Name of Student	**Jai Navani**
Class Roll No	**D15A 30**
D.O.P.	**20/03/25**
D.O.S.	**27/03/25**
Sign and Grade	

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?

MongoDB is a **NoSQL database** designed for scalability, flexibility, and high performance. Key features include:

a. Document-Oriented Storage

- Stores data in **JSON-like documents** (BSON format).
- Documents can have nested structures, making it flexible for complex data.

b. Schema Flexibility

- No fixed schema: Each document in a collection can have a different structure.
- Easy to evolve the data model over time.

c. Scalability

- **Horizontal Scaling**: Supports **sharding** to distribute data across multiple servers.
- **Vertical Scaling**: Can handle large datasets and high traffic.

d. High Performance

- Indexing for fast querying.
- In-memory storage for frequently accessed data.

e. Replication

- Provides **high availability** through replica sets (multiple copies of data on different servers).

f. Aggregation Framework

- Powerful tools for data analysis and transformation (e.g., `$group`, `$match`, `$sort`).

g. Geospatial Support

- Built-in support for geospatial queries (e.g., finding locations within a radius).

h. Ad Hoc Queries

- Supports dynamic queries on documents using a rich query language.

i. Multi-Document ACID Transactions

- Ensures data consistency across multiple operations.

j. Cloud Integration

- Works seamlessly with MongoDB Atlas (fully managed cloud database service).

B. What are Documents and Collections in MongoDB?

a. Documents

A document is a basic unit of data in MongoDB, stored in BSON (Binary JSON) format.

Example:

```
{
  "_id": ObjectId("64b8f1a2e4b0a1a2b3c4d5e6"),
```

```
  "name": "John Doe",
  "age": 30,
  "address": {
    "city": "New York",
    "state": "NY"
  }
}
```

Documents are self-describing: Fields and values are stored together.

b. Collections

A collection is a group of documents.

Similar to a table in relational databases, but without a fixed schema.

Example: A users collection can store multiple user documents.

C. When to use MongoDB?

MongoDB is ideal for:

a. Unstructured or Semi-Structured Data

- When the data schema is not fixed or evolves over time.

b. High Write and Read Throughput

- Applications requiring fast read/write operations (e.g., real-time analytics).

c. Scalable Applications

- Applications that need to scale horizontally (e.g., social media platforms).

d. Hierarchical Data

- When data has nested structures (e.g., product catalogs, user profiles).

e. Real-Time Applications

- Use cases like IoT, gaming, and live feeds.

f. Prototyping and Agile Development

- Flexible schema allows quick iterations.

D. What is Sharding in MongoDB?

Sharding is a method for horizontal scaling in MongoDB. It distributes data across multiple servers (called shards) to handle large datasets and high traffic.

How Sharding Works

Shard Key: A field in the document is chosen as the shard key (e.g., user_id).

Shards: Data is split into chunks based on the shard key and distributed across shards.

Query Routing: A mongos (router) directs queries to the appropriate shard.

Benefits of Sharding

Scalability: Distributes load across multiple servers.

Performance: Improves read/write throughput.

Storage: Handles large datasets that exceed a single server’s capacity.

