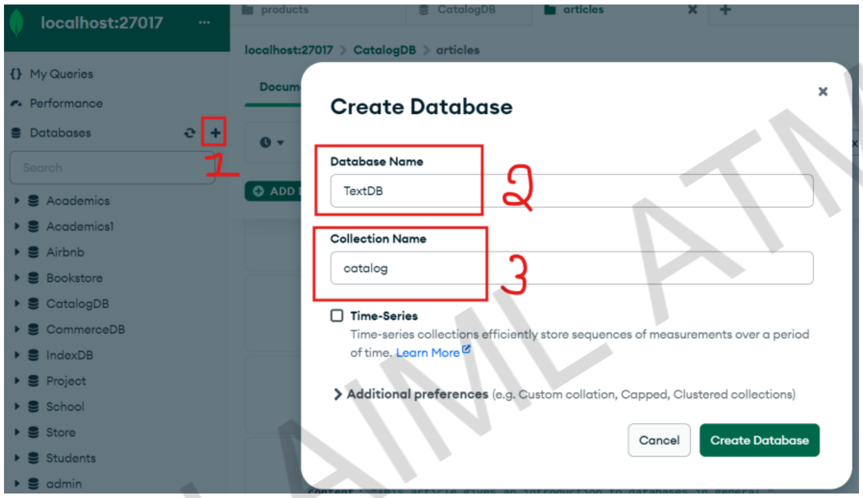
Program 10

Develop an aggregation pipeline to illustrate Text search on Catalog data collection.

Create a database **TextDB** and collection **catalog** in Mongo IDE.



Create Database

Database Name: TextDB

Collection Name: catalog

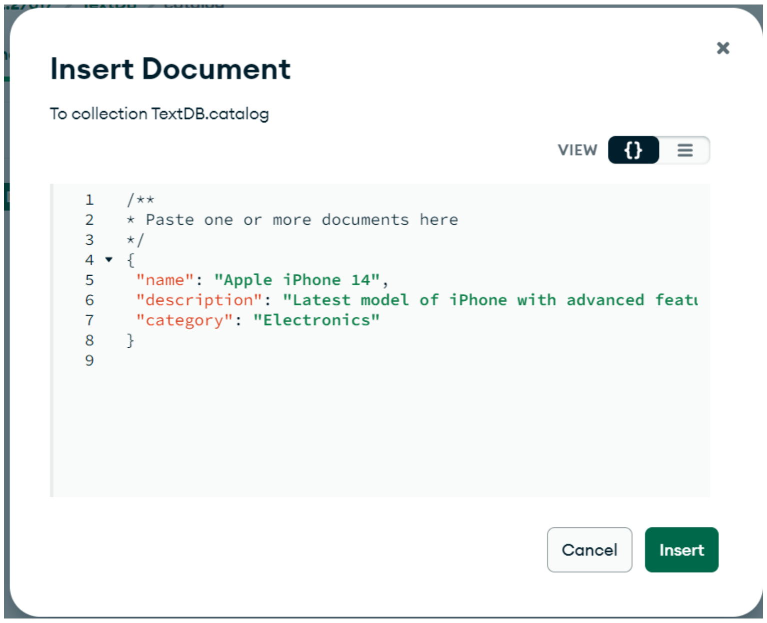
Time-Series
Time-series collections efficiently store sequences of measurements over a period of time. Learn More

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

Cancel | Create Database

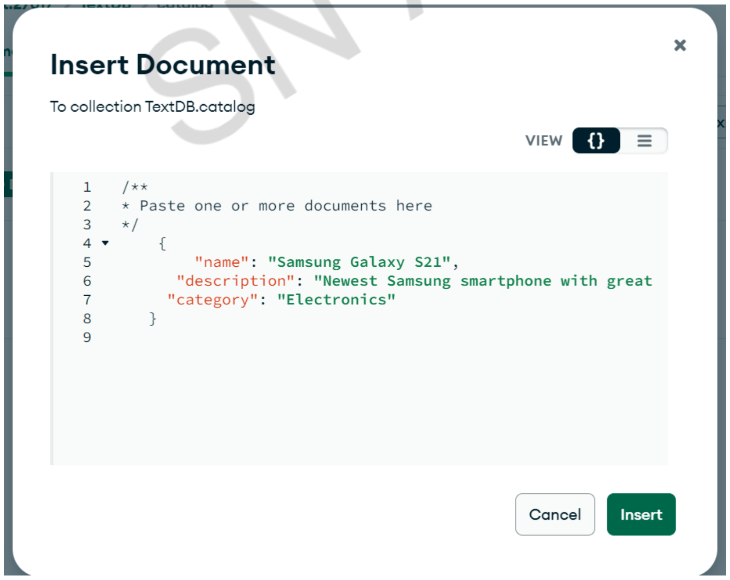
Add the following documents in the **catalog collection** in MongoDB Shell.

```
{
  "name": "Apple iPhone 14",
  "description": "Latest model of iPhone with advanced features",
  "category": "Electronics"
}
```



