

## EX.NO.10 INSTALLATION OF MONGODB AND CREATION OF COLLECTIONS

### Aim:

To install MongoDB and create a collection with it.

### Description:

MongoDB is an open-source document database and leading NoSQL database. MongoDB is written in C++. It is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON like documents with optional schemas.

### Procedure:

#### Installation:

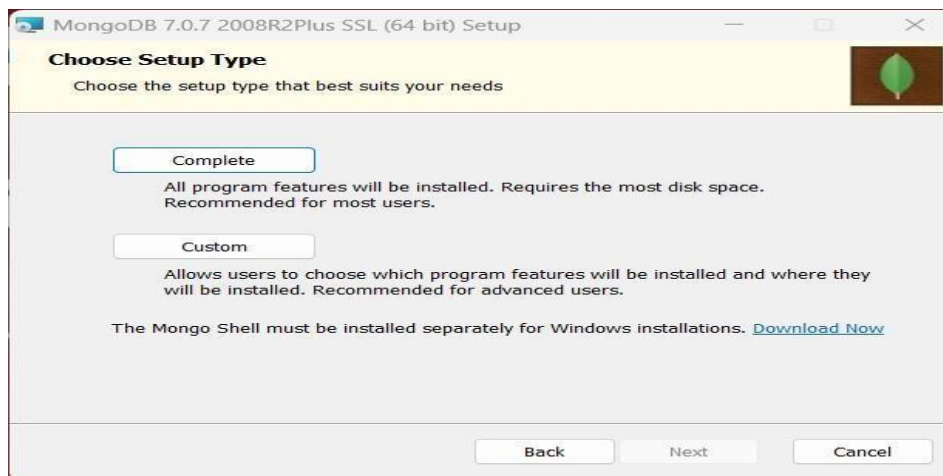
**STEP 1:** Go to <https://www.mongodb.com/download-center/community>. Select the necessary configuration and click on the Download button.

**STEP 2:** In the installation wizard, Click Next.

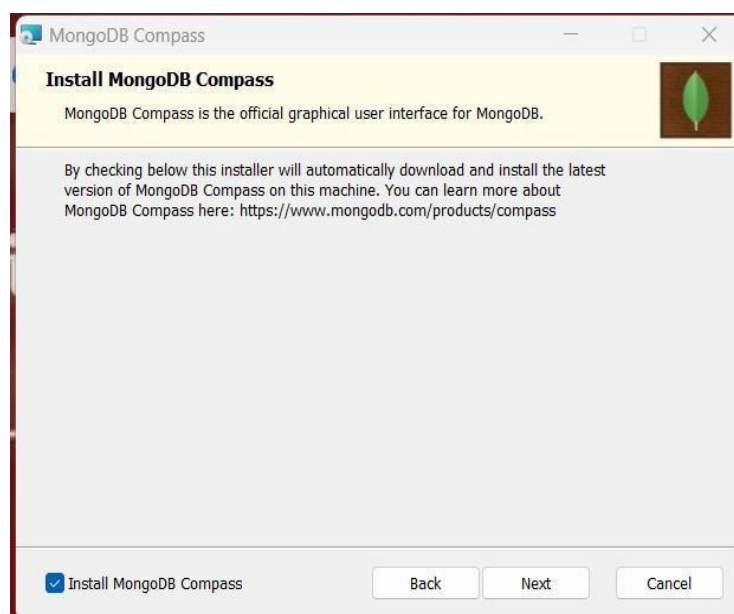
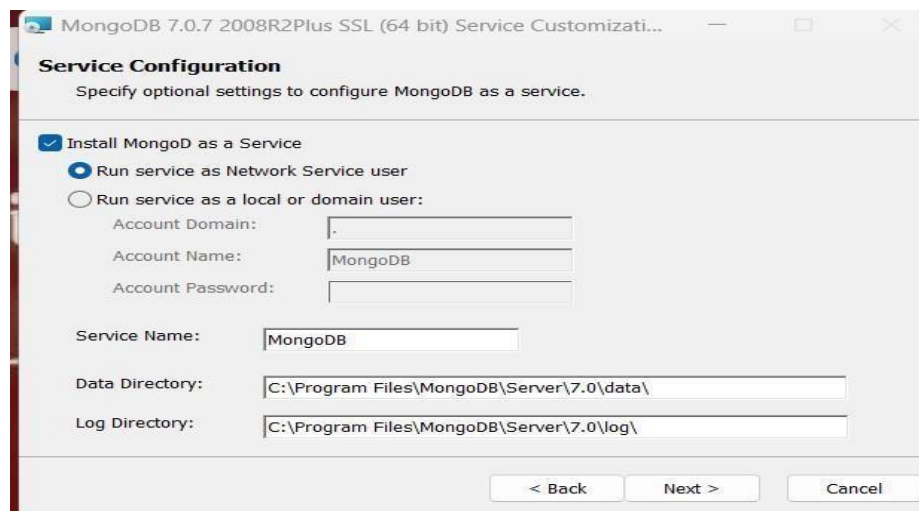


