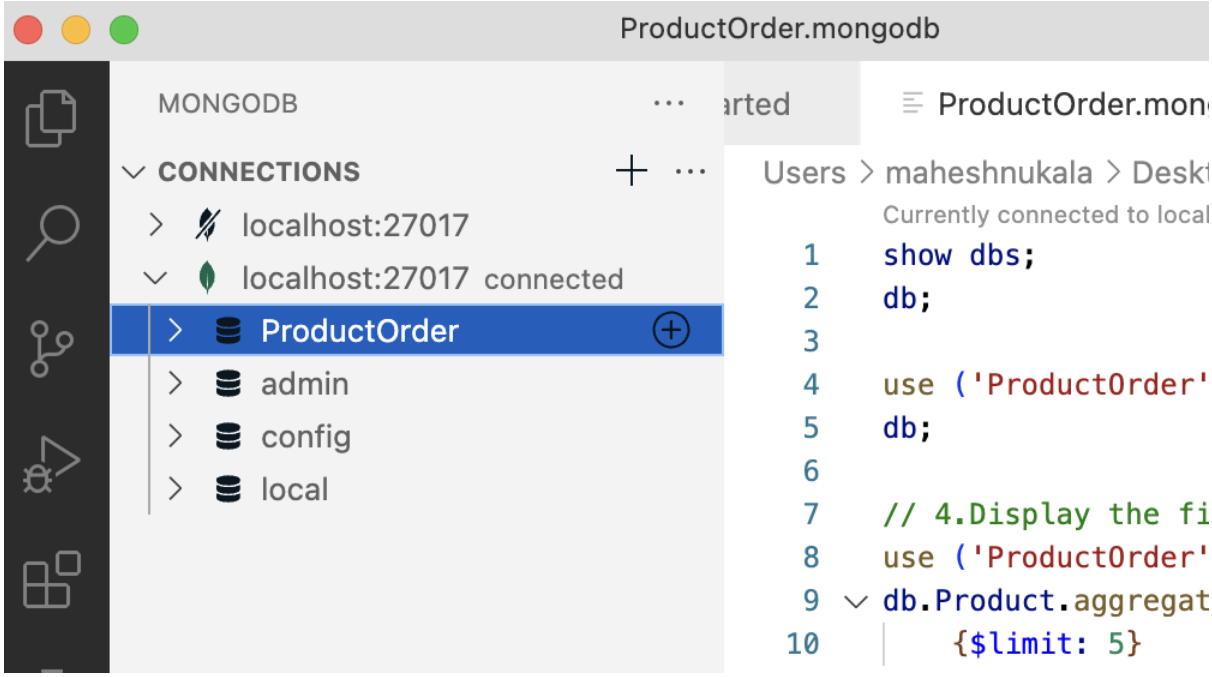
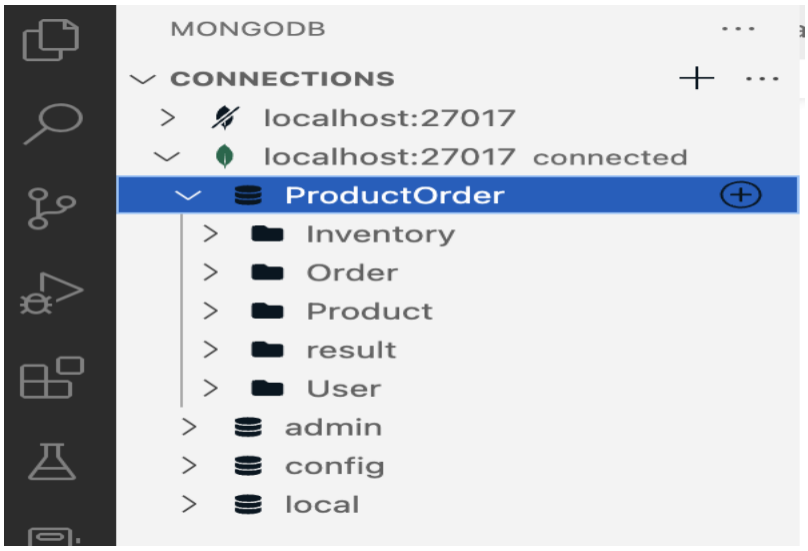
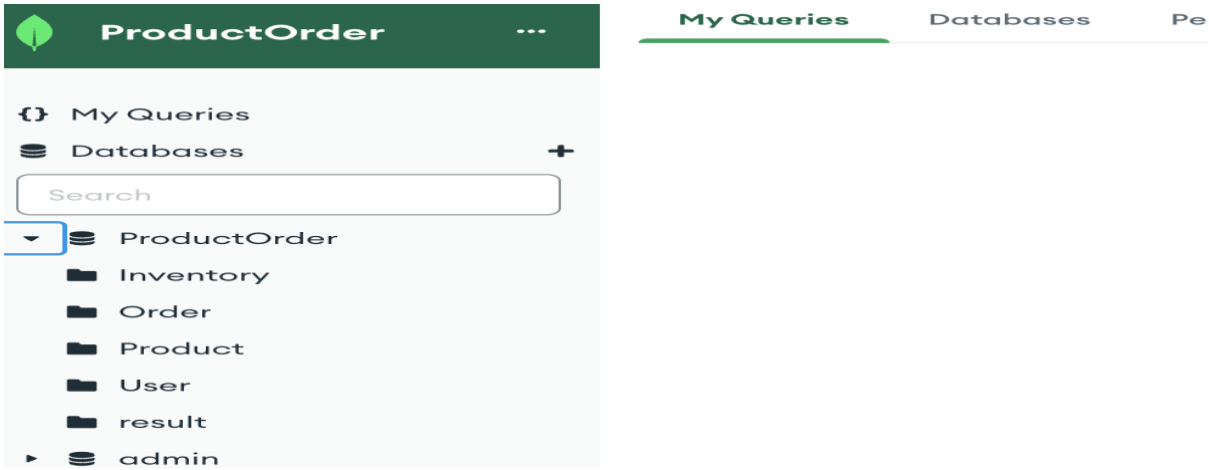
1. Open VS Code and connect to MONGODB



