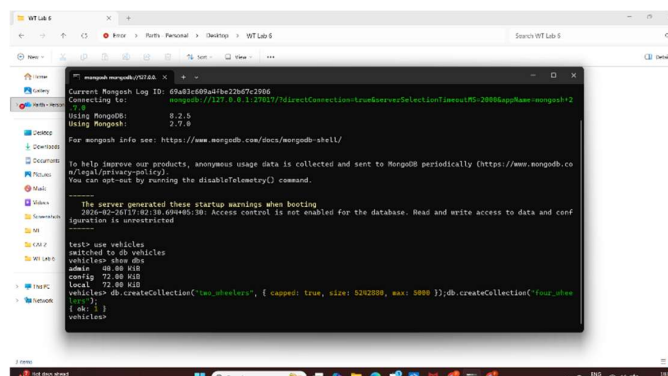
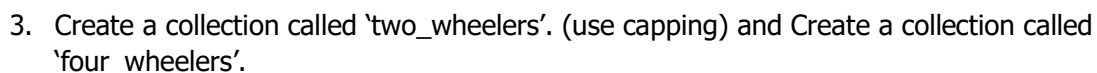
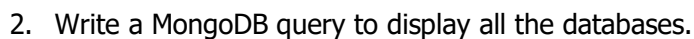


**Date:** 26-02-2026  
**School:** SCOPE  
**Reg. no.:** 23BCE7353

1. Use MongoDB to implement the following DB operations
  1. Create a database called 'vehicles' and *write* a MongoDB query to select database as "vehicles".



## Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Date: 26-02-2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Parth Kanzariya

Reg. no.: 23BCE7353

4. Add 5 two-wheeler details to the collection named 'two\_wheelers'. Each document consists of following fields as bike\_name, model (gear or gearless), category (100cc, 125cc, 150cc, 200cc), colors\_available (red, black, blue, sport red etc) as array, manufacturer, performance (out of 10), timestamp (date and year release) and price.

```

config 72.80 MB
local 72.80 MB
vehicles> db.createCollection("two_wheelers", { capped: true, size: 5242880, max: 5000 });db.createCollection("four_whe
[ ok ]
vehicles> db.two_wheelers.insertMany([
  { bike_name: "Bajaj Pulsar 150", model: "gear", category: "150cc", colors_available: ["black", "silver"], manufacturer:
"bajaj", performance: 7, timestamp: new Date("2022-01-15"), price: 75000 },
  { bike_name: "Hero Splendor 125", model: "gearless", category: "125cc", colors_available: ["blue", "red", "white"], manufact
urer: "hero", performance: 6, timestamp: new Date("2022-05-20"), price: 52000 },
  { bike_name: "Royal Enfield Hunter 350", model: "gear", category: "180cc", colors_available:
["black", "silver"], manufacturer: "royal", performance: 7, timestamp: new Date("2022-01-15"), price: 75000 },
  { bike_name: "TVS Apache 160", model: "gear", category: "160cc", colors_available: ["red", "blue"], manufacturer: "tvs",
performance: 8, timestamp: new Date("2022-05-20"), price: 62000 },
  { bike_name: "Pulsar 180", model: "gear", category: "180cc", colors_available: ["black", "blue"], manufacturer: "bajaj",
performance: 8, timestamp: new Date("2022-05-20"), price: 62000 },
  { bike_name: "KTM Duke 200", model: "gear", category: "200cc", colors_available: ["red", "black"], man
ufacturer: "ktm", performance: 10, timestamp: new Date("2022-12-01"), price: 150000 },
  { bike_name: "Jupiter 125", model: "gearless", category: "125cc", colors_available: ["white", "blue", "purple"], manufacturer: "aps", performance: 4, timestamp:
new Date("2024-02-14"), price: 88000 }]);
acknowledged: true,
insertedIds: [
  ObjectId("60a83e9af9a3f22b67c2987"),
  ObjectId("60a83e9af9a3f22b67c2988"),
  ObjectId("60a83e9af9a3f22b67c2989"),
  ObjectId("60a83e9af9a3f22b67c298a"),
  ObjectId("60a83e9af9a3f22b67c298b")
]
vehicles>