4) **Output:**

```
> db.products.insertMany([
    { ProductID: 1, ProductName: "Laptop", Category: "Electronics", Price: 999.99, Stock: 10 },
    { ProductID: 2, ProductName: "Smartphone", Category: "Electronics", Price: 699.99, Stock: 15 },
    { ProductID: 3, ProductName: "Headphones", Category: "Electronics", Price: 89.99, Stock: 30 },
    { ProductID: 4, ProductName: "T-Shirt", Category: "Clothing", Price: 19.99, Stock: 50 },
    { ProductID: 5, ProductName: "Jeans", Category: "Clothing", Price: 49.99, Stock: 40 },
    { ProductID: 6, ProductName: "Keyboard", Category: "Electronics", Price: 59.99, Stock: 25 },
    { ProductID: 7, ProductName: "Monitor", Category: "Electronics", Price: 199.99, Stock: 20 },
    { ProductID: 8, ProductName: "Mouse", Category: "Electronics", Price: 29.99, Stock: 35 },
    { ProductID: 9, ProductName: "Sneakers", Category: "Footwear", Price: 89.99, Stock: 30 },
    { ProductID: 10, ProductName: "Tablet", Category: "Electronics", Price: 299.99, Stock: 18 }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('67db928350f0aee17851d428'),
      '1': ObjectId('67db928350f0aee17851d429'),
      '2': ObjectId('67db928350f0aee17851d42a'),
      '3': ObjectId('67db928350f0aee17851d42b'),
      '4': ObjectId('67db928350f0aee17851d42c'),
      '5': ObjectId('67db928350f0aee17851d42d'),
      '6': ObjectId('67db928350f0aee17851d42e'),
      '7': ObjectId('67db928350f0aee17851d42f'),
      '8': ObjectId('67db928350f0aee17851d430'),
      '9': ObjectId('67db928350f0aee17851d431')
    }
  }
```

```
> db.products.find()
< {
    _id: ObjectId('67db928350f0aee17851d428'),
    ProductID: 1,
    ProductName: 'Laptop',
    Category: 'Electronics',
    Price: 999.99,
    Stock: 10
  }
  {
    _id: ObjectId('67db928350f0aee17851d429'),
    ProductID: 2,
    ProductName: 'Smartphone',
    Category: 'Electronics',
    Price: 699.99,
    Stock: 15
  }
```

```
{
  _id: ObjectId('67db928350f0aee17851d42a'),
  ProductID: 3,
  ProductName: 'Headphones',
  Category: 'Electronics',
  Price: 89.99,
  Stock: 30
}
{
  _id: ObjectId('67db928350f0aee17851d42b'),
  ProductID: 4,
  ProductName: 'T-Shirt',
  Category: 'Clothing',
  Price: 19.99,
  Stock: 50
}
{
  _id: ObjectId('67db928350f0aee17851d42c'),
  ProductID: 5,
  ProductName: 'Jeans',
  Category: 'Clothing',
  Price: 49.99,
  Stock: 40
}
```

```
{
  _id: ObjectId('67db928350f0aee17851d42d'),
  ProductID: 6,
  ProductName: 'Keyboard',
  Category: 'Electronics',
  Price: 59.99,
  Stock: 25
}
{
  _id: ObjectId('67db928350f0aee17851d42e'),
  ProductID: 7,
  ProductName: 'Monitor',
  Category: 'Electronics',
  Price: 199.99,
  Stock: 20
}
{
  _id: ObjectId('67db928350f0aee17851d42f'),
  ProductID: 8,
  ProductName: 'Mouse',
  Category: 'Electronics',
  Price: 29.99,
  Stock: 35
}
```

```
> db.products.find({ Category: "Electronics" })
< {
    _id: ObjectId('67db928350f0aee17851d428'),
    ProductID: 1,
    ProductName: 'Laptop',
    Category: 'Electronics',
    Price: 999.99,
    Stock: 10
  }
  {
    _id: ObjectId('67db928350f0aee17851d429'),
    ProductID: 2,
    ProductName: 'Smartphone',
    Category: 'Electronics',
    Price: 699.99,
    Stock: 15
  }
```

```
{
  _id: ObjectId('67db928350f0aee17851d42a'),
  ProductID: 3,
  ProductName: 'Headphones',
  Category: 'Electronics',
  Price: 89.99,
  Stock: 30
}
{
  _id: ObjectId('67db928350f0aee17851d42d'),
  ProductID: 6,
  ProductName: 'Keyboard',
  Category: 'Electronics',
  Price: 59.99,
  Stock: 25
}
{
  _id: ObjectId('67db928350f0aee17851d42e'),
  ProductID: 7,