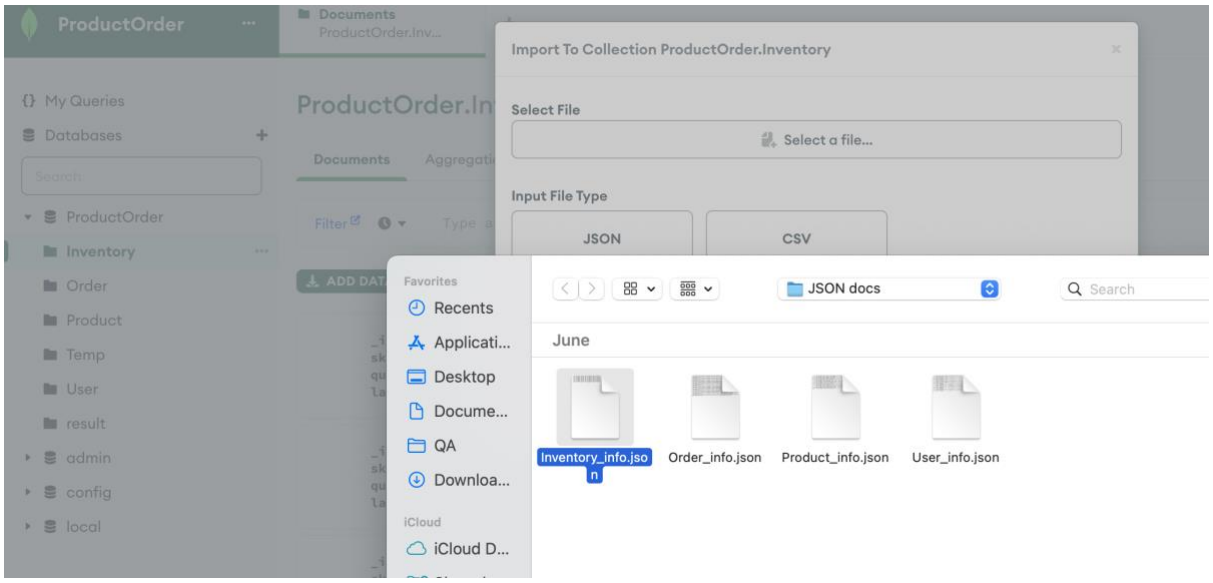
2. Create a database "ProductOrder" and create collections "Product","Inventory","User", and "Order" in it.





