

## Experiment 6: MongoDB

1) **Aim:** To study CRUD operations in MongoDB

2) **Problem Statement:**

A) Create a database, create a collection, insert data, query and manipulate data using various MongoDB operations.

1. Create a database named "inventory".
2. Create a collection named "products" with the fields: (ProductID, ProductName, Category, Price, Stock).
3. Insert 10 documents into the "products" collection.
4. Display all the documents in the "products" collection.
5. Display all the products in the "Electronics" category.
6. Display all the products in ascending order of their names.
7. Display the details of the first 5 products.
8. Display the categories of products with a specific name.
9. Display the number of products in the "Electronics" category.
10. Display all the products without showing the "\_id" field.
11. Display all the distinct categories of products.
12. Display products in the "Electronics" category with prices greater than 50 but less than 100.
13. Change the price of a product.
14. Delete a particular product entry.

3) **Theory:**

A. Describe some of the features of MongoDB?

- **Flexible Schema:** MongoDB is schema-less, meaning it can store documents with different structures in the same collection.
- **Scalability:** It supports horizontal scaling using sharding.
- **High Performance:** Efficient for read and write operations.
- **Replication:** Provides data redundancy and high availability using replica sets.
- **Indexing:** Supports various types of indexes for efficient query execution.
- **Aggregation Framework:** Allows powerful data aggregation and transformation.

## B. What are Documents and Collections in MongoDB?

- **Document:** A document in MongoDB is a JSON-like data structure called BSON (Binary JSON). It consists of field-value pairs, similar to a row in a relational database.
- **Collection:** A collection is a group of MongoDB documents, equivalent to a table in relational databases. Documents within a collection can have varying structures.

## C. When to use MongoDB?

- When dealing with large volumes of unstructured or semi-structured data.
- For applications requiring horizontal scalability.
- When frequent schema changes are expected.
- For real-time analytics and content management systems.

## D. What is Sharding in MongoDB?

- **Sharding** is a method of horizontally partitioning data across multiple servers to handle large datasets.
- MongoDB uses **shards** to store subsets of data, ensuring improved read and write performance.
- A **Shard Key** is used to distribute data evenly across shards.

#### 4) Output:

##### 1) Create a database and collection

## Create Database

Database Name

Collection Name

☐ **Time-Series**  
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

➤ **Additional preferences** (e.g. Custom collation, Clustered collections)

**i** Before MongoDB can save your new database, a collection name must also be specified at the time of creation. [More Information](#)

Cancel Create Database

## Create Collection

Collection Name

☐ **Time-Series**  
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)



➤ **Additional preferences** (e.g. Custom collation, Clustered collections)

Cancel Create Collection



## 2) Insert Data

### Insert Document

To collection Inventory.products

VIEW  

```
1 {  
2   "ProductID": 1,  
3   "ProductName": "Laptop",  
4   "Category": "Electronics",  
5   "Price": 900,  
6   "Stock": 15  
7 }
```

Cancel

Insert

MongoDB Compass - Inventory/Inventory.products

Connections Edit View Collection Help

Compass

My Queries

CONNECTIONS (3)

