

## Experiment No: 08

**Title:** NoSQL Database-MongoDB

**Aim:** Installation of MongoDB. Explore MongoDB and its basics.

### Theory:

#### What is NoSQL?

NoSQL databases (AKA "not only SQL") store data differently than relational tables. NoSQL databases come in a variety of types based on their data model. The main types are [document](#), key-value, wide-column, and graph. They provide flexible schemas and scale easily with large amounts of big data and high user loads.

#### Types of NoSQL Database:

- Document-based databases
- Key-value stores
- Column-oriented databases
- Graph-based databases

Here's a brief overview of each type of NoSQL database:

##### 1. Document-based databases:

- Stores data as **JSON-like documents** (e.g., BSON in MongoDB).
- Each document can contain different structures (e.g., fields, nested data).
- Great for handling unstructured or semi-structured data.
- **Example:** MongoDB, CouchDB.

##### 2. Key-value stores:

- Stores data as **key-value pairs**, where a unique key is mapped to a specific value.
- Simple, fast, and efficient for retrieving values based on keys.
- **Example:** Redis, Amazon DynamoDB.

##### 3. Column-oriented databases:

- Data is stored in columns rather than rows, optimizing for **read/write operations** on large datasets.
- Ideal for use cases with vast amounts of data and frequent read/write operations.
- **Example:** Apache Cassandra, HBase.

#### **4. Graph-based databases:**

- Focus on **relationships between data** using nodes, edges, and properties.
- Ideal for use cases with highly interconnected data like social networks or recommendation engines.
- **Example:** Neo4j, ArangoDB.

#### **NoSQL Database-MongoDB**

MongoDB is a popular open-source, document-based NoSQL database designed to store and manage large volumes of unstructured or semi-structured data. Unlike traditional relational databases, MongoDB uses a flexible, schema-less structure, allowing it to handle varied data types and adapt easily to changes.

#### **Key Features of MongoDB:**

##### **1. Document-Oriented Storage:**

- Stores data in BSON (Binary JSON) format, allowing flexible schema design.
- Documents are grouped in collections, where each document can have a different structure.

##### **2. Flexible Schema:**

- MongoDB supports dynamic schemas, allowing documents in a collection to have different fields and structures.
- This makes it easy to evolve the database schema without downtime.

##### **3. Scalability:**

- Horizontal scaling using sharding allows MongoDB to distribute large datasets across multiple servers, ensuring efficient scaling.
- Ideal for applications needing to scale out across multiple servers.

##### **4. Indexing:**

- MongoDB provides efficient indexing on any field within a document to improve query performance.
- Supports various types of indexes, including compound, text, and geospatial indexes.

