REPORT ON:
MONGO DB

REPORT BY: BHAGYA SREE G

MONGO DB

A GIANT DATABASE

PREFACE :

Mongo DB is an extremely versatile noSQl database that offers performance,scalability,and reability of data. It has become one of the leading noSQL database systems used for storing extremely large datasets.mongo DB is an open source document-oriented database system.

FEATURES :

➢Scalability

➢Full cloud-based developer data platform

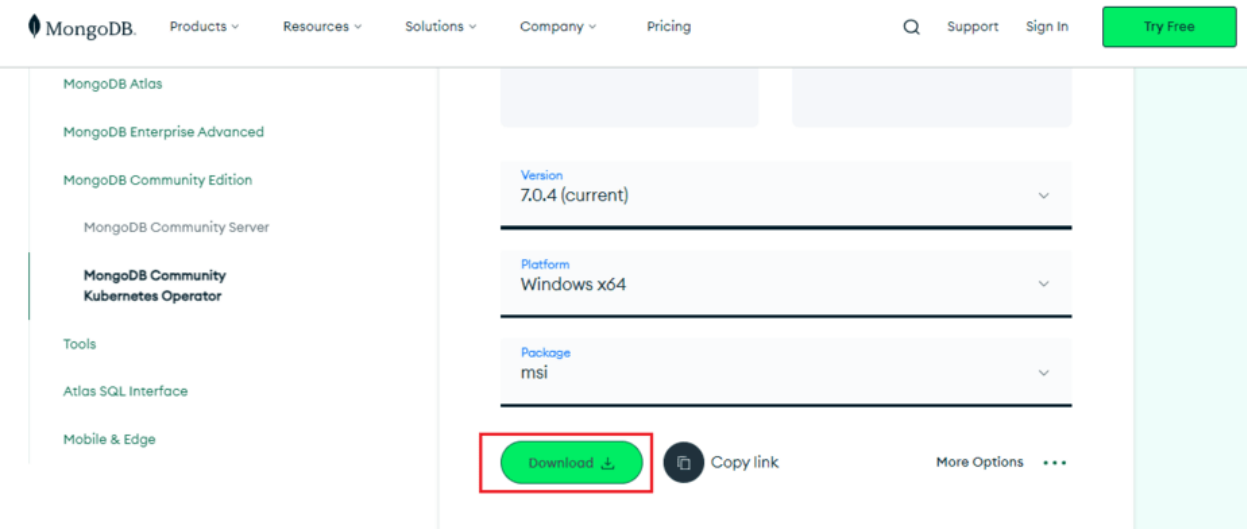
➢Full cloud-based developer data platform

➢Geo spatial indexing

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**.

