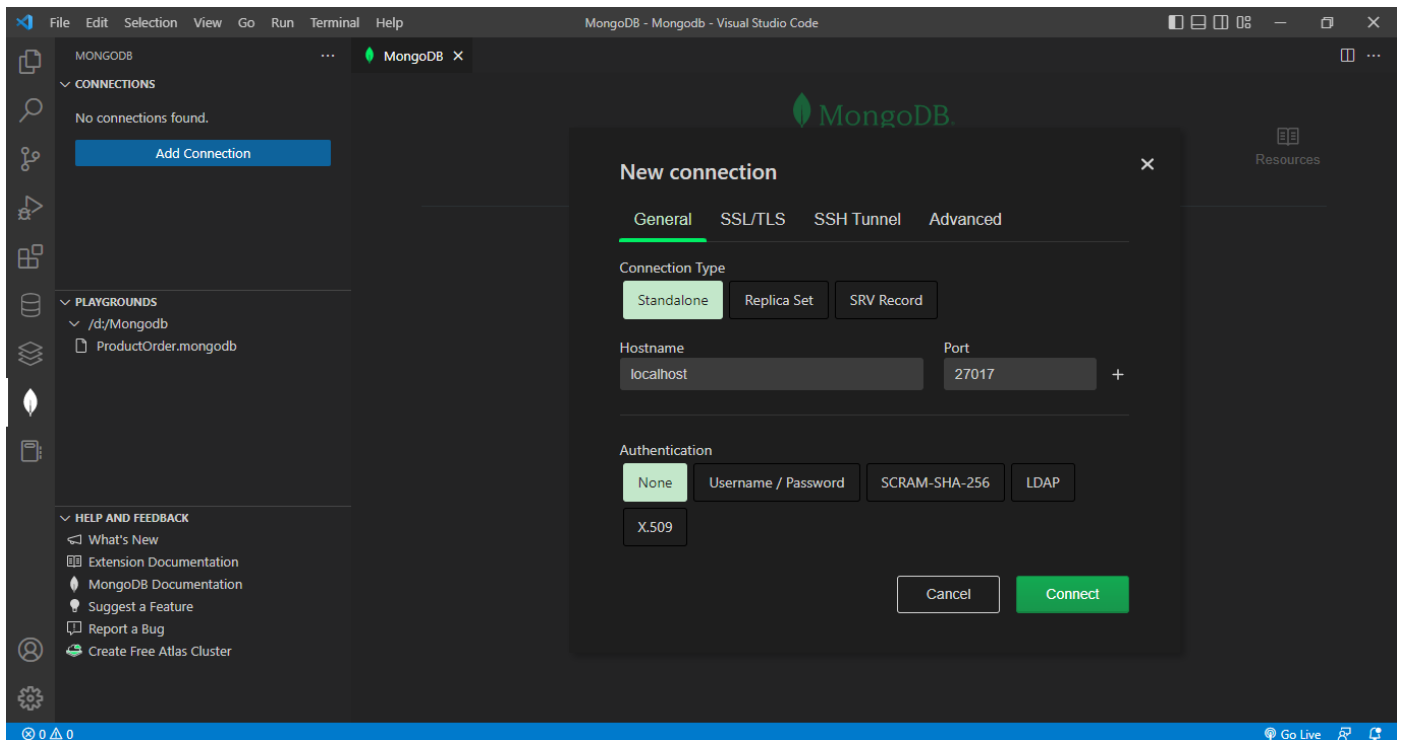
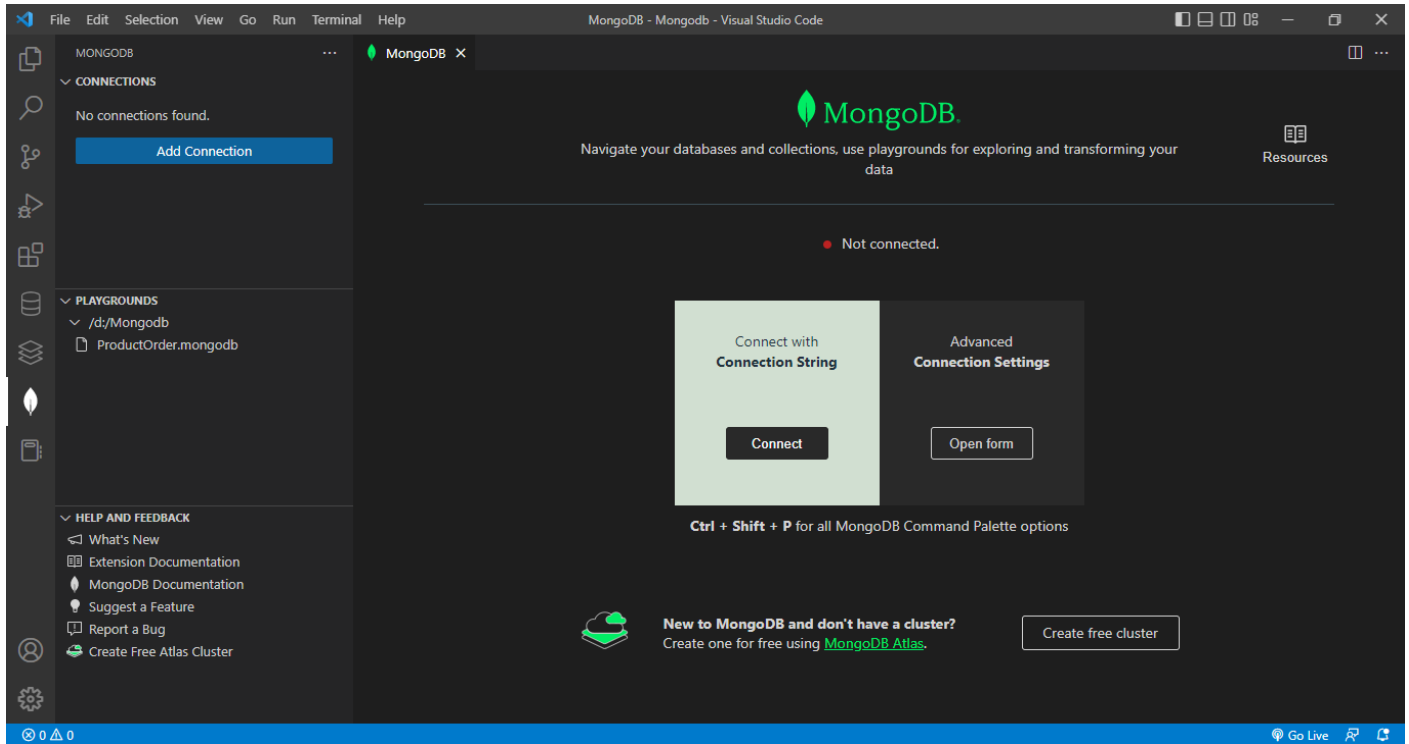
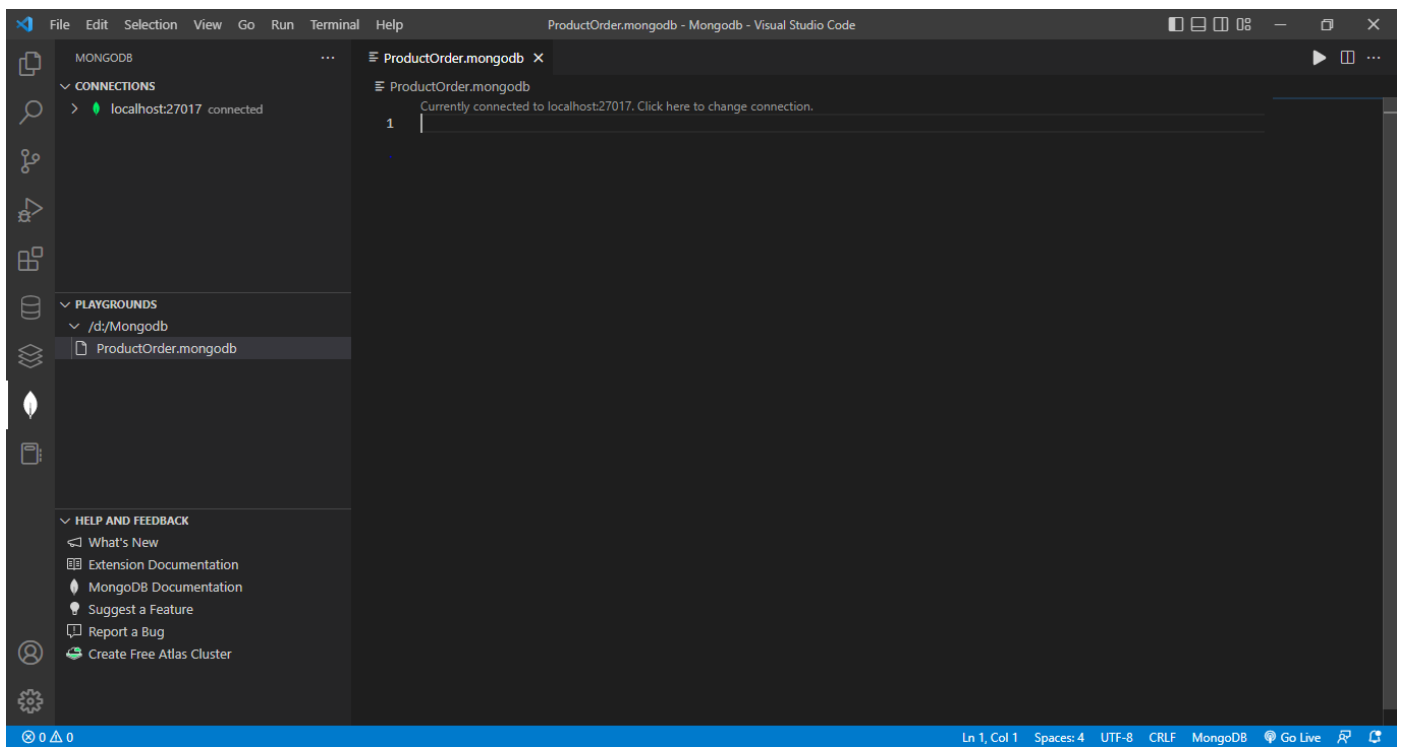
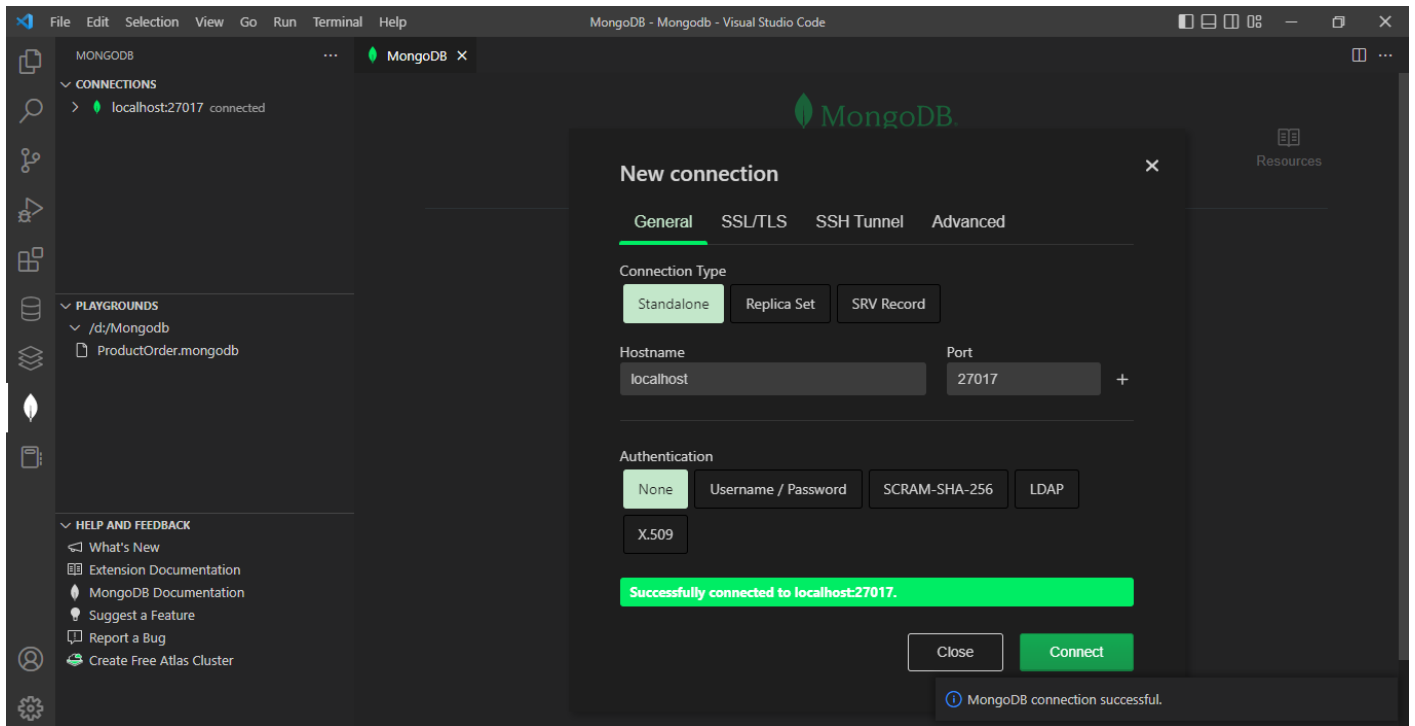