Search connections

Inventory

Inventory

products

admin

config

local

personality\_db

Predictor\_db

univota.ylp7g.mongodb.net

products

Inventory > Inventory > products

Documents 10 Aggregations Schema Indexes 1 Validation

{ "Category": "Electronics" }

Generate query + Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

25 1 - 5 of 5

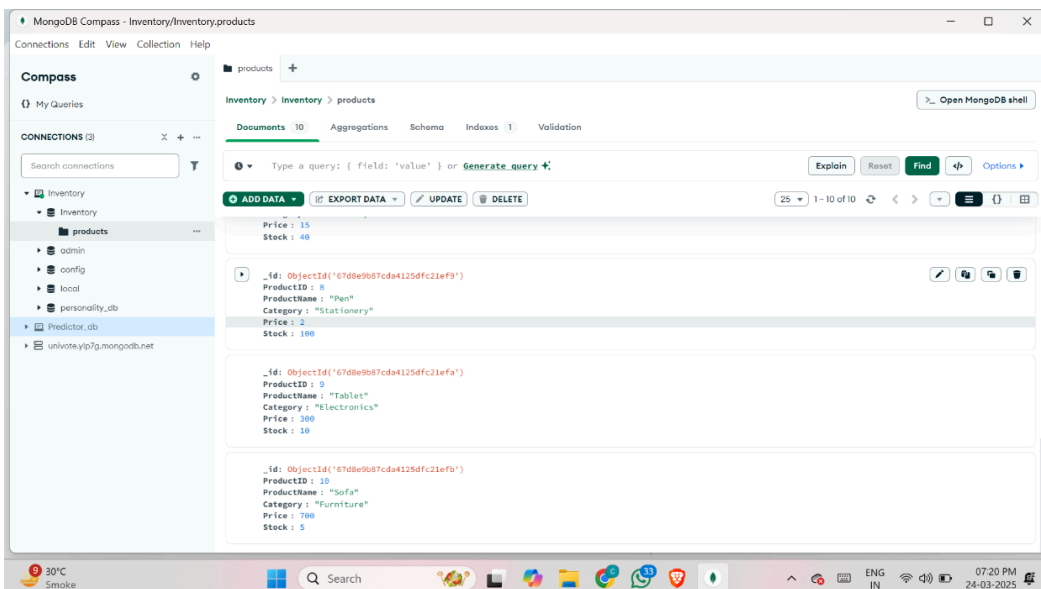
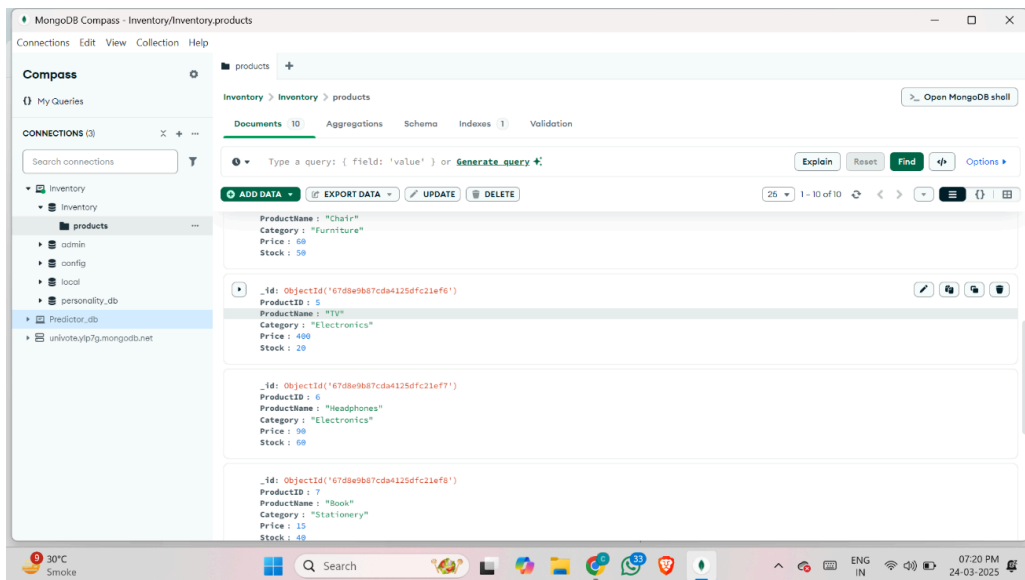
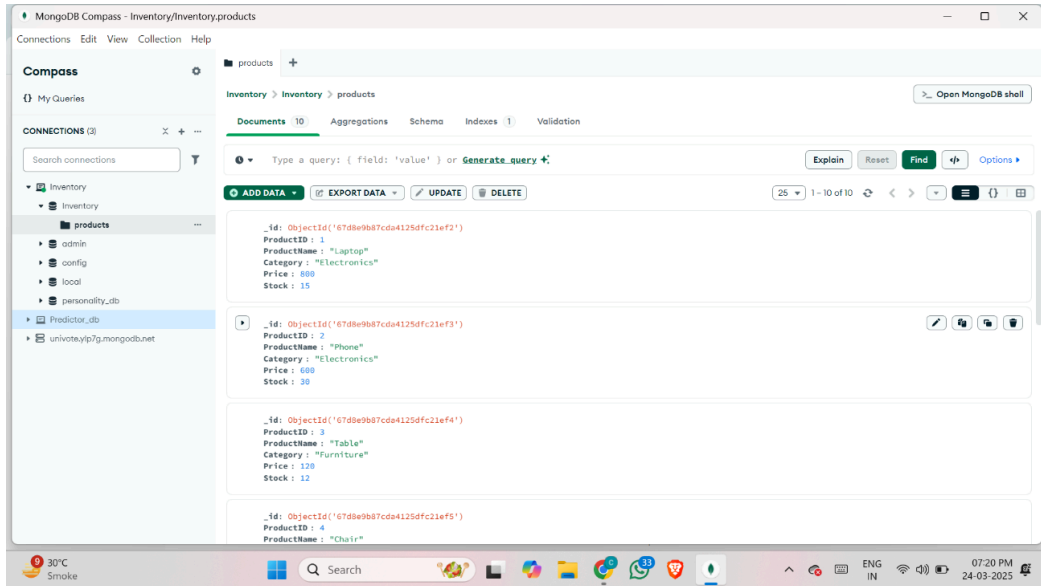
\_id: ObjectId('67d8e9b87cda4125dfc21ef2')  
ProductID: 1  
ProductName: "Laptop"  
Category: "Electronics"  
Price: 900  
Stock: 15

