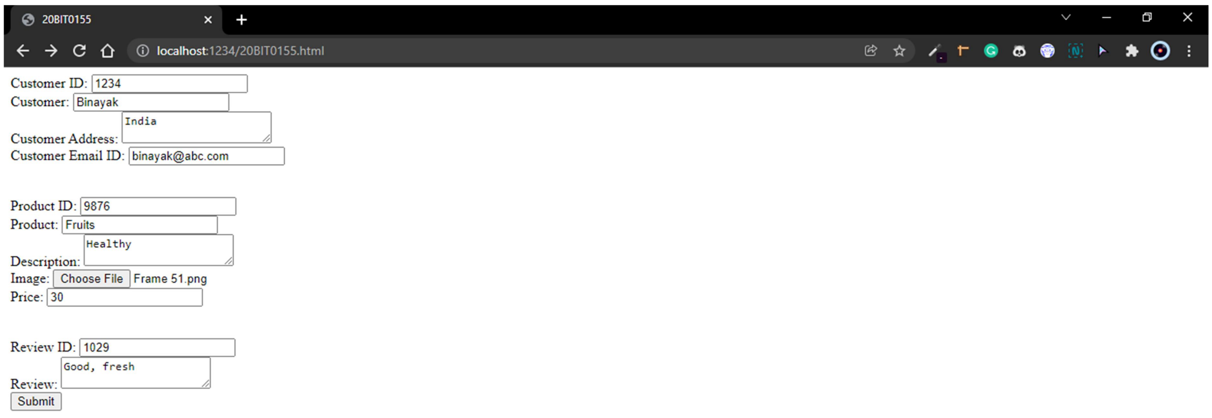

File Edit Selection View Go Run Terminal Help
20BIT0155.js - Lab FAT - Visual Studio Code
20BIT0155.html U
20BIT0155.js U
20BIT0155.html > html > body > form#myForm > div
20BIT0155.js > app.get('/20BIT0155.html') callback
DEBUG CONSOLE OUTPUT PROBLEMS TERMINAL
powershell
master* 0↓ 2↑ ⊗ 0 ⚠ 0 tabnine Ln 17, Col 57 Spaces: 4 UTF-8 CRLF JavaScript

```html
        <input type="text" name="prodname"
    </div>
    <div>
        <label for="desp">Description:</la
        <textarea name="desp" id="desp"></
    </div>
    <div>
        <label for="image">Image:</label>
        <input type="file" id="image" name
    </div>
    <div>
        <label for="price">Price:</label>
        <input type="text" name="price" ic
    </div>

    <br /><br />
```

```javascript
const u = bodyParser.urlencoded({ extended: false })

app.use(express.static(path.join(__dirname, 'html')))
app.use(express.static(path.join(__dirname, 'css')))
app.use(express.static(path.join(__dirname, 'assets')))

app.get('/20BIT0155.html', (req, res) => {
    res.sendFile(path.join(__dirname, '20BIT0155.html'))
})

app.post('/', u, function (req, res) {
    var product = {
        product_id: req.body.prodid,
        name: req.body.prodname,
        description: req.body.desp,
        image:req.body.image
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS D:\VIT\Semester_III\Web_Tech_A2;L17,18\WebTech_L17_18\Lab FAT> node 20BIT0155.js
Running...
```

Form

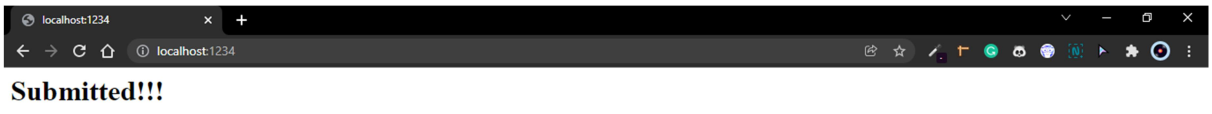


Customer ID: 1234
Customer: Binayak
Customer Address: India
Customer Email ID: binayak@abc.com

Product ID: 9876
Product: Fruits
Description: Healthy
Image: Choose File Frame 51.png
Price: 30

