

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Date: 26-02-2026

Faculty Name: Prof. S.Gopikrishnan

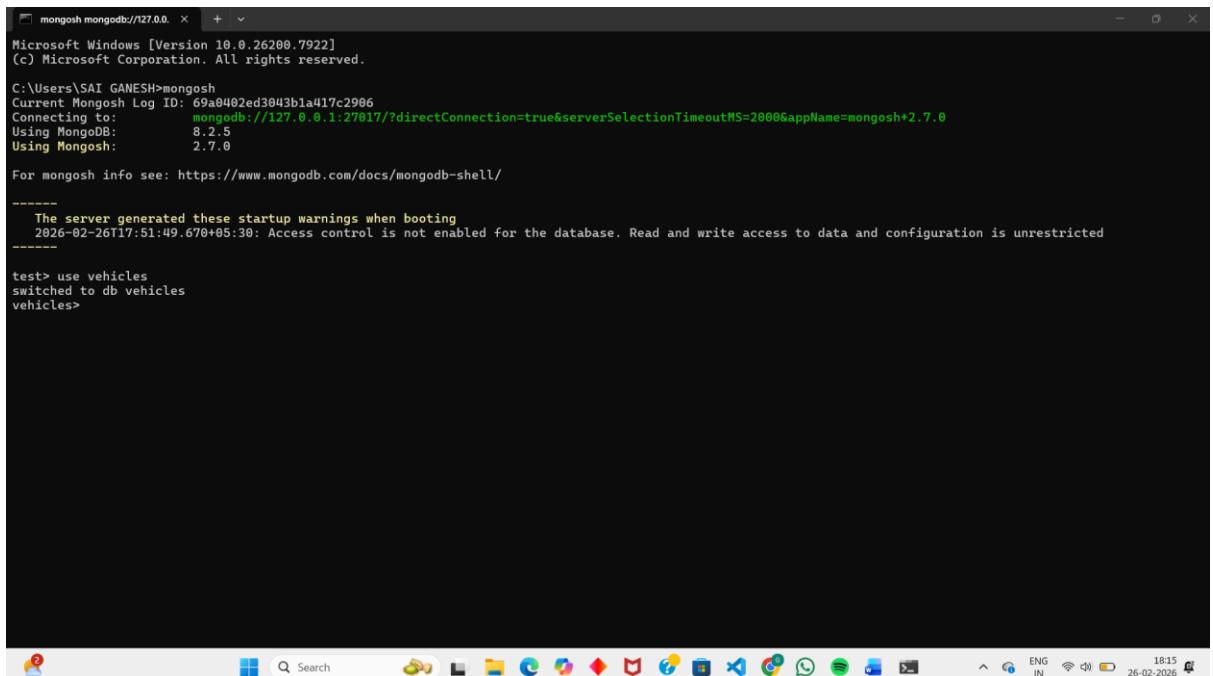
School: SCOPE

Student name: M.V.S.K GANESH

Reg. no.: 23BCE8536

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”.



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Microsoft Windows [Version 10.0.26200.7922]
(c) Microsoft Corporation. All rights reserved.