\_id: ObjectId('67d8e9b87cda4125dfc21ef3')  
ProductID: 2  
ProductName: "Phone"  
Category: "Electronics"  
Price: 600  
Stock: 30

\_id: ObjectId('67d8e9b87cda4125dfc21ef6')  
ProductID: 5  
ProductName: "TV"  
Category: "Electronics"  
Price: 400  
Stock: 20

\_id: ObjectId('67d8e9b87cda4125dfc21ef7')  
ProductID: 6  
ProductName: "Headphones"  
Category: "Electronics"

### 3) Display all Documents



## 4) Display all Products in the Electronics Category

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: 'Inventory' database with 'products' collection. The main panel shows a query filter: `{ "Category": "Electronics" }`. The results table displays 5 products:

#	products	_id ObjectId	ProductID Int32	ProductName String	Category String	Price Int32	Stock :
1		ObjectId('67d8e9b87cd412...')	1	"Laptop"	"Electronics"	800	15
2		ObjectId('67d8e9b87cd412...')	2	"Phone"	"Electronics"	600	30
3		ObjectId('67d8e9b87cd412...')	5	"TV"	"Electronics"	400	20
4		ObjectId('67d8e9b87cd412...')	6	"Headphones"	"Electronics"	90	60
5		ObjectId('67d8e9b87cd412...')	9	"Tablet"	"Electronics"	300	10

## 5) Display Products in Ascending Order of Names

The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: 'Inventory' database with 'products' collection. The main panel shows a query filter: `{ }`. The 'Sort' field is set to `"ProductName": 1`. The results table displays 10 products sorted by name in ascending order:

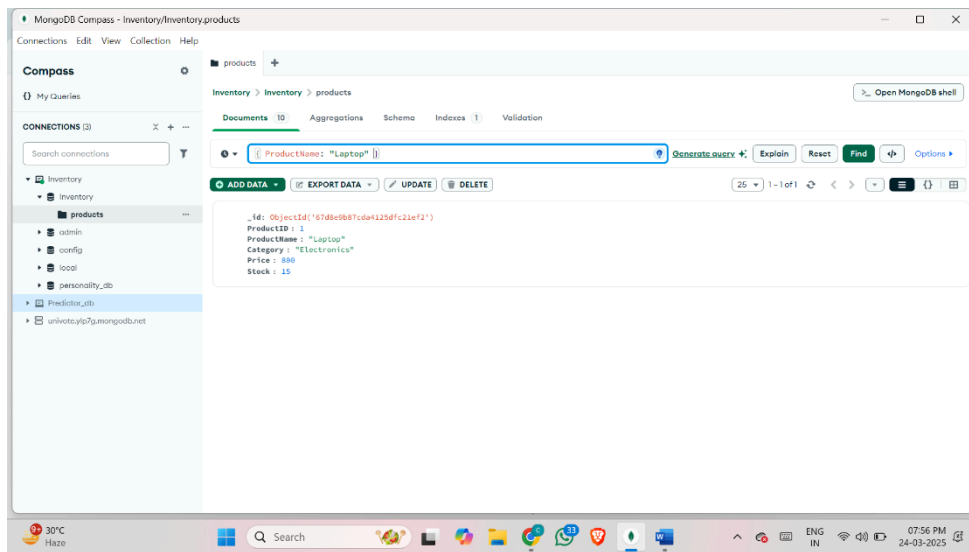
#	products	_id ObjectId	ProductID Int32	ProductName String	Category String	Price Int32	Stock :
1		ObjectId('67d8e9b87cd412...')	7	"Book"	"Stationery"	15	40
2		ObjectId('67d8e9b87cd412...')	4	"Chair"	"Furniture"	60	50
3		ObjectId('67d8e9b87cd412...')	6	"Headphones"	"Electronics"	90	60
4		ObjectId('67d8e9b87cd412...')	1	"Laptop"	"Electronics"	800	15
5		ObjectId('67d8e9b87cd412...')	8	"Pen"	"Stationery"	2	100
6		ObjectId('67d8e9b87cd412...')	2	"Phone"	"Electronics"	600	30
7		ObjectId('67d8e9b87cd412...')	10	"Sofa"	"Furniture"	700	5
8		ObjectId('67d8e9b87cd412...')	5	"TV"	"Electronics"	400	20
9		ObjectId('67d8e9b87cd412...')	3	"Table"	"Furniture"	120	12
10		ObjectId('67d8e9b87cd412...')	9	"Tablet"	"Electronics"	300	10