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.



Same as above pic added all the files into the collections

Inventory data added:

Documents
ProductOrder.Inv...

ProductOrder.Inventory

Documents | Aggregations | Schema | Explain Plan | Indexes | Validati

Filter | Type a query: { field: 'value' }

ADD DATA | EXPORT COLLECTION

```
quantity: 100
last_updated: "2021-11-18 18:19:27"

_id: ObjectId('6386f6d87c17a8c4b99d430e')
sku: "SNY-12002"
quantity: 80
last_updated: "2021-12-01 23:12:45"

_id: ObjectId('6386f6d87c17a8c4b99d430f')
sku: "SMG-21001"
quantity: 400
last_updated: "2021-10-31 19:30:18"

_id: ObjectId('6386f6d87c17a8c4b99d4310')
sku: "LLG-32001"
quantity: 450
last_updated: "2021-11-28 12:34:56"

_id: ObjectId('6386f6d87c17a8c4b99d4311')
sku: "PNS-18001"
quantity: 500
last_updated: "2021-12-01 11:34:21"

_id: ObjectId('6386f6d87c17a8c4b99d4312')
```

Product data added

ProductOrder.Product

Documents | Aggregations | Schema | Explain Plan | Ind

Filter | Type a query: { field: 'value' }

ADD DATA | EXPORT COLLECTION

```
_id: ObjectId('6386f6bd7c17a8c4b99d4302')
sku: "SNY-11001"
code: "Sony-01"
price: 100000
created: "2021-08-09 12:32:56"
last_updated: "2021-08-09 12:32:56"
brand: "Sony"
model: "Bravia-X"
warranty: 5

_id: ObjectId('6386f6bd7c17a8c4b99d4303')
sku: "SNY-12002"
code: "Sony-02"
price: 120000
created: "2021-09-19 08:23:45"
```

6386f6bd7c17a8c4b99d4303

Order data added

ProductOrder.Order

Documents | Aggregations | Schema | Explain Plan

Filter | Type a query: { field: 'value' }

ADD DATA | EXPORT COLLECTION

```
_id: ObjectId('6386f6e27c17a8c4b99d4318')
created: "2021-12-06 23:12:09"
last_updated: "2021-12-07 20:30:01"
> items: Array
total_price: 100000
discount: 10000
net_price: 90000
status: 1
user_email: "sudha.nat@yourmail.com"
```

```
_id: ObjectId('6386f6e27c17a8c4b99d4319')
created: "2021-12-07 12:18:01"
last_updated: "2021-12-07 23:59:59"
> items: Array
total_price: 192000
```

