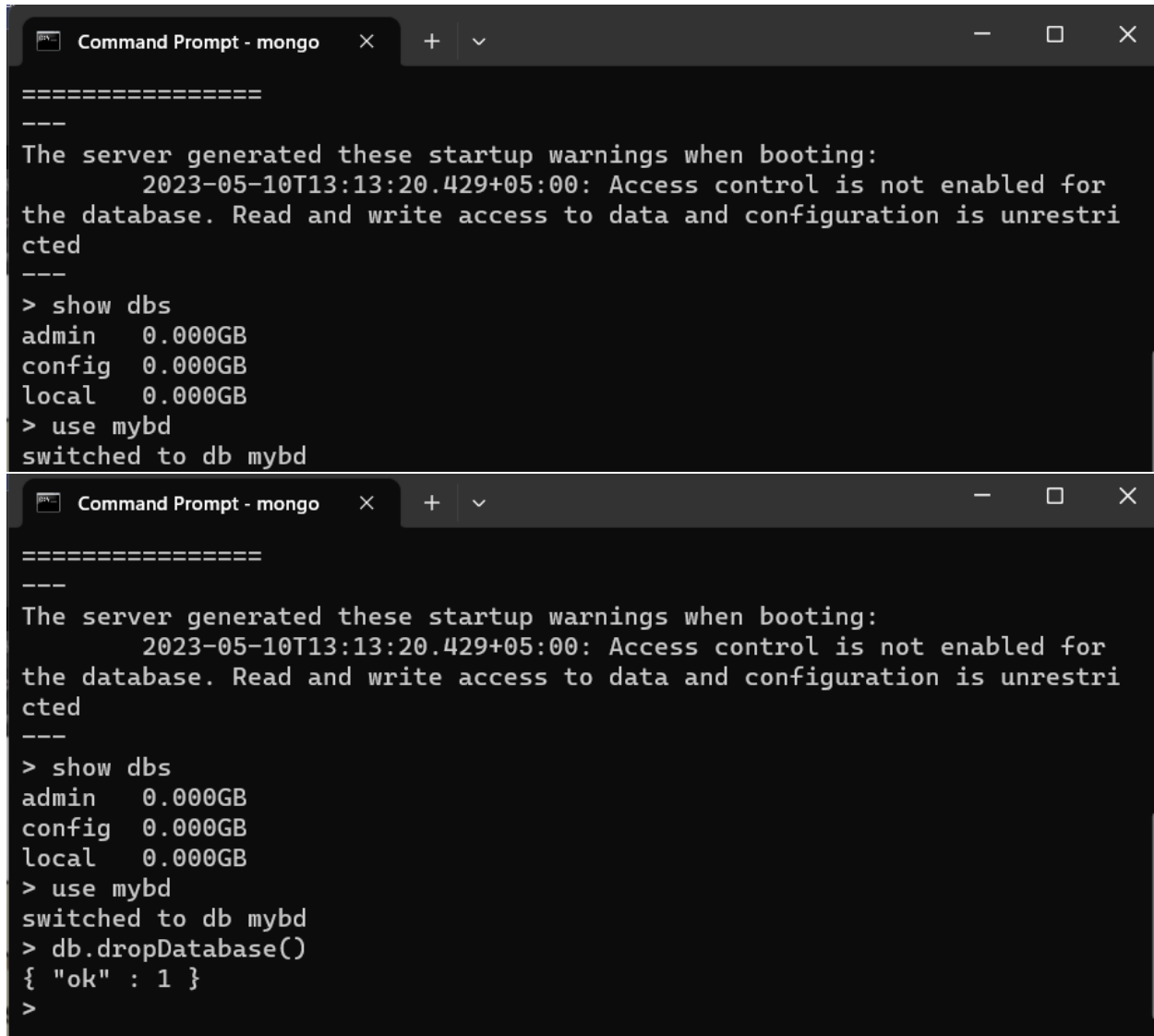


# Lab 12

Ahmed Baig (20i-1884)



The image displays two screenshots of a MongoDB Command Prompt window. The top screenshot shows the initial state of the database with three databases: admin (0.000GB), config (0.000GB), and local (0.000GB). The user has switched to the 'mybd' database. The bottom screenshot shows the same state, but with the 'mybd' database dropped, resulting in a confirmation message: { "ok" : 1 }.

```
Command Prompt - mongo
=====
---
The server generated these startup warnings when booting:
    2023-05-10T13:13:20.429+05:00: Access control is not enabled for
the database. Read and write access to data and configuration is unres-
tricted
---
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> use mybd
switched to db mybd

Command Prompt - mongo
=====
---
The server generated these startup warnings when booting:
    2023-05-10T13:13:20.429+05:00: Access control is not enabled for
the database. Read and write access to data and configuration is unres-
tricted
---
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
> use mybd
switched to db mybd
> db.dropDatabase()
{ "ok" : 1 }
>
```

```
Command Prompt - mongo x + v
> db.createCollection(name)
uncaught exception: ReferenceError: name is not defined :
@(shell):1:1
> db.createCollection(name string)
uncaught exception: SyntaxError: missing ) after argument list :
@(shell):1:25
> db.createCollection(myCollection)
uncaught exception: ReferenceError: myCollection is not defined :
@(shell):1:1
> use test
switched to db test
> db.createCollection("myCollection")
{ "ok" : 1 }
> show collections
myCollection
>
```

```
Command Prompt - mongo x + v
@(shell):1:1
> db.createCollection(name string)
uncaught exception: SyntaxError: missing ) after argument list :
@(shell):1:25
> db.createCollection(myCollection)
uncaught exception: ReferenceError: myCollection is not defined :
@(shell):1:1
> use test
switched to db test
> db.createCollection("myCollection")
{ "ok" : 1 }
> show collections
myCollection
> db.myCollection.drop()
true
>
```

```
Command Prompt - mongo x + v
=====
---
The server generated these startup warnings when booting:
  2023-05-10T13:13:20.429+05:00: Access control is not enabled for the database. Read and write access to data and
  configuration is unrestricted
---
> db.mycol.insert({ _id: ObjectId("123456789012345678901234"), title: 'Hello DB', likes: 100 })
WriteResult({ "nInserted" : 1 })
>
```

```
Command Prompt - mongo x + v
The server generated these startup warnings when booting:
  2023-05-10T13:13:20.429+05:00: Access control is not enabled for the database. Read and write access to data and
  configuration is unrestricted
---
> db.mycol.insert({ _id: ObjectId("123456789012345678901234"), title: 'Hello DB', likes: 100 })
WriteResult({ "nInserted" : 1 })
> db.mycol.find()
{ "_id" : ObjectId("123456789012345678901234"), "title" : "Hello DB", "likes" : 100 }
>

Command Prompt - mongo x + v
> db.mycol.find()
{ "_id" : ObjectId("123456789012345678901234"), "title" : "Hello DB", "likes" : 100 }
> db.mycol.find().pretty()
{
  "_id" : ObjectId("123456789012345678901234"),
  "title" : "Hello DB",
  "likes" : 100
}
>
```

Query 1: Write a query from employee collection where salary is between 1000 and 2000 and employee belongs to department 10.

```
db.employee.find({
  salary: { $gte: 1000, $lte: 2000 },
  department: 10
})
```

Query 2: Write a query from employee collection 10 or 20. where employee belongs to department.

```
db.employee.find({
  department: { $in: [10, 20] }
})
```