```

5. Add 5 four-wheeler details to the collection named 'four\_wheelers'. Each document consists of following fields as vehicle\_name, model (commercial or own), category (car, lorry, bus, mini truck, heavy truck, containers), variants (vxi, zxi, petrol, diesel etc) as array, manufacturer, performance (out of 10), timestamp (date and year release) and price.

```

[ ok ]
vehicles> db.four_wheelers.insertMany([
  { vehicle_name: "Swift", model: "own", category: "car", variants: ["vxi", "zxi", "petrol"], manufacturer: "Maruti",
performance: 8, timestamp: new Date("2022-05-20"), price: 80000 },
  { vehicle_name: "Vitara", model: "commercial", category: "mini truck", variants: ["diesel", "cng"], manufacturer: "Ta
ta", performance: 7, timestamp: new Date("2022-11-20"), price: 1800000 },
  { vehicle_name: "Tata Nano", model: "commercial", category: "heavy truck", variants: ["diesel"], manufacturer: "Tat
a", performance: 9, timestamp: new Date("2021-05-15"), price: 2500000 },
  { vehicle_name: "Mahindra XUV 300", model: "own", variants: ["petrol", "diesel", "cng"], manufacturer: "Mahindr
a", performance: 9, timestamp: new Date("2023-10-26"), price: 1600000 },
  { vehicle_name: "Jaguar XE", model: "commercial", category: "bus", variants: ["diesel"], manufacturer: "Jaguar", performance: 4, timestamp: new Date("2020-05-12"), price: 2200000 }
]);
acknowledged: true,
insertedIds: [
  ObjectId("60a83e9af9a3f22b67c298c"),
  ObjectId("60a83e9af9a3f22b67c298d"),
  ObjectId("60a83e9af9a3f22b67c298e"),
  ObjectId("60a83e9af9a3f22b67c298f"),
  ObjectId("60a83e9af9a3f22b67c2990")
]
vehicles>

```

**Date:** 26-02-2026  
**School:** SCOPE  
**Reg. no.:** 23BCE7353

6. Write a MongoDB query to display all documents available in two\_wheelers and four\_wheelers.

[illegible]

7. Write a MongoDB query to display only vehicle name and price in all the collection of the database

[illegible]

8. Write a MongoDB query to display two\_wheelers from a particular company

The screenshot shows a Windows desktop with a taskbar at the bottom. The taskbar includes icons for Edge, File Explorer, and several other applications. The main window is MongoDB Compass, displaying a JSON document for a vehicle. The document is as follows:

```
{
  "category": "1601a",
  "color": "red",
  "color_available": "1601a",
  "manufacturer": "Honda",
  "performance": 1,
  "timestamp": "2022-05-27T00:00:00.000Z",
  "price": 220000
}
```

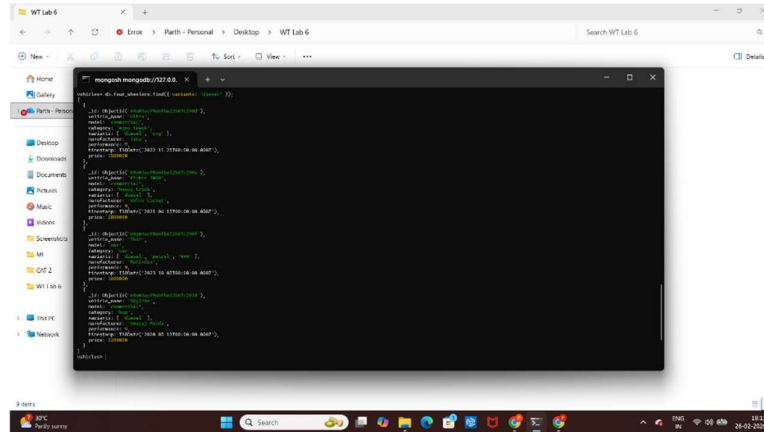
The document is part of a collection named 'vehicles' in a database named '1601a'. The document is the first in the collection, with a '\_id' of 0. The document is also part of a collection named 'vehicles' in a database named '1601a'.