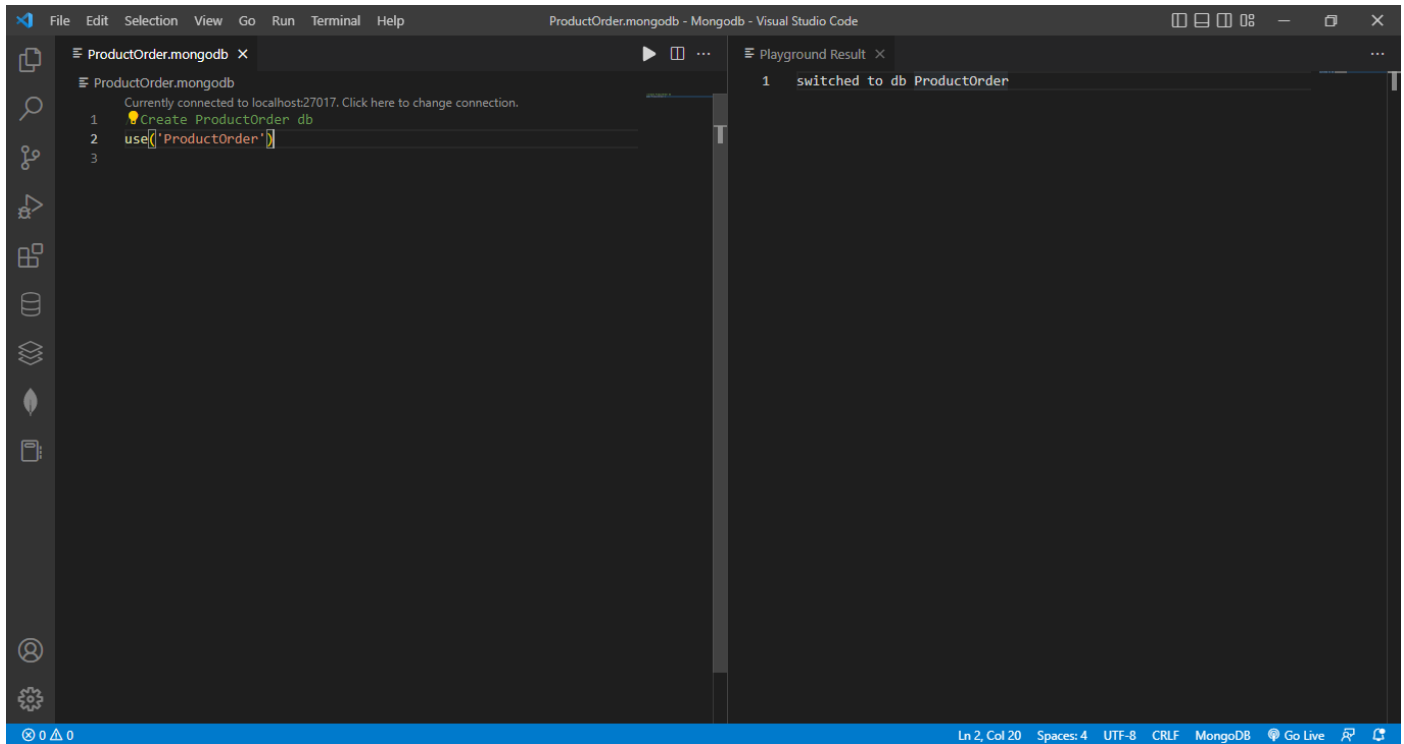
# Analyzing shipment data of an ecommerce firm using MONGODB

## 1. Open VS Code and connect to MONGODB





2. Create a database "ProductOrder" and create collections "Product","Inventory","User", and "Order" in it.



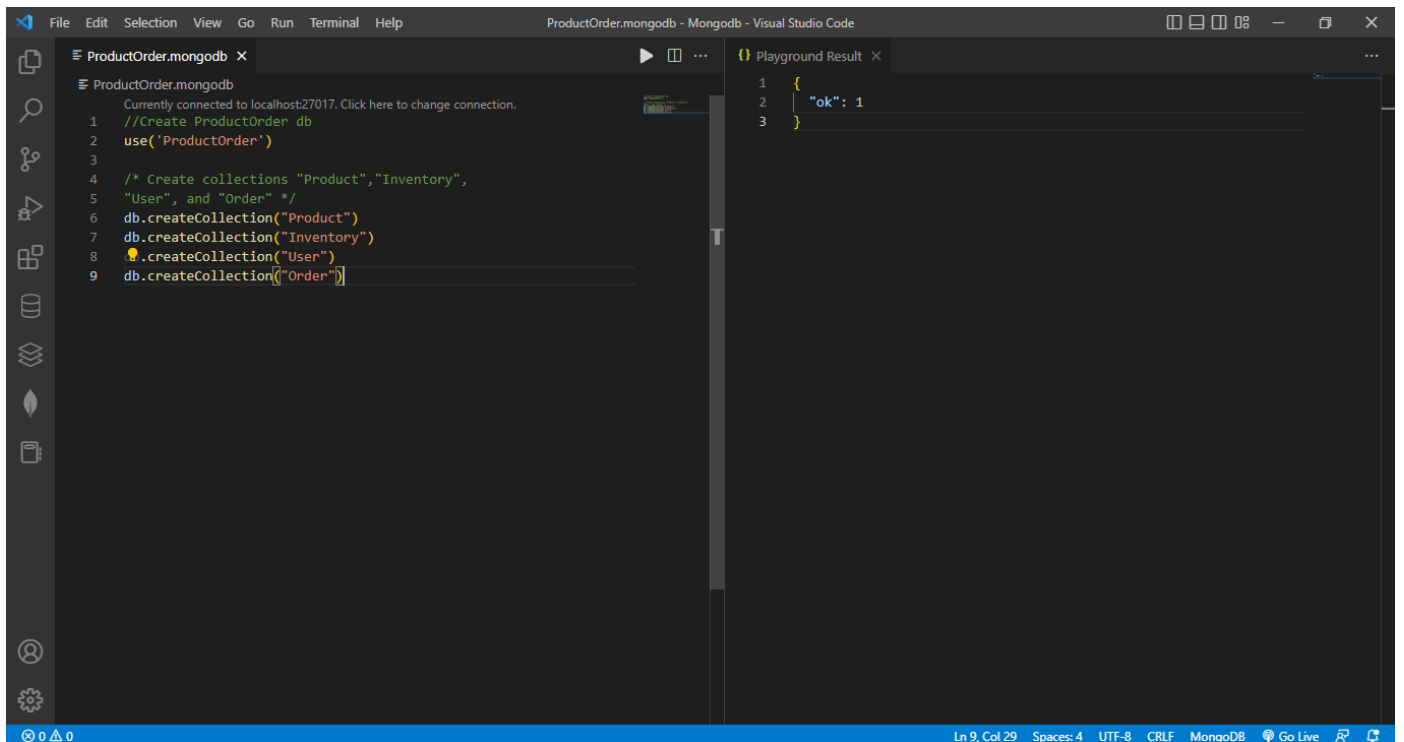
The screenshot shows the Visual Studio Code interface with the MongoDB Playground extension. The left pane, titled "ProductOrder.mongodb", contains the following code:

```
1 //Create ProductOrder db
2 use('ProductOrder')
3
```

The right pane, titled "Playground Result", shows the output of the command:

```
1 switched to db ProductOrder
```

The status bar at the bottom indicates the file is "Ln 2, Col 20" and the language is "MongoDB".



The screenshot shows the Visual Studio Code interface with the MongoDB Playground extension. The left pane, titled "ProductOrder.mongodb", contains the following code:

```
1 //Create ProductOrder db
2 use('ProductOrder')
3
4 /* Create collections "Product","Inventory",
5    "User", and "Order" */
6 db.createCollection("Product")
7 db.createCollection("Inventory")
8 db.createCollection("User")
9 db.createCollection("Order")
```

The right pane, titled "Playground Result", shows the output of the commands:

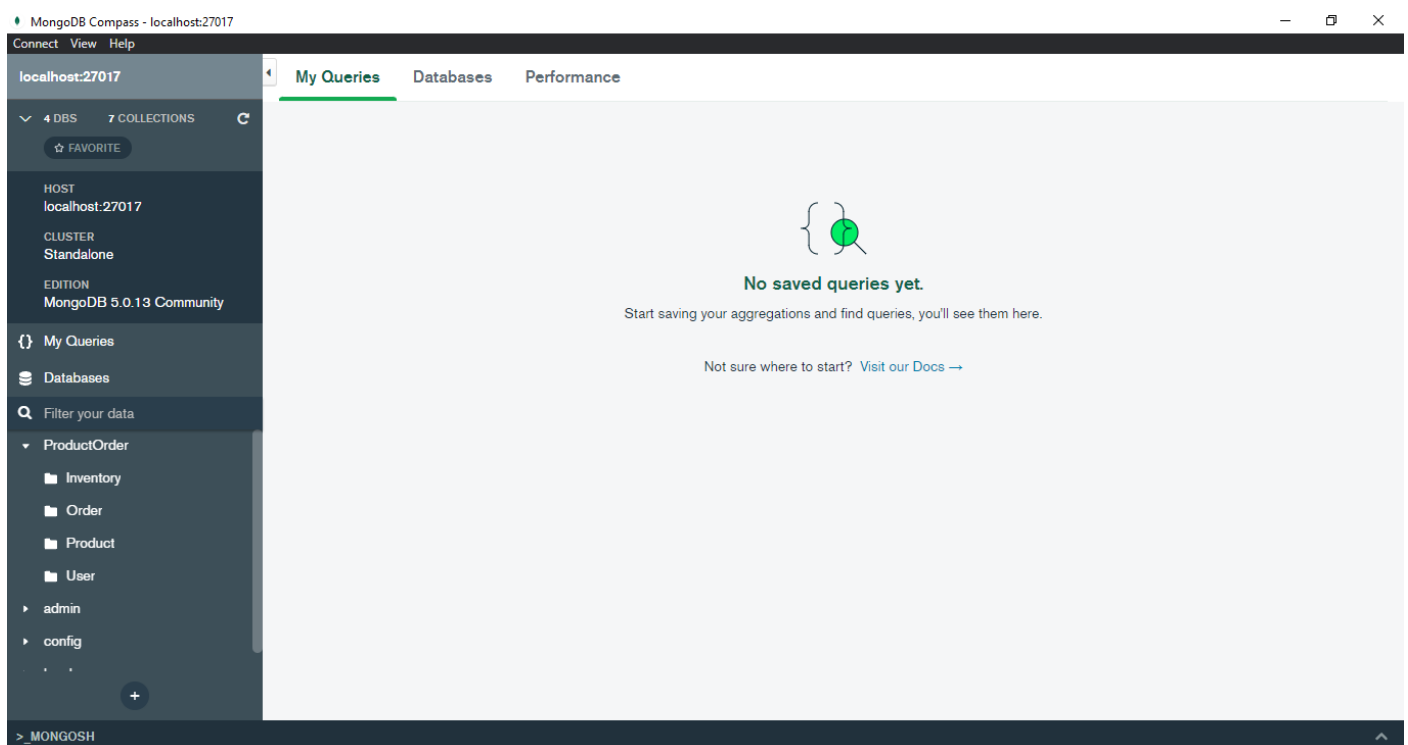
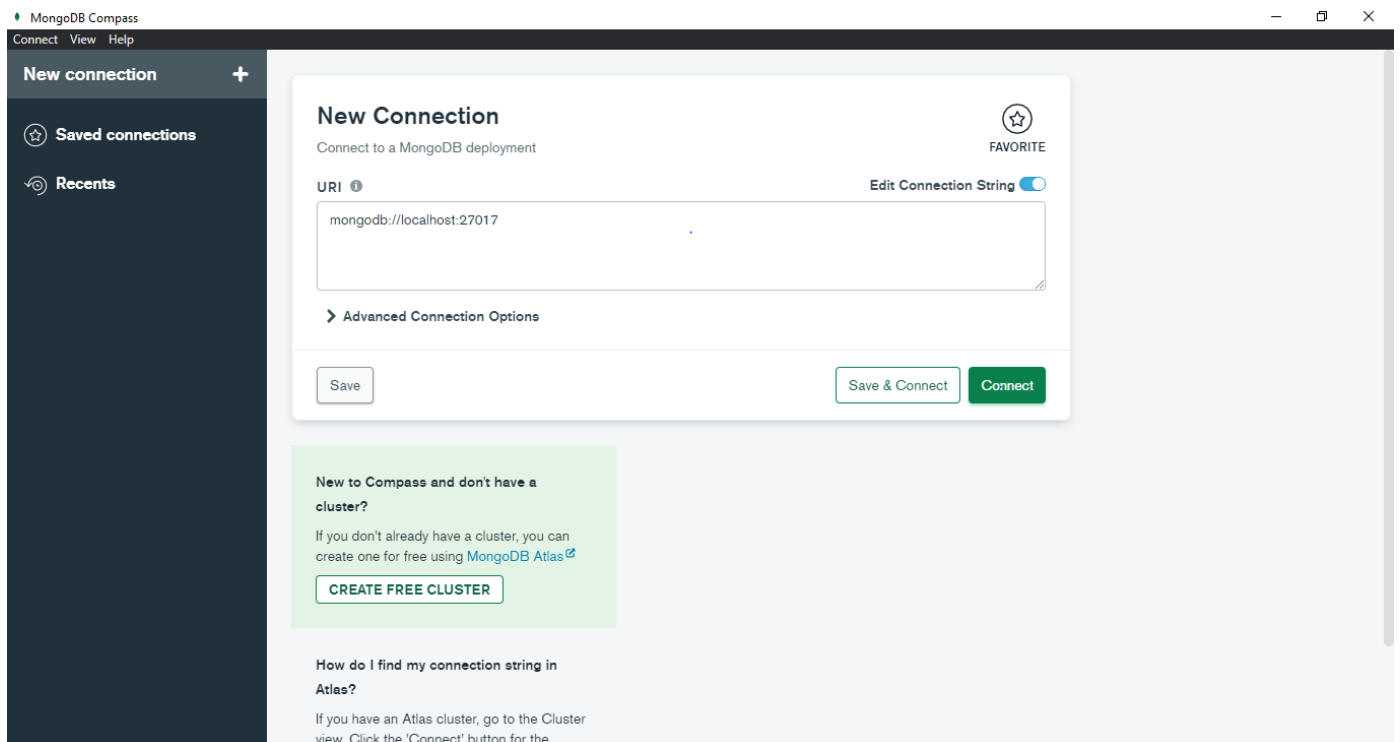
```
1 {
2   "ok": 1
3 }
```