User data added

ProductOrder.User

ProductOrder.User

Documents | Aggregations | Schema | Explain Plan | In

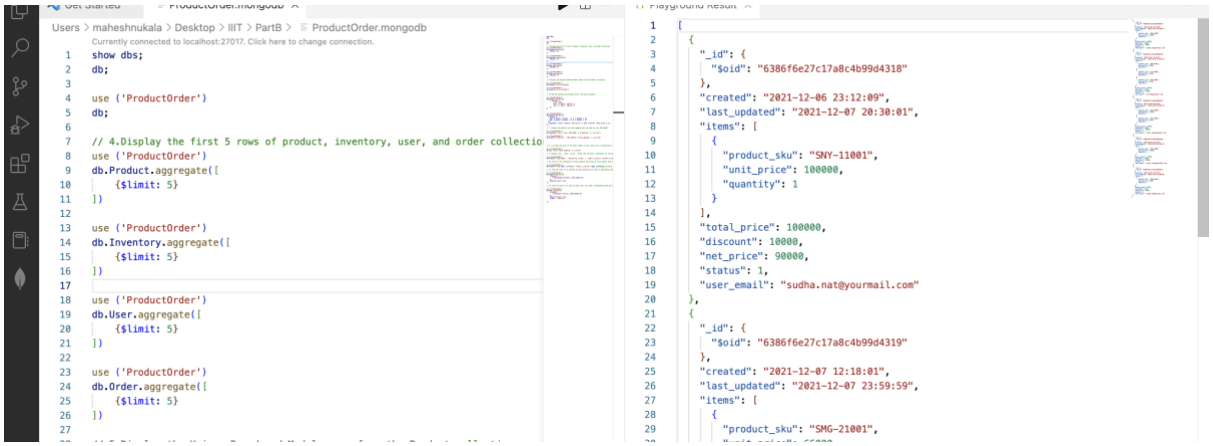
Filter | Type a query: { field: 'value' }

ADD DATA | EXPORT COLLECTION

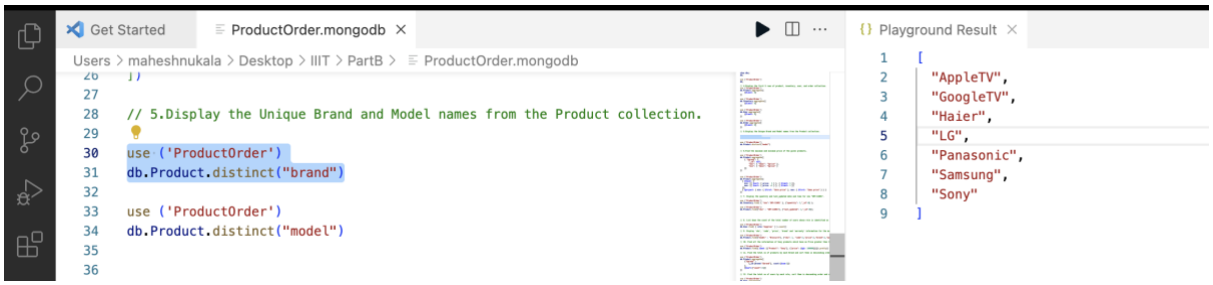
```
_id: ObjectId('6386f6ee7c17a8c4b99d4324')
name: "Anjali Gupta"
email: "anjali.gupta@zestmail.com"
created: "2020-12-08 13:01:56"
last_accessed: "2021-12-04 13:23:00"
role: "Customer"
```

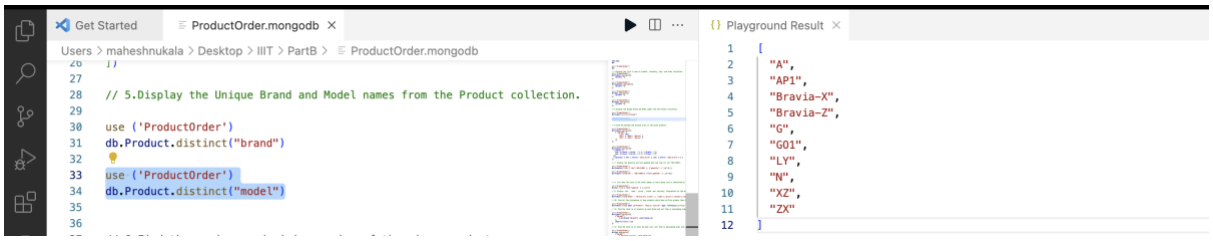
```
_id: ObjectId('6386f6ee7c17a8c4b99d4325')
```