Lab Sheet 6: MongoDB Basic commands

**Branch/ Class:** B.Tech/M.Tech  
**Faculty Name:** Prof. S.Gopikrishnan  
**Student name:** Parth Kanzariya

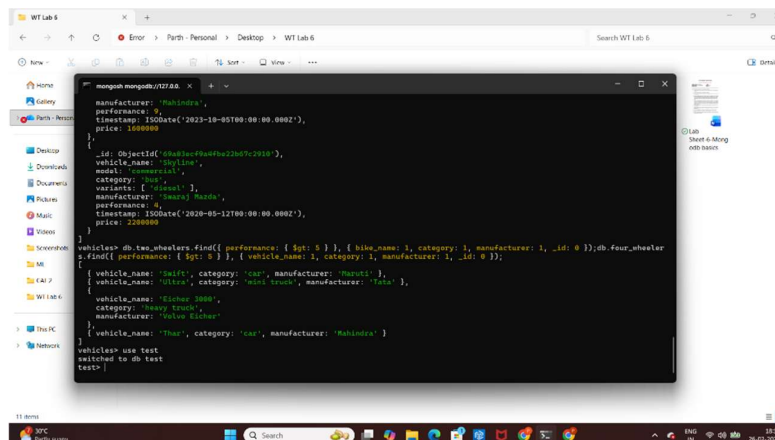
**Date:** 26-02-2026  
**School:** SCOPE  
**Reg. no.:** 23BCE7353

9. Write a MongoDB query to display four\_wheelers available in diesel variants



```
mongo> use four_wheelers
switched to db four_wheelers
mongo> find(
  {
    "manufacturer": "Mahindra",
    "performance": 5,
    "timestamp": ISODate("2023-10-05T00:00:00Z"),
    "price": 100000
  },
  {
    "_id": ObjectId("60a8ba94aef8a22b67c2010"),
    "vehicle_name": "Spline",
    "model": "Commercial",
    "category": "bus",
    "manufacturer": "Mahindra",
    "performance": 5,
    "timestamp": ISODate("2023-10-05T00:00:00Z"),
    "price": 100000
  },
  {
    "_id": ObjectId("60a8ba94aef8a22b67c2011"),
    "vehicle_name": "Spline",
    "model": "Commercial",
    "category": "bus",
    "manufacturer": "Mahindra",
    "performance": 5,
    "timestamp": ISODate("2023-10-05T00:00:00Z"),
    "price": 100000
  },
  {
    "_id": ObjectId("60a8ba94aef8a22b67c2012"),
    "vehicle_name": "Spline",
    "model": "Commercial",
    "category": "bus",
    "manufacturer": "Mahindra",
    "performance": 5,
    "timestamp": ISODate("2023-10-05T00:00:00Z"),
    "price": 100000
  },
  {
    "_id": ObjectId("60a8ba94aef8a22b67c2013"),
    "vehicle_name": "Spline",
    "model": "Commercial",
    "category": "bus",
    "manufacturer": "Mahindra",
    "performance": 5,
    "timestamp": ISODate("2023-10-05T00:00:00Z"),
    "price": 100000
  }
)
```