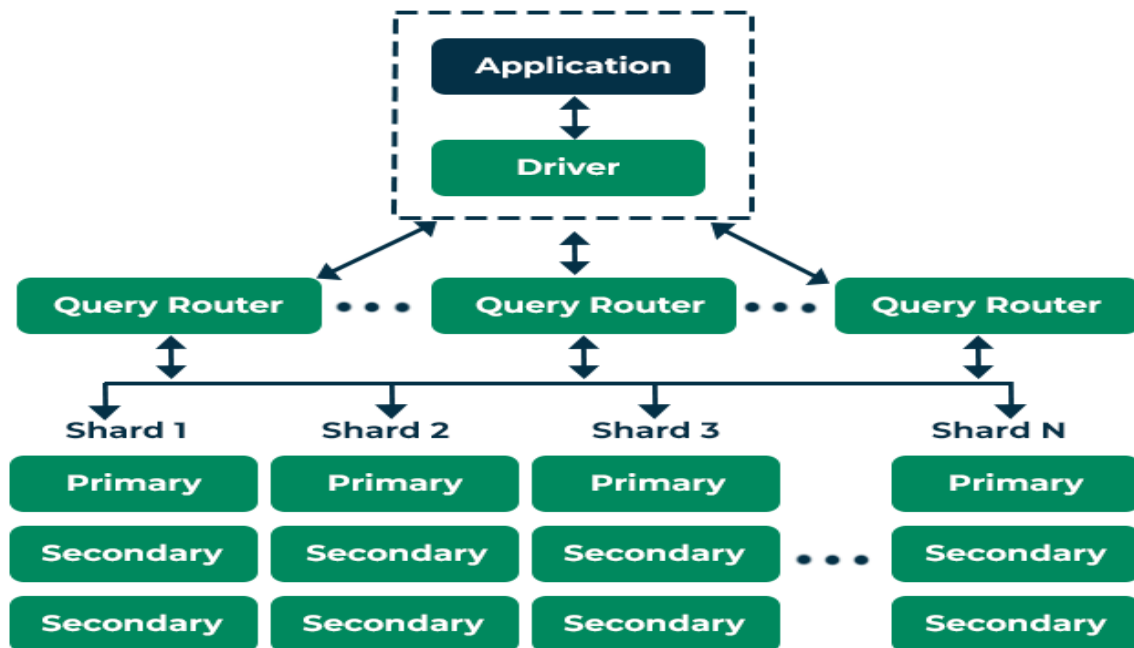
##### **5. Aggregation Framework:**

- A powerful feature that allows data processing and transformation through pipelines.
- It can be used for filtering, grouping, and summarizing data, similar to SQL queries with GROUP BY and SUM.

##### **6. Replication:**

- MongoDB ensures high availability with replica sets, where multiple copies of data are stored across different nodes.
- Automatic failover ensures data durability and fault tolerance.

### MongoDB Architecture: -



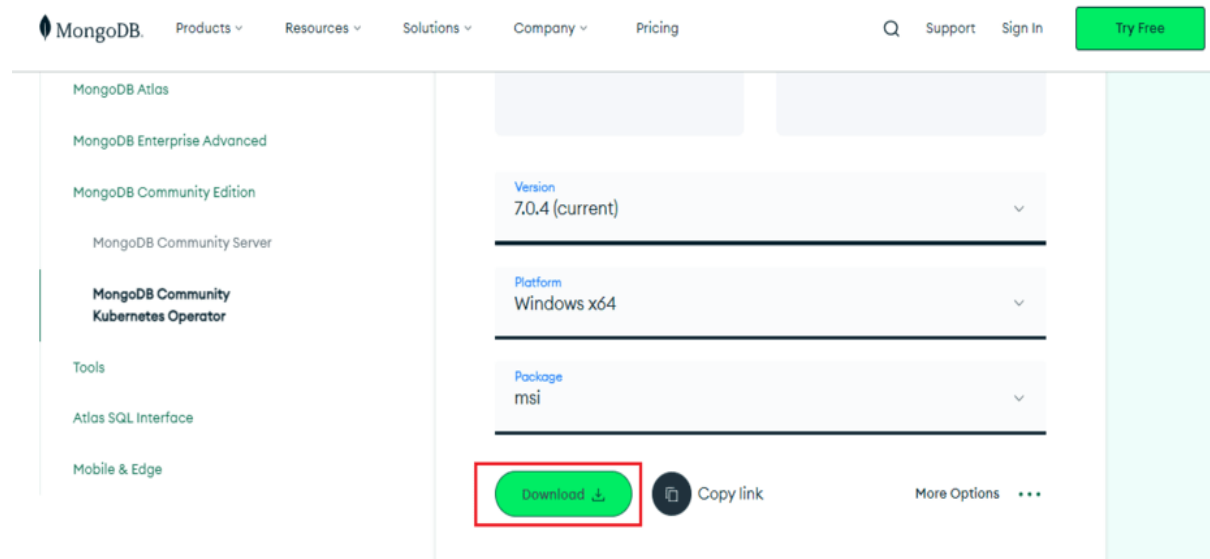
- 1. Drivers:** MongoDB provides drivers for various programming languages like Python, Java, .NET, and Node.js, which allow applications to communicate with the database.
- 2. Storage Engine:** The storage engine is responsible for managing how data is stored, retrieved, and updated. MongoDB supports multiple storage engines, including WiredTiger and In-Memory.
- 3. Sharding:** This is the process of distributing data across multiple servers to ensure horizontal scalability. Each shard holds a subset of the data.
- 4. Replication:** MongoDB uses replica sets to provide redundancy and high availability. A replica set consists of multiple copies of the data, with one primary node and multiple secondary nodes.
- 5. Query Engine:** The query engine processes and executes queries. It translates highlevel queries into low-level operations that interact with the storage engine.
- 6. Management Tools:** MongoDB provides various tools for monitoring, backup, and performance tuning

## Installation of MongoDB

### Steps to Install MongoDB on Windows using MSI

To install MongoDB on Windows, first, download the MongoDB server and then install the MongoDB shell. The Steps below explain the installation process in detail and provide the required resources for the smooth **download and install MongoDB**.

