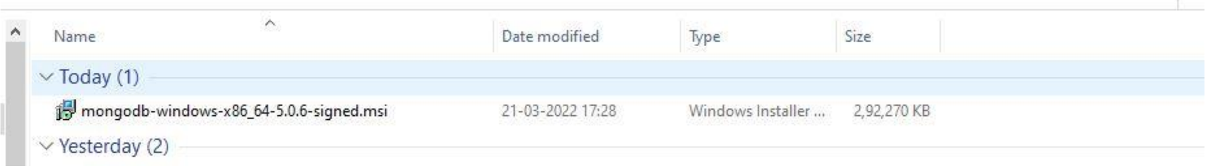
BIBD MINI PROJECT

-MongoDB

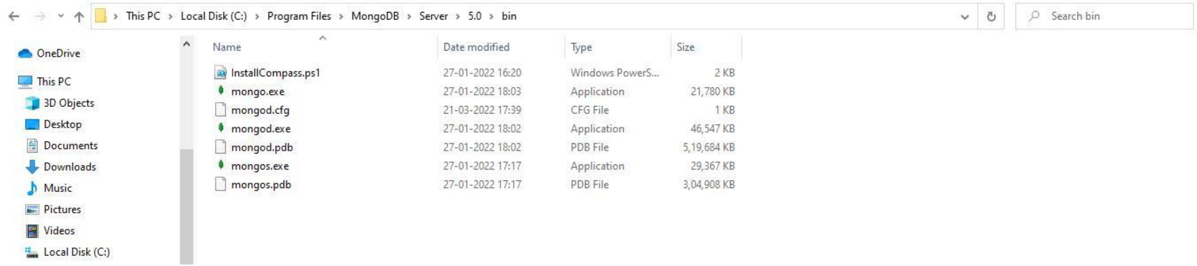
STEP 1: We first need to install MongoDB in our pc.

We will do so by installing the installer.



Name	Date modified	Type	Size
Today (1)			
mongodb-windows-x86_64-5.0.6-signed.msi	21-03-2022 17:28	Windows Installer ...	2,92,270 KB
Yesterday (2)			

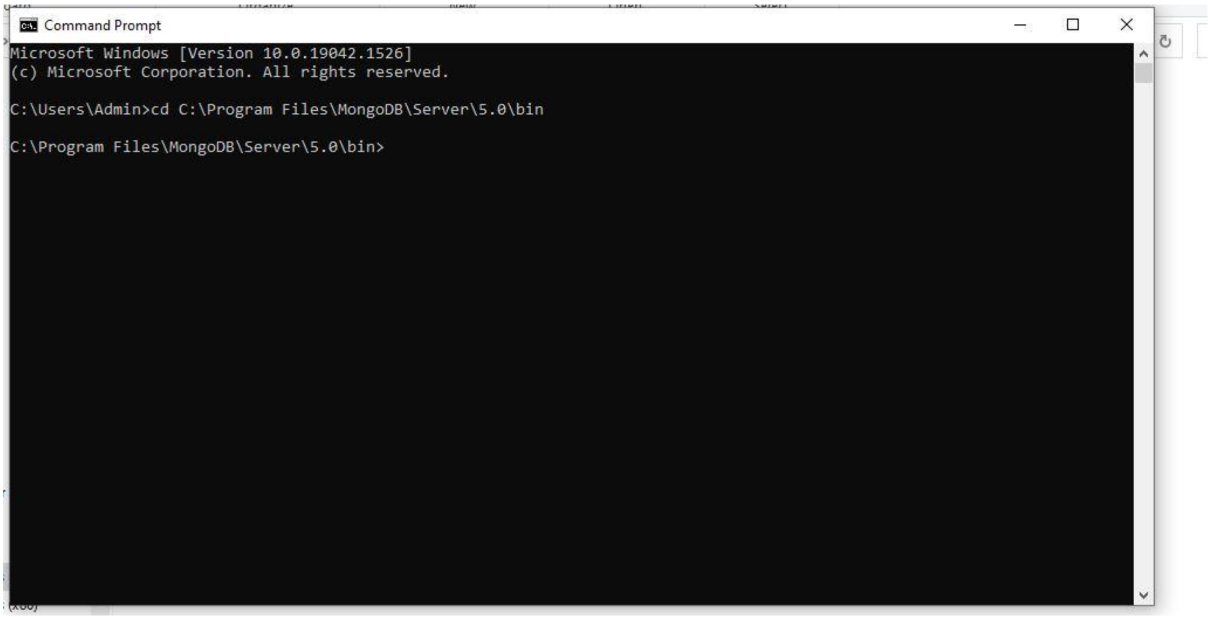
STEP2: Then after installing installer successfully, we will run it on our pc and it will be downloaded successfully. In this case it is there on C drive.



This PC › Local Disk (C:) › Program Files › MongoDB › Server › 5.0 › bin

Name	Date modified	Type	Size
InstallCompass.ps1	27-01-2022 16:20	Windows PowerS...	2 KB
mongo.exe	27-01-2022 18:03	Application	21,780 KB
mongod.cfg	21-03-2022 17:39	CFG File	1 KB
mongod.exe	27-01-2022 18:02	Application	46,547 KB
mongod.pdb	27-01-2022 18:02	PDB File	5,19,684 KB
mongos.exe	27-01-2022 17:17	Application	29,367 KB
mongos.pdb	27-01-2022 17:17	PDB File	3,04,908 KB

STEP 3: After that we will open Command Prompt and we will assign the path folder on cmd.