10. Write a MongoDB query to display vehicles name, category and manufacturer details whose rating is more than 5.



```
mongo> use four_wheelers
switched to db four_wheelers
mongo> find(
  {
    "manufacturer": "Mahindra",
    "performance": 5,
    "timestamp": ISODate("2023-10-05T00:00:00Z"),
    "price": 100000
  },
  {
    "_id": ObjectId("60a8ba94aef8a22b67c2010"),
    "vehicle_name": "Spline",
    "model": "Commercial",
    "category": "bus",
    "manufacturer": "Mahindra",
    "performance": 5,
    "timestamp": ISODate("2023-10-05T00:00:00Z"),
    "price": 100000
  },
  {
    "_id": ObjectId("60a8ba94aef8a22b67c2011"),
    "vehicle_name": "Spline",
    "model": "Commercial",
    "category": "bus",
    "manufacturer": "Mahindra",
    "performance": 5,
    "timestamp": ISODate("2023-10-05T00:00:00Z"),
    "price": 100000
  },
  {
    "_id": ObjectId("60a8ba94aef8a22b67c2012"),
    "vehicle_name": "Spline",
    "model": "Commercial",
    "category": "bus",
    "manufacturer": "Mahindra",
    "performance": 5,
    "timestamp": ISODate("2023-10-05T00:00:00Z"),
    "price": 100000
  },
  {
    "_id": ObjectId("60a8ba94aef8a22b67c2013"),
    "vehicle_name": "Spline",
    "model": "Commercial",
    "category": "bus",
    "manufacturer": "Mahindra",
    "performance": 5,
    "timestamp": ISODate("2023-10-05T00:00:00Z"),
    "price": 100000
  }
)
```

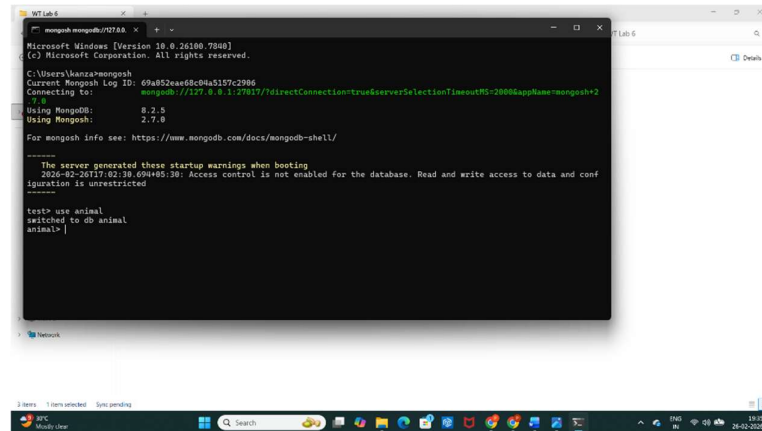
Lab Sheet 6: MongoDB Basic commands

**Branch/ Class:** B.Tech/M.Tech  
**Faculty Name:** Prof. S.Gopikrishnan  
**Student name:** Parth Kanzariya

**Date:** 26-02-2026  
**School:** SCOPE  
**Reg. no.:** 23BCE7353

2. Use MongoDB to implement the following DB operations for a Zoo

1. Create a database called 'animal' and *write* a MongoDB query to select database as 'animal'.



```
Microsoft Windows [Version 10.0.26100.7890]
(c) Microsoft Corporation. All rights reserved.

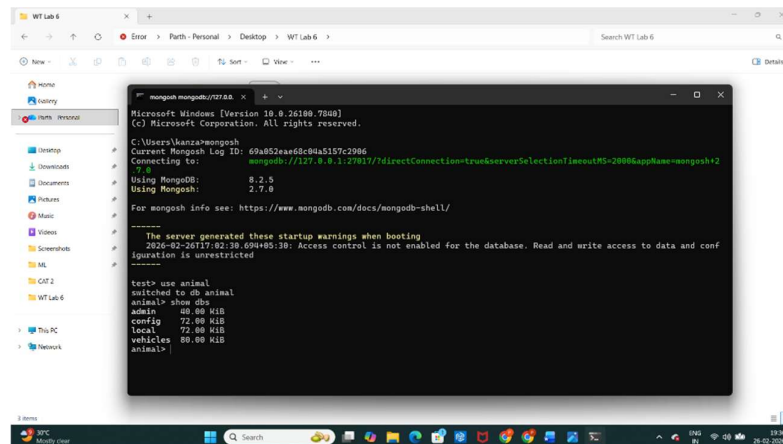
C:\Users\kanza>mongosh
Current MongoDB Log ID: 69a852eae68c04a5157c2986
Connecting to: mongod://127.0.0.1:27027/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Using MongoDB: 8.2.5
Using Mongosh: 2.7.0

For mongosh info see: https://www.mongodb.com/docs/mongosh-shell/

-----
The server generated these startup warnings when booting
2026-02-26T17:02:30.694+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> use animal
switched to db animal
animal>
```