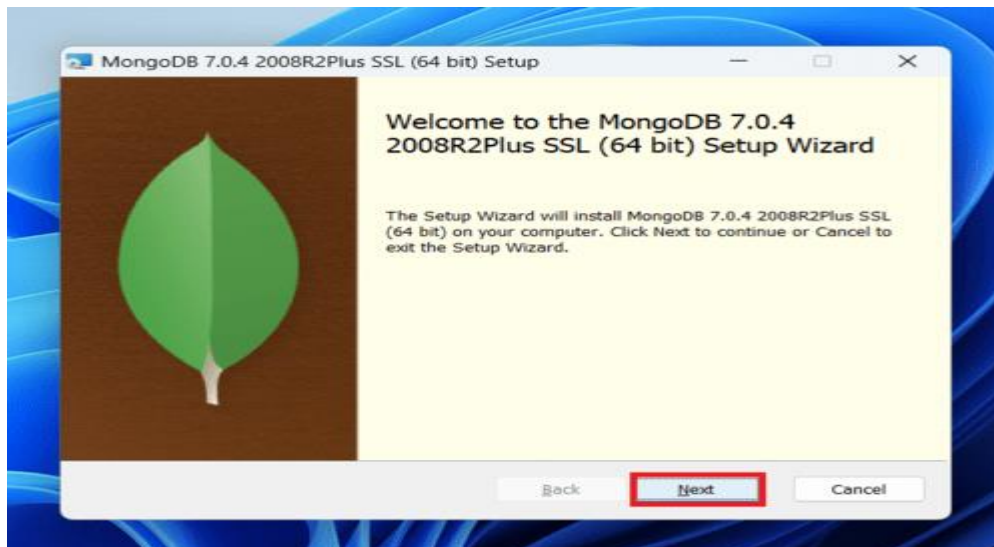
**Step 1:** Go to the [MongoDB Download Center](#) to download the MongoDB Community Server.



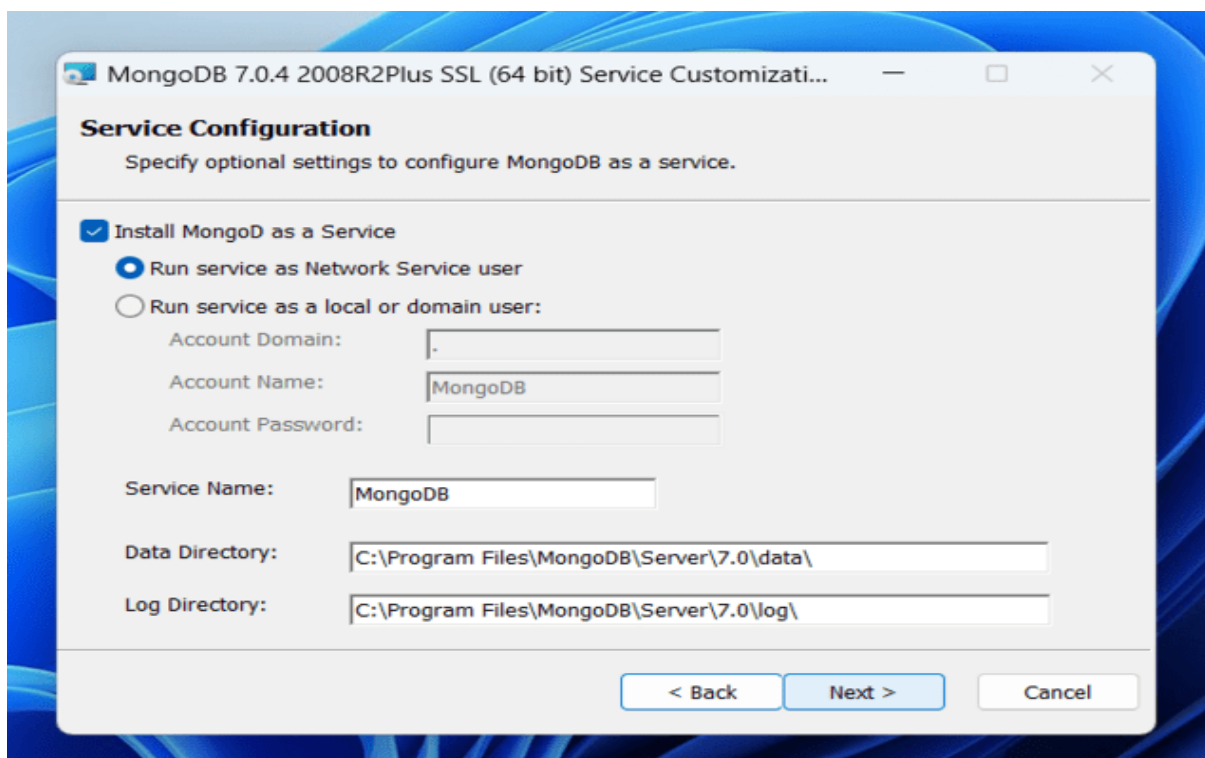
Here, You can select any version, Windows, and package according to your requirement. For Windows, we need to choose:

- **Version: 7.0.4**
- **OS: Windows x64**
- **Package: msi**

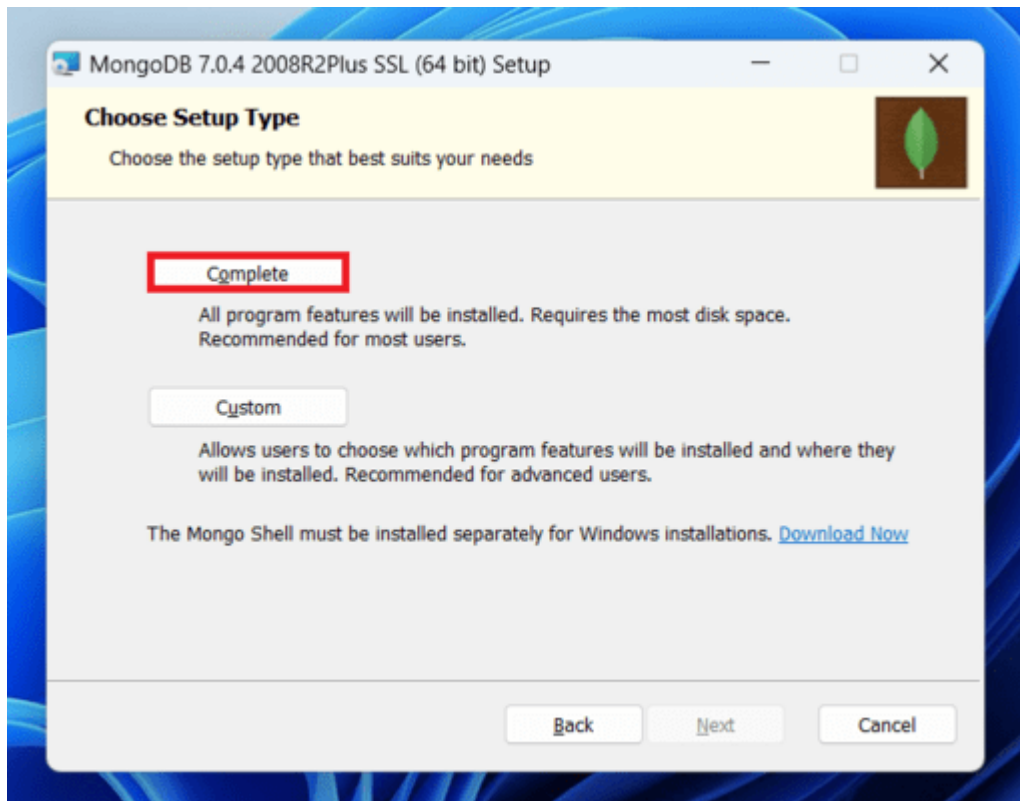
**Step 2:** When the download is complete open the msi file and click the *next button* in the startup screen:



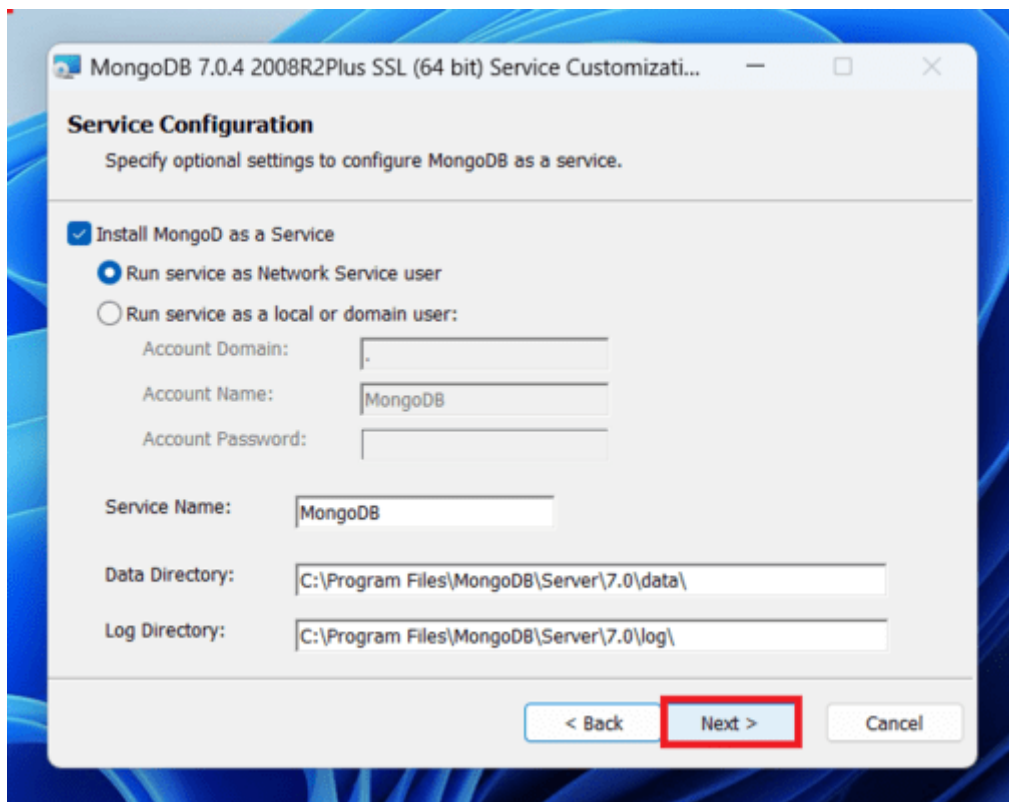
**Step 3:** Now accept the **End-User License Agreement** and click the next button:



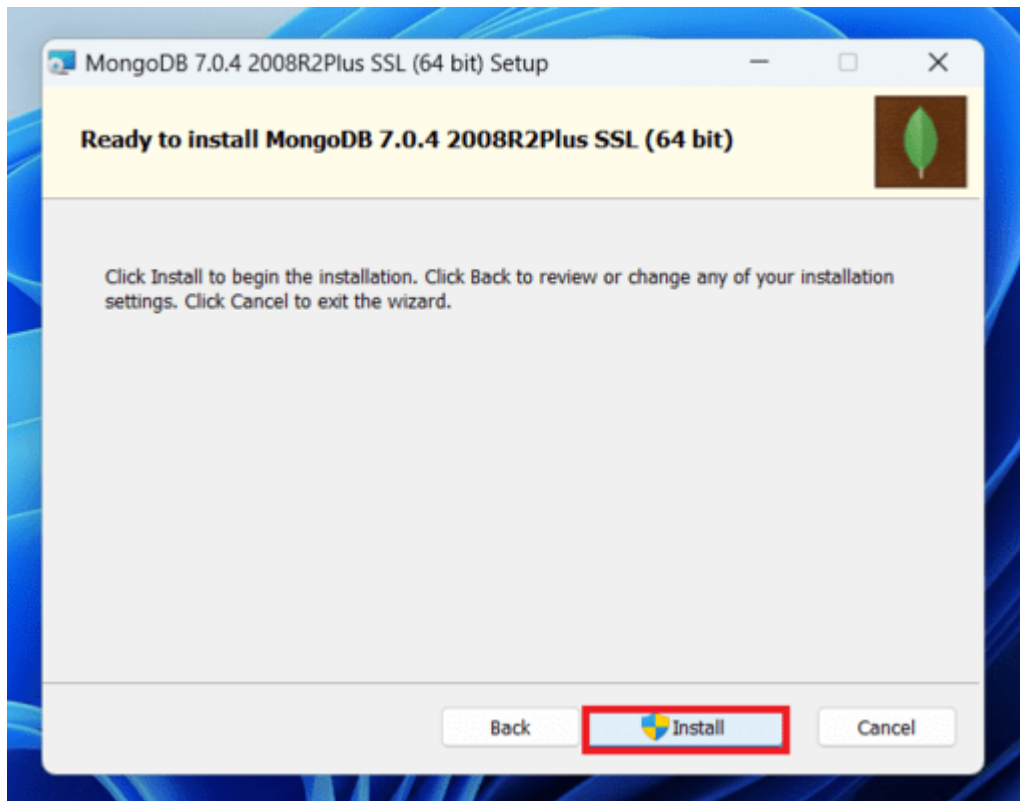
**Step 4:** Now select the **complete option** to install all the program features. Here, if you can want to install only selected program features and want to select the location of the installation, then use the **Custom option**:



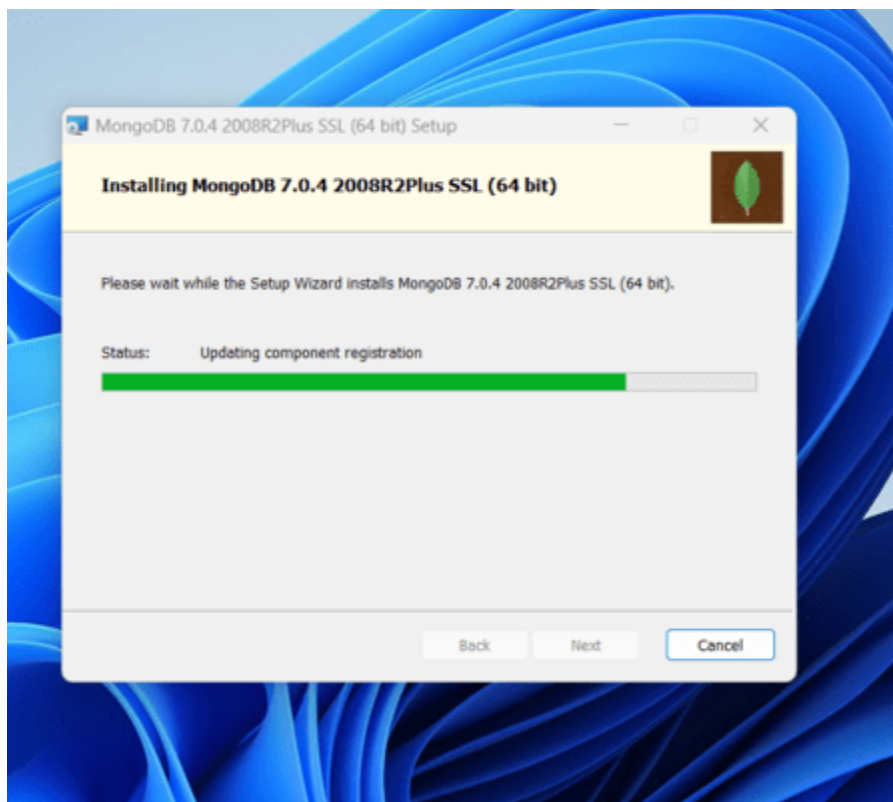
**Step 5:** Select “Run service as Network Service user” and copy the path of the data directory. Click Next:



**Step 6:** Click the **Install** button to start the MongoDB installation process:

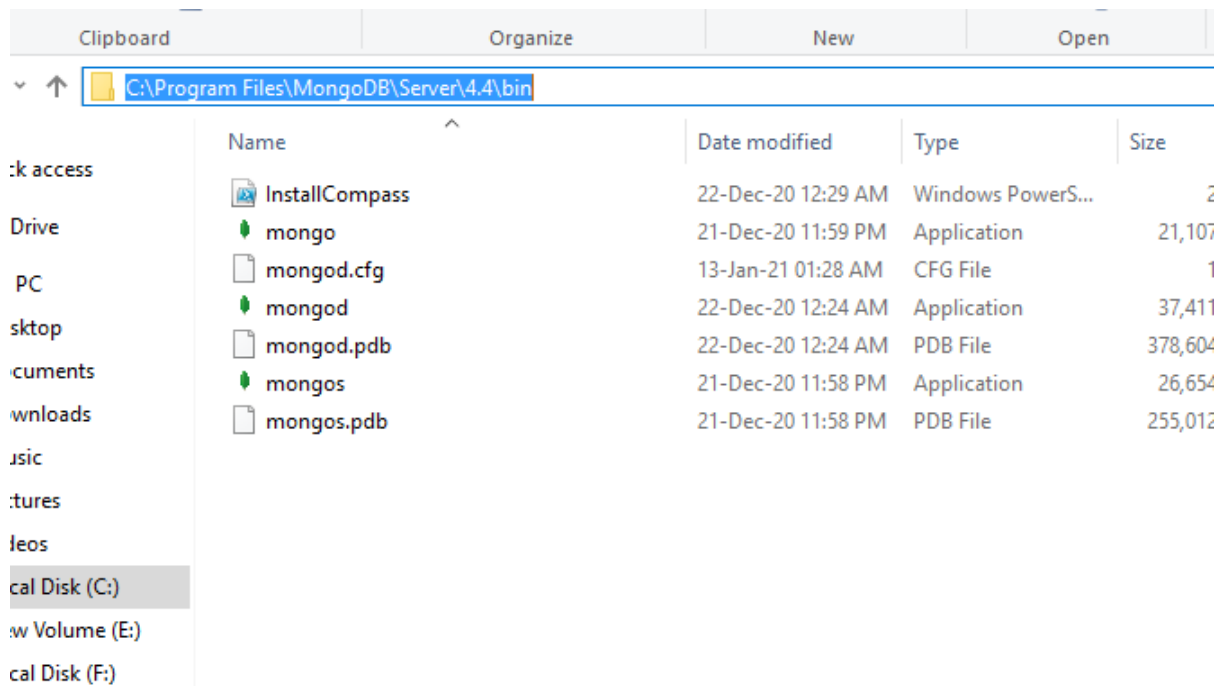


**Step 7:** After clicking on the install button installation of MongoDB begins:

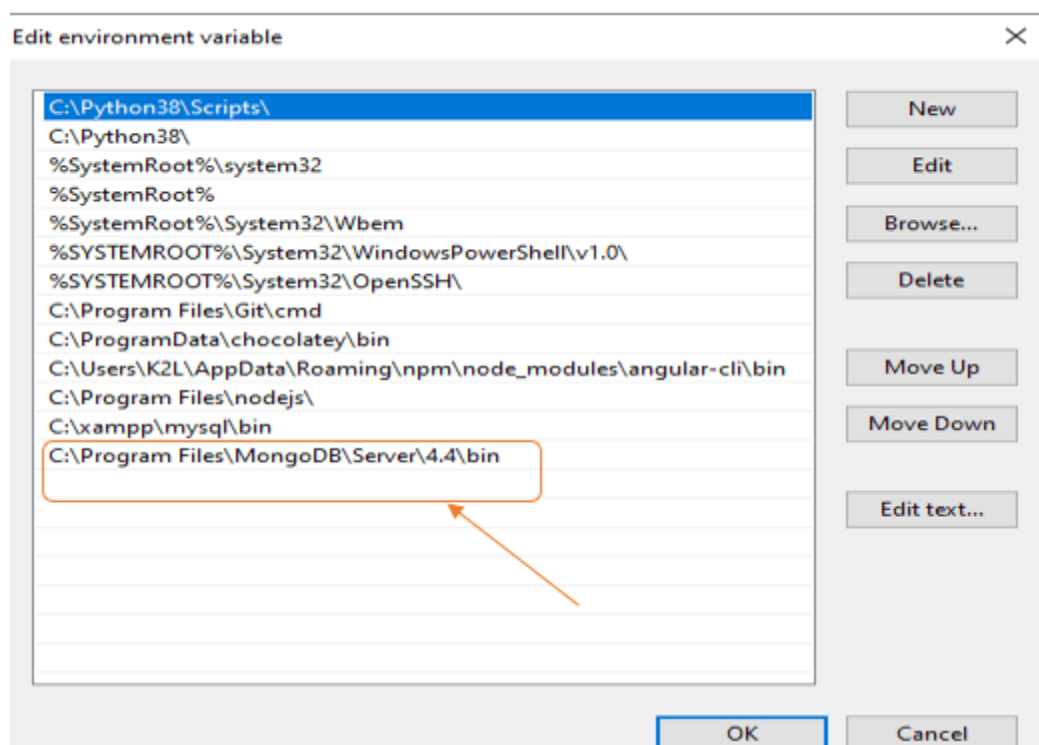


**Step 8:** Now click the **Finish** button to complete the MongoDB installation process:

**Step 9:** Now we go to the location where MongoDB installed in step 5 in your system and copy the **bin** path:



**Step 10:** Now, to create an environment variable open system **properties** >> **Environment Variable** >> **System variable** >> **path** >> **Edit Environment variable** and paste the copied link to your environment system and **click Ok**:



**Step 11:** After setting the environment variable, we will run the MongoDB server, i.e. **mongod**. So, open the **command prompt** and run the following command:

**mongod**

When you run this command you will get an error i.e. **C:/data/db/ not found**.



**Step 12:** Now, Open C drive and create a folder named “data” inside this folder create another folder named “db”. After creating these folders. Again open the command prompt and run the following command:

### **mongod**

Now, this time the MongoDB server(i.e., mongod) will run successfully.

```
C:\Users\Nikhil Chhipa>mongod
{"t":{"$date":"2021-01-31T00:56:54.081+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":
ify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2021-01-31T00:56:54.087+05:30"},"s":"W",  "c":"ASIO",    "id":22601,   "ctx":
}
{"t":{"$date":"2021-01-31T00:56:54.088+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I",  "c":"STORAGE",  "id":4615611, "ctx":
bPath":"C:/data/db/","architecture":"64-bit","host":"DESKTOP-L9MUQ7N"}}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":
rgetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":
gitVersion":"913d6b62acfb344dde1b116f4161360acd8fd13","modules":[],"allocator":"tcmalloc",
}}}}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":
ndows 10","version":"10.0 (build 14393)"}
{"t":{"$date":"2021-01-31T00:56:54.090+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":
{"t":{"$date":"2021-01-31T00:56:54.157+05:30"},"s":"I",  "c":"STORAGE",  "id":22270,   "ctx":
:{"dbpath":"C:/data/db/","storageEngine":"wiredTiger"}}
{"t":{"$date":"2021-01-31T00:56:54.158+05:30"},"s":"I",  "c":"STORAGE",  "id":22315,   "ctx":
ize=1491M,session_max=33000,eviction=(threads_min=4,threads_max=4),config_base=false,statist
le_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statisti
ess],"}
{"t":{"$date":"2021-01-31T00:56:54.395+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":
95788][3708:140713908197088], txn-recover: [WT_VERB_RECOVERY_PROGRESS] Recovering log 20 thr
{"t":{"$date":"2021-01-31T00:56:54.631+05:30"},"s":"I",  "c":"STORAGE",  "id":22430,   "ctx":
```

### **Run mongo Shell**

**Step 13:** Now we are going to connect our server (mongod) with the mongo shell. So, keep that mongod window and open a new command prompt window and write **mongo**. Now, our mongo shell will successfully connect to the mongod.

**Important Point:** Please do not close the mongod window if you close this window your server will stop working and it will not able to connect with the mongo shell.

```
MySQL 8.0 Command Line Cli  x  +  v
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.39 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| geeks4geeks |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql> |
```

Now, you are ready to write queries in the mongo Shell.