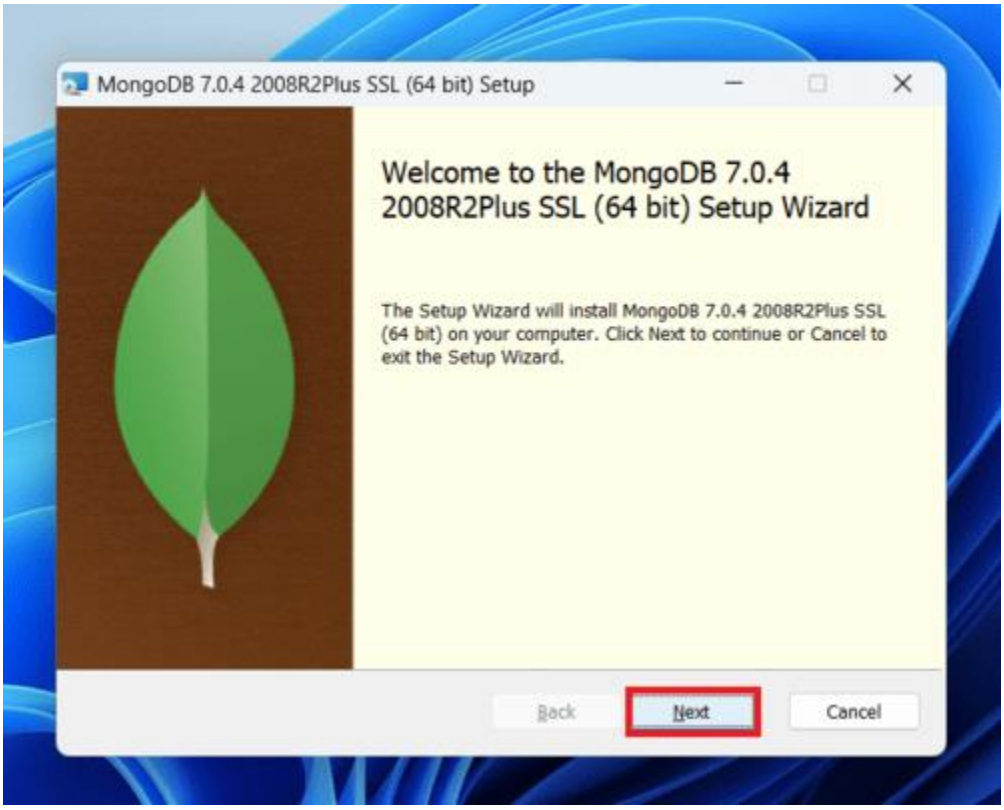
Step 1: Go to the [MongoDB Download Center](#) to download the MongoDB Community Server.



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

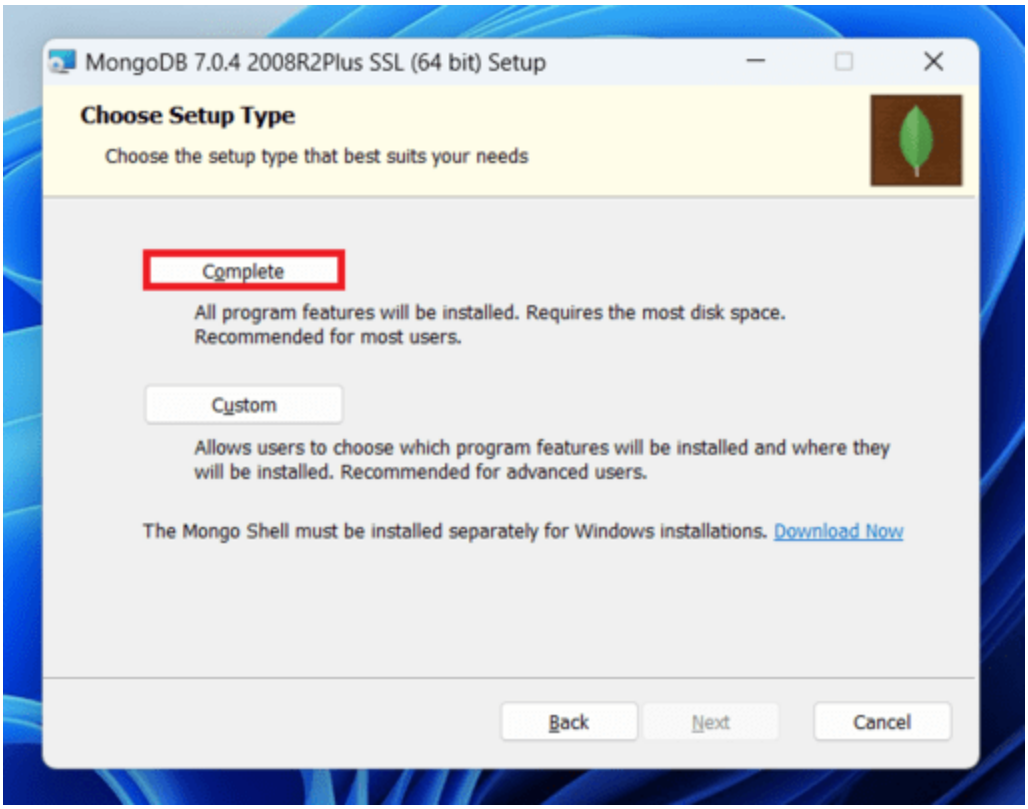
- Version: 7.0.4
- OS: Windows x64
- Package: msi