2. Write a MongoDB query to display all the databases.



```
Microsoft Windows [Version 10.0.26100.7890]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kanza>mongosh
Current MongoDB Log ID: 69a852eae68c04a5157c2986
Connecting to: mongod://127.0.0.1:27027/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Using MongoDB: 8.2.5
Using Mongosh: 2.7.0

For mongosh info see: https://www.mongodb.com/docs/mongosh-shell/

-----
The server generated these startup warnings when booting
2026-02-26T17:02:30.694+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

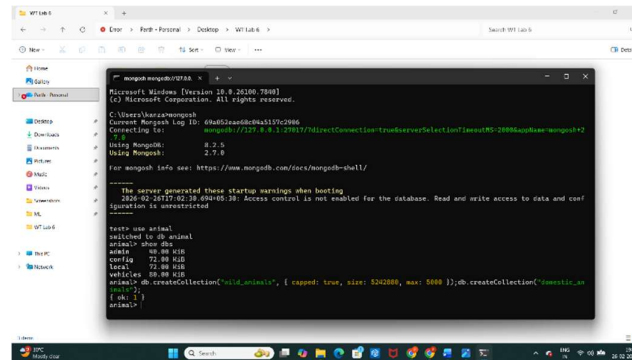
test> use animal
switched to db animal
animal> show dbs
admin    48.00 KiB
config  72.00 KiB
local   72.00 KiB
vehicles 80.00 KiB
animal>
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech  
Faculty Name: Prof. S.Gopikrishnan  
Student name: Parth Kanzariya

Date: 26-02-2026  
School: SCOPE  
Reg. no.: 23BCE7353

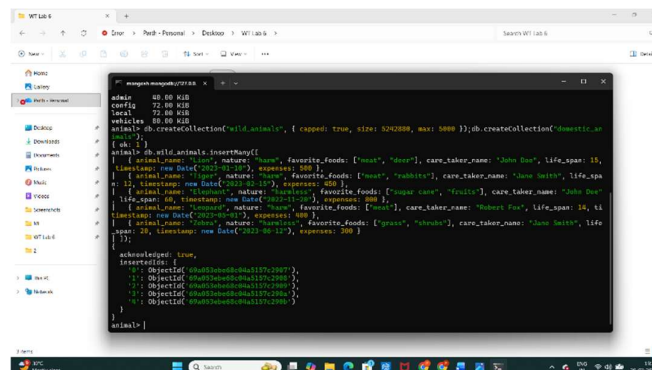
3. Create a collection called 'wild\_animals'.(use capping) and Create a collection called 'domestic\_animals'.



```
Microsoft Windows [Version 10.0.26100.7800]
(c) Microsoft Corporation. All rights reserved.

C:\Users\harsh>mongo
MongoDB Shell
> use animal
switched to db animal
> createCollection('wild_animals', { capped: true, size: 5242880, max: 5000 })
{
  "ok" : 1
}
```

4. Add 5 wild\_animal details to the collection named 'wild\_animals'. Each document consists of following fields as animal\_name, nature (harm or harmless), favorite\_foods (meat, rabbits, deer etc) as array, care\_taker\_name, life span (in years), timestamp (when the animal registered at the Zoo) and expenses.



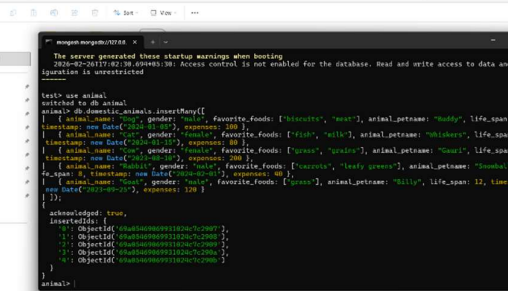
```
Microsoft Windows [Version 10.0.26100.7800]
(c) Microsoft Corporation. All rights reserved.