Accept the user agreement and Select the setup type as Complete and click Next.

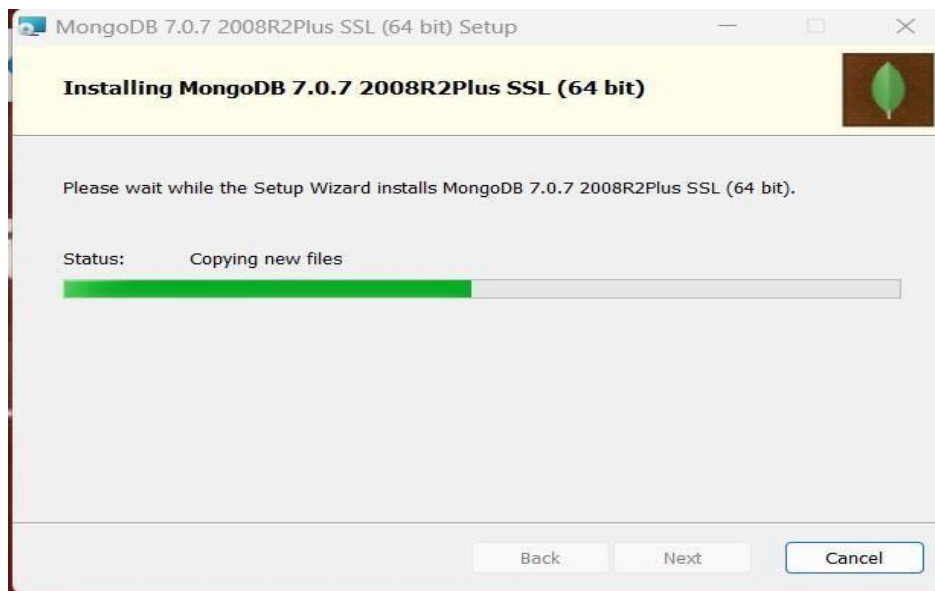
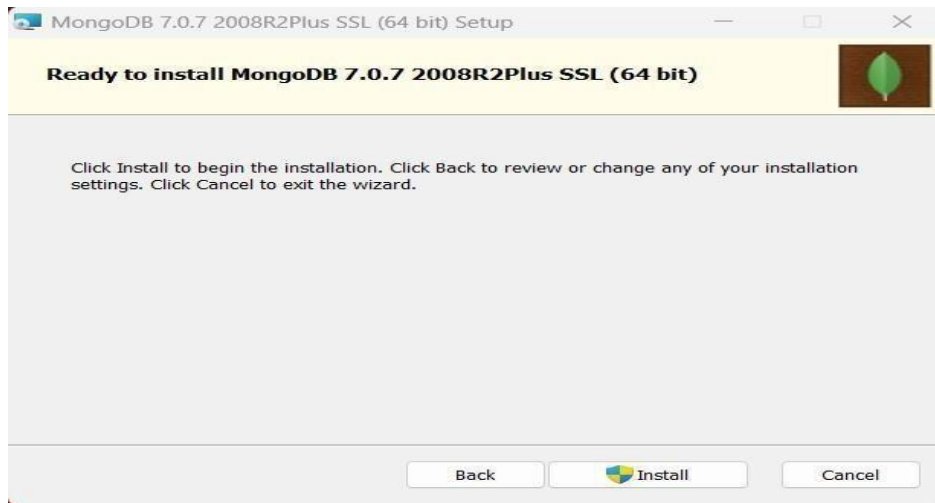


**STEP 3:**

Click the Next button in the upcoming two windows.