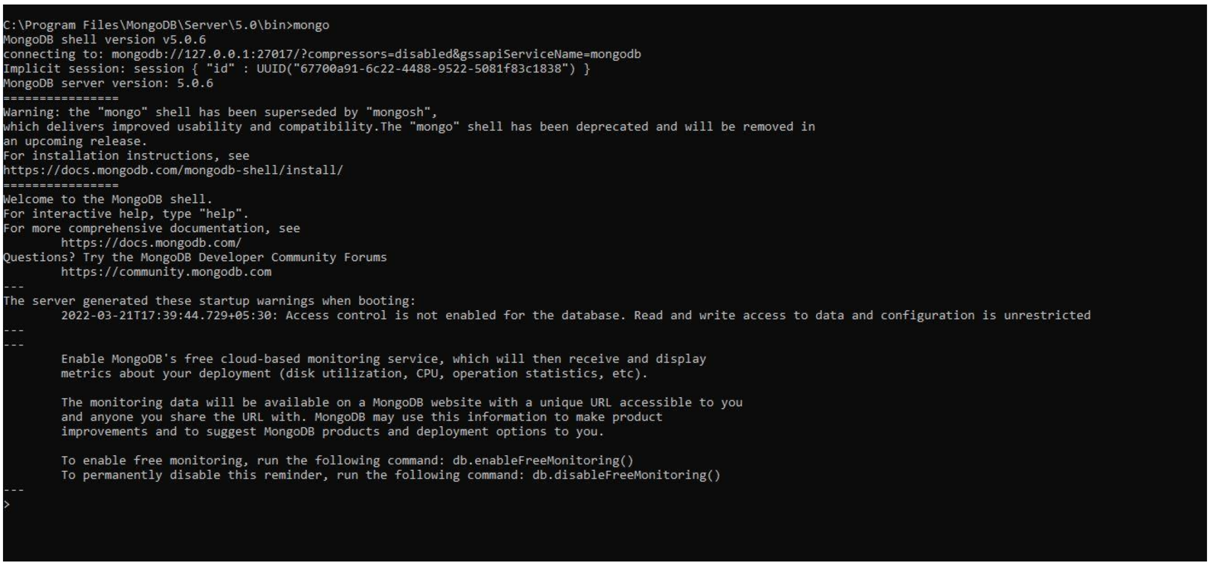


```
Microsoft Windows [Version 10.0.19042.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>cd C:\Program Files\MongoDB\Server\5.0\bin

C:\Program Files\MongoDB\Server\5.0\bin>
```

STEP 4: After this we will type command:

mongo

It is used to connect to the database server on localhost.

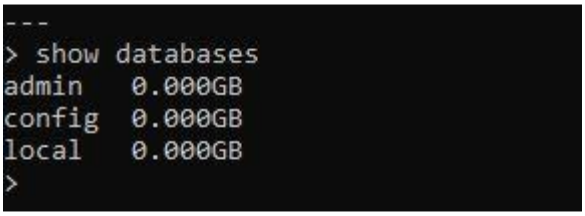


```
C:\Program Files\MongoDB\Server\5.0\bin>mongo
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("67700a91-6c22-4488-9522-5081f83c1838") }
MongoDB server version: 5.0.6
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
        https://community.mongodb.com
---
The server generated these startup warnings when booting:
        2022-03-21T17:39:44.729+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>
```

Step5:To show databases, command will be:

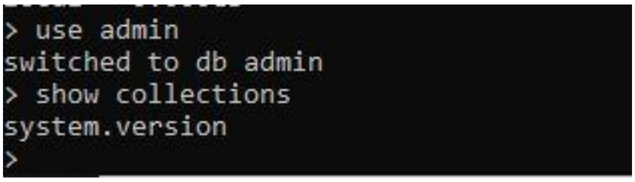
Show databases

Databases in the server will be shown



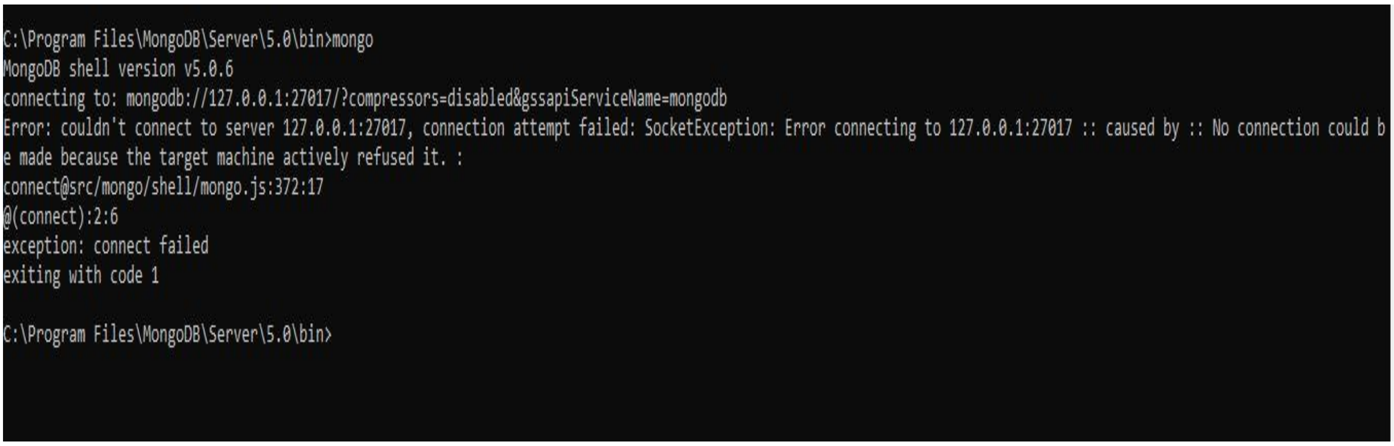
```
---
> show databases
admin   0.000GB
config  0.000GB
local   0.000GB
>
```

Can use the database admin here and then we will show the collections in the database.



