| NAME | Konika Khurana |
| --- | --- |
| UID | 23BCS12925 |
| CLASS | 622-A |

➢ NodeJs PRACTISE 5.3

- CODE

catalog.js (Main server):

```
// catalog.js
const mongoose = require('mongoose');
const Product = require('./models/Product');

// MongoDB connection
mongoose.connect('mongodb+srv://alpha:alpha2025@cluster0.gttfsb5.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0')
.then(() => console.log(' Connected to MongoDB'))
.catch(err => console.error(' Error:', err.message));

// Insert sample products
async function seedData() {
  await Product.deleteMany(); // Clear old data

  const products = [
    {
      name: "T-Shirt",
      price: 499,
      category: "Clothing",
      variants: [
        { color: "Red", size: "M", stock: 30 },
        { color: "Blue", size: "L", stock: 25 },
```

```
        { color: "Black", size: "S", stock: 20 }
      ]
    },
    {
      name: "Sneakers",
      price: 2999,
      category: "Footwear",
      variants: [
        { color: "White", size: "8", stock: 15 },
        { color: "Black", size: "9", stock: 10 }
      ]
    },
    {
      name: "Laptop Bag",
      price: 1499,
      category: "Accessories",
      variants: [
        { color: "Gray", size: "Medium", stock: 40 },
        { color: "Blue", size: "Large", stock: 12 }
      ]
    }
  ];

  await Product.insertMany(products);
  console.log("Sample data inserted successfully!");
  process.exit();
}

seedData();
```

/models/Product.js :

```
// models/Product.js
const mongoose = require('mongoose');

const VariantSchema = new mongoose.Schema({
  color: { type: String, required: true },
  size: { type: String, required: true },
  stock: { type: Number, required: true, min: 0 }
});

const ProductSchema = new mongoose.Schema({
  name: { type: String, required: true, trim: true },
```

```
  price: { type: Number, required: true, min: 0 },
  category: { type: String, required: true, trim: true },
  variants: [VariantSchema],  // nested array of variant documents
  createdAt: { type: Date, default: Date.now }
});

module.exports = mongoose.model('Product', ProductSchema);
```