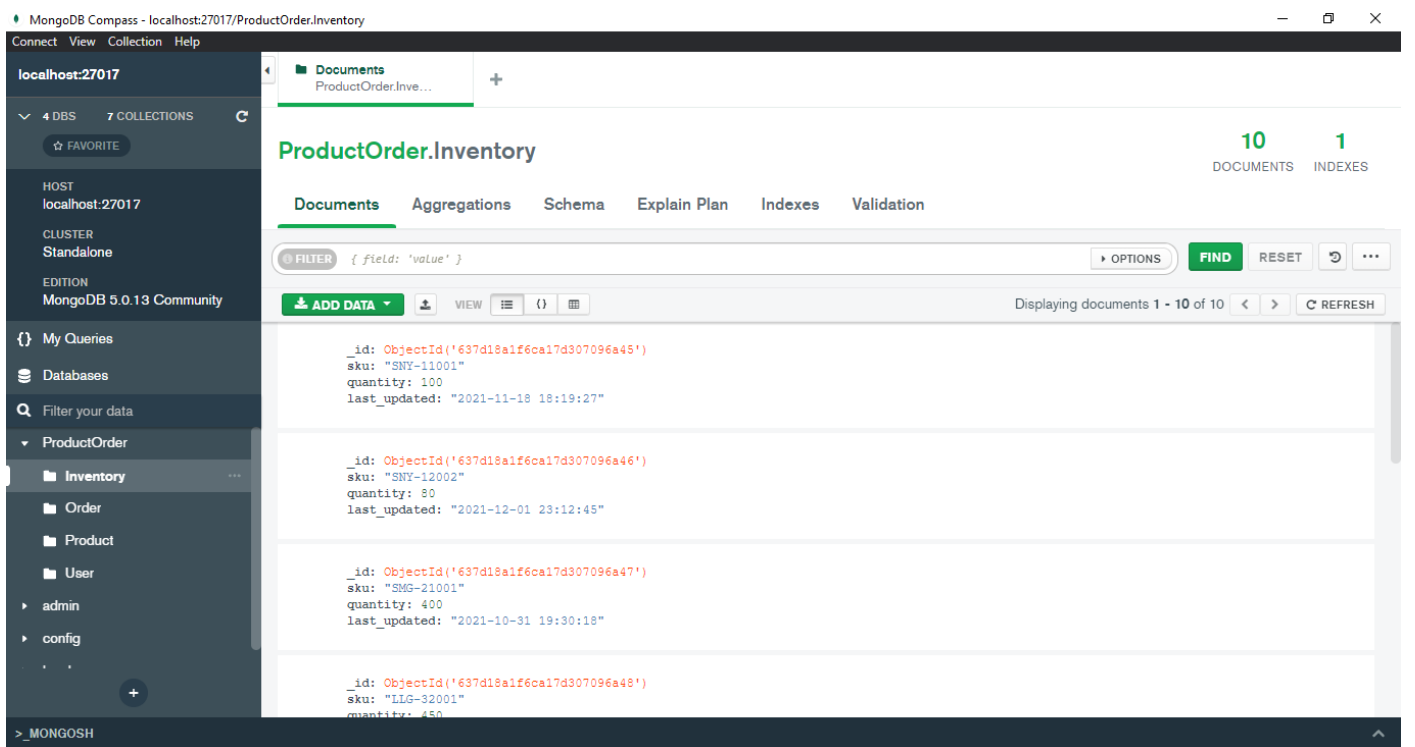
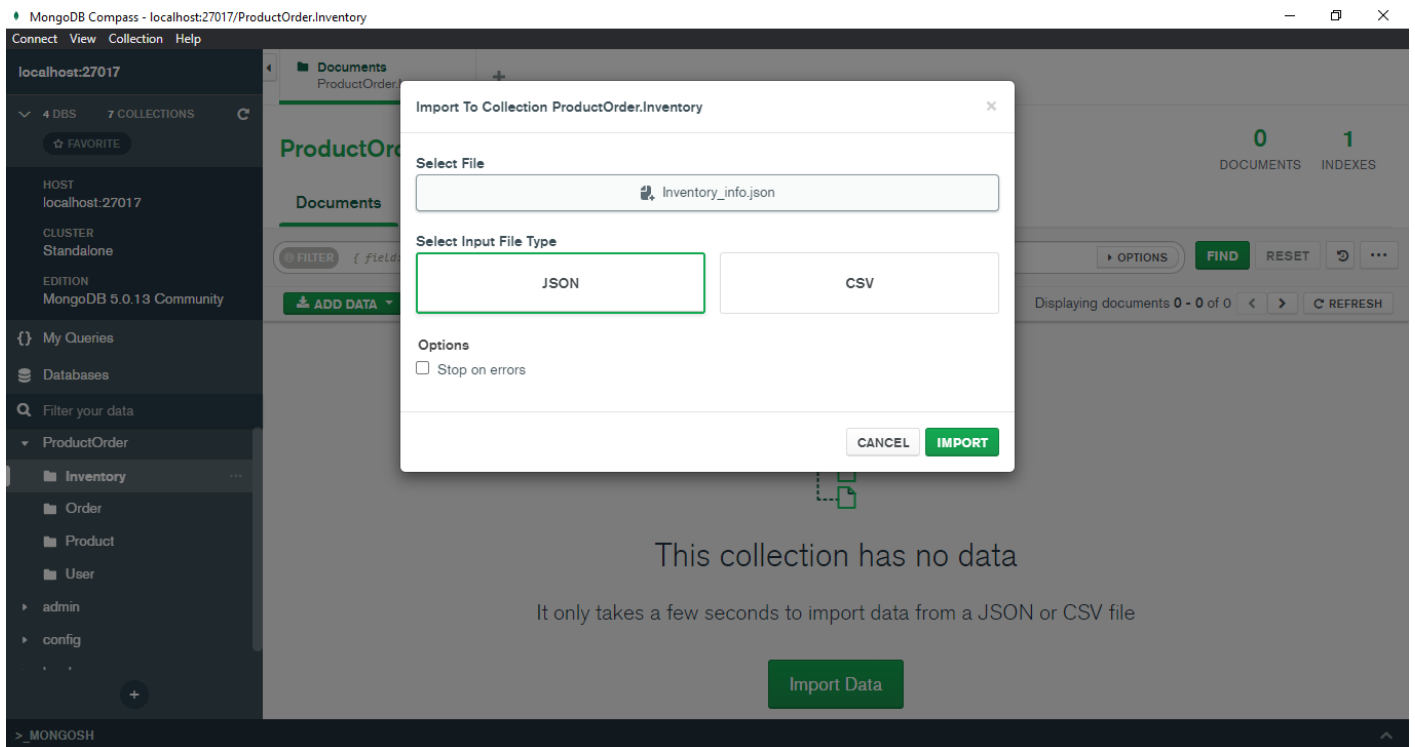
The status bar at the bottom indicates the file is "Ln 9, Col 29" and the language is "MongoDB".

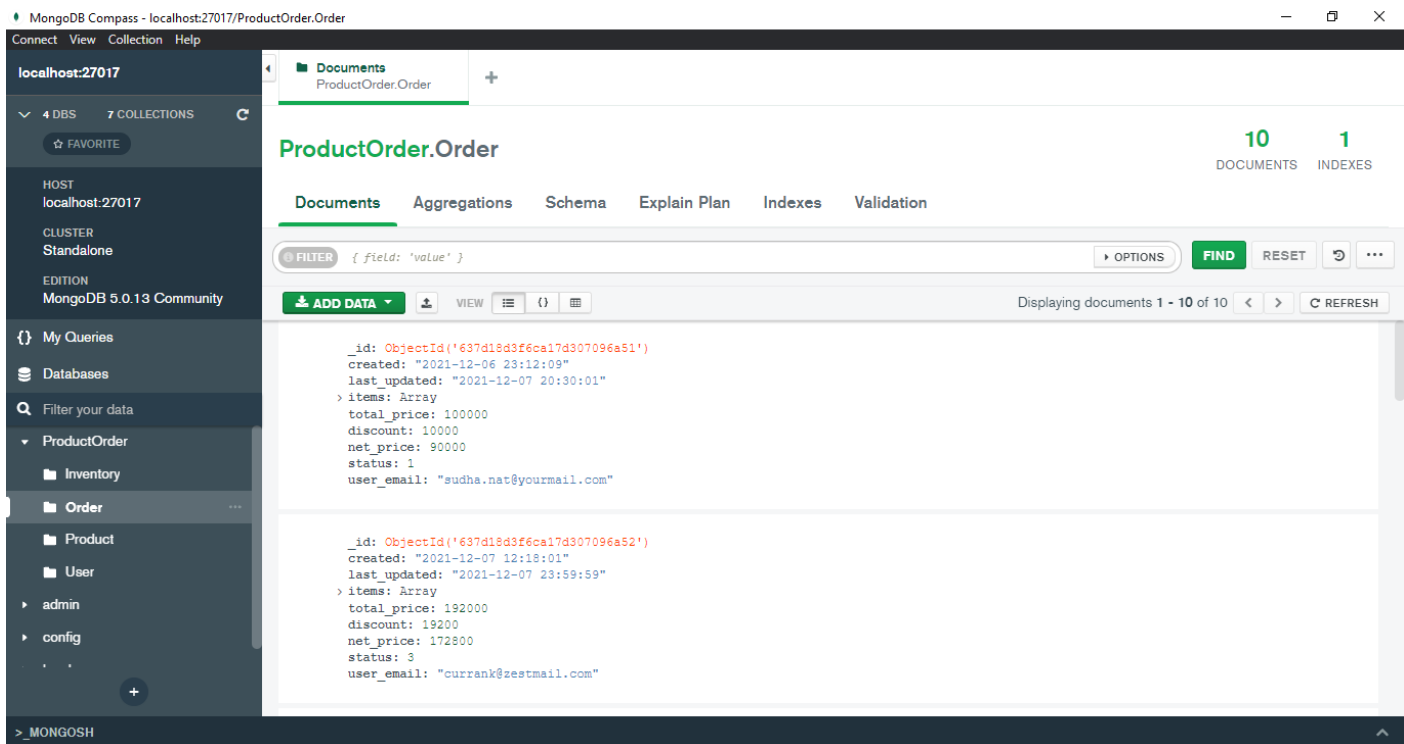
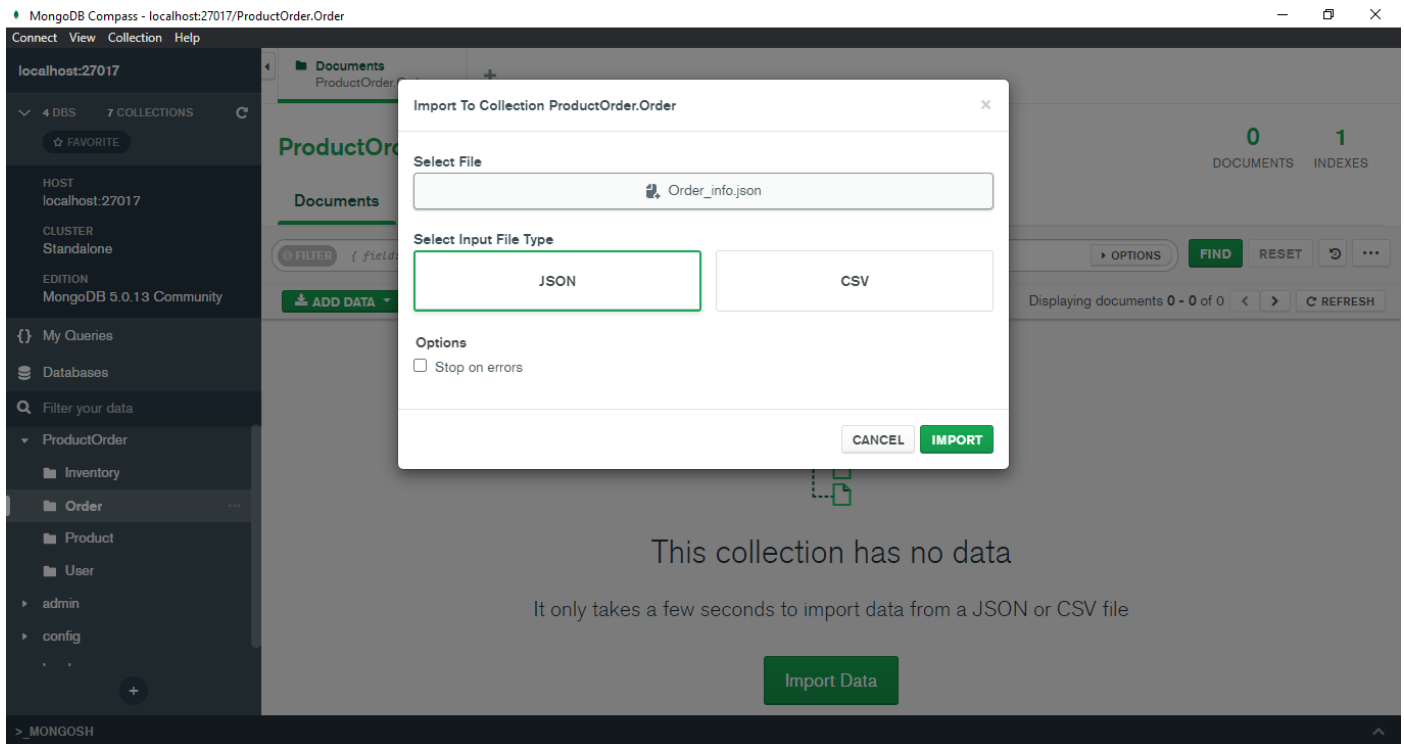
### 3. Open MongoDBCompass and navigate to the "ProductOrder" database.