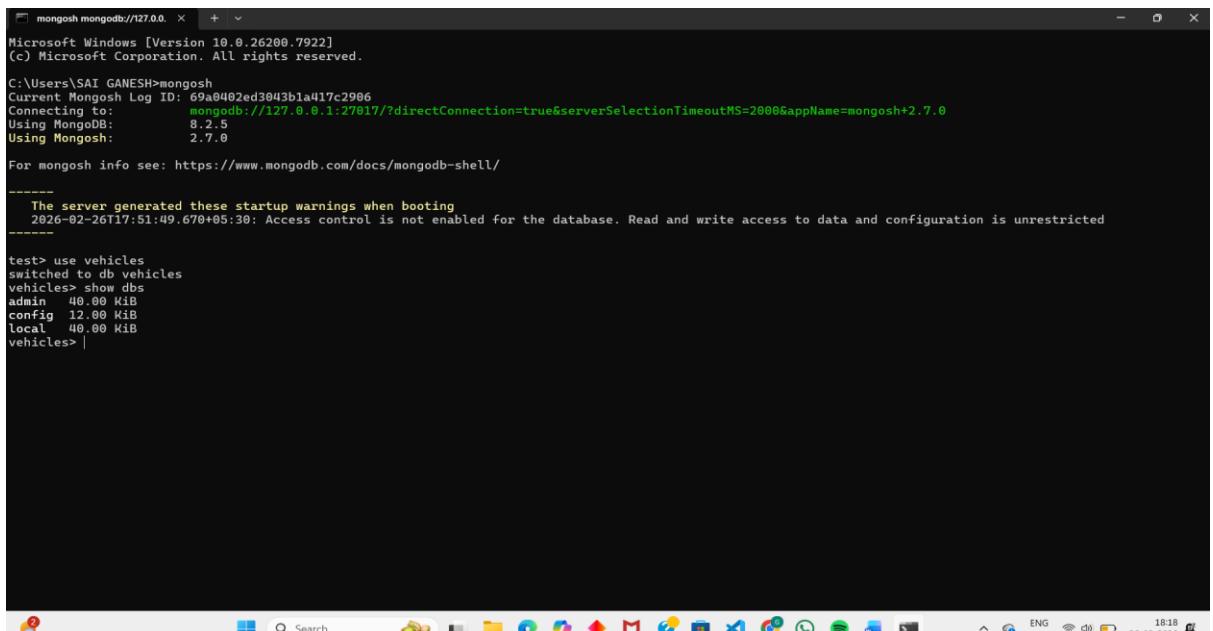
C:\Users\SAI GANESH>mongosh
Current Mongosh Log ID: 69a0402ed3043b1a417c2906
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Using MongoDB:      8.2.5
Using Mongosh:      2.7.0

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

-----
The server generated these startup warnings when booting
2026-02-26T17:51:49.670+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> use vehicles
switched to db vehicles
vehicles>
```

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



```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
Microsoft Windows [Version 10.0.26200.7922]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SAI GANESH>mongosh
Current Mongosh Log ID: 69a0402ed3043b1a417c2906
Connecting to:      mongodb://127.0.0.1:27017/?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/mongodb-shell/

-----
The server generated these startup warnings when booting
2026-02-26T17:51:49.670+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

test> use vehicles
switched to db vehicles
vehicles> show dbs
admin 40.00 KiB
config 12.00 KiB
local 40.00 KiB
vehicles|
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Date: 26-02-2026

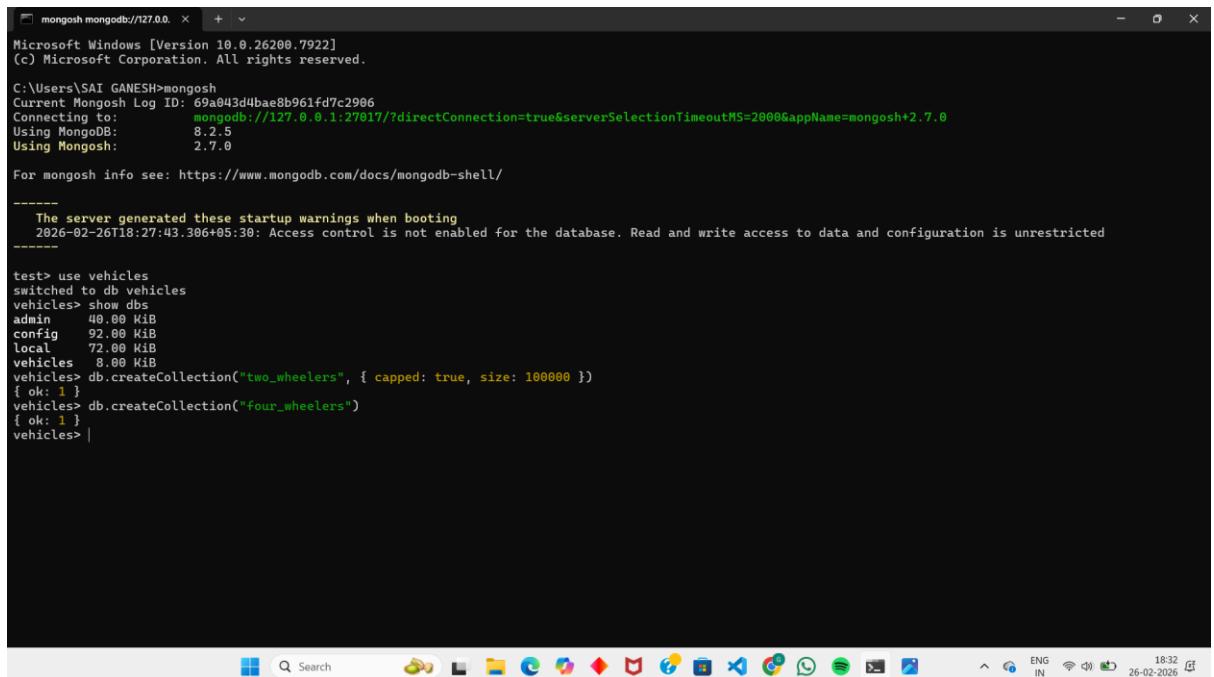
Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: M.V.S.K GANESH