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



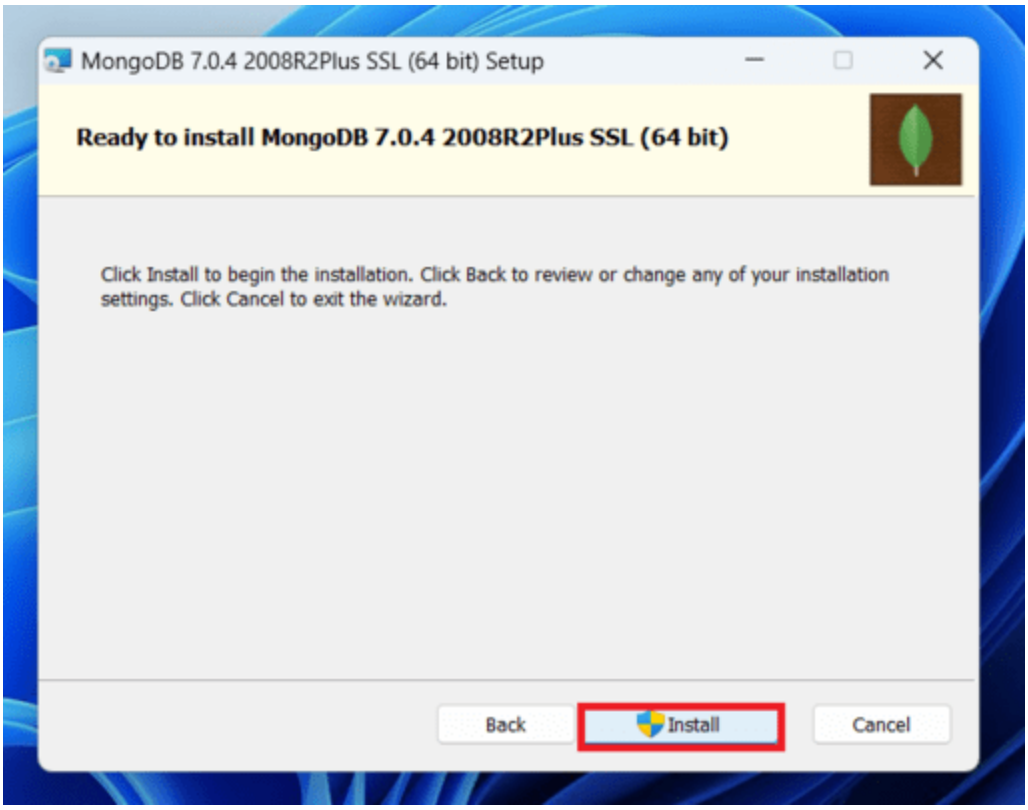
Step 3: Now accept the End-User License Agreement and click the next button:

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:

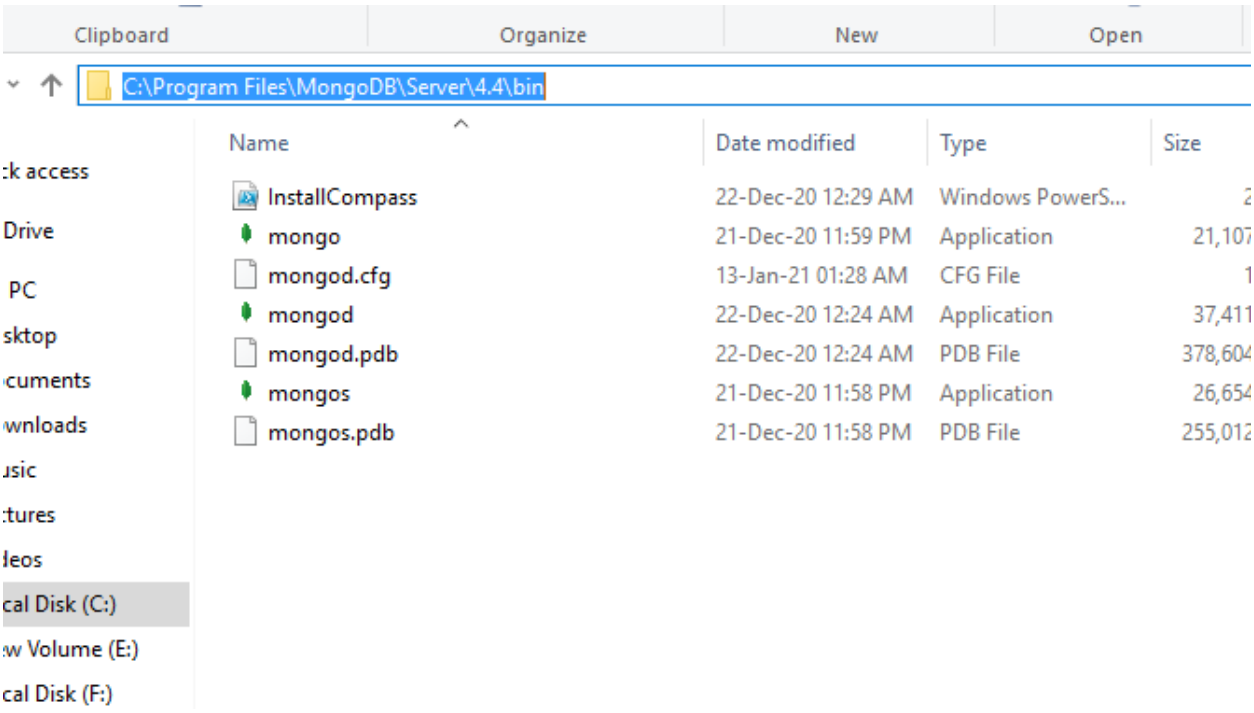
Step 6: Click the *Install button* to start the MongoDB installation process:



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.

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.

The `mongo` Shell

The `mongo` shell is an interactive JavaScript interface to MongoDB. You can use the `mongo` shell to query and update data as well as perform administrative operations.

NOTE

The following document pertains to the `mongo` shell included in the MongoDB Server Download. For information on the new MongoDB Shell, `mongosh`, refer to the mongosh Documentation.

To understand the differences between the two shells, see Comparison of the `mongo` Shell and `mongosh`.

Download the `mongo` Shell

The `mongo` shell is included as part of the MongoDB server installation. If you have already installed the server, the `mongo` shell is installed to the same location as the server binary.

Alternatively, if you would like to download the `mongo` shell separately from the MongoDB Server, you can install the shell as a standalone package by following these steps:

1. Access the Download Center for your Edition of MongoDB:
    - MongoDB Community Download Center
    - MongoDB Enterprise Download Center
2. Select your preferred **Version** and **Platform** from the dropdowns.
3. Select the **Package** to download according to your platform
4. Copy the `mongo` shell from the archive to a location on your filesystem.

For additional installation guidance specific to your platform, or to install the `mongo` shell as part of a MongoDB Server installation, see the installation guide for your platform.

LOADING DOCUMENT

- **Download the student csv from this link**
- **Import the data to the collection created link**
- **You should be able to see the uploaded data in mongo compass**

Few Commands to test after connections

Command	Notes
db.foo.batchInsert([{"_id" : 0}, {"_id" : 1}, {"_id" : 2}])	Insert more than one document
db.foo.find()	Print all rows
db.foo.remove()	Remove foo table

DOCUMENTS

At the heart of MongoDB is the document: an ordered set of keys with associated values.