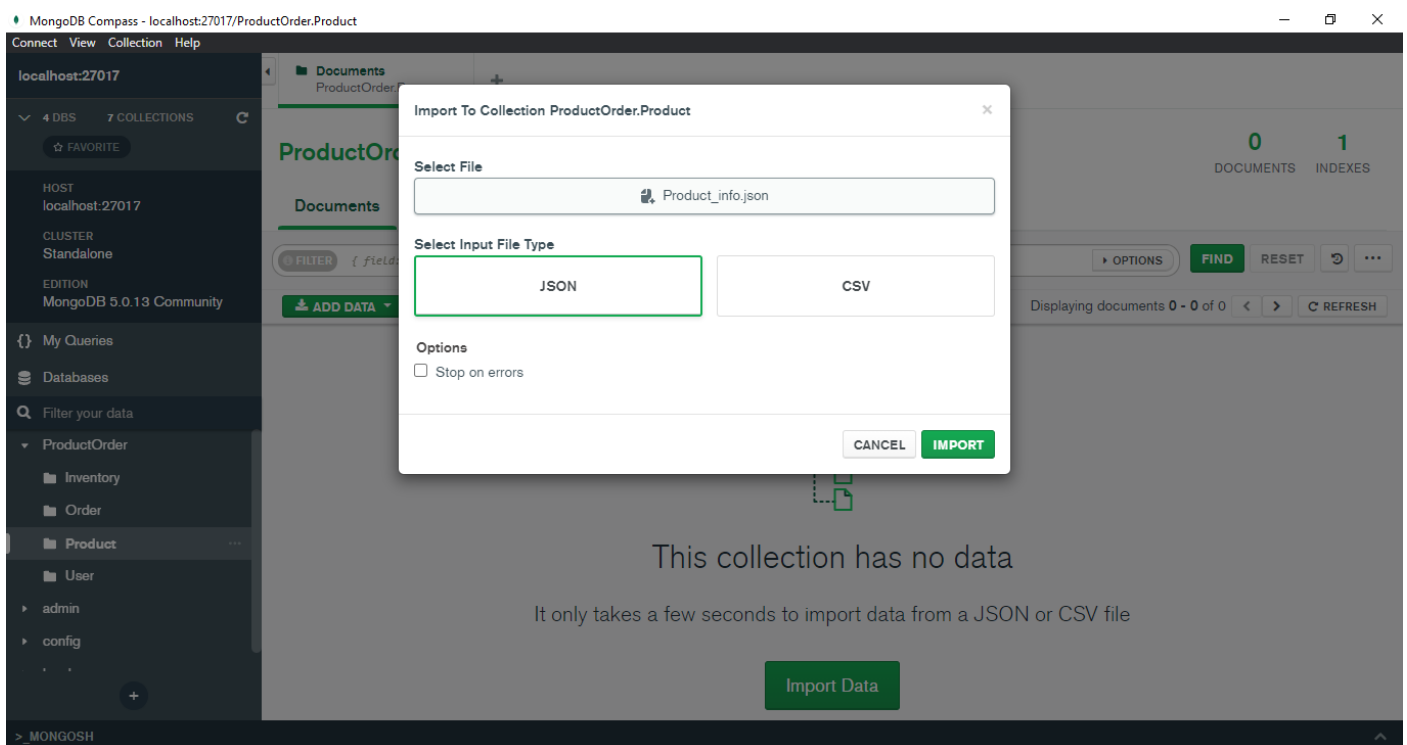
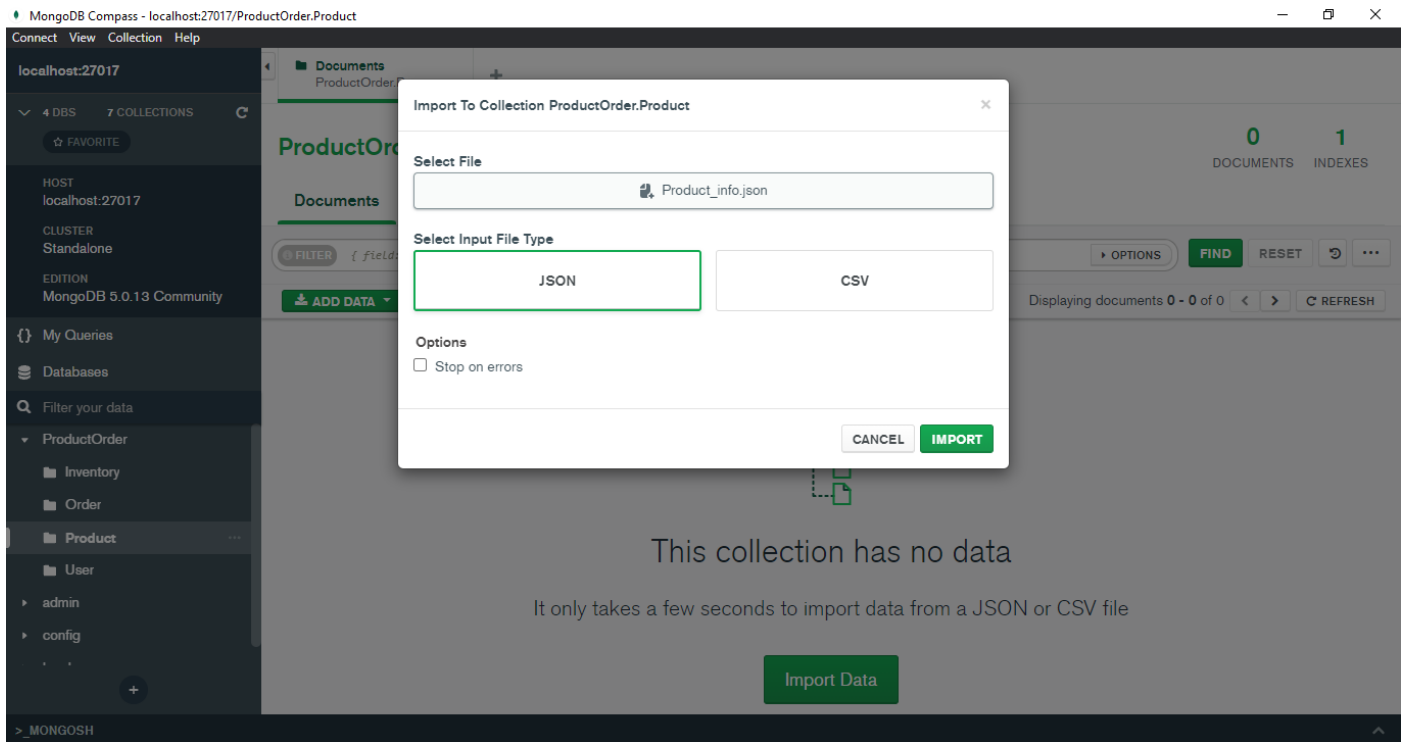
- i) Add "Product\_info.json" file into the "Product" collection.
- ii) Add "Inventory\_info.json" file into "Inventory" collection.
- iii) Add "User\_info.json" file into the "User" collection.
- iv) Add "Order\_info.json" file into "Order" collection.

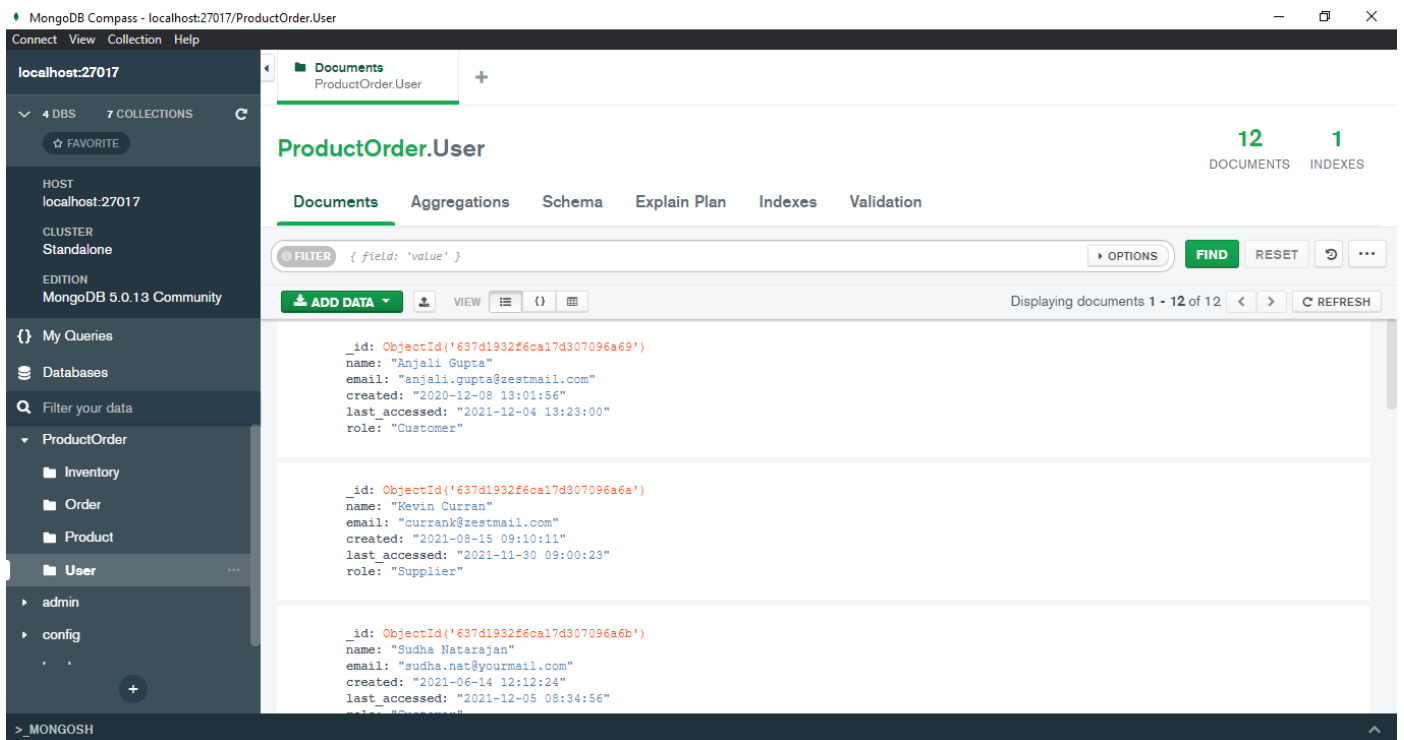
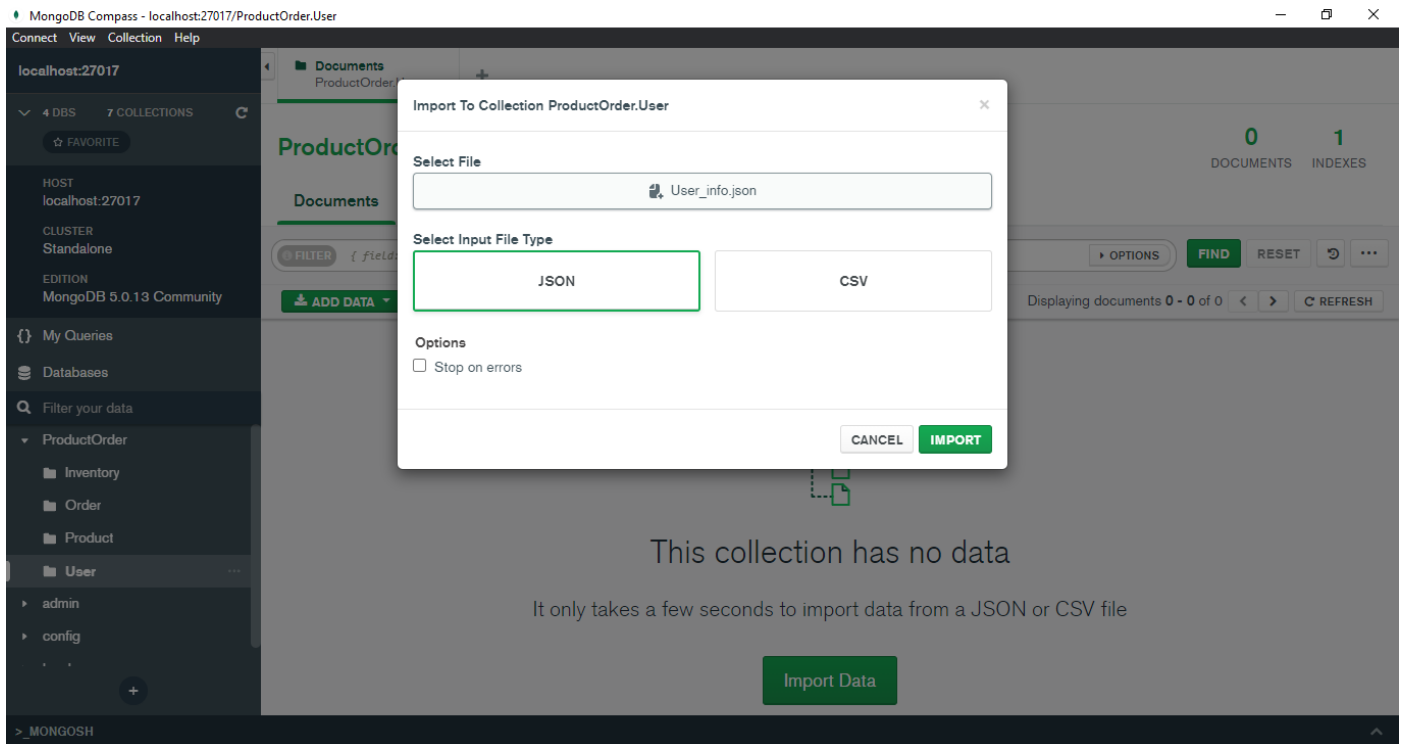
Note: These JSON files have restricted access. Hence, not uploaded to the repository.