Reg. no.: 23BCE8536

3. Create a collection called 'two_wHEELERS'. (use capping) and Create a collection called 'four_wHEELERS'.



The screenshot shows a Windows desktop environment with a terminal window open. The terminal window title is "mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0". The terminal content shows the following MongoDB shell session:

```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.7.0
C:\Users\SAI GANESH>mongosh
Current Mongosh Log ID: 69ab043d4bae88961fd7c2906
(c) Microsoft Corporation. All rights reserved.

Using Mongosh: 2.7.0
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
-----
The server generated these startup warnings when booting
2026-02-26T18:27:43.306+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
test> use vehicles
switched to db vehicles
vehicles> show dbs
admin      40.00 KiB
config     92.00 KiB
local      72.00 KiB
vehicles    8.00 KiB
vehicles> db.createCollection("two_wHEELERS", { capped: true, size: 100000 })
{ ok: 1 }
vehicles> db.createCollection("four_wHEELERS")
{ ok: 1 }
vehicles> |
```

The taskbar at the bottom of the screen shows various application icons, and the system tray indicates the date and time as 26-02-2026 18:32.

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: M.V.S.K GANESH

Reg. no.: 23BCE8536

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.

```

mongosh mongoDB://127.0.0.1:27017
vehicles> db.two_wheelers.insertMany([
  {
    bike_name: "Shine",
    model: "gear",
    category: "125cc",
    colors_available: ["red", "black", "blue"],
    manufacturer: "Honda",
    performance: 9,
    timestamp: new Date("2022-03-10"),
    price: 60000
  },
  {
    bike_name: "Activa",
    model: "gearless",
    category: "110cc",
    colors_available: ["white", "grey"],
    manufacturer: "Honda",
    performance: 8,
    timestamp: new Date("2021-06-15"),
    price: 80000
  },
  {
    bike_name: "Pulsar 150",
    model: "gear",
    category: "150cc",
    colors_available: ["black", "sport red"],
    manufacturer: "Bajaj",
    performance: 9,
    timestamp: new Date("2023-01-20"),
    price: 120000
  },
  {
    bike_name: "Apache RTR 160",
    model: "gear",
    category: "160cc",
    colors_available: ["blue", "black"],
    manufacturer: "TVS",
    performance: 8,
    timestamp: new Date("2022-09-05"),
    price: 125000
  },
  {
    bike_name: "Jupiter",
    model: "gearless",
    category: "110cc",
    colors_available: ["silver", "brown"],
    manufacturer: "TVS",
    performance: 6,
    timestamp: new Date("2020-11-11"),
    price: 90000
  }
])
{
  acknowledged: true,
  insertedIds: [
    '5f9a04581bae8b961fd7c2907',
    '5f9a04581bae8b961fd7c2908',
    '5f9a04581bae8b961fd7c2909',
    '5f9a04581bae8b961fd7c290a',
    '5f9a04581bae8b961fd7c290b'
  ]
}
vehicles>