## *Example Queries (In MongoDB Atlas Filter):*

### a) Retrieve all products

{ }

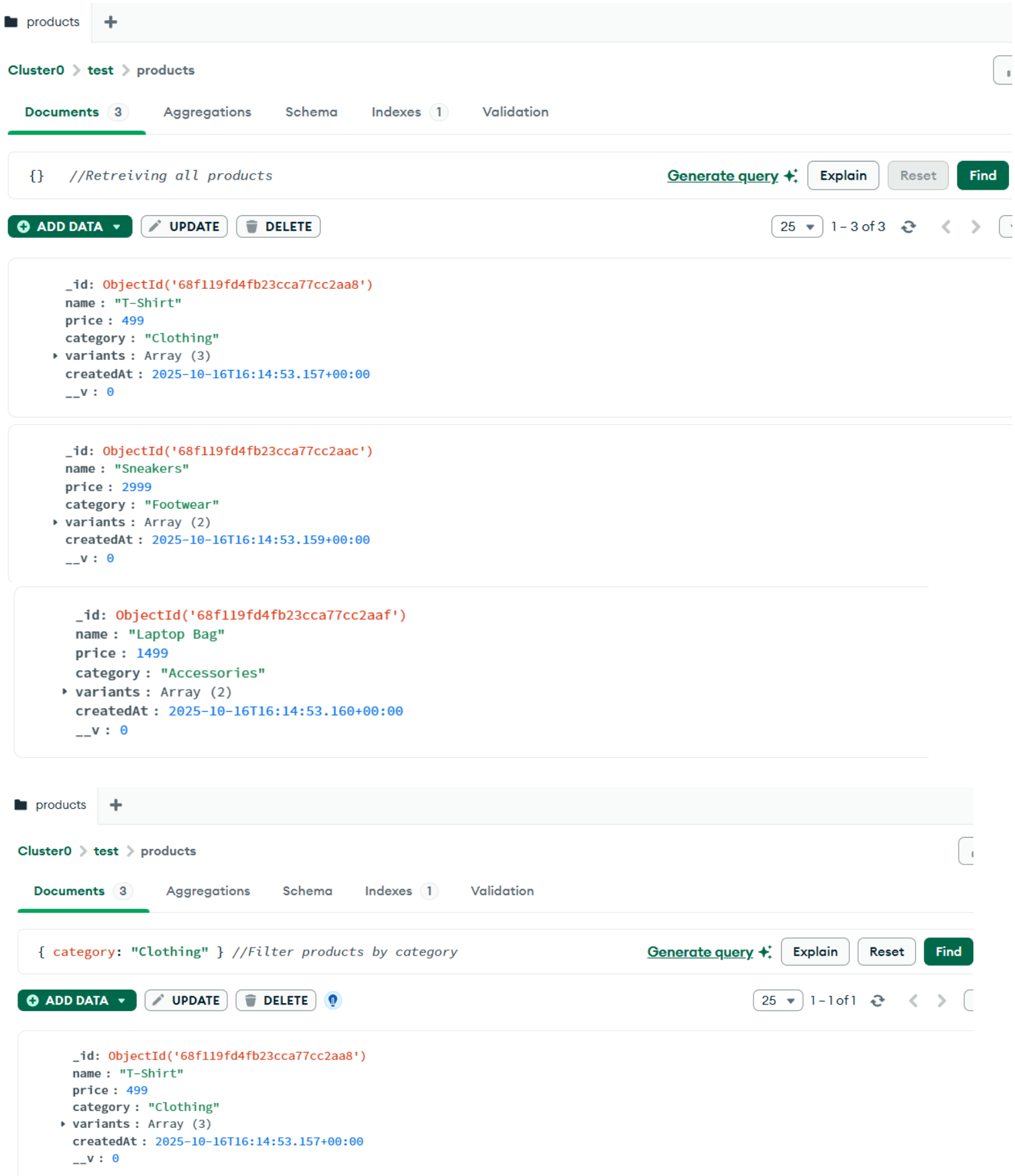
### b) Filter products by category

{ category: "Clothing" }

### c) Find products that have a variant in size “M”

{ "variants.size": "M" }

OUTPUT


products
Cluster0 > test > products
Documents 3
Aggregations
Schema
Indexes 1
Validation
{} //Retreiving all products
Generate query
Explain
Reset
Find
ADD DATA
UPDATE
DELETE
25
1 – 3 of 3
_id: ObjectId('68f119fd4fb23cca77cc2aa8')
name : "T-Shirt"
price : 499
category : "Clothing"
variants : Array (3)
createdAt : 2025-10-16T16:14:53.157+00:00
__v : 0
_id: ObjectId('68f119fd4fb23cca77cc2aac')
name : "Sneakers"
price : 2999
category : "Footwear"
variants : Array (2)
createdAt : 2025-10-16T16:14:53.159+00:00
__v : 0
_id: ObjectId('68f119fd4fb23cca77cc2aaf')
name : "Laptop Bag"
price : 1499
category : "Accessories"
variants : Array (2)
createdAt : 2025-10-16T16:14:53.160+00:00
__v : 0
products
Cluster0 > test > products
Documents 3
Aggregations
Schema
Indexes 1
Validation
{ category: "Clothing" } //Filter products by category
Generate query
Explain
Reset
Find
ADD DATA
UPDATE
DELETE
25
1 – 1 of 1
_id: ObjectId('68f119fd4fb23cca77cc2aa8')
name : "T-Shirt"
price : 499
category : "Clothing"
variants : Array (3)
createdAt : 2025-10-16T16:14:53.157+00:00
__v : 0

products +

Cluster0 > test > products

Documents 3 | Aggregations | Schema | Indexes 1 | Validation

```
{ "variants.color": "Blue" } //Filter products by color
```

Generate query | Explain | Reset | Find

ADD DATA | UPDATE | DELETE

25 | 1 – 2 of 2

```
_id: ObjectId('68f119fd4fb23cca77cc2aa8')
name : "T-Shirt"
price : 499
category : "Clothing"
▸ variants : Array (3)
createdAt : 2025-10-16T16:14:53.157+00:00
__v : 0
```

```
_id: ObjectId('68f119fd4fb23cca77cc2aaf')
name : "Laptop Bag"
price : 1499
category : "Accessories"
▸ variants : Array (2)
createdAt : 2025-10-16T16:14:53.160+00:00
__v : 0
```