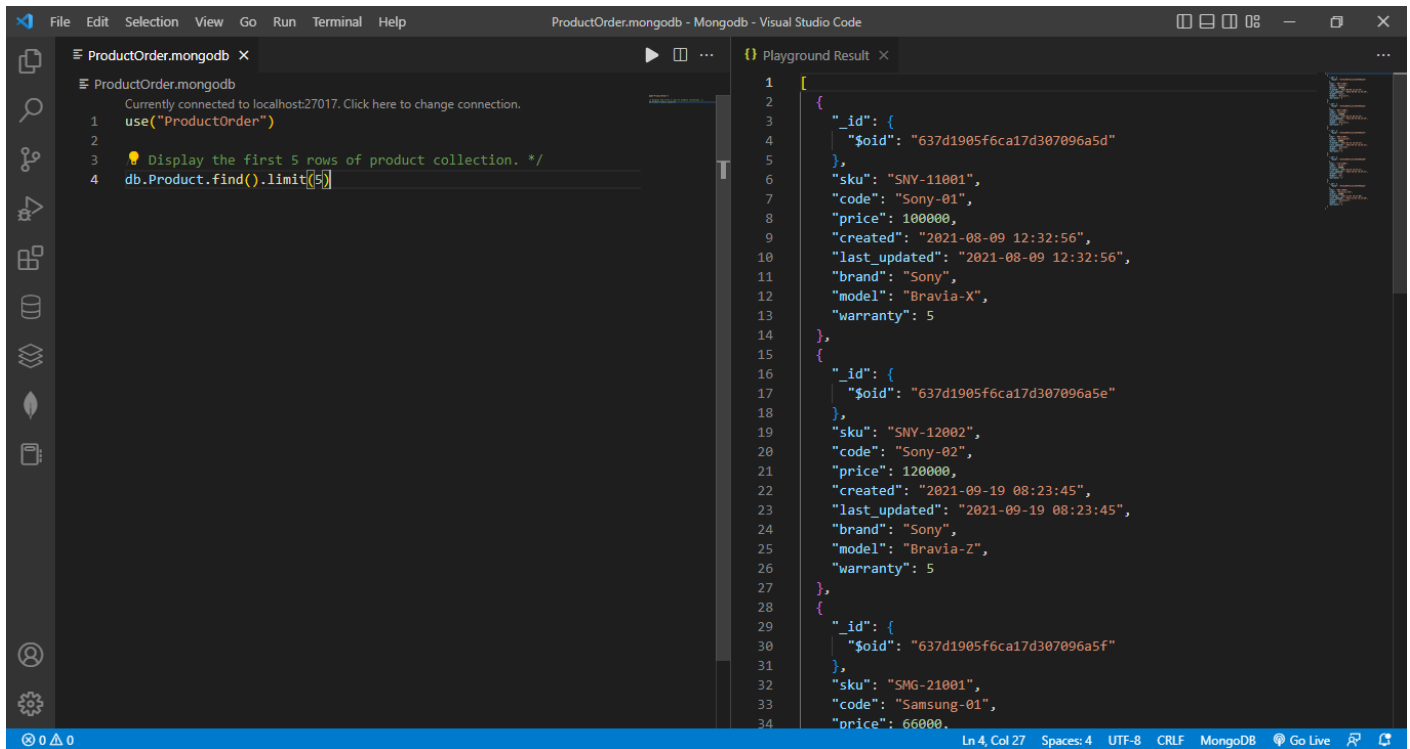








## 4. Display the first 5 rows of product, inventory, user, and order collection

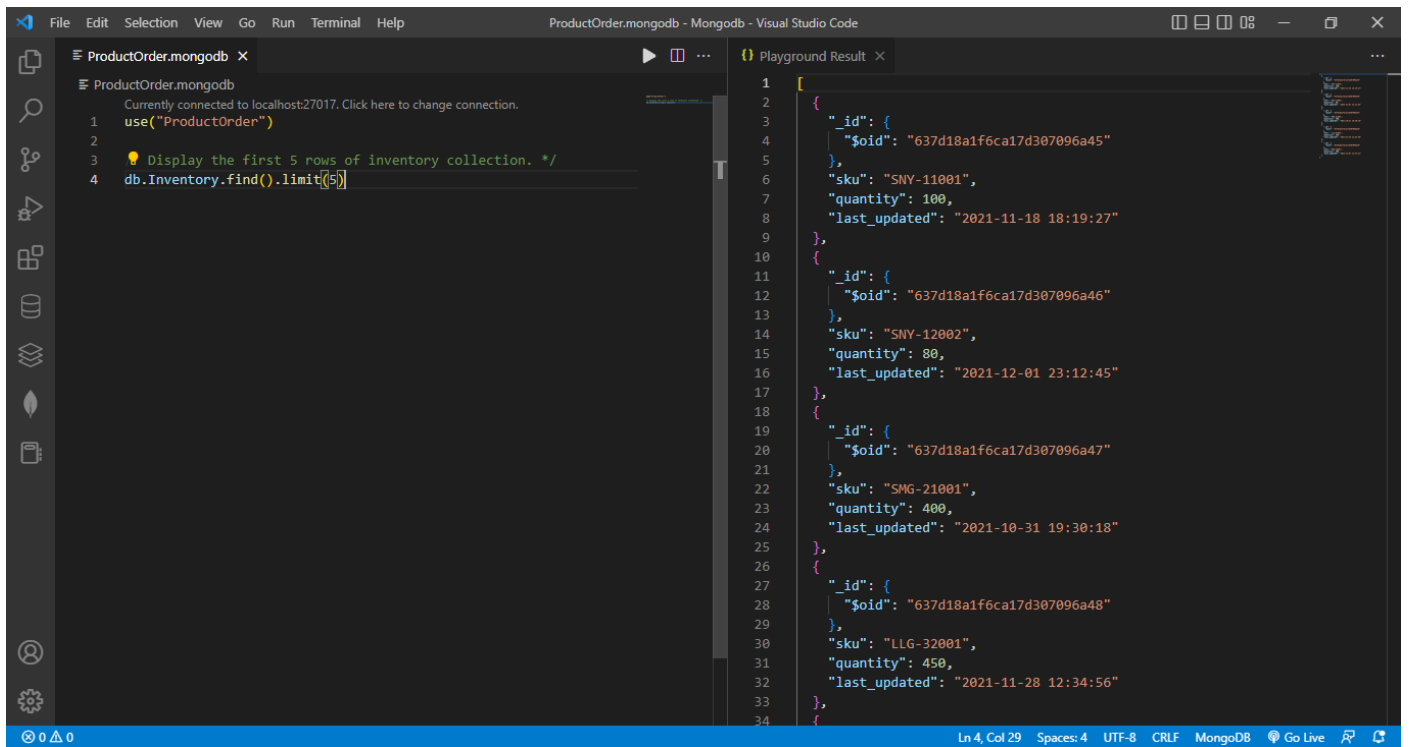


The screenshot shows the Visual Studio Code interface with the MongoDB Playground extension. The left sidebar displays the file explorer with 'ProductOrder.mongodb' selected. The main editor shows the MongoDB Playground script with the following code:

```
1 use("ProductOrder")
2
3 // Display the first 5 rows of product collection. */
4 db.Product.find().limit(5)
```

The right sidebar shows the 'Playground Result' tab, displaying the first 5 rows of the product collection:

```
1 {
2   "_id": {
3     "$oid": "637d1905f6ca17d307096a5d"
4   },
5   "sku": "SNY-11001",
6   "code": "Sony-01",
7   "price": 100000,
8   "created": "2021-08-09 12:32:56",
9   "last_updated": "2021-08-09 12:32:56",
10  "brand": "Sony",
11  "model": "Bravia-X",
12  "warranty": 5
13 },
14 {
15   "_id": {
16     "$oid": "637d1905f6ca17d307096a5e"
17   },
18   "sku": "SNY-12002",
19   "code": "Sony-02",
20   "price": 120000,
21   "created": "2021-09-19 08:23:45",
22   "last_updated": "2021-09-19 08:23:45",
23   "brand": "Sony",
24   "model": "Bravia-Z",
25   "warranty": 5
26 },
27 {
28   "_id": {
29     "$oid": "637d1905f6ca17d307096a5f"
30   },
31   "sku": "SMG-21001",
32   "code": "Samsung-01",
33   "price": 66000,
34 }
```



The screenshot shows the Visual Studio Code interface with the MongoDB Playground extension. The left sidebar displays the file explorer with 'ProductOrder.mongodb' selected. The main editor shows the MongoDB Playground script with the following code:

```
1 use("ProductOrder")
2
3 // Display the first 5 rows of inventory collection. */
4 db.Inventory.find().limit(5)
```

The right sidebar shows the 'Playground Result' tab, displaying the first 5 rows of the inventory collection:

```
1 {
2   "_id": {
3     "$oid": "637d18a1f6ca17d307096a45"
4   },
5   "sku": "SNY-11001",
6   "quantity": 100,
7   "last_updated": "2021-11-18 18:19:27"
8 },
9 {
10   "_id": {
11     "$oid": "637d18a1f6ca17d307096a46"
12   },
13   "sku": "SNY-12002",
14   "quantity": 80,
15   "last_updated": "2021-12-01 23:12:45"
16 },
17 {
18   "_id": {
19     "$oid": "637d18a1f6ca17d307096a47"
20   },
21   "sku": "SMG-21001",
22   "quantity": 400,
23   "last_updated": "2021-10-31 19:30:18"
24 },
25 {
26   "_id": {
27     "$oid": "637d18a1f6ca17d307096a48"
28   },
29   "sku": "LLG-32001",
30   "quantity": 450,
31   "last_updated": "2021-11-28 12:34:56"
32 },
33 {
34 }
```

```
File Edit Selection View Go Run Terminal Help ProductOrder.mongodb - MongoDB - Visual Studio Code

ProductOrder.mongodb X
ProductOrder.mongodb
Currently connected to localhost:27017. Click here to change connection.
1 use("ProductOrder")
2
3 ⚡ Display the first 5 rows of user collection. */
4 db.User.find().limit(5)

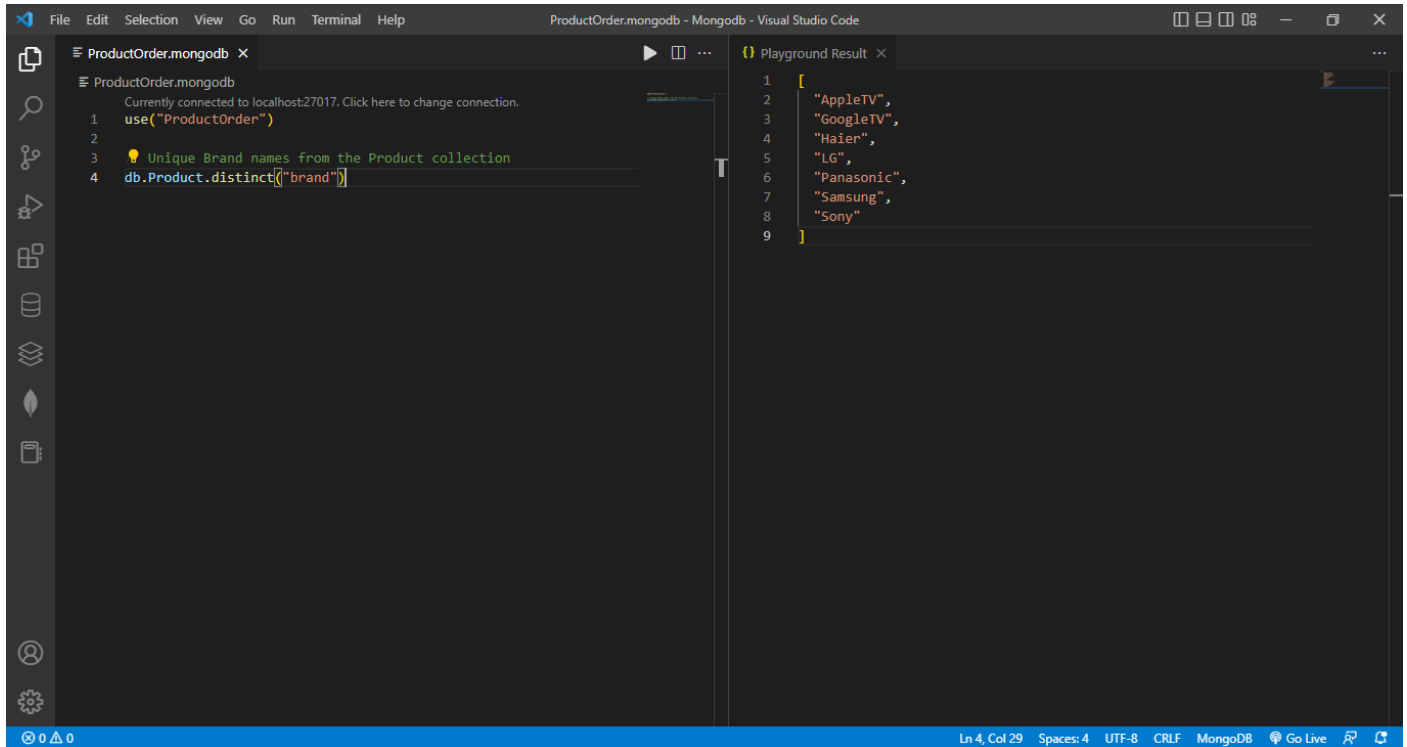
Playground Result X
1 {
2   {
3     "_id": {
4       "$oid": "637d1932f6ca17d307096a69"
5     },
6     "name": "Anjali Gupta",
7     "email": "anjali.gupta@zestmail.com",
8     "created": "2020-12-08 13:01:56",
9     "last_accessed": "2021-12-04 13:23:00",
10    "role": "Customer"
11  },
12  {
13    "_id": {
14      "$oid": "637d1932f6ca17d307096a6a"
15    },
16    "name": "Kevin Curran",
17    "email": "currank@zestmail.com",
18    "created": "2021-08-15 09:10:11",
19    "last_accessed": "2021-11-30 09:00:23",
20    "role": "Supplier"
21  },
22  {
23    "_id": {
24      "$oid": "637d1932f6ca17d307096a6b"
25    },
26    "name": "Sudha Natarajan",
27    "email": "sudha.nat@yourmail.com",
28    "created": "2021-06-14 12:12:24",
29    "last_accessed": "2021-12-05 08:34:56",
30    "role": "Customer"
31  },
32  {
33    "_id": {
34      "$oid": "637d1932f6ca17d307096a6c"
```

```
File Edit Selection View Go Run Terminal Help ProductOrder.mongodb - MongoDB - Visual Studio Code

ProductOrder.mongodb X
ProductOrder.mongodb
Currently connected to localhost:27017. Click here to change connection.
1 use("ProductOrder")
2
3 ⚡ Display the first 5 rows of order collection. */
4 db.Order.find().limit(5)

Playground Result X
1 {
2   {
3     "_id": {
4       "$oid": "637d18d3f6ca17d307096a51"
5     },
6     "created": "2021-12-06 23:12:09",
7     "last_updated": "2021-12-07 20:30:01",
8     "items": [
9       {
10        "product_sku": "SNV-11001",
11        "unit_price": 100000,
12        "quantity": 1
13      }
14    ],
15    "total_price": 100000,
16    "discount": 10000,
17    "net_price": 90000,
18    "status": 1,
19    "user_email": "sudha.nat@yourmail.com"
20  },
21  {
22    "_id": {
23      "$oid": "637d18d3f6ca17d307096a52"
24    },
25    "created": "2021-12-07 12:18:01",
26    "last_updated": "2021-12-07 23:59:59",
27    "items": [
28      {
29        "product_sku": "SMG-21001",
30        "unit_price": 66000,
31        "quantity": 1
32      },
33      {
34        "product_sku": "LLG-32001",
```