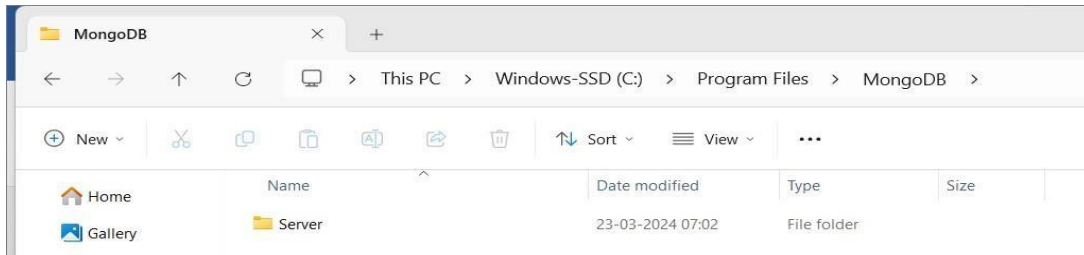
Click on the install button to start the installation.

**STEP 4:**

Click on the Finish button to finish the set up.



**STEP 5:** MongoDB requires a data directory under the C drive which will have a Server file. Create it.



**STEP 8:** In the command prompt, enter the following command “cd C:\Program Files\MongoDB\Server\7.0\bin”. Once inside the folder, type “mongod” command.

```
C:\Program Files\MongoDB\Server\7.0\bin>mongod
{"t":{"sdate":"2024-06-07T14:04:53.974+05:30"},"s":"I",
  "c":"CONTROL",
  "id":23285,
  "ctx":{"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}},
  "c":"NETWORK",
  "id":4925701,
  "ctx":{"thread1","msg":"Initialized wire specification, attributes: {spec: {incomingExternalClient: {minWireVersion: 0, maxWireVersion: 21}, outgoingExternalClient: {minWireVersion: 0, maxWireVersion: 21}, internalClient: {true}}}}"},
  "c":"NETWORK",
  "id":4648602,
  "ctx":{"thread1","msg":"Implicit TCP FastOpen in use."}},
  "c":"REPL",
  "id":5123908,
  "ctx":{"thread1","msg":"Successfully registered PrimaryOnlyService, attributes: {service: 'TenantMigrationDonorService', namespace: 'config.tenantmigrationdonors'}"}},
  "c":"REPL",
  "id":5123908,
  "ctx":{"thread1","msg":"Successfully registered PrimaryOnlyService, attributes: {service: 'TenantMigrationRecipientService', namespace: 'config.tenantmigrationrecipients'}"}},
  "c":"CONTROL",
  "id":5945603,
  "ctx":{"thread1","msg":"Multi threading initialized"}},
  "c":"TENANT_M",
  "id":7891680,
  "ctx":{"thread1","msg":"Starting TenantMigrationAccessLockerRegistry"}},
  "c":"INIT",
  "id":4615011,
  "ctx":{"initandlisten","msg":"MongoDB starting", "attr":{"pid":12836, "port":27017, "dbPath":"C:/data/db/", "architecture": "64-bit", "host": "LAPTOP-731FUPV8"}},
  "c":"CONTROL",
  "id":23398,
  "ctx":{"initandlisten","msg":"Target operating system minimum version", "attr":{"targetMinOS": "Windows 7/Windows Server 2008 R2"}},
  "c":"CONTROL",
  "id":23403,
  "ctx":{"initandlisten","msg":"Build Info", "attr":{"buildInfo":{"version": "7.0.8", "gitVersion": "c5d3e55ba3d98e2f4d76sec4533809a0a0a0", "modules": {}, "allocator": "tcmalloc", "environment": {"distro": "windows", "distarch": "x86_64", "target_arch": "x86_64"}}}}},
  "c":"CONTROL",
  "id":51765,
  "ctx":{"initandlisten","msg":"Operating System", "attr":{"os":{"name": "Microsoft Windows 10", "version": "10.0 (build 22H2)"}}}},
  "c":"CONTROL",
  "id":21951,
  "ctx":{"initandlisten","msg":"Options set by command line", "attr":{"options": {}}}),
  "c":"CONTROL",
  "id":28557,
  "ctx":{"initandlisten","msg":"DBException in initandlisten, terminating", "attr":{"error": {"NonExistentPath: Data directory C:/data/db/ not found. Create the missing directory or specify another path using (1) the --dbpath command line option, or (2) by adding the --storage.dbPath option in the configuration file."}}}}},
  "c":"REPL",
  "id":4790602,
  "ctx":{"initandlisten","msg":"Attempting to enter quiesce mode"}},
  "c":"REPL",
  "id":6371601,
  "ctx":{"initandlisten","msg":"Shutting down the FLE Crud thread pool"}},
  "c":"REPL",
  "id":4780901,
  "ctx":{"initandlisten","msg":"Shutting down the MirrorMaster"}},
  "c":"COMMAND",
  "id":4780902,
  "ctx":{"initandlisten","msg":"Shutting down the WaitFormajerityService"}},