## 6) Display First 5 Products

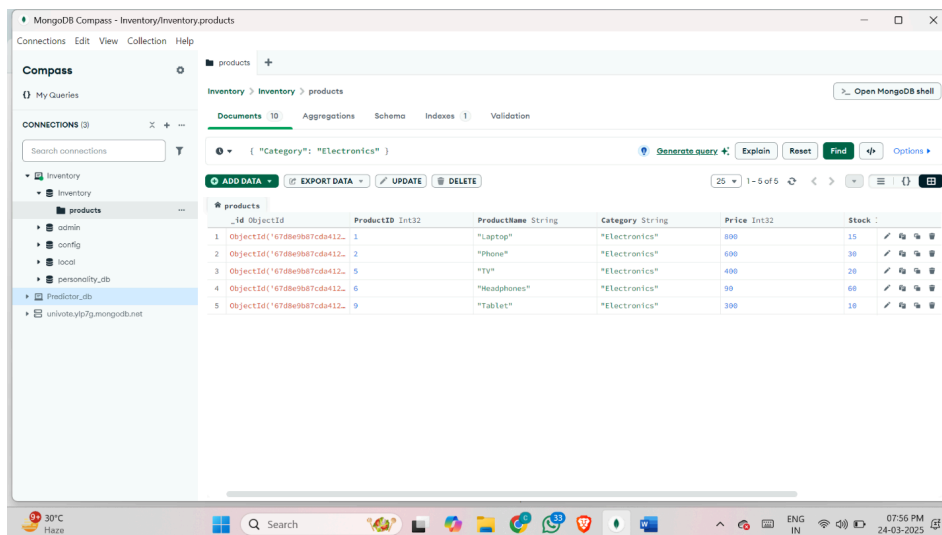
The screenshot shows the MongoDB Compass interface. The left sidebar displays the database structure: 'Inventory' database with 'products' collection. The main panel shows a query filter: `{ }`. The 'Limit' field is set to 5. The results table displays the first 5 products:

#	products	_id ObjectId	ProductID Int32	ProductName String	Category String	Price Int32	Stock :
1		ObjectId('67d8e9b87cd412...')	1	"Laptop"	"Electronics"	800	15
2		ObjectId('67d8e9b87cd412...')	2	"Phone"	"Electronics"	600	30
3		ObjectId('67d8e9b87cd412...')	3	"Table"	"Furniture"	120	12
4		ObjectId('67d8e9b87cd412...')	4	"Chair"	"Furniture"	60	50
5		ObjectId('67d8e9b87cd412...')	5	"TV"	"Electronics"	400	20