## 5. Display the Unique Brand and Model names from the Product collection

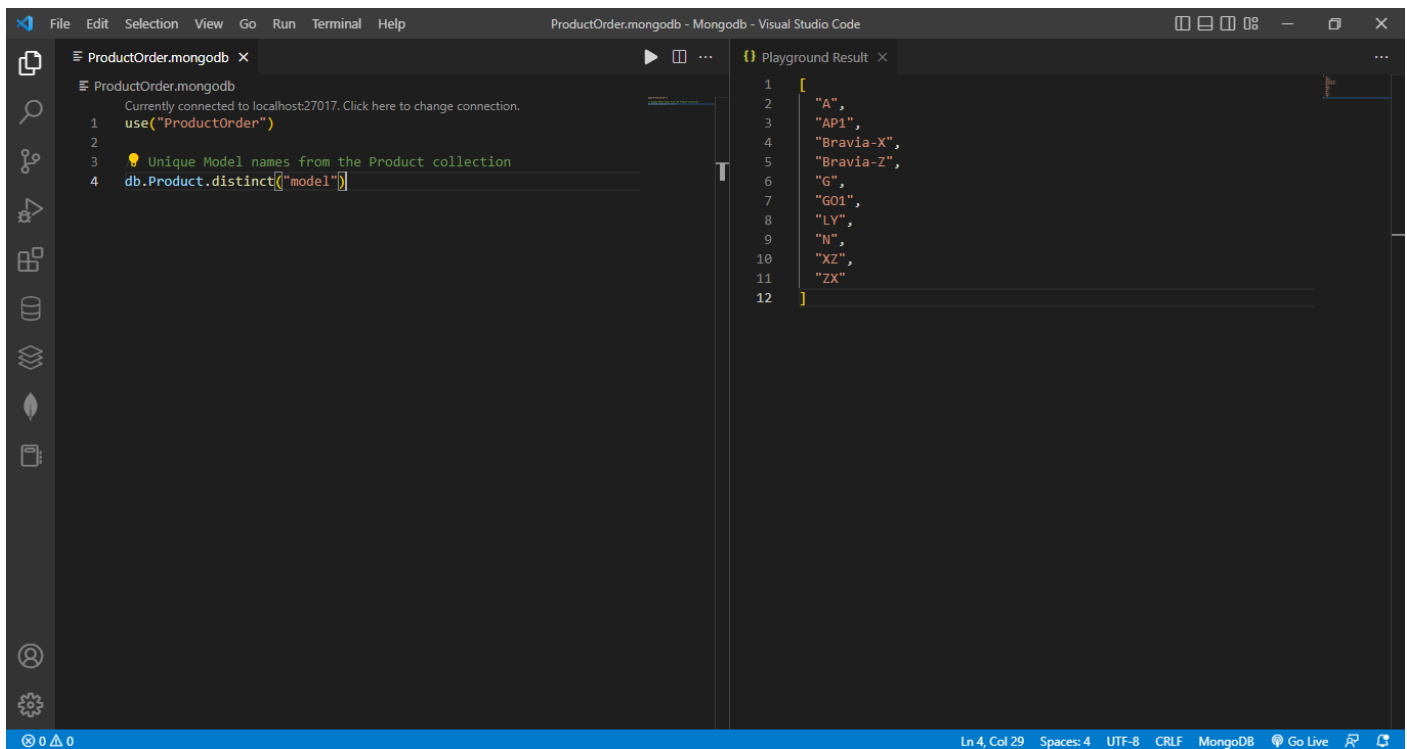


The screenshot shows the Visual Studio Code interface with the MongoDB Playground extension. The left pane shows the MongoDB Explorer with the 'ProductOrder.mongodb' database selected. The right pane shows the Playground Result with the following JSON output:

```
1 [
2   "AppleTV",
3   "GoogleTV",
4   "Haier",
5   "LG",
6   "Panasonic",
7   "Samsung",
8   "Sony"
9 ]
```

The code editor on the left contains the following JavaScript code:

```
1 use("ProductOrder")
2
3 // Unique Brand names from the Product collection
4 db.Product.distinct("brand")
```



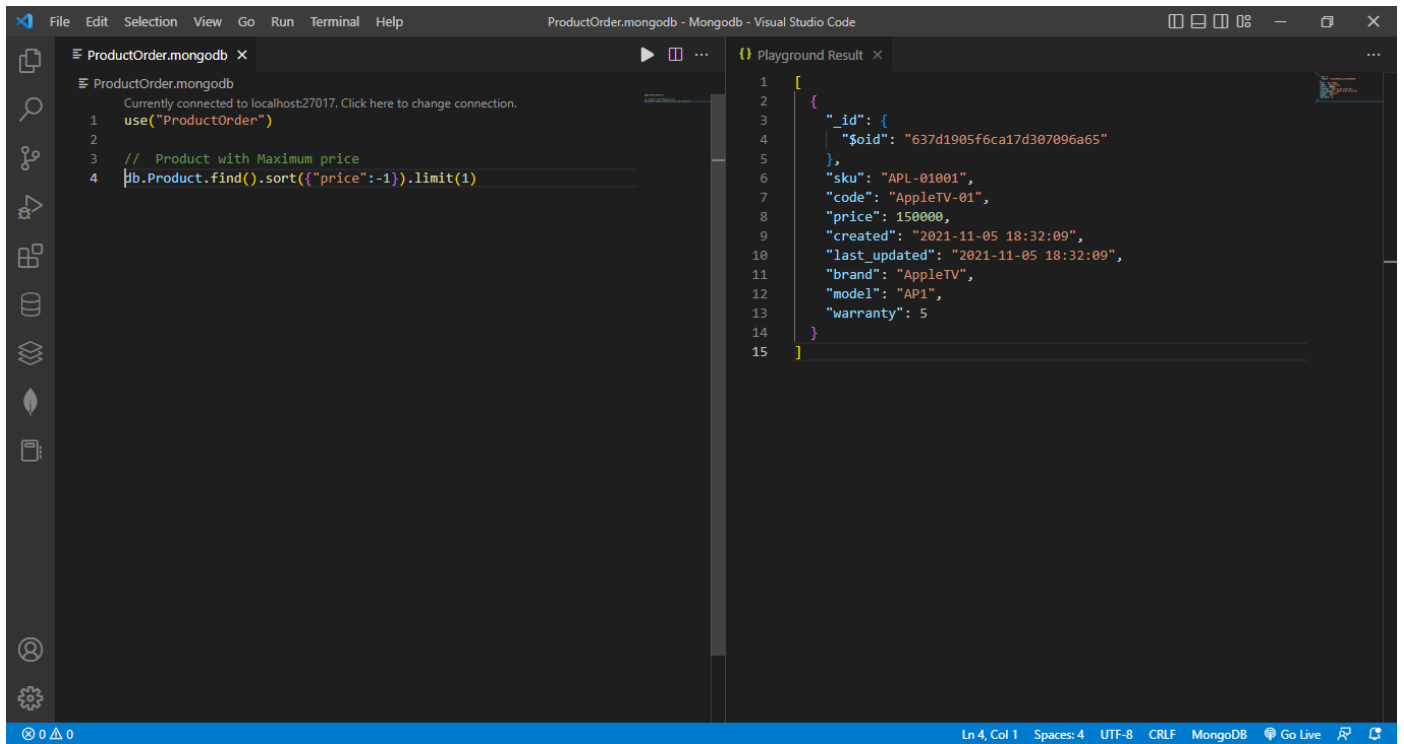
The screenshot shows the Visual Studio Code interface with the MongoDB Playground extension. The left pane shows the MongoDB Explorer with the 'ProductOrder.mongodb' database selected. The right pane shows the Playground Result with the following JSON output:

```
1 [
2   "A",
3   "AP1",
4   "Bravia-X",
5   "Bravia-Z",
6   "G",
7   "G01",
8   "LV",
9   "N",
10  "XZ",
11  "ZX"
12 ]
```

The code editor on the left contains the following JavaScript code:

```
1 use("ProductOrder")
2
3 // Unique Model names from the Product collection
4 db.Product.distinct("model")
```

## 6. Find the maximum and minimum price of the given products.

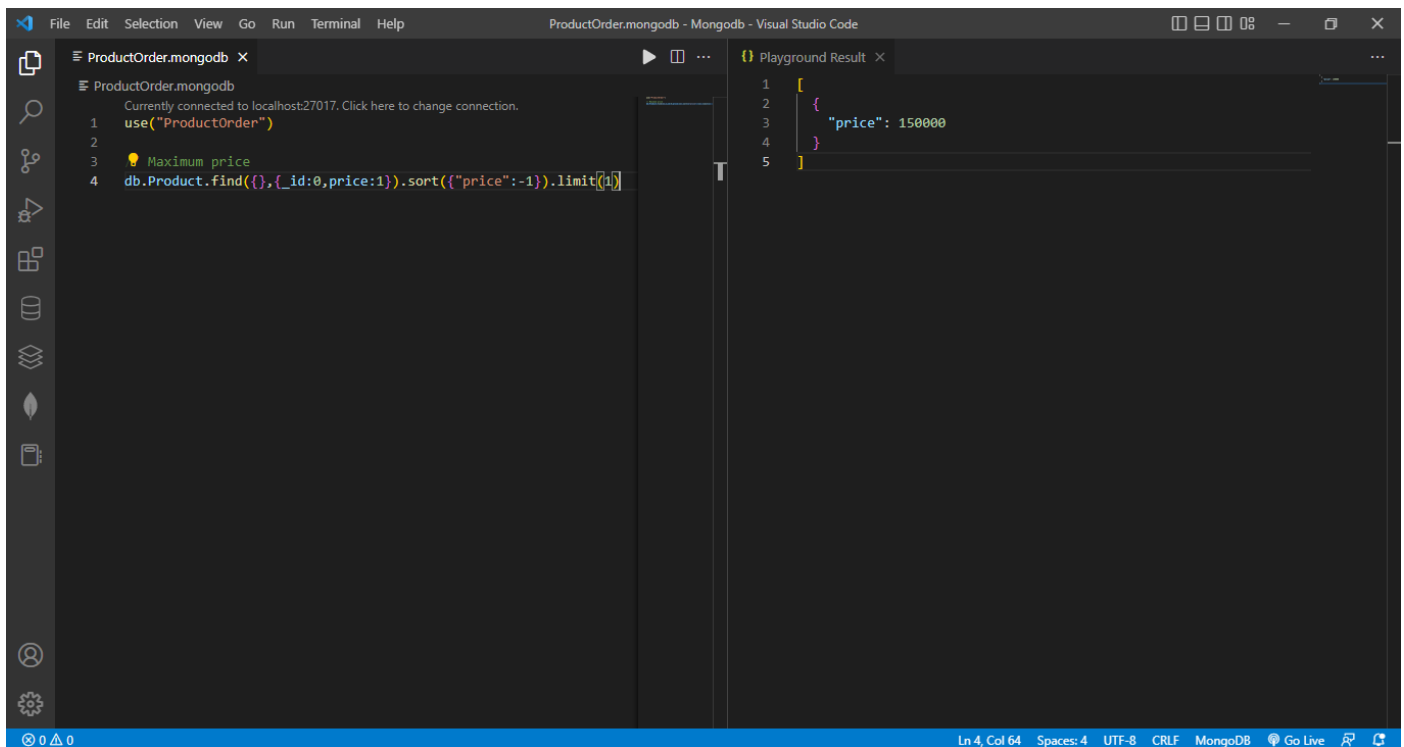


The screenshot shows the Visual Studio Code interface with a MongoDB Playground tab. The code in the editor is as follows:

```
1 use("ProductOrder")
2
3 // Product with Maximum price
4 db.Product.find().sort({"price":-1}).limit(1)
```

The Playground Result tab displays a JSON document representing the product with the highest price:

```
1 [
2   {
3     "_id": {
4       "$oid": "637d1905f6ca17d307096a65"
5     },
6     "sku": "APL-01001",
7     "code": "AppleTV-01",
8     "price": 150000,
9     "created": "2021-11-05 18:32:09",
10    "last_updated": "2021-11-05 18:32:09",
11    "brand": "AppleTV",
12    "model": "AP1",
13    "warranty": 5
14  }
15 ]
```



The screenshot shows the Visual Studio Code interface with a MongoDB Playground tab. The code in the editor is as follows:

```
1 use("ProductOrder")
2
3 // Maximum price
4 db.Product.find({}, {_id:0, price:1}).sort({"price":-1}).limit(1)
```