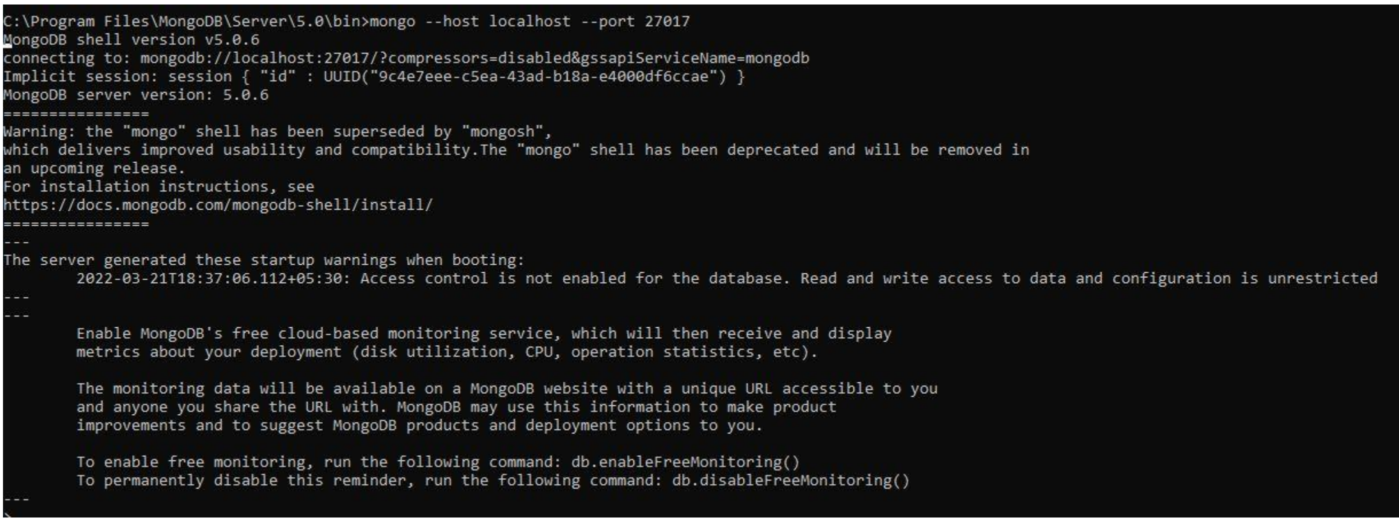
```
> use admin
switched to db admin
> show collections
system.version
>
```

Step 6:Now in case we stopped the MongoDB Server we will face this



```
C:\Program Files\MongoDB\Server\5.0\bin>mongo
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Error: couldn't connect to server 127.0.0.1:27017, connection attempt failed: SocketException: Error connecting to 127.0.0.1:27017 :: caused by :: No connection could b
e made because the target machine actively refused it. :
connect@src/mongo/shell/mongo.js:372:17
@(connect):2:6
exception: connect failed
exiting with code 1

C:\Program Files\MongoDB\Server\5.0\bin>
```

Step 7: Then again we will start MongoDB server and then we will try to connect to server again. We have done it by using localhost and port number that is 27017 by default.

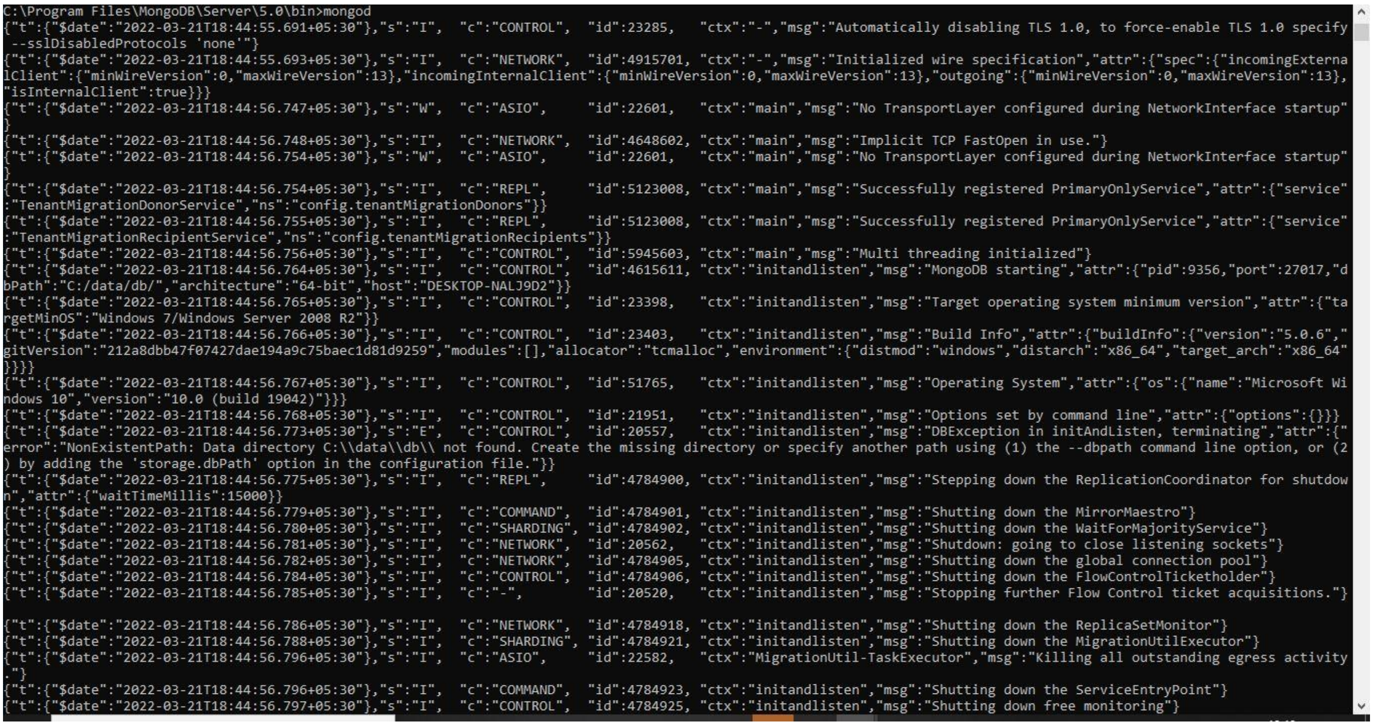


```
C:\Program Files\MongoDB\Server\5.0\bin>mongo --host localhost --port 27017
MongoDB shell version v5.0.6
connecting to: mongodb://localhost:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("9c4e7eee-c5ea-43ad-b18a-e4000df6ccae") }
MongoDB server version: 5.0.6
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2022-03-21T18:37:06.112+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>
```

It has successfully connected to the server.