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

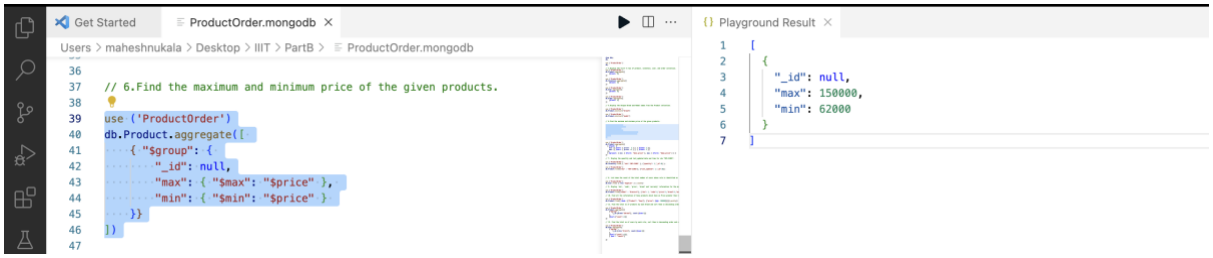


5. Display the Unique Brand and Model names from the Product collection





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



```
use ('ProductOrder')
db.Product.aggregate([
  { $facet: {
    min: [{ $sort: { price: 1 } }, { $limit: 1 }],
    max: [{ $sort: { price: -1 } }, { $limit: 1 }]
  }},
  { $project: { min: { $first: "$min.price" }, max: { $first: "$max.price" } }
])
```

```
[
  {
    "min": 62000,
    "max": 150000
  }
]
```

7. Display the quantity and last_updated date and time for sku "SNY-11001".

```
// 7. Display the quantity and last_updated date and time for sku "SNY-11001".

use ('ProductOrder')
db.Inventory.find( { "sku":'SNY-11001' }, {"quantity": 1,"_id":0} );

use ('ProductOrder')
db.Product.find({"sku" : "SNY-11001"}, {"last_updated": 1,"_id":0});
```

```
[
  {
    "quantity": 100
  }
]
```

```
// 7. Display the quantity and last_updated date and time for sku "SNY-11001".

use ('ProductOrder')
db.Inventory.find( { "sku":'SNY-11001' }, {"quantity": 1,"_id":0} );

use ('ProductOrder')
db.Product.find({"sku" : "SNY-11001"}, {"last_updated": 1,"_id":0});
```

```
[
  {
    "last_updated": "2021-08-09 12:32:56"
  }
]
```

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

```
// 8. List down the count of the total number of users whose role is identifie

use ('ProductOrder')
db.User.find( { role:'Supplie
```

MongoDB: Run All or Selection From Playground

```
3
```

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

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

use ('ProductOrder')
db.Product.find({"model" : "Bravia-X"}, {"sku": 1, "code":1,"price":1,"brand":1,"warranty":1,"_id":0});

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

```
[
  {
    "sku": "SNY-11001",
    "code": "Sony-01",
    "price": 100000,
    "brand": "Sony",
    "warranty": 5
  }
]
```

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

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

use ('ProductOrder')
db.Product.find({ $and: [{"brand": "Sony"}, {"price": {$gt: 100000}}]})

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

use ('ProductOrder')
db.Product.aggregate([
    {"$group" :
        {_id:{brand:"$brand"}, count:{$sum:1}}
    },
    {$sort:{"count":-1}}
```