The representation of a document varies by programming language, but most languages

have a data structure that is a natural fit, such as a map, hash, or dictionary. In JavaScript,

for example, documents are represented as objects:

```
{"greeting" : "Hello, world!"}
```

This simple document contains a single key, "greeting", with a value of "Hello,

world!". Most documents will be more complex than this simple one and often will

contain multiple key/value pairs:

```
{"greeting" : "Hello, world!", "foo" : 3}
```

In this example the value for "greeting" is a string,

whereas the value for "foo" is an integer.

NOTABLE EXCEPTIONS:

• Keys must not contain the character \0 (the null character). This character is used to signify the end of a key.

• The . and $ characters have some special properties and should be used only in certain circumstances.

MongoDB is type-sensitive and case-sensitive. For example, these documents are

distinct:

{"foo" : 3}

{"foo" : "3"}

as are as these:

{"foo" : 3}

{"Foo" : 3}

A final important thing to note is that documents in MongoDB cannot contain duplicate keys. For example, the following is not a legal document:

{"greeting" : "Hello, world!", "greeting" : "Hello, MongoDB!"}

Key/value pairs in documents are ordered: {"x" : 1, "y" : 2} is not the same as

{"y" : 2, "x" : 1}. Field order does not usually matter and you should not design.

COLLECTIONS

A collection is a group of documents. If a document is the MongoDB analog of a row in a relational database, then a collection can be thought of as the analog to a table.

Dynamic Schemas

Collections have dynamic schemas. This means that the documents within a single collection can have any number of different "shapes."

For example, both of the following documents could be stored in a single collection:

```
{"greeting" : "Hello, world!"}

{"foo" : 5}
```

NAMING

A collection is identified by its name.

• The empty string ("") is not a valid collection name.

• Collection names may not contain the character \0 (the null character) because this delineates the end of a collection name.

• You should not create any collections that start with system., a prefix reserved for internal collections. For example, the system users collection contains the database's users, and the system namespaces collection contains information about all of the database's collections.

• User-created collections should not contain the reserved character $ in the name. The various drivers available for the database do support using $ in collection names because some system-generated collections contain it.

DATA BASES

In addition to grouping documents by collection, MongoDB groups collections into databases. A single instance of MongoDB can host several databases, each grouping together zero or more collections. A database has its own permissions, and each database is stored in separate files on disk. A good rule of thumb is to store all data for a single application in the same database. Separate databases are useful when storing data for several application or users on the same MongoDB server.Like collections, databases are identified by name with the following restrictions:

• The empty string ("") is not a valid database name.

• A database name cannot contain any of these characters: /, \, ., ", *, <, >, :, |, ?, $, (a single space), or \0 (the null character). Basically, stick with alphanumeric ASCII.

• Database names are case-sensitive, even on non-case-sensitive filesystems. To keep things simple, try to just use lowercase characters.

• Database names are limited to a maximum of 64 bytes.

DATA TYPES

MongoDB supports a wide range of data types as values in documents.

Basic Data Types

NULL : Null can be used to represent both a null value and a nonexistent field.

```
{"x" : null}
```

BOOLEAN : There is a boolean type, which can be used for the values true and false.

```
{"x" : true}
```

NUMBER : The shell defaults to using 64-bit floating point numbers. Thus, these numbers look "normal" in the shell.

```
{"x" : 3.14}
```

For integers, use the Number Int or Number Long classes, which represent 4-byte or 8-byte signed integers, respectively.

```
{"x" : NumberInt("3")}
{"x" : NumberLong("3")}
```

STRING : Any string of UTF-8 characters can be represented using the string type.

```
{"x" : "foobar"}
```

DATE : Dates are stored as milliseconds since the epoch. The time zone is not stored.

```
{"x" : new Date()}
```

ARRAYS: Sets or lists of values can be represented as arrays:

```
{"x" : ["a", "b", "c"]}
```

DATA TYPE

Basically each document will be in JSON format which will be as follows.

Where each attributes inside can be of multiple data types.