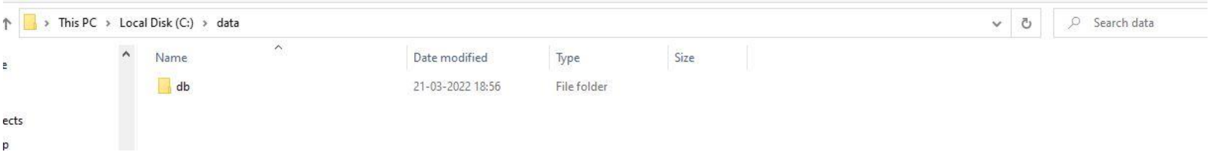
Step 8 : Now by stopping server again, we will put the another command which is a server program.

mongod



```
C:\Program Files\MongoDB\Server\5.0\bin>mongod
{"t":{"$date":"2022-03-21T18:44:55.691+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"-","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify
 --sslDisabledProtocols 'none'"}
{"t":{"$date":"2022-03-21T18:44:55.693+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"-","msg":"Initialized wire specification","attr":{"spec":{"incomingExterna
lClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},
"isInternalClient":true}}}
{"t":{"$date":"2022-03-21T18:44:56.747+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"
}
{"t":{"$date":"2022-03-21T18:44:56.748+05:30"},"s":"I",  "c":"NETWORK",  "id":4648602, "ctx":"main","msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2022-03-21T18:44:56.754+05:30"},"s":"W",  "c":"ASIO",     "id":22601,   "ctx":"main","msg":"No TransportLayer configured during NetworkInterface startup"
}
{"t":{"$date":"2022-03-21T18:44:56.754+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"main","msg":"Successfully registered PrimaryOnlyService","attr":{"service"
:"TenantMigrationDonorService","ns":"config.tenantMigrationDonors"}}
{"t":{"$date":"2022-03-21T18:44:56.755+05:30"},"s":"I",  "c":"REPL",     "id":5123008, "ctx":"main","msg":"Successfully registered PrimaryOnlyService","attr":{"service"
:"TenantMigrationRecipientService","ns":"config.tenantMigrationRecipients"}}
{"t":{"$date":"2022-03-21T18:44:56.756+05:30"},"s":"I",  "c":"CONTROL",  "id":5945603, "ctx":"main","msg":"Multi threading initialized"}
{"t":{"$date":"2022-03-21T18:44:56.764+05:30"},"s":"I",  "c":"CONTROL",  "id":4615611, "ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":9356,"port":27017,"d
bPath":"C:/data/db/","architecture":"64-bit","host":"DESKTOP-NALJ9D2"}}
{"t":{"$date":"2022-03-21T18:44:56.765+05:30"},"s":"I",  "c":"CONTROL",  "id":23398,   "ctx":"initandlisten","msg":"Target operating system minimum version","attr":{"ta
rgetMinOS":"Windows 7/Windows Server 2008 R2"}}
{"t":{"$date":"2022-03-21T18:44:56.766+05:30"},"s":"I",  "c":"CONTROL",  "id":23403,   "ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"5.0.6","
gitVersion":"212a8dbb47f07427dae194a9c75baec1d81d9259","modules":[],"allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"
}}}}
{"t":{"$date":"2022-03-21T18:44:56.767+05:30"},"s":"I",  "c":"CONTROL",  "id":51765,   "ctx":"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Wi
ndows 10","version":"10.0 (build 19042)"}}}
{"t":{"$date":"2022-03-21T18:44:56.768+05:30"},"s":"I",  "c":"CONTROL",  "id":21951,   "ctx":"initandlisten","msg":"Options set by command line","attr":{"options":{}}}
{"t":{"$date":"2022-03-21T18:44:56.773+05:30"},"s":"E",  "c":"CONTROL",  "id":20557,   "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."}}
{"t":{"$date":"2022-03-21T18:44:56.775+05:30"},"s":"I",  "c":"REPL",     "id":4784900, "ctx":"initandlisten","msg":"Stepping down the ReplicationCoordinator for shutdow
n","attr":{"waitTimeMillis":15000}}
{"t":{"$date":"2022-03-21T18:44:56.779+05:30"},"s":"I",  "c":"COMMAND",  "id":4784901, "ctx":"initandlisten","msg":"Shutting down the MirrorMaestro"}
{"t":{"$date":"2022-03-21T18:44:56.780+05:30"},"s":"I",  "c":"SHARDING", "id":4784902, "ctx":"initandlisten","msg":"Shutting down the WaitForMajorityService"}
{"t":{"$date":"2022-03-21T18:44:56.781+05:30"},"s":"I",  "c":"NETWORK",  "id":20562,   "ctx":"initandlisten","msg":"Shutdown: going to close listening sockets"}
{"t":{"$date":"2022-03-21T18:44:56.782+05:30"},"s":"I",  "c":"NETWORK",  "id":4784905, "ctx":"initandlisten","msg":"Shutting down the global connection pool"}
{"t":{"$date":"2022-03-21T18:44:56.784+05:30"},"s":"I",  "c":"CONTROL",  "id":4784906, "ctx":"initandlisten","msg":"Shutting down the FlowControlTicketholder"}
{"t":{"$date":"2022-03-21T18:44:56.785+05:30"},"s":"I",  "c":"-",        "id":20520,   "ctx":"initandlisten","msg":"Stopping further Flow Control ticket acquisitions."}

{"t":{"$date":"2022-03-21T18:44:56.786+05:30"},"s":"I",  "c":"NETWORK",  "id":4784918, "ctx":"initandlisten","msg":"Shutting down the ReplicaSetMonitor"}
{"t":{"$date":"2022-03-21T18:44:56.788+05:30"},"s":"I",  "c":"SHARDING", "id":4784921, "ctx":"initandlisten","msg":"Shutting down the MigrationUtilExecutor"}
{"t":{"$date":"2022-03-21T18:44:56.796+05:30"},"s":"I",  "c":"ASIO",     "id":22582,   "ctx":"MigrationUtil-TaskExecutor","msg":"Killing all outstanding egress activity
."}
{"t":{"$date":"2022-03-21T18:44:56.796+05:30"},"s":"I",  "c":"COMMAND",  "id":4784923, "ctx":"initandlisten","msg":"Shutting down the ServiceEntryPoint"}
{"t":{"$date":"2022-03-21T18:44:56.797+05:30"},"s":"I",  "c":"CONTROL",  "id":4784925, "ctx":"initandlisten","msg":"Shutting down free monitoring"}
```