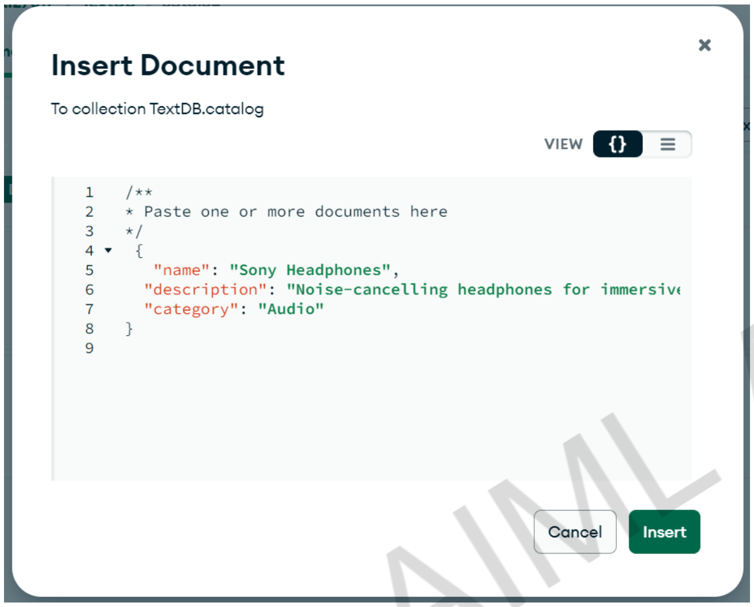
```
/**
* Paste one or more documents here
*/
{
  "name": "Apple iPhone 14",
  "description": "Latest model of iPhone with advanced featu
  "category": "Electronics"
}
```

```
{
    "name": "Samsung Galaxy S21",
   "description": "Newest Samsung smartphone with great camera",
  "category": "Electronics"
}
```



```
/**
* Paste one or more documents here
*/
    {
        "name": "Samsung Galaxy S21",
      "description": "Newest Samsung smartphone with great
     "category": "Electronics"
    }
```

```
{
  "name": "Sony Headphones",
  "description": "Noise-cancelling headphones for immersive sound",
  "category": "Audio"
}
```


Insert Document

To collection TextDB.catalog

```
/**
 * Paste one or more documents here
 */
  {
    "name": "Sony Headphones",
   "description": "Noise-cancelling headphones for immersive
   "category": "Audio"
 }
```

In MongoShell

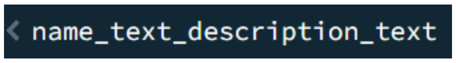
```
>use TextDB
```

Create a Text Index

First, create a text index on the ‘`name`’ and ‘`description`’ fields.

```
db.catalog.createIndex({ name: "text", description: "text" });
```

Output:


```
name_text_description_text
```

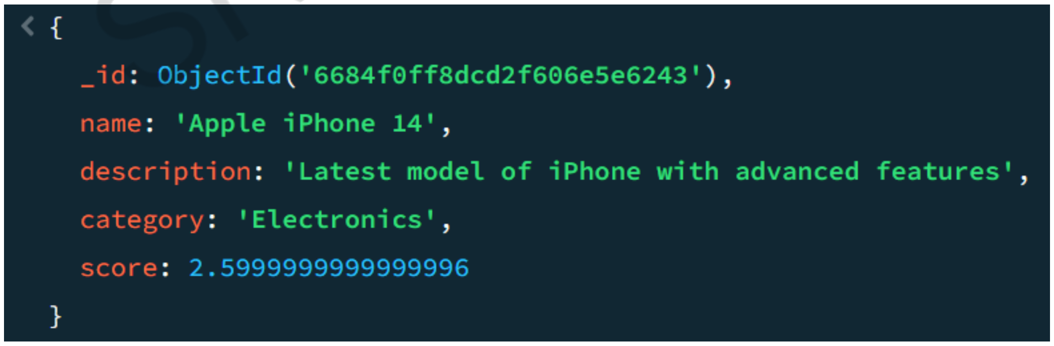
Define the Aggregation Pipeline

Now, create an aggregation pipeline to perform the text search and process the results. Below is an example pipeline:

```
db.catalog.aggregate([
 // Stage 1: Match documents containing the search term
 {
  $match: {
   $text: { $search: " Apple iPhone 14" }
  }
 },
 // Stage 2: Project only required fields
 {
  $project: {
   _id: 0,
   name: 1,
   description: 1,
   category: 1

   // Add more fields as needed
  }
 }
])
```

Output:



```
< {
    _id: ObjectId('6684f0ff8dcd2f606e5e6243'),
    name: 'Apple iPhone 14',
    description: 'Latest model of iPhone with advanced features',
    category: 'Electronics',
    score: 2.5999999999999996
  }
```