The Playground Result tab displays a JSON document representing the product with the highest price:

```
1 [
2   {
3     "price": 150000
4   }
5 ]
```

```
ProductOrder.mongodb - MongoDB - Visual Studio Code

ProductOrder.mongodb x
ProductOrder.mongodb
Currently connected to localhost:27017. Click here to change connection.
1 use("ProductOrder")
2
3 ⚡ Product with Minimum price
4 db.Product.find().sort({"price":1}).limit(1)
```

```
Playground Result x
1 [
2   {
3     "_id": {
4       "$oid": "637d1905f6ca17d307096a61"
5     },
6     "sku": "PNS-18001",
7     "code": "Panasonic-01",
8     "price": 62000,
9     "created": "2021-11-01 19:21:00",
10    "last_updated": "2021-11-01 19:21:00",
11    "brand": "Panasonic",
12    "model": "LY",
13    "warranty": 3
14  }
15 ]
```

Ln 4, Col 45 Spaces: 4 UTF-8 CRLF MongoDB Go Live

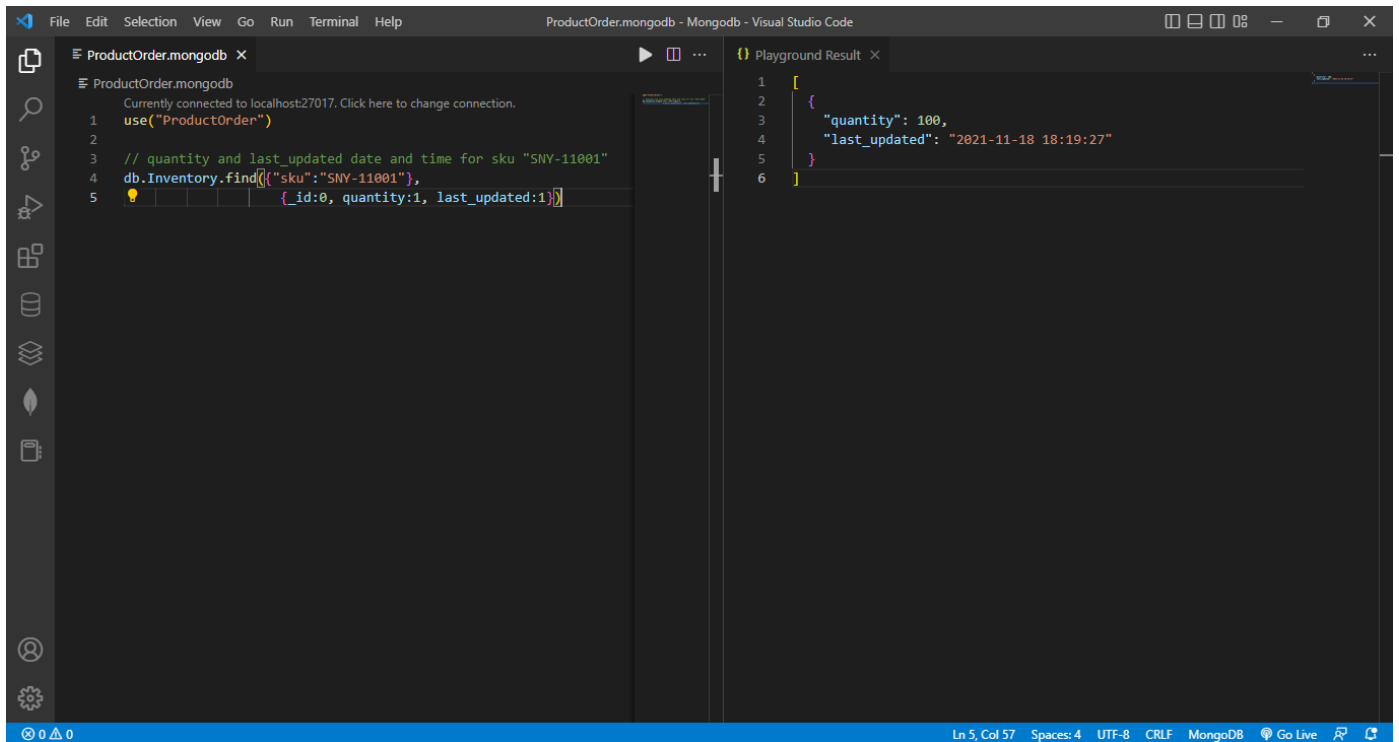
```
ProductOrder.mongodb - MongoDB - Visual Studio Code

ProductOrder.mongodb x
ProductOrder.mongodb
Currently connected to localhost:27017. Click here to change connection.
1 use("ProductOrder")
2
3 ⚡ Minimum price
4 db.Product.find({}, {_id:0, price:1}).sort({"price":1}).limit(1)
```

```
Playground Result x
1 [
2   {
3     "price": 62000
4   }
5 ]
```

Ln 4, Col 63 Spaces: 4 UTF-8 CRLF MongoDB Go Live

7. Display the quantity and last\_updated date and time for sku "SNY-11001"

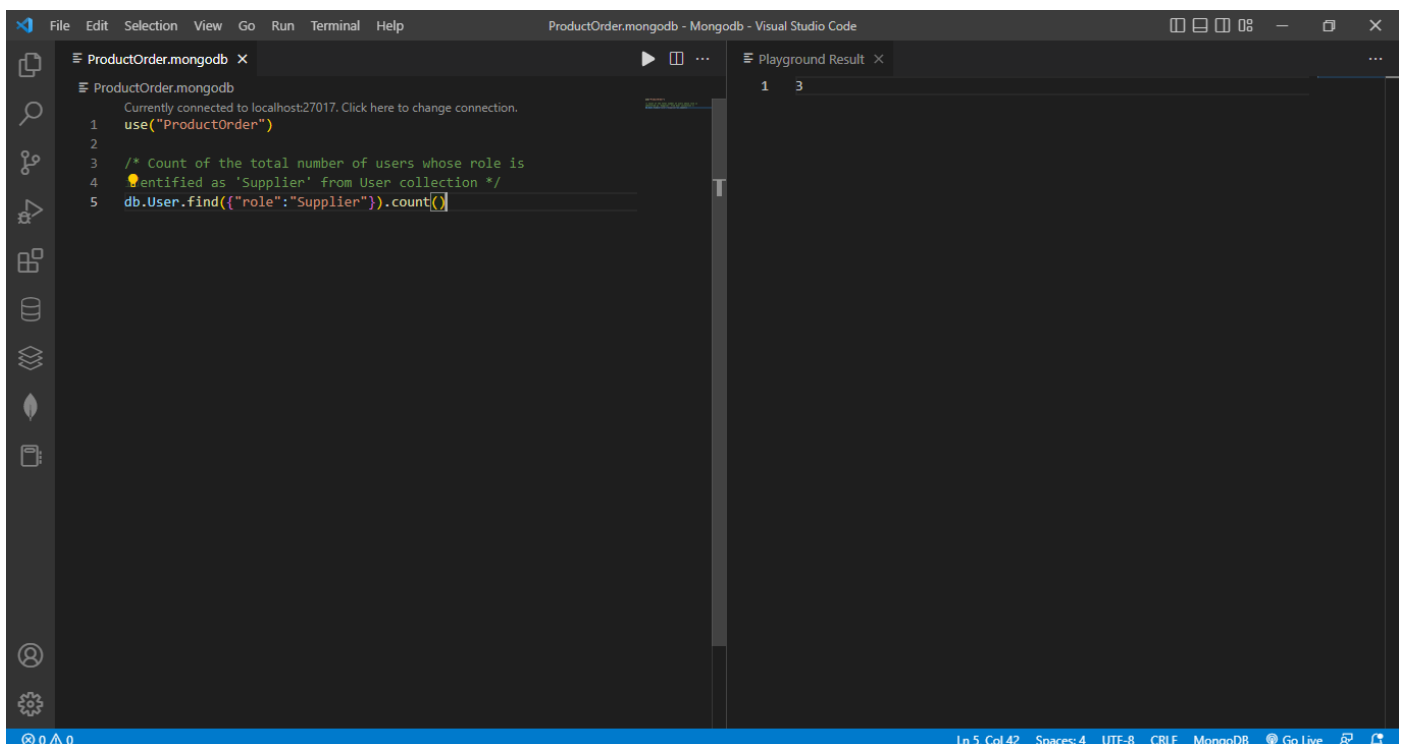


The screenshot shows the Visual Studio Code interface with the MongoDB Playground extension. The left pane displays the code for connecting to the database and running a query. The right pane shows the results of the query in a JSON array format.

```
ProductOrder.mongodb
1 use("ProductOrder")
2
3 // quantity and last updated date and time for sku "SNY-11001"
4 db.Inventory.find({"sku":"SNY-11001"},
5   [{"_id:0, quantity:1, last_updated:1}])
```

```
Playground Result
1 [
2   {
3     "quantity": 100,
4     "last_updated": "2021-11-18 18:19:27"
5   }
6 ]
```

8. List down the count of the total number of users whose role is identified as 'Supplier' from User collection

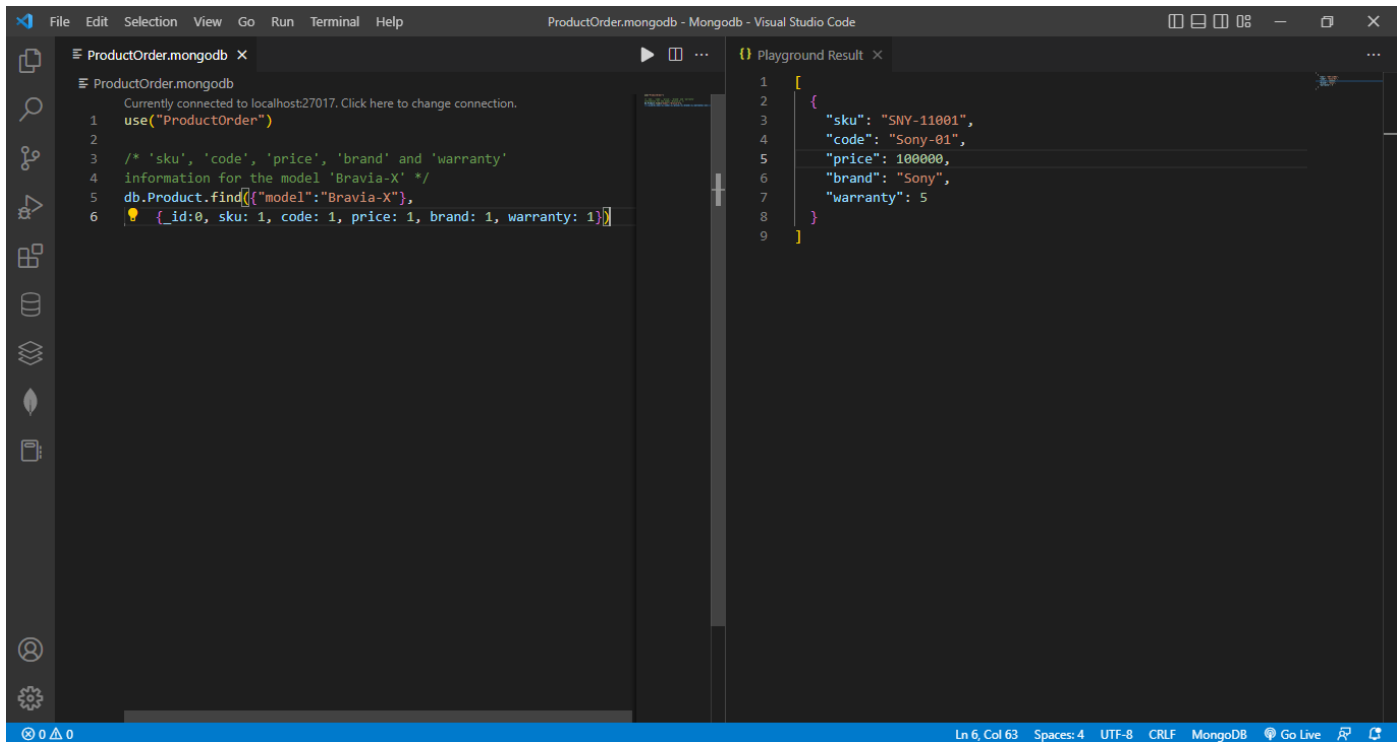


The screenshot shows the Visual Studio Code interface with the MongoDB Playground extension. The left pane displays the code for connecting to the database and running a count query. The right pane shows the result of the query as a single number.

```
ProductOrder.mongodb
1 use("ProductOrder")
2
3 /* Count of the total number of users whose role is
4   identified as 'Supplier' from User collection */
5 db.User.find({"role":"Supplier"}).count()
```

```
Playground Result
1 3
```

9. Display 'sku', 'code', 'price', 'brand' and 'warranty' information for the model 'Bravia-X'



The screenshot shows the Visual Studio Code interface with a file named 'ProductOrder.mongodb'. The code in the editor is as follows:

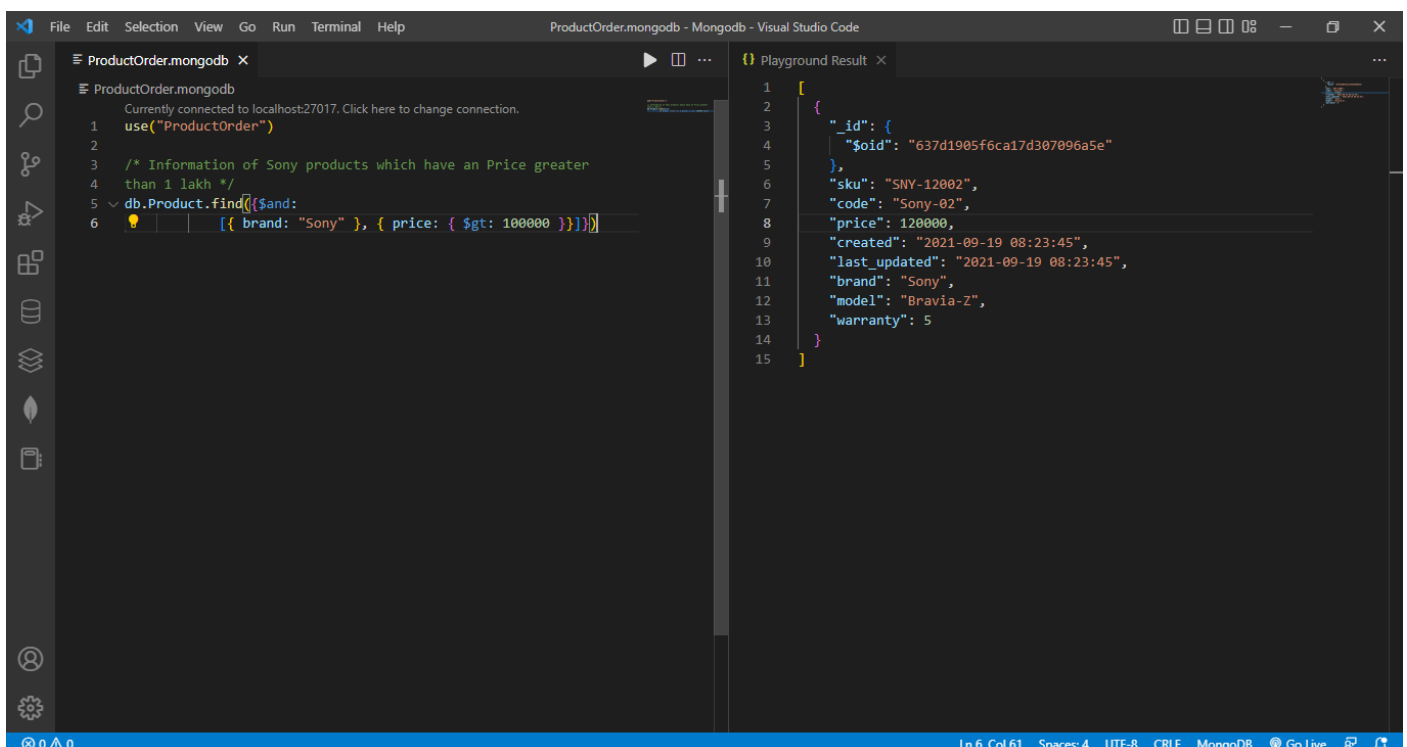
```
1 use("ProductOrder")
2
3 /* 'sku', 'code', 'price', 'brand' and 'warranty'
4 information for the model 'Bravia-X' */
5 db.Product.find({"model": "Bravia-X"},
6 [{"_id:0, sku: 1, code: 1, price: 1, brand: 1, warranty: 1}])
```

The 'Playground Result' panel on the right displays the output of the query:

```
1 [
2   {
3     "sku": "SNY-11001",
4     "code": "Sony-01",
5     "price": 100000,
6     "brand": "Sony",
7     "warranty": 5
8   }
9 ]
```

The status bar at the bottom indicates 'Ln 6, Col 63', 'Spaces: 4', 'UTF-8', 'CRLF', 'MongoDB', and 'Go Live'.

10. Find all the information of Sony products which have an Price greater than 1 lakh



The screenshot shows the Visual Studio Code interface with a file named 'ProductOrder.mongodb'. The code in the editor is as follows:

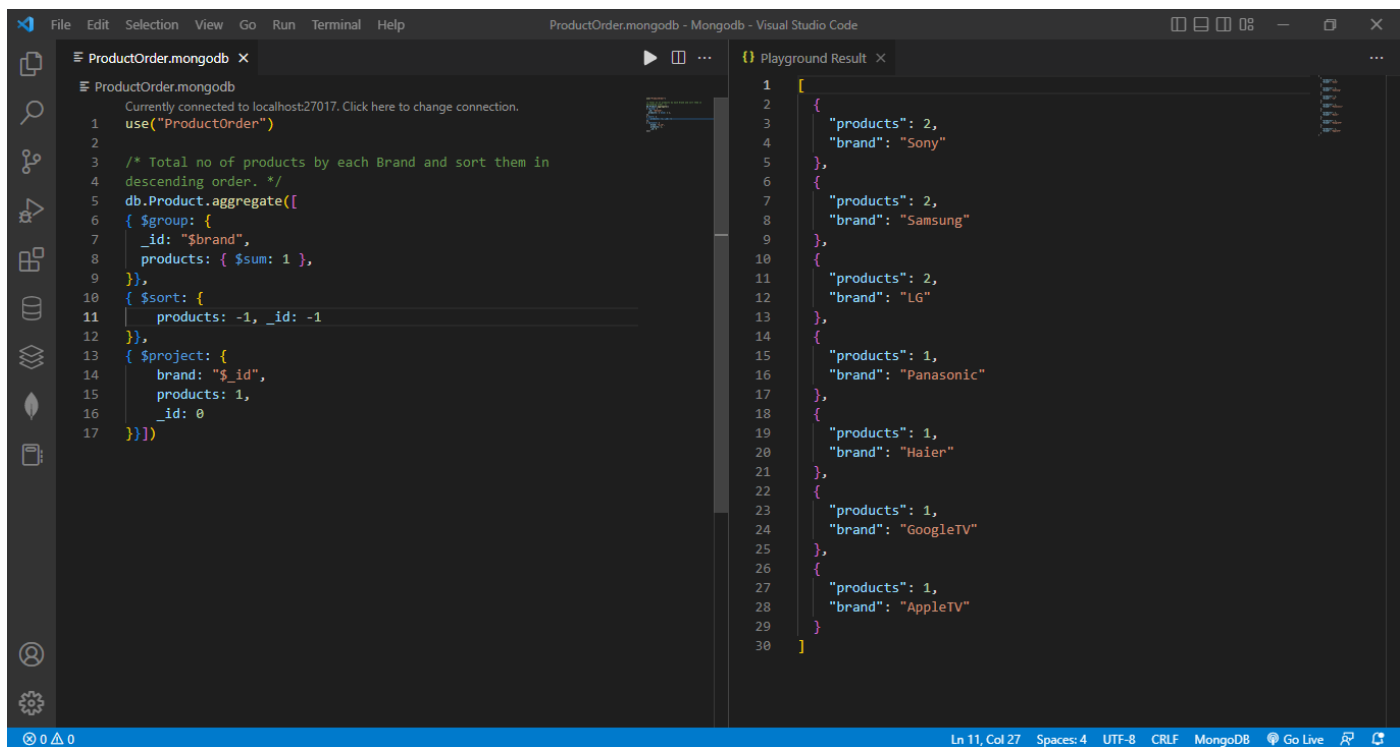
```
1 use("ProductOrder")
2
3 /* Information of Sony products which have an Price greater
4 than 1 lakh */
5 db.Product.find({$and:
6 [{"brand": "Sony"}, {"price": {"$gt: 100000}}]})
```

The 'Playground Result' panel on the right displays the output of the query:

```
1 {
2   "_id": {
3     "$oid": "637d1905f6ca17d307096a5e"
4   },
5   "sku": "SNY-12002",
6   "code": "Sony-02",
7   "price": 120000,
8   "created": "2021-09-19 08:23:45",
9   "last_updated": "2021-09-19 08:23:45",
10  "brand": "Sony",
11  "model": "Bravia-Z",
12  "warranty": 5
13 }
14 ]
15
```

The status bar at the bottom indicates 'Ln 6, Col 61', 'Spaces: 4', 'UTF-8', 'CRLF', 'MongoDB', and 'Go Live'.

11. Find the total no of products by each Brand and sort them in descending order.

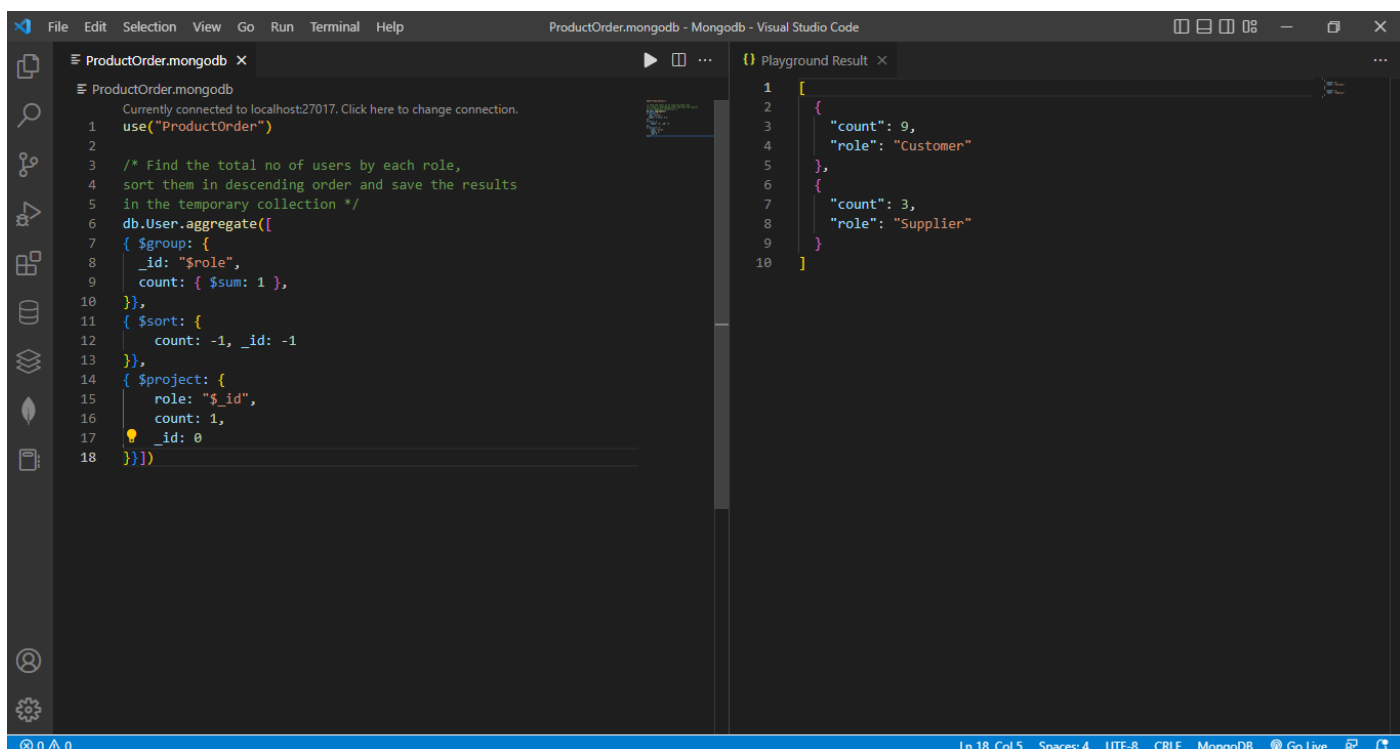


The screenshot shows the Visual Studio Code interface with a MongoDB connection to localhost:27017. The left pane displays a MongoDB aggregation query in the 'ProductOrder.mongodb' file. The query uses the 'Product' collection and the 'aggregate' method to group products by brand, sort them in descending order of product count, and project the brand and product count. The right pane shows the 'Playground Result' output, which is a JSON array of objects, each containing a brand name and its corresponding product count.

```
1 use("ProductOrder")
2
3 /* Total no of products by each Brand and sort them in
4 descending order. */
5 db.Product.aggregate([
6   { $group: {
7     _id: "$brand",
8     products: { $sum: 1 },
9   }},
10  { $sort: {
11    products: -1, _id: -1
12  }},
13  { $project: {
14    brand: "$_id",
15    products: 1,
16    _id: 0
17  }}])
```

```
1 [
2   {
3     "products": 2,
4     "brand": "Sony"
5   },
6   {
7     "products": 2,
8     "brand": "Samsung"
9   },
10  {
11    "products": 2,
12    "brand": "LG"
13  },
14  {
15    "products": 1,
16    "brand": "Panasonic"
17  },
18  {
19    "products": 1,
20    "brand": "Haier"
21  },
22  {
23    "products": 1,
24    "brand": "GoogleTV"
25  },
26  {
27    "products": 1,
28    "brand": "AppleTV"
29  }
30 ]
```

12. Find the total no of users by each role, sort them in descending order and save the results in the temporary collection



The screenshot shows the Visual Studio Code interface with a MongoDB connection to localhost:27017. The left pane displays a MongoDB aggregation query in the 'ProductOrder.mongodb' file. The query uses the 'User' collection and the 'aggregate' method to group users by role, sort them in descending order of user count, and project the role and user count. The right pane shows the 'Playground Result' output, which is a JSON array of objects, each containing a role name and its corresponding user count.

```
1 use("ProductOrder")
2
3 /* Find the total no of users by each role,
4 sort them in descending order and save the results
5 in the temporary collection */
6 db.User.aggregate([
7   { $group: {
8     _id: "$role",
9     count: { $sum: 1 },
10   }},
11  { $sort: {
12    count: -1, _id: -1
13  }},
14  { $project: {
15    role: "$_id",
16    count: 1,
17    _id: 0
18  }}])
```

```
1 [
2   {
3     "count": 9,
4     "role": "Customer"
5   },
6   {
7     "count": 3,
8     "role": "Supplier"
9   }
10 ]
```



ProductOrder.mongodb - Visual Studio Code

ProductOrder.mongodb

Currently connected to localhost:27017. Click here to change connection.

```
1 use("ProductOrder")
2
3 /* Find the total no of users by each role,
4 sort them in descending order and save the results
5 in the temporary collection */
6 db.User.aggregate([
7   { $group: {
8     _id: "$role",
9     count: { $sum: 1 }},
10  },
11  { $sort: {
12    count: -1, _id: -1
13  }},
14  { $project: {
15    role: "$_id",
16    count: 1,
17    _id: 0
18  }},
19  // write output as a new collection
20  {
21    $out: "user_count_by_role"
22  })
23
24 db.user_count_by_role.find()
```

Playground Result

```
1 [
2   {
3     Edit Document
4     "_id": {
5       "$oid": "637fbf07158d341a5c46633a"
6     },
7     "count": 9,
8     "role": "Customer"
9   },
10  {
11    Edit Document
12    "_id": {
13      "$oid": "637fbf07158d341a5c46633b"
14    },
15    "count": 3,
16    "role": "Supplier"
17  }
18 ]
```

Ln 24, Col 29 Spaces: 4 UTF-8 CRLF MongoDB Go Live