  "c":"NETWORK",
  "id":28652,
  "ctx":{"initandlisten","msg":"Shutting down the global connection pool"}},
  "c":"NETWORK",
  "id":4780905,
  "ctx":{"initandlisten","msg":"Shutting down the FlowControlTicketHolder"}},
  "c":"CONTROL",
  "id":4780906,
  "ctx":{"initandlisten","msg":"Shutting down the FlowControlTicketHolder"}},
  "c":"CONTROL",
  "id":28652,
  "ctx":{"initandlisten","msg":"Stopping further Flow control ticket acquisitions."}},
  "c":"NETWORK",
  "id":4780910,
  "ctx":{"initandlisten","msg":"Shutting down the ReplicasetMonitor"}},
  "c":"SHARDING",
  "id":4780921,
  "ctx":{"initandlisten","msg":"Shutting down the MigrationUtilExecutor"}},
  "c":"ASIO",
  "id":28582,
  "ctx":{"initandlisten","msg":"Killing all outstanding express activity."}},
  "c":"COMMAND",
  "id":4780923,
  "ctx":{"initandlisten","msg":"Shutting down the ServiceEntryPoint"}},
  "c":"SHARDING",
  "id":4780922,
  "ctx":{"initandlisten","msg":"Shutting down the TLM monitor"}},
  "c":"CONTROL",
  "id":6278511,
  "ctx":{"initandlisten","msg":"Shutting down the Change Stream Expired Pre-images Remover"}},
  "c":"CONTROL",
  "id":4780929,
  "ctx":{"initandlisten","msg":"Acquiring the global lock for shutdown"}},
  "c":"CONTROL",
  "id":4780931,
  "ctx":{"initandlisten","msg":"Dropping the scope cache for shutdown"}},
  "c":"CONTROL",
  "id":28568,
  "ctx":{"initandlisten","msg":"Now exiting"}},
  "c":"CONTROL",
  "id":8423484,
  "ctx":{"initandlisten","msg":"mongod shutdown complete", "attr":{"summary": {"Statistics": {"Enter terminal shutdown": "0 ms", "Stop down the replication coordinator for shutdown": "0 ms", "Time spent in quiesce mode": "0 ms", "Shut down FLE Crud subsystem": "2 ms", "Shut down MirrorMaster": "0 ms", "Shut down WaitFormajerityService": "0 ms", "Shut down the transport layer": "1 ms", "Shut down the global connection pool": "1 ms", "Shut down the flow control ticket holder": "1 ms", "Shut down the replica set monitor": "1 ms", "Shut down the migration util executor": "2 ms", "Shut down the TLM monitor": "0 ms", "Shut down expired pre-images and documents removers": "0 ms", "Wait for the oplog cap maintainer thread to stop": "0 ms", "Shut down full-time data capture": "0 ms", "ShutdownTask total elapsed time": "20 ms"}}}}}},
  "c":"INIT",
  "id":23138,
  "ctx":{"initandlisten","msg":"Shutting down", "attr":{"exitCode":100}}}
```

**STEP 9:** After the service is started open another command prompt and move to the directory specified before. Type “mongo” in order to open the mongo shell.

MongoDB is installed successfully.

### Creation of collection:

1. Create database using “use databasename” command.
2. Create a collection and insert a value using

```
test> show dbs
admin 40.00 KiB
config 60.00 KiB
local 40.00 KiB
userdb 8.00 KiB
test> use vehicle
switched to db vehicle

vehicle> db.createCollection("employee");
{ ok: 1 }
vehicle> show dbs
admin 40.00 KiB
config 72.00 KiB
local 40.00 KiB
userdb 8.00 KiB
vehicle 8.00 KiB

3.
vehicle> db.employee.insertOne({Emp_id: 1, "Emp_Name": "Rith1", "Phone_no": 9345542103, "salary": 50000})
{ acknowledged: true, insertedId: ObjectId('661268dcefc8f645d016c9b5') }

4. “db.collectionname.insert({"key":value,"key": "value"})” command.
```

### Result:

Thus, the MongoDB was installed and the collections are created successfully.