  ProductName: 'Monitor',
  Category: 'Electronics',
  Price: 199.99,
  Stock: 20
}
```

```
> db.products.find().sort({ ProductName: 1 })
< {
    _id: ObjectId('67db928350f0aee17851d42a'),
    ProductID: 3,
    ProductName: 'Headphones',
    Category: 'Electronics',
    Price: 89.99,
    Stock: 30
  }
  {
    _id: ObjectId('67db928350f0aee17851d42c'),
    ProductID: 5,
    ProductName: 'Jeans',
    Category: 'Clothing',
    Price: 49.99,
    Stock: 40
  }
  {
    _id: ObjectId('67db928350f0aee17851d42d'),
    ProductID: 6,
    ProductName: 'Keyboard',
    Category: 'Electronics',
    Price: 59.99,
    Stock: 25
  }
```

```
{
  _id: ObjectId('67db928350f0aee17851d428'),
  ProductID: 1,
  ProductName: 'Laptop',
  Category: 'Electronics',
  Price: 999.99,
  Stock: 10
}
{
  _id: ObjectId('67db928350f0aee17851d42e'),
  ProductID: 7,
  ProductName: 'Monitor',
  Category: 'Electronics',
  Price: 199.99,
  Stock: 20
}
{
  _id: ObjectId('67db928350f0aee17851d42f'),
  ProductID: 8,
  ProductName: 'Mouse',
  Category: 'Electronics',
  Price: 29.99,
  Stock: 35
}
{
```

```
> db.products.find().limit(5)
< {
    _id: ObjectId('67db928350f0aee17851d428'),
    ProductID: 1,
    ProductName: 'Laptop',
    Category: 'Electronics',
    Price: 999.99,
    Stock: 10
  }
  {
    _id: ObjectId('67db928350f0aee17851d429'),
    ProductID: 2,
    ProductName: 'Smartphone',
    Category: 'Electronics',
    Price: 699.99,
    Stock: 15
  }
  {
    _id: ObjectId('67db928350f0aee17851d42a'),
    ProductID: 3,
    ProductName: 'Headphones',
    Category: 'Electronics',
    Price: 89.99,
    Stock: 30
  }
```

```
{
  _id: ObjectId('67db928350f0aee17851d42b'),
  ProductID: 4,
  ProductName: 'T-Shirt',
  Category: 'Clothing',
  Price: 19.99,
  Stock: 50
}
{
  _id: ObjectId('67db928350f0aee17851d42c'),
  ProductID: 5,
  ProductName: 'Jeans',
  Category: 'Clothing',
  Price: 49.99,
  Stock: 40
}
```

```
> db.products.find({ ProductName: "Laptop" }, { Category: 1, _id: 0 })
< {
    Category: 'Electronics'
  }
> db.products.countDocuments({ Category: "Electronics" })
< 7
> db.products.find({}, { _id: 0 })
< {
    ProductID: 1,
    ProductName: 'Laptop',
    Category: 'Electronics',
    Price: 999.99,
    Stock: 10
  }
  {
    ProductID: 2,
    ProductName: 'Smartphone',
    Category: 'Electronics',
    Price: 699.99,
    Stock: 15
  }
  {
    ProductID: 3,
    ProductName: 'Headphones',
    Category: 'Electronics',
    Price: 89.99,
    Stock: 30
```

```
{
  ProductID: 4,
  ProductName: 'T-Shirt',
  Category: 'Clothing',
  Price: 19.99,
  Stock: 50
}
{
  ProductID: 5,
  ProductName: 'Jeans',
  Category: 'Clothing',
  Price: 49.99,
  Stock: 40
}
{
  ProductID: 6,
  ProductName: 'Keyboard',
  Category: 'Electronics',
  Price: 59.99,
  Stock: 25
}
{
  ProductID: 7,
  ProductName: 'Monitor',
  Category: 'Electronics',
  Price: 199.99,
  Stock: 20
```


db.products.distinct("Category")
['Clothing', 'Electronics', 'Footwear']
db.products.find({
Category: "Electronics",
Price: { $gt: 50, $lt: 100 }
})
{
_id: ObjectId('67db928350f0aee17851d42a'),
ProductID: 3,
ProductName: 'Headphones',
Category: 'Electronics',
Price: 89.99,
Stock: 30
}
{
_id: ObjectId('67db928350f0aee17851d42d'),
ProductID: 6,
ProductName: 'Keyboard',
Category: 'Electronics',
Price: 59.99,
Stock: 25
}

```
> db.products.updateOne(
    { ProductName: "Mouse" },
    { $set: { Price: 35.00 } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.products.deleteOne({ ProductName: "Sneakers" })
< {
    acknowledged: true,
    deletedCount: 1
  }
test>
```