```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Date: 26-02-2026

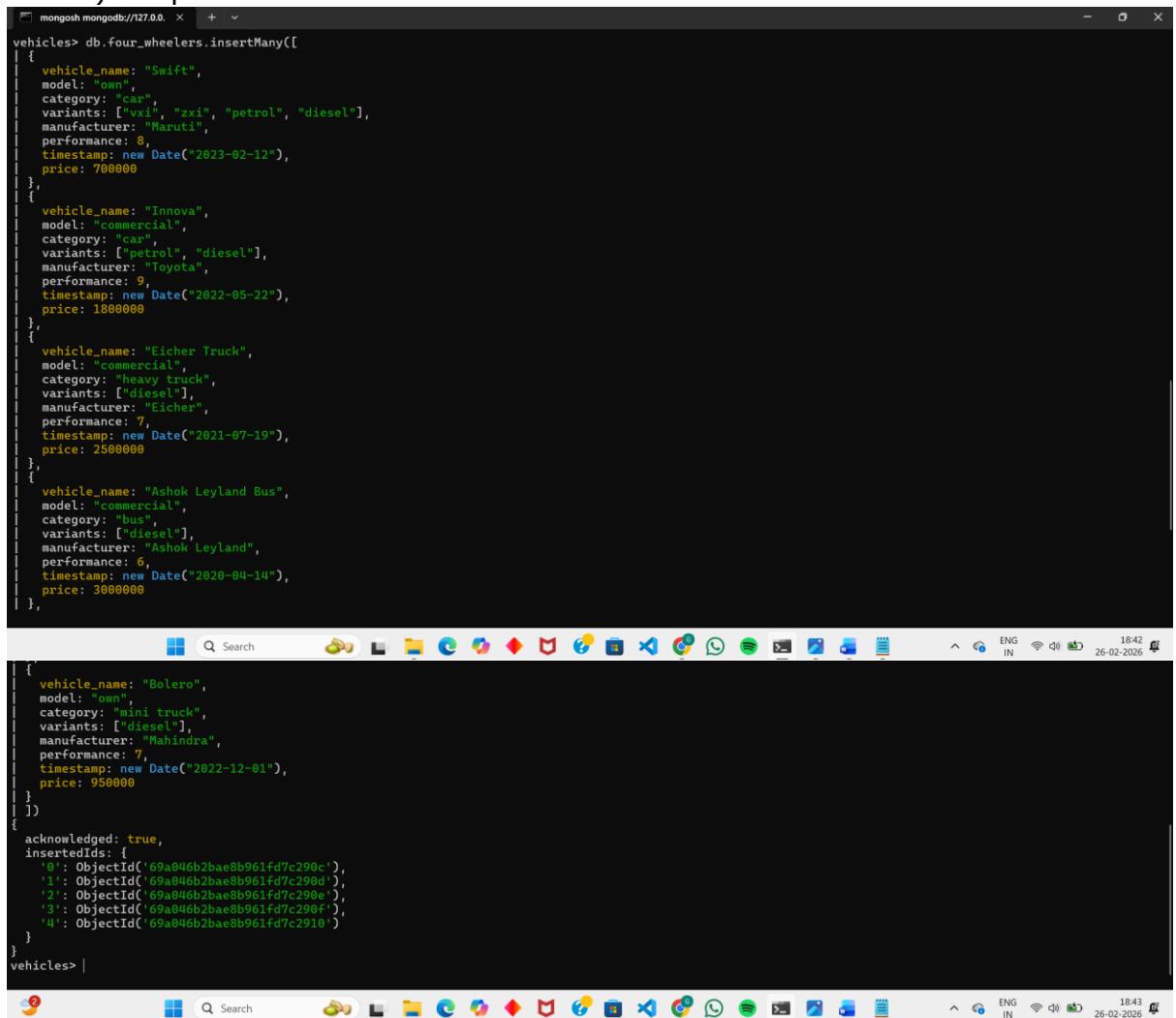
Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: M.V.S.K GANESH