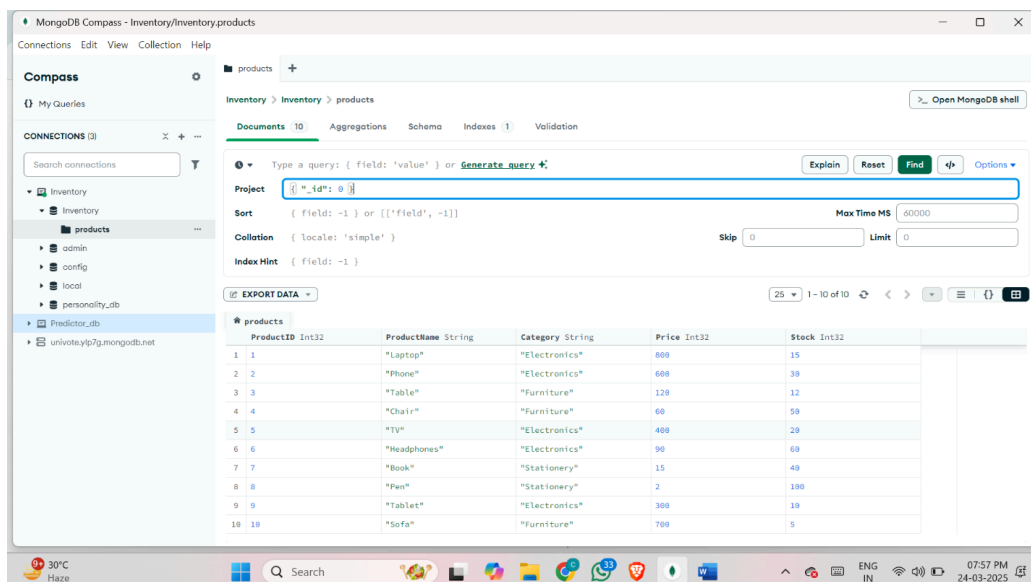
## 7) Display Products with a Specific Name



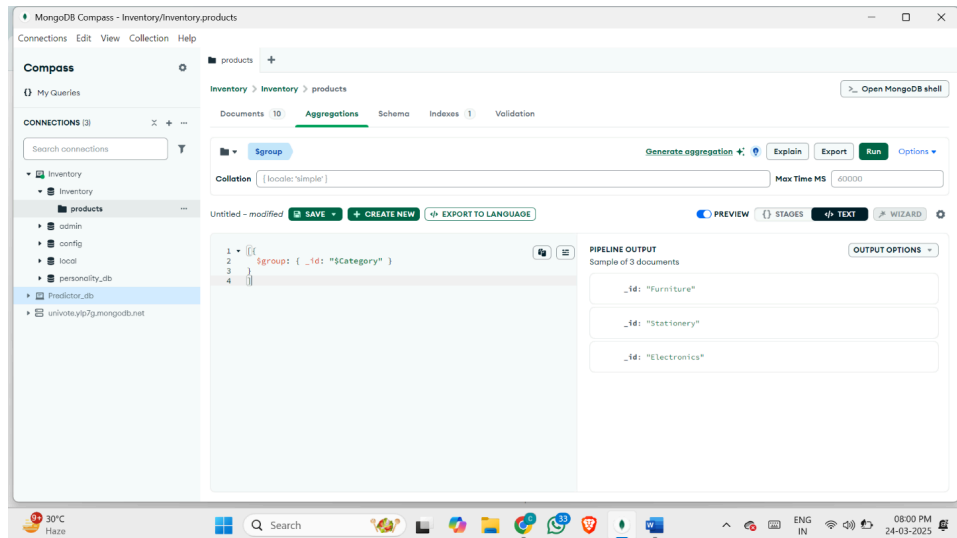
## 8) Count Products in Electronics Category