Step 9: Now we faced an error. We need to create a new folder in C drive.

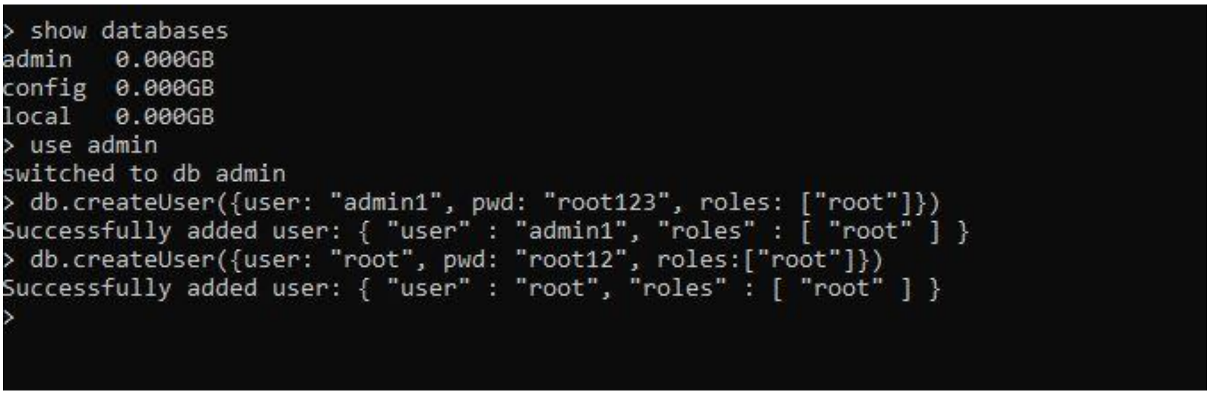


This PC > Local Disk (C:) > data

Name	Date modified	Type	Size
db	21-03-2022 18:56	File folder	

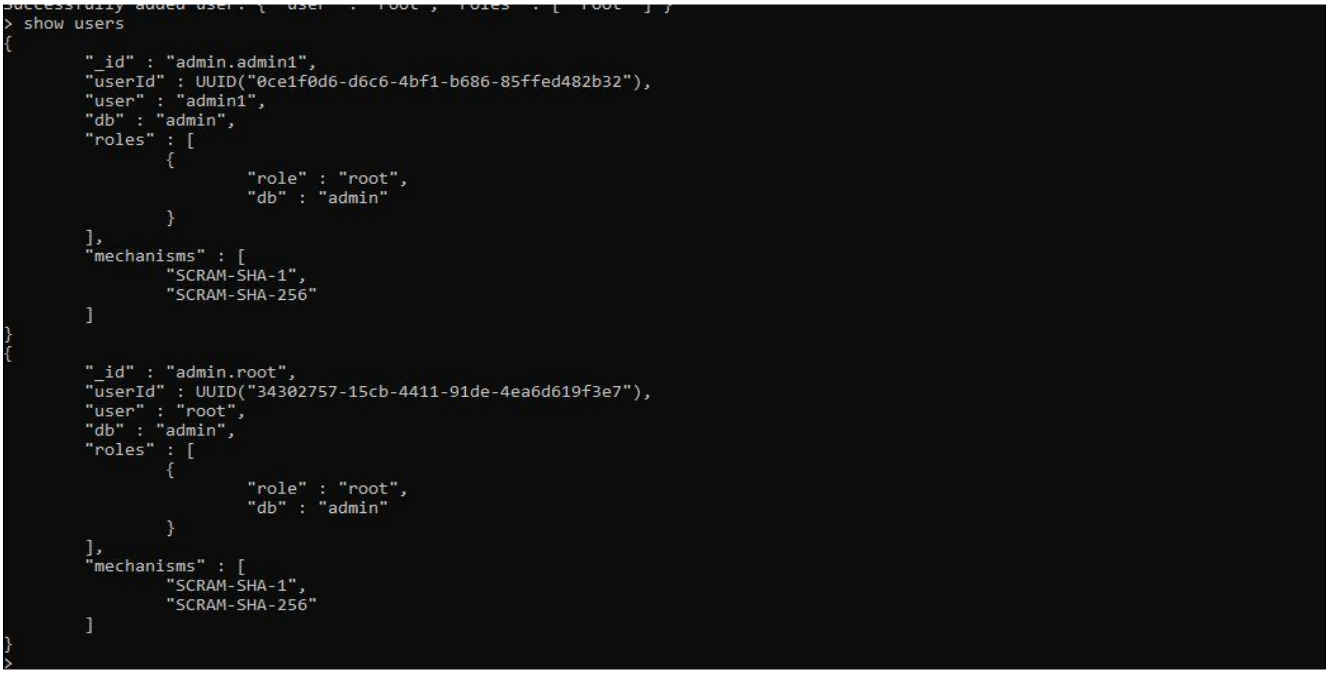
We will run the command mongod again

Step 10: Now we will be creating a new user in admin database



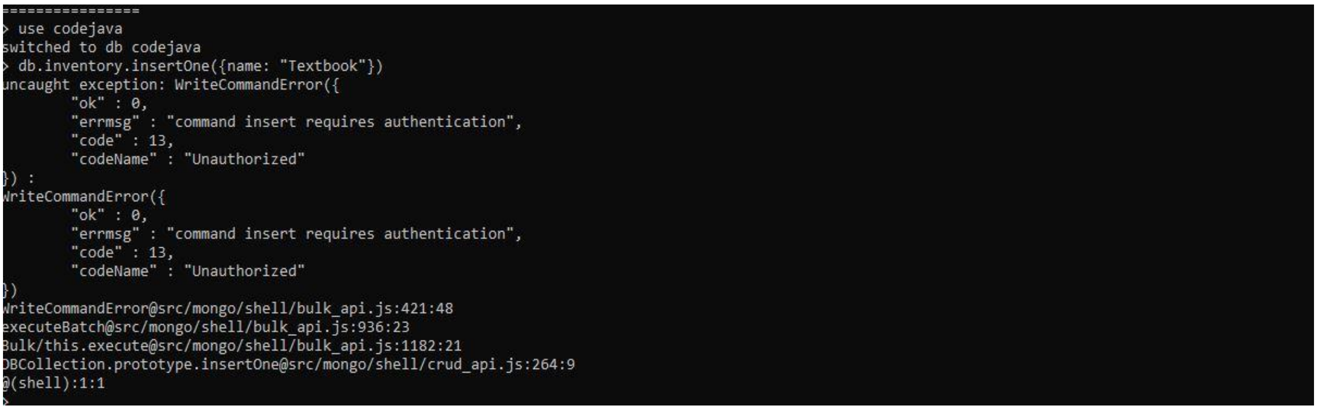
```
> show databases
admin   0.000GB
config  0.000GB
local   0.000GB
> use admin
switched to db admin
> db.createUser({user: "admin1", pwd: "root123", roles: ["root"]})
Successfully added user: { "user" : "admin1", "roles" : [ "root" ] }
> db.createUser({user: "root", pwd: "root12", roles:["root"]})
Successfully added user: { "user" : "root", "roles" : [ "root" ] }
>
```

Showing the users



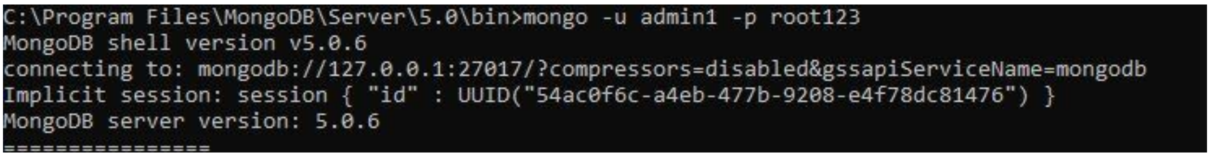
```
Successfully added user: { "user" : "root", "roles" : [ "root" ] }
> show users
{
        "_id" : "admin.admin1",
        "userId" : UUID("0ce1f0d6-d6c6-4bf1-b686-85ffed482b32"),
        "user" : "admin1",
        "db" : "admin",
        "roles" : [
                {
                        "role" : "root",
                        "db" : "admin"
                }
        ],
        "mechanisms" : [