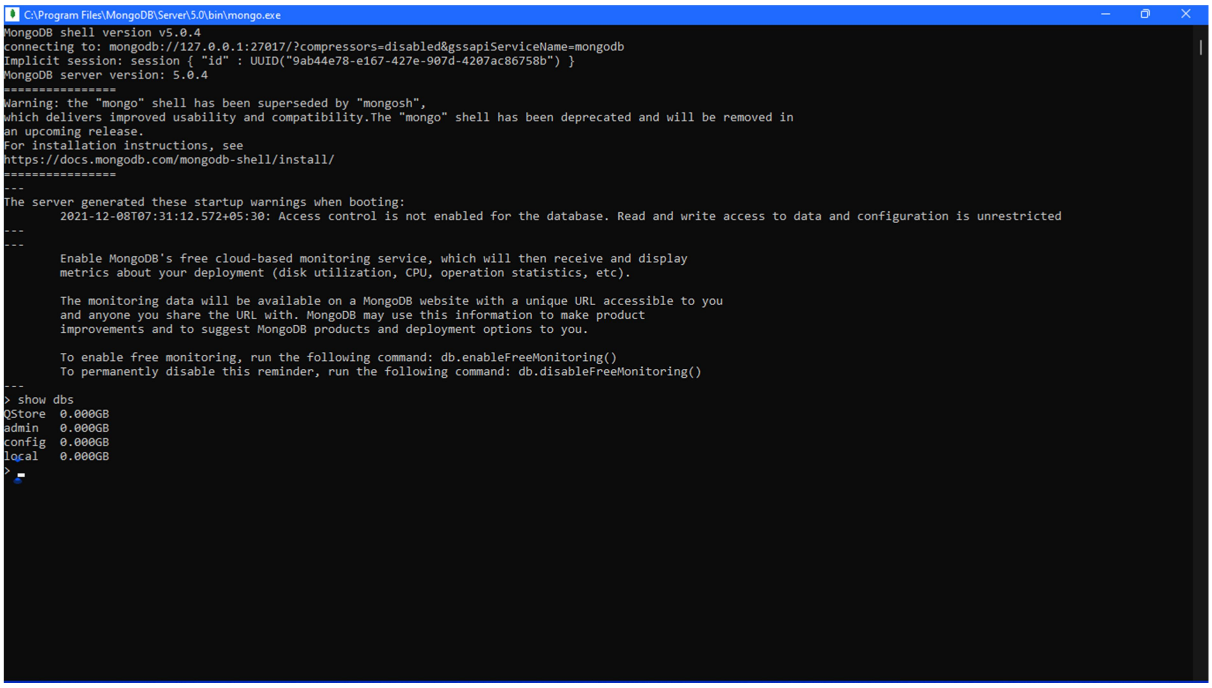
Review ID: 1029
Review: Good, fresh

Confirmation of submission


localhost:1234

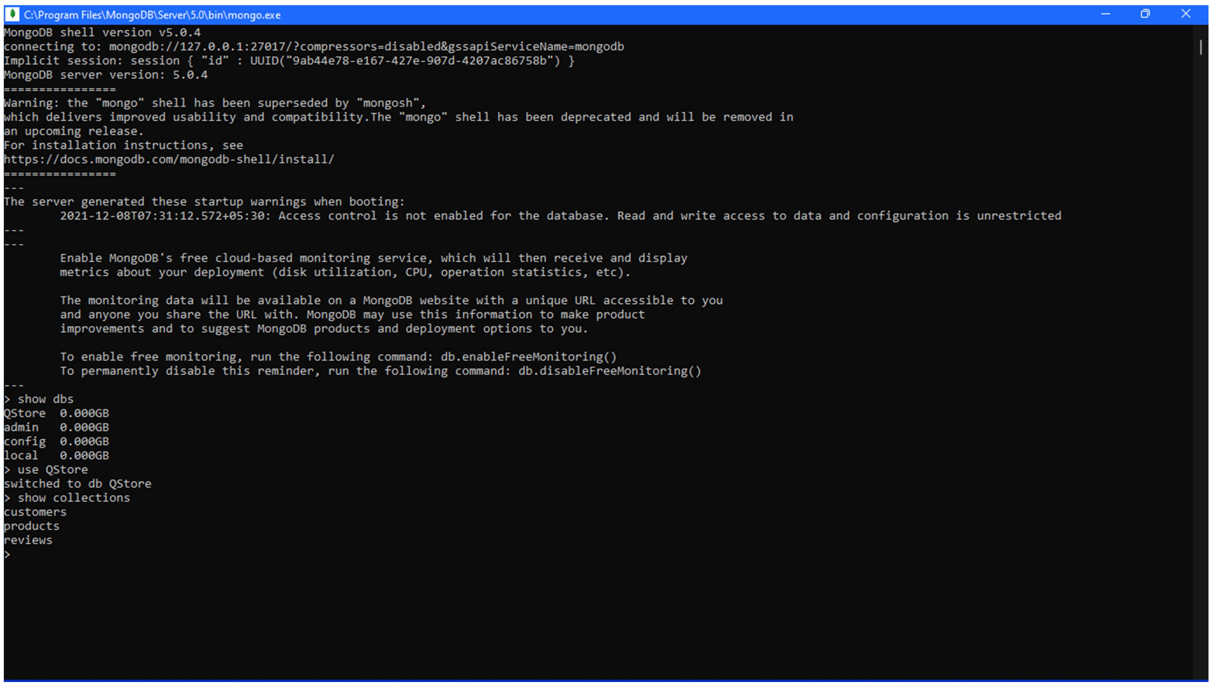
Submitted!!!