C:\Users\harsh>mongo
MongoDB Shell
> use animal
switched to db animal
> insertMany()
{
  "insertedCount" : 5,
  "upsertedCount" : 0,
  "upsertedIds" : {},
  "insertedIds" : [
    ObjectId("60b1b0b0b0b0b0b0b0b0b0b0"),
    ObjectId("60b1b0b0b0b0b0b0b0b0b0b1"),
    ObjectId("60b1b0b0b0b0b0b0b0b0b0b2"),
    ObjectId("60b1b0b0b0b0b0b0b0b0b0b3"),
    ObjectId("60b1b0b0b0b0b0b0b0b0b0b4")
  ]
}
```

**Date:** 26-02-2026

**School:** SCOPE

**Reg. no.:** 23BCE7353

- 
- The screenshot shows a Windows 10 desktop environment. In the background, a web browser window is open, displaying a search result for "WT Lab 6" from a site named "Search WT Lab 6". The browser's address bar shows the URL "http://localhost:3000/".
- In the foreground, a terminal window titled "mongodb-mongoose@WTLab6" is open. It displays the output of a command to generate startup warnings for a database. The output shows a list of animals with their IDs, names, genders, favorite foods, and lifespans. The animals are:
- Animal ID: 1, Name: "Maddy", Gender: "female", Favorite Foods: ["fish", "milk"], Life Span: 33, Timestamp: 2020-09-20T00:00:00.000Z
  - Animal ID: 2, Name: "Whiskers", Gender: "male", Favorite Foods: ["fish", "milk"], Life Span: 30, Timestamp: 2020-09-20T00:00:00.000Z
  - Animal ID: 3, Name: "Gracie", Gender: "female", Favorite Foods: ["milk", "grain"], Life Span: 28, Timestamp: 2020-09-20T00:00:00.000Z
  - Animal ID: 4, Name: "Greenie", Gender: "male", Favorite Foods: ["milk", "green"], Life Span: 32, Timestamp: 2020-09-20T00:00:00.000Z
  - Animal ID: 5, Name: "Billy", Gender: "male", Favorite Foods: ["milk"], Life Span: 12, Timestamp: 2020-09-20T00:00:00.000Z
- The terminal window also shows a message indicating that the server generated these startup warnings when booting, and that the database is not enabled for the database. The terminal window is titled "mongodb-mongoose@WTLab6" and the command being executed is "node server.js".

- [illegible]

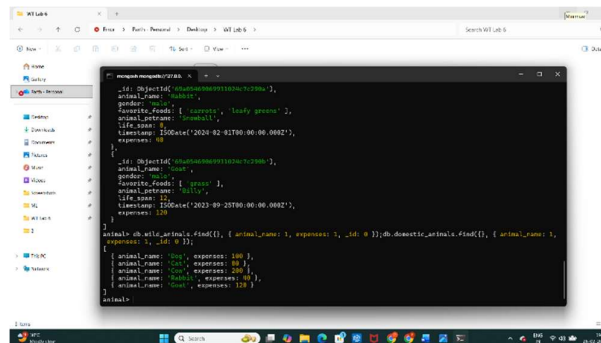


Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech  
Faculty Name: Prof. S.Gopikrishnan  
Student name: Parth Kanzariya

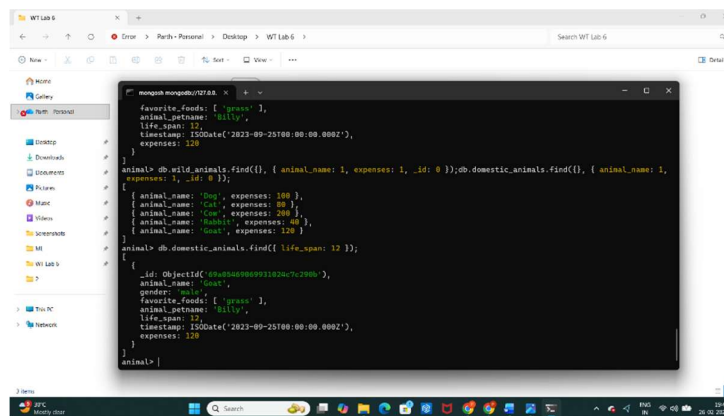
Date: 26-02-2026  
School: SCOPE  
Reg. no.: 23BCE7353

7. Write a MongoDB query to display only animal name and expenses in all the collection of the database



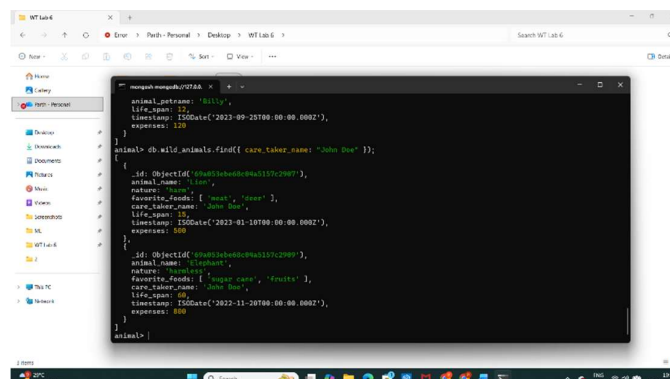
```
use animals
db.getCollectionNames()
[ 'animals', 'wild_animals', 'domestic_animals' ]
db.animals.find( { animal_name: 1, expenses: 1, _id: 0 });
db.wild_animals.find( { animal_name: 1, expenses: 1, _id: 0 });
db.domestic_animals.find( { animal_name: 1, expenses: 1, _id: 0 });
```

8. Write a MongoDB query to display domestic\_animals whose life is a particular year