                "SCRAM-SHA-1",
                "SCRAM-SHA-256"
        ]
}
{
        "_id" : "admin.root",
        "userId" : UUID("34302757-15cb-4411-91de-4ea6d619f3e7"),
        "user" : "root",
        "db" : "admin",
        "roles" : [
                {
                        "role" : "root",
                        "db" : "admin"
                }
        ],
        "mechanisms" : [
                "SCRAM-SHA-1",
                "SCRAM-SHA-256"
        ]
}
>
```

Step 11: To create a database and insert a document to collection



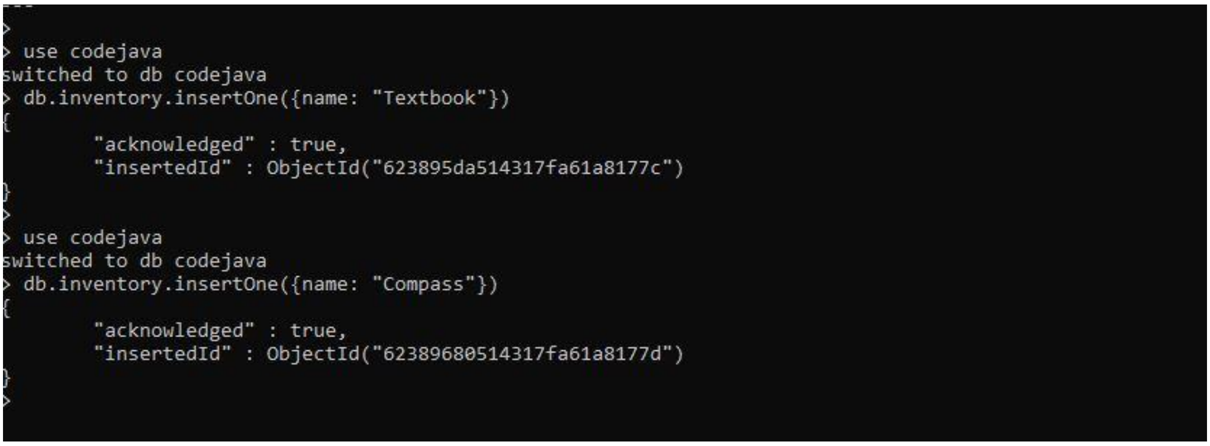
```
================
> use codejava
switched to db codejava
> db.inventory.insertOne({name: "Textbook"})
uncaught exception: WriteCommandError({
        "ok" : 0,
        "errmsg" : "command insert requires authentication",
        "code" : 13,
        "codeName" : "Unauthorized"
}) :
WriteCommandError({
        "ok" : 0,
        "errmsg" : "command insert requires authentication",
        "code" : 13,
        "codeName" : "Unauthorized"
})
WriteCommandError@src/mongo/shell/bulk_api.js:421:48
executeBatch@src/mongo/shell/bulk_api.js:936:23
Bulk/this.execute@src/mongo/shell/bulk_api.js:1182:21
DBCollection.prototype.insertOne@src/mongo/shell/crud_api.js:264:9
@(shell):1:1
>
```