Database created by nodejs


C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe

```
MongoDB shell version v5.0.4
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("9ab44e78-e167-427e-907d-4207ac86758b") }
MongoDB server version: 5.0.4
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2021-12-08T07:31:12.572+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show dbs
QStore  0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
>
```

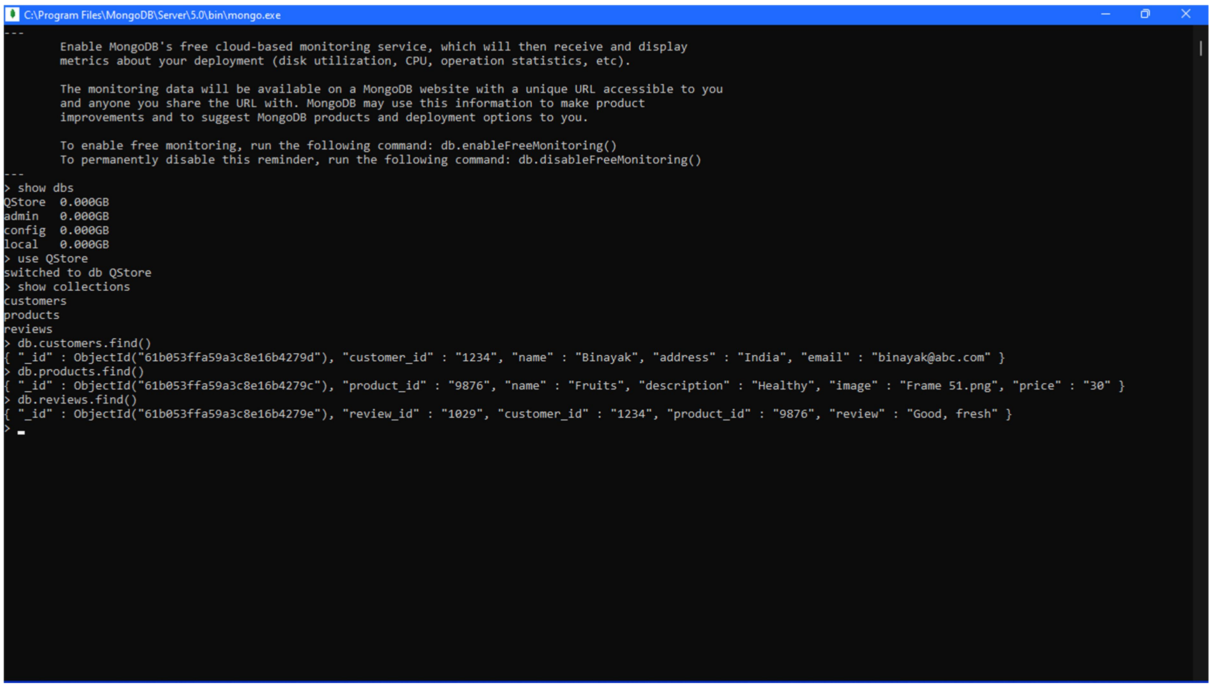
Three collections also made



C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe

```
MongoDB shell version v5.0.4
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("9ab44e78-e167-427e-907d-4207ac86758b") }
MongoDB server version: 5.0.4
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2021-12-08T07:31:12.572+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show dbs
QStore  0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
> use QStore
switched to db QStore
> show collections
customers
products
reviews
>
```