```
use animals
db.getCollectionNames()
[ 'animals', 'wild_animals', 'domestic_animals' ]
db.animals.find( { animal_name: 1, expenses: 1, _id: 0 });
db.wild_animals.find( { animal_name: 1, expenses: 1, _id: 0 });
db.domestic_animals.find( { animal_name: 1, expenses: 1, _id: 0 });
db.animals.find( { life_span: 12 });
```

9. Write a MongoDB query to display wild\_animals available under a particular care\_taker



```
use animals
db.getCollectionNames()
[ 'animals', 'wild_animals', 'domestic_animals' ]
db.animals.find( { care_taker_name: 'John Doe' });
```



Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Date: 26-02-2026

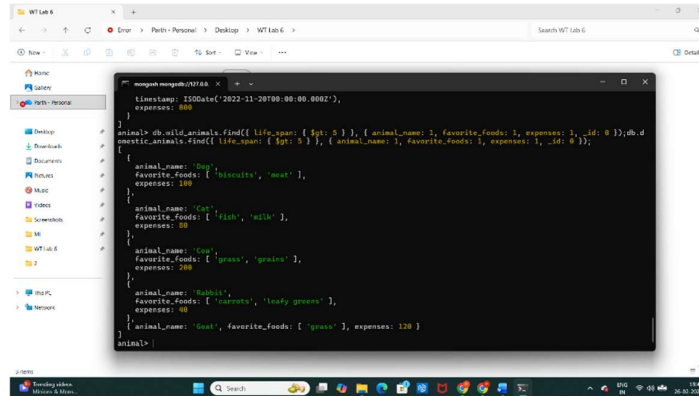
Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Parth Kanzariya

Reg. no.: 23BCE7353

10. Write a MongoDB query to display animal name, favorite\_foods and expenses details whose lifespan is more than 5 years.



```
mongo> use animals
animals> db.animals.find({lifespan: { $gt: 5 } }, { animal_name: 1, favorite_foods: 1, expenses: 1, _id: 0 });
{
  animal_name: "Cat",
  favorite_foods: [ "hiscuit", "meat" ],
  expenses: 100
},
{
  animal_name: "Cat",
  favorite_foods: [ "fish", "milk" ],
  expenses: 80
},
{
  animal_name: "Cat",
  favorite_foods: [ "grass", "grains" ],
  expenses: 100
},
{
  animal_name: "Rabbit",
  favorite_foods: [ "bushes", "leafy greens" ],
  expenses: 40
},
{
  animal_name: "Goat",
  favorite_foods: [ "grass" ],
  expenses: 120
}
animals>
```