Step 12: As it is showing that it is unauthorized, we need to provide username and password.



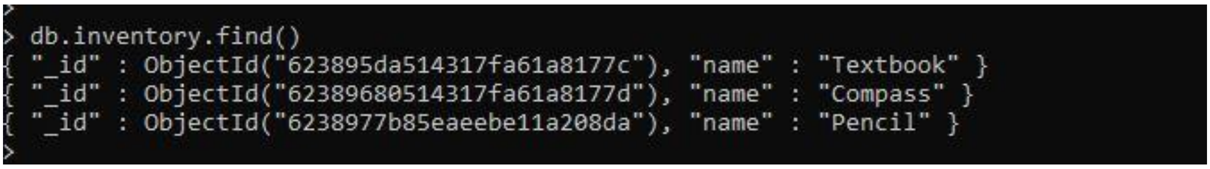
```
C:\Program Files\MongoDB\Server\5.0\bin>mongo -u admin1 -p root123
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("54ac0f6c-a4eb-477b-9208-e4f78dc81476") }
MongoDB server version: 5.0.6
================
```

Now to check that it is successful.



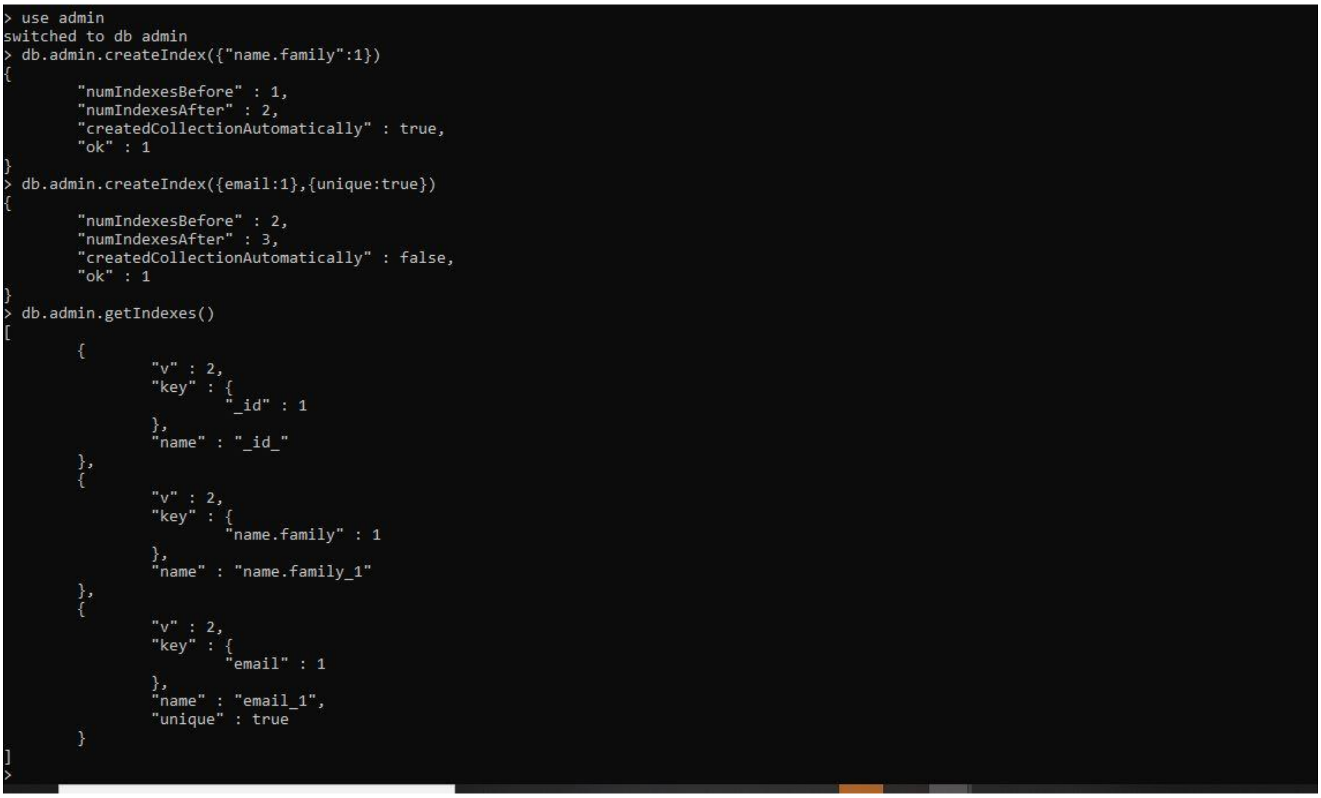
```
---
>
> use codejava
switched to db codejava
> db.inventory.insertOne({name: "Textbook"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("623895da514317fa61a8177c")
}
>
> use codejava
switched to db codejava
> db.inventory.insertOne({name: "Compass"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("62389680514317fa61a8177d")
}
>
```

Step 13: Now in order to find inventory.