```
[
  {
    "_id": {
      "$oid": "6386f6bd7c17a8c4b99d4303"
    },
    "sku": "SNY-12002",
    "code": "Sony-02",
    "price": 120000,
    "created": "2021-09-19 08:23:45",
    "last_updated": "2021-09-19 08:23:45",
    "brand": "Sony",
    "model": "Bravia-Z",
    "warranty": 5
  }
]
```

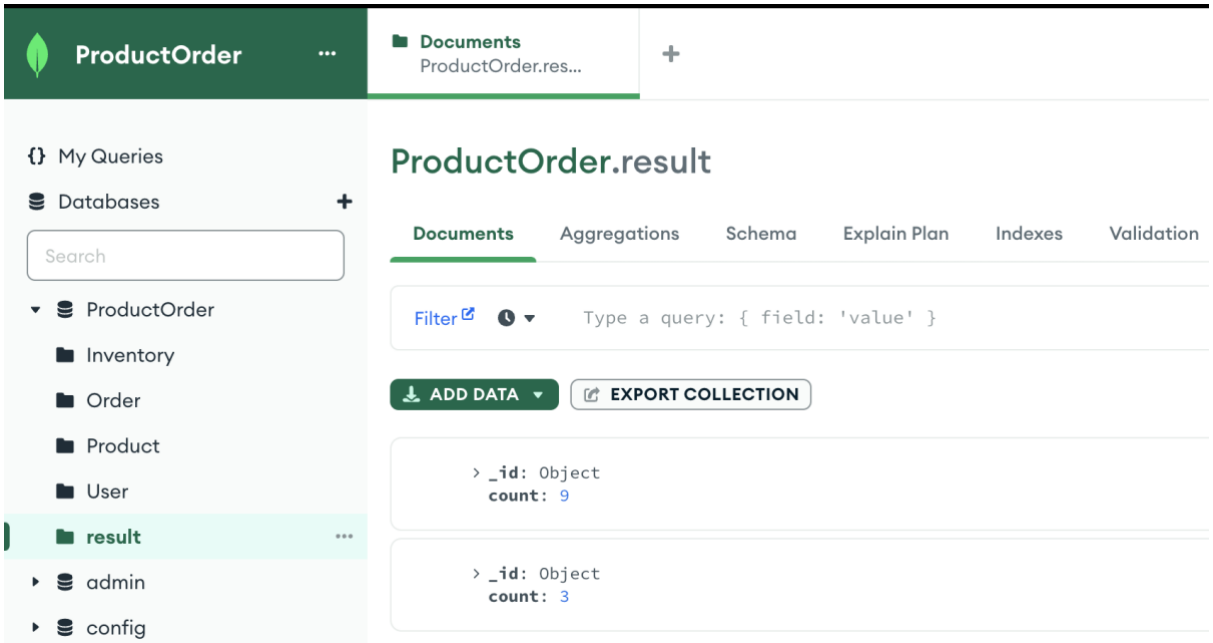
11. Find the total no of products by each Brand and sort them in descending order.

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

use ('ProductOrder')
db.Product.aggregate([
    {"$group" :
        {_id:{brand:"$brand"}, count:{$sum:1}}
    },
    {$sort:{"count":-1}}
])

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

use ('ProductOrder')
db.User.aggregate([
    {"$group" :
        {_id:{role:"$role"}, count:{$sum:1}}
    },
    {$sort:{"count":-1}},
    { $out : "result"}
])
```

```
[
  {
    "_id": {
      "brand": "Samsung"
    },
    "count": 2
  },
  {
    "_id": {
      "brand": "LG"
    },
    "count": 2
  },
  {
    "_id": {
      "brand": "Sony"
    },
    "count": 2
  },
  {
    "_id": {
      "brand": "AppleTV"
    },
    "count": 1
  },
  {
    "_id": {
      "brand": "GoogleTV"
    },
    "count": 1
  },
  {
    "_id": {
      "brand": "Haier"
    },
    "count": 1
  },
  {
    "_id": {
      "brand": "Panasonic"
    },
    "count": 1
  }
]
```

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

```
// 12. Find the total no of users by each role, sor

use ('ProductOrder')
db.User.aggregate([
    {"$group" :
        {_id:{role:"$role"}, count:{$sum:1}}
    },
    {$sort:{"count":-1}},
    { $out : "result"}
])
```


ProductOrder
Documents
ProductOrder.res...
My Queries
Databases
Search
ProductOrder
Inventory
Order
Product
User
result
admin
config
ProductOrder.result
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
Filter
Type a query: { field: 'value' }
ADD DATA
EXPORT COLLECTION
_id: Object
count: 9
_id: Object
count: 3