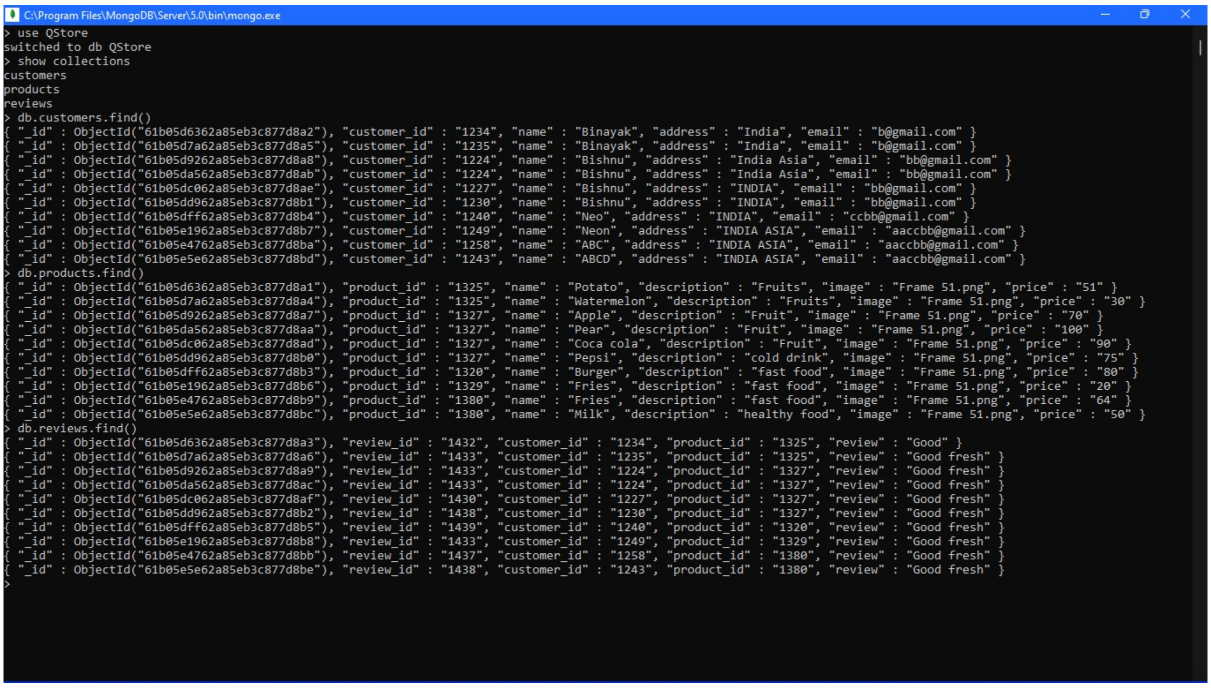
First set of data



C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe

```
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show dbs
QStore  0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
> use QStore
switched to db QStore
> show collections
customers
products
reviews
> db.customers.find()
{ "_id" : ObjectId("61b053ffa59a3c8e16b4279d"), "customer_id" : "1234", "name" : "Binayak", "address" : "India", "email" : "binayak@abc.com" }
> db.products.find()
{ "_id" : ObjectId("61b053ffa59a3c8e16b4279c"), "product_id" : "9876", "name" : "Fruits", "description" : "Healthy", "image" : "Frame 51.png", "price" : "30" }
> db.reviews.find()
{ "_id" : ObjectId("61b053ffa59a3c8e16b4279e"), "review_id" : "1029", "customer_id" : "1234", "product_id" : "9876", "review" : "Good, fresh" }
> _
```