Reg. no.: 23BCE8536

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.



```

mongosh mongodb://127.0.0.1:27017
{
  vehicle_name: "Swift",
  model: "own",
  category: "car",
  variants: ["vxi", "zxi", "petrol", "diesel"],
  manufacturer: "Maruti",
  performance: 8,
  timestamp: new Date("2023-02-12"),
  price: 700000
},
{
  vehicle_name: "Innova",
  model: "commercial",
  category: "car",
  variants: ["petrol", "diesel"],
  manufacturer: "Toyota",
  performance: 9,
  timestamp: new Date("2022-05-22"),
  price: 1800000
},
{
  vehicle_name: "Eicher Truck",
  model: "commercial",
  category: "heavy truck",
  variants: ["diesel"],
  manufacturer: "Eicher",
  performance: 7,
  timestamp: new Date("2021-07-19"),
  price: 2500000
},
{
  vehicle_name: "Ashok Leyland Bus",
  model: "commercial",
  category: "bus",
  variants: ["diesel"],
  manufacturer: "Ashok Leyland",
  performance: 6,
  timestamp: new Date("2020-04-14"),
  price: 3000000
},
{
  vehicle_name: "Bolero",
  model: "own",
  category: "mini truck",
  variants: ["diesel"],
  manufacturer: "Mahindra",
  performance: 7,
  timestamp: new Date("2022-12-01"),
  price: 950000
}
]

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('69a046b2bae8b961fd7c290c'),
    '1': ObjectId('69a046b2bae8b961fd7c290d'),
    '2': ObjectId('69a046b2bae8b961fd7c290e'),
    '3': ObjectId('69a046b2bae8b961fd7c290f'),
    '4': ObjectId('69a046b2bae8b961fd7c2910')
  }
}

vehicles> |
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Date: 26-02-2026

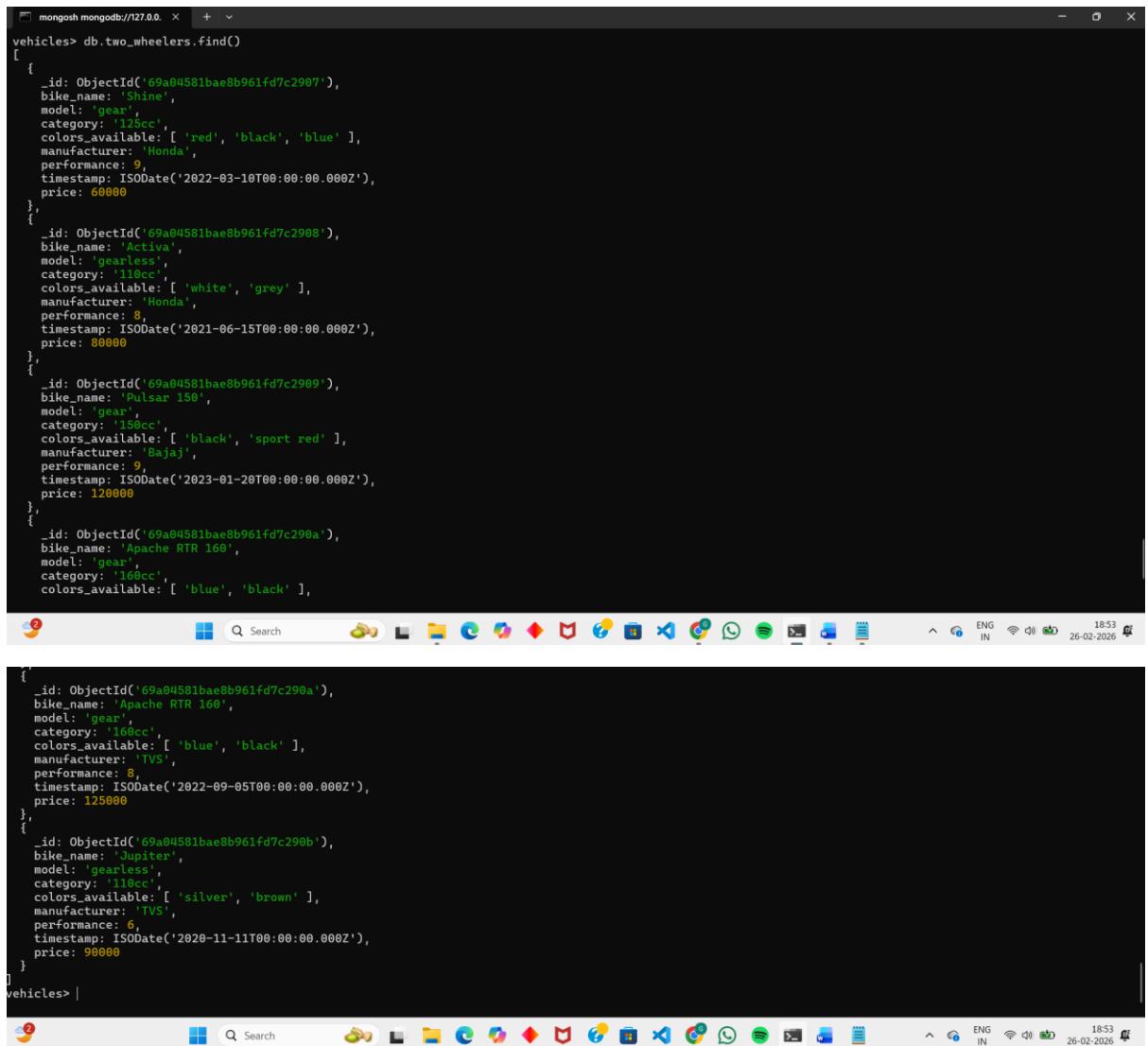
Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: M.V.S.K GANESH

Reg. no.: 23BCE8536

6. Write a MongoDB query to display all documents available in two_wheelers and four_wheelers.