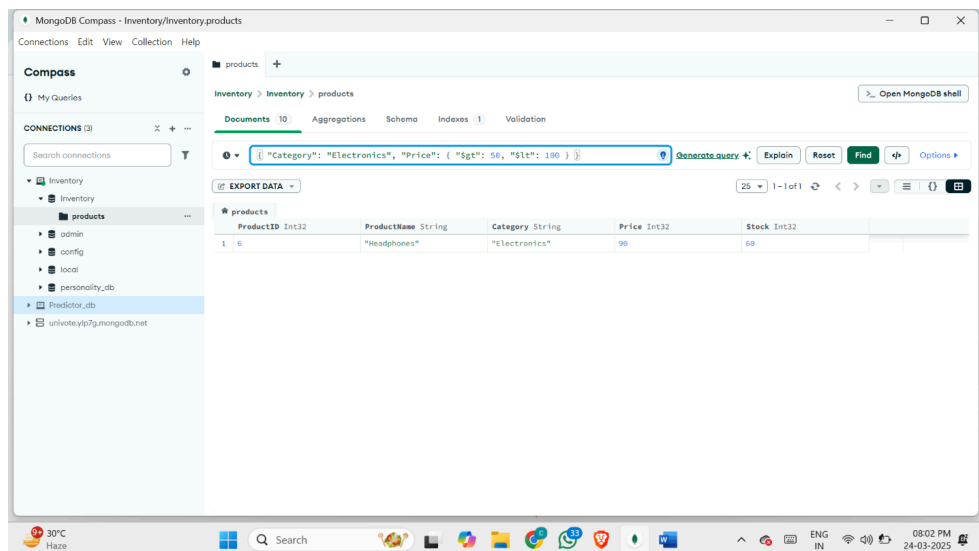
## 9) Hide the “\_id” Field



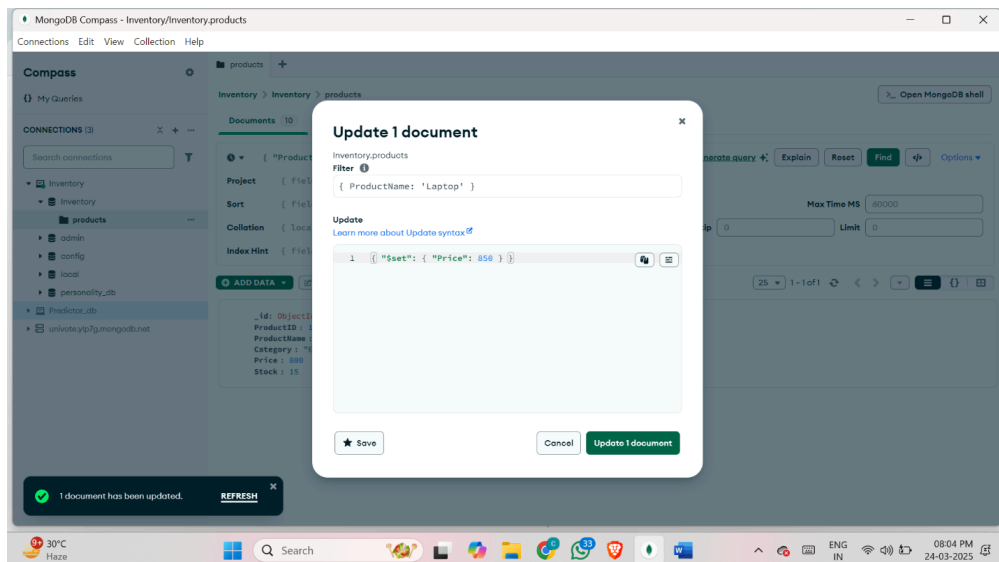
## 10) Display Distinct Categories



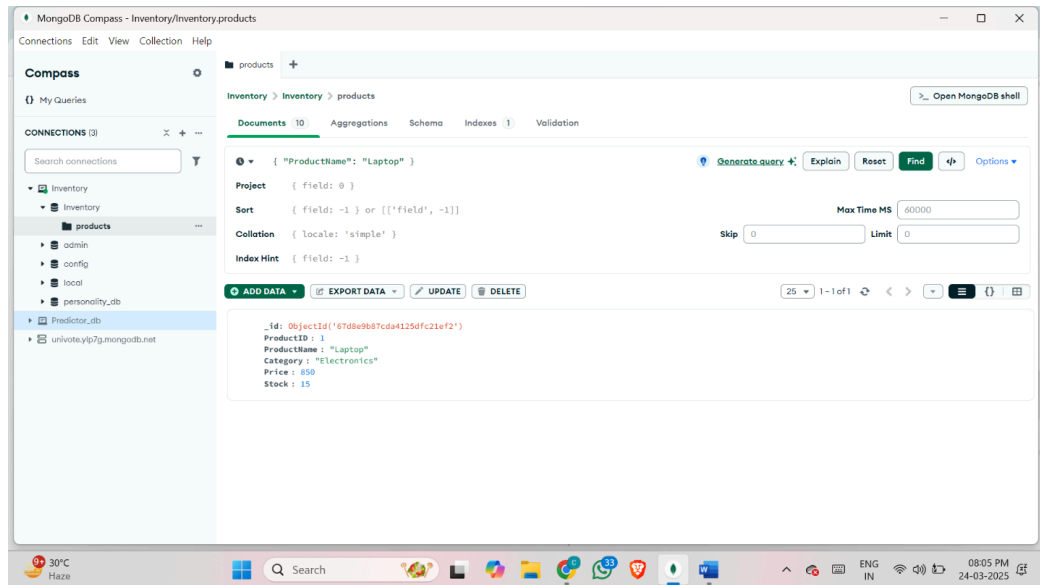
## 11) Display Products in Electronics Category with Price > 50 and < 100



## 12) Change the Price of a Product







### 13) Delete a Product