Ten data added by form via nodejs



C:\Program Files\MongoDB\Server\5.0\bin\mongo.exe

```
> use QStore
switched to db QStore
> show collections
customers
products
reviews
> db.customers.find()
{ "_id" : ObjectId("61b05d6362a85eb3c877d8a2"), "customer_id" : "1234", "name" : "Binayak", "address" : "India", "email" : "b@gmail.com" }
{ "_id" : ObjectId("61b05d7a62a85eb3c877d8a5"), "customer_id" : "1235", "name" : "Binayak", "address" : "India", "email" : "b@gmail.com" }
{ "_id" : ObjectId("61b05d9262a85eb3c877d8a8"), "customer_id" : "1224", "name" : "Bishnu", "address" : "India Asia", "email" : "bb@gmail.com" }
{ "_id" : ObjectId("61b05da562a85eb3c877d8ab"), "customer_id" : "1224", "name" : "Bishnu", "address" : "India Asia", "email" : "bb@gmail.com" }
{ "_id" : ObjectId("61b05dc062a85eb3c877d8ae"), "customer_id" : "1227", "name" : "Bishnu", "address" : "INDIA", "email" : "bb@gmail.com" }
{ "_id" : ObjectId("61b05dd962a85eb3c877d8b1"), "customer_id" : "1230", "name" : "Bishnu", "address" : "INDIA", "email" : "bb@gmail.com" }
{ "_id" : ObjectId("61b05dff62a85eb3c877d8b4"), "customer_id" : "1240", "name" : "Neo", "address" : "INDIA", "email" : "ccbb@gmail.com" }
{ "_id" : ObjectId("61b05e1962a85eb3c877d8b7"), "customer_id" : "1249", "name" : "Neon", "address" : "INDIA ASIA", "email" : "aaccbb@gmail.com" }
{ "_id" : ObjectId("61b05e4762a85eb3c877d8ba"), "customer_id" : "1258", "name" : "ABC", "address" : "INDIA ASIA", "email" : "aaccbb@gmail.com" }
{ "_id" : ObjectId("61b05e5e62a85eb3c877d8bd"), "customer_id" : "1243", "name" : "ABCD", "address" : "INDIA ASIA", "email" : "aaccbb@gmail.com" }
> db.products.find()
{ "_id" : ObjectId("61b05d6362a85eb3c877d8a1"), "product_id" : "1325", "name" : "Potato", "description" : "Fruits", "image" : "Frame 51.png", "price" : "51" }
{ "_id" : ObjectId("61b05d7a62a85eb3c877d8a4"), "product_id" : "1325", "name" : "Watermelon", "description" : "Fruits", "image" : "Frame 51.png", "price" : "30" }
{ "_id" : ObjectId("61b05d9262a85eb3c877d8a7"), "product_id" : "1327", "name" : "Apple", "description" : "Fruit", "image" : "Frame 51.png", "price" : "70" }
{ "_id" : ObjectId("61b05da562a85eb3c877d8aa"), "product_id" : "1327", "name" : "Pear", "description" : "Fruit", "image" : "Frame 51.png", "price" : "100" }
{ "_id" : ObjectId("61b05dc062a85eb3c877d8ad"), "product_id" : "1327", "name" : "Coca cola", "description" : "Fruit", "image" : "Frame 51.png", "price" : "90" }
{ "_id" : ObjectId("61b05dd962a85eb3c877d8b0"), "product_id" : "1327", "name" : "Pepsi", "description" : "cold drink", "image" : "Frame 51.png", "price" : "75" }
{ "_id" : ObjectId("61b05dff62a85eb3c877d8b3"), "product_id" : "1320", "name" : "Burger", "description" : "fast food", "image" : "Frame 51.png", "price" : "80" }
{ "_id" : ObjectId("61b05e1962a85eb3c877d8b6"), "product_id" : "1329", "name" : "Fries", "description" : "fast food", "image" : "Frame 51.png", "price" : "20" }
{ "_id" : ObjectId("61b05e4762a85eb3c877d8b9"), "product_id" : "1380", "name" : "Fries", "description" : "fast food", "image" : "Frame 51.png", "price" : "64" }
{ "_id" : ObjectId("61b05e5e62a85eb3c877d8bc"), "product_id" : "1380", "name" : "Milk", "description" : "healthy food", "image" : "Frame 51.png", "price" : "50" }
> db.reviews.find()
{ "_id" : ObjectId("61b05d6362a85eb3c877d8a3"), "review_id" : "1432", "customer_id" : "1234", "product_id" : "1325", "review" : "Good" }
{ "_id" : ObjectId("61b05d7a62a85eb3c877d8a6"), "review_id" : "1433", "customer_id" : "1235", "product_id" : "1325", "review" : "Good fresh" }
{ "_id" : ObjectId("61b05d9262a85eb3c877d8a9"), "review_id" : "1433", "customer_id" : "1224", "product_id" : "1327", "review" : "Good fresh" }
{ "_id" : ObjectId("61b05da562a85eb3c877d8ac"), "review_id" : "1433", "customer_id" : "1224", "product_id" : "1327", "review" : "Good fresh" }
{ "_id" : ObjectId("61b05dc062a85eb3c877d8af"), "review_id" : "1430", "customer_id" : "1227", "product_id" : "1327", "review" : "Good fresh" }
{ "_id" : ObjectId("61b05dd962a85eb3c877d8b2"), "review_id" : "1438", "customer_id" : "1230", "product_id" : "1327", "review" : "Good fresh" }
{ "_id" : ObjectId("61b05dff62a85eb3c877d8b5"), "review_id" : "1439", "customer_id" : "1240", "product_id" : "1320", "review" : "Good fresh" }
{ "_id" : ObjectId("61b05e1962a85eb3c877d8b8"), "review_id" : "1433", "customer_id" : "1249", "product_id" : "1329", "review" : "Good fresh" }
{ "_id" : ObjectId("61b05e4762a85eb3c877d8bb"), "review_id" : "1437", "customer_id" : "1258", "product_id" : "1380", "review" : "Good fresh" }
{ "_id" : ObjectId("61b05e5e62a85eb3c877d8be"), "review_id" : "1438", "customer_id" : "1243", "product_id" : "1380", "review" : "Good fresh" }
>
```