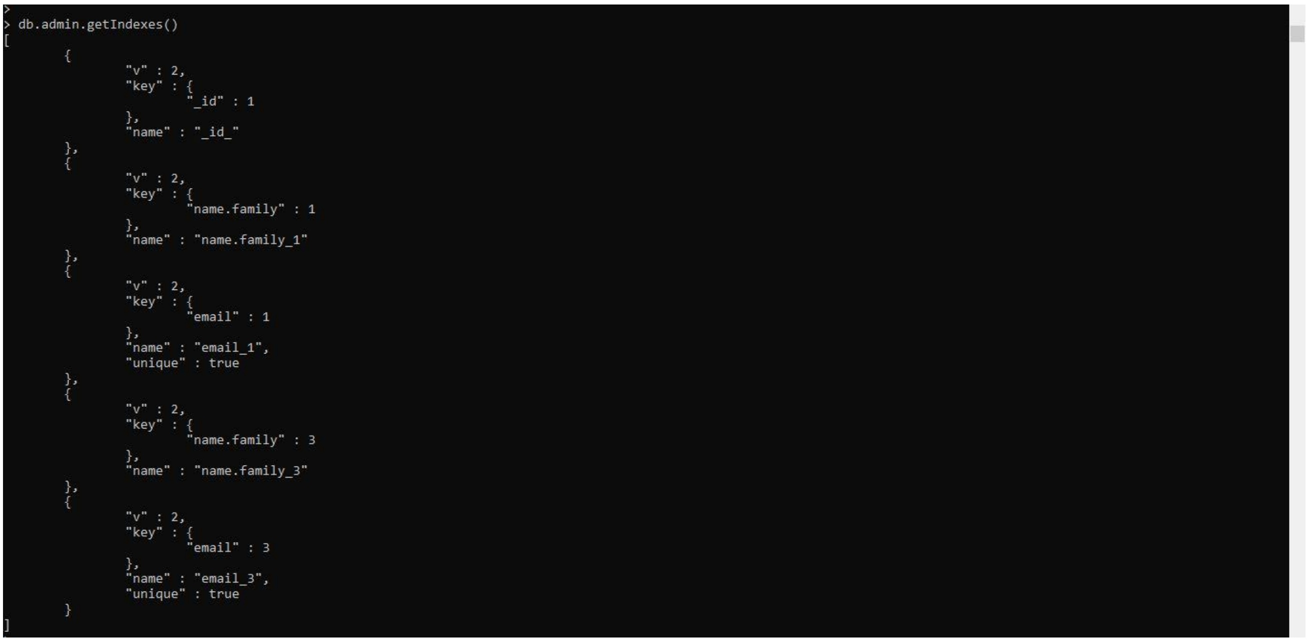
```
>
> db.inventory.find()
{ "_id" : ObjectId("623895da514317fa61a8177c"), "name" : "Textbook" }
{ "_id" : ObjectId("62389680514317fa61a8177d"), "name" : "Compass" }
{ "_id" : ObjectId("6238977b85eaeebe11a208da"), "name" : "Pencil" }
>
```

Step 14: Now we will create an index . We have also created an unique index that allows us to ensure that there is at most 1 record in the collection with a given value for field.

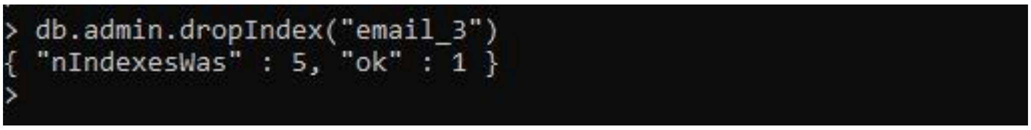


```
> use admin
switched to db admin
> db.admin.createIndex({"name.family":1})
{
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "createdCollectionAutomatically" : true,
        "ok" : 1
}
> db.admin.createIndex({email:1},{unique:true})
{
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 3,
        "createdCollectionAutomatically" : false,
        "ok" : 1
}
> db.admin.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "v" : 2,
                "key" : {
                        "name.family" : 1
                },
                "name" : "name.family_1"
        },
        {
                "v" : 2,
                "key" : {
                        "email" : 1
                },
                "name" : "email_1",
                "unique" : true
        }
]
>
```

Step 15:Now in order to drop an index we will use.

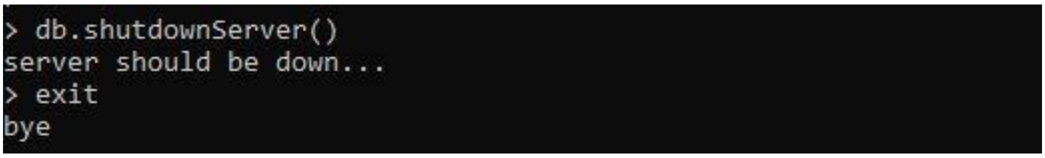


```
>
> db.admin.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "v" : 2,
                "key" : {
                        "name.family" : 1
                },
                "name" : "name.family_1"
        },
        {
                "v" : 2,
                "key" : {
                        "email" : 1
                },
                "name" : "email_1",
                "unique" : true
        },
        {
                "v" : 2,
                "key" : {
                        "name.family" : 3
                },
                "name" : "name.family_3"
        },
        {
                "v" : 2,
                "key" : {
                        "email" : 3
                },
                "name" : "email_3",
                "unique" : true
        }
]
```



```
> db.admin.dropIndex("email_3")
{ "nIndexesWas" : 5, "ok" : 1 }
>
```

Step 16: To shutdown the server we will use



```
> db.shutdownServer()
server should be down...
> exit
bye
```