```
mongosh mongodb://127.0.0.1:27017
db.two_wHEELERS.find()
[{
  _id: ObjectId('69a04581bae8b961fd7c2907'),
  bike_name: 'Shine',
  model: 'gear',
  category: '125cc',
  colors_available: [ 'red', 'black', 'blue' ],
  manufacturer: 'Honda',
  performance: 9,
  timestamp: ISODate('2022-03-10T00:00:00.000Z'),
  price: 60000
},
{
  _id: ObjectId('69a04581bae8b961fd7c2908'),
  bike_name: 'Activa',
  model: 'gearless',
  category: '110cc',
  colors_available: [ 'white', 'grey' ],
  manufacturer: 'Honda',
  performance: 8,
  timestamp: ISODate('2021-06-15T00:00:00.000Z'),
  price: 80000
},
{
  _id: ObjectId('69a04581bae8b961fd7c2909'),
  bike_name: 'Pulsar 150',
  model: 'gear',
  category: '150cc',
  colors_available: [ 'black', 'sport red' ],
  manufacturer: 'Bajaj',
  performance: 9,
  timestamp: ISODate('2023-01-20T00:00:00.000Z'),
  price: 120000
},
{
  _id: ObjectId('69a04581bae8b961fd7c290a'),
  bike_name: 'Apache RTR 160',
  model: 'gear',
  category: '160cc',
  colors_available: [ 'blue', 'black' ],
  manufacturer: 'TVS',
  performance: 8,
  timestamp: ISODate('2022-09-05T00:00:00.000Z'),
  price: 125000
},
{
  _id: ObjectId('69a04581bae8b961fd7c290b'),
  bike_name: 'Jupiter',
  model: 'gearless',
  category: '110cc',
  colors_available: [ 'silver', 'brown' ],
  manufacturer: 'TVS',
  performance: 6,
  timestamp: ISODate('2020-11-11T00:00:00.000Z'),
  price: 90000
}
]
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: M.V.S.K GANESH

Reg. no.: 23BCE8536

```
vehicles> db.four_wheelers.find()
[
  {
    _id: ObjectId('69a046b2bae8b961fd7c290c'),
    vehicle_name: 'Swift',
    model: 'own',
    category: 'car',
    variants: [ 'xi', 'xsi', 'petrol', 'diesel' ],
    manufacturer: 'Maruti',
    performance: 8,
    timestamp: ISODate('2023-02-12T00:00:00.000Z'),
    price: 700000
  },
  {
    _id: ObjectId('69a046b2bae8b961fd7c290d'),
    vehicle_name: 'Innova',
    model: 'commercial',
    category: 'car',
    variants: [ 'petrol', 'diesel' ],
    manufacturer: 'Toyota',
    performance: 9,
    timestamp: ISODate('2022-05-22T00:00:00.000Z'),
    price: 1800000
  },
  {
    _id: ObjectId('69a046b2bae8b961fd7c290e'),
    vehicle_name: 'Eicher Truck',
    model: 'commercial',
    category: 'heavy truck',
    variants: [ 'diesel' ],
    manufacturer: 'Eicher',
    performance: 7,
    timestamp: ISODate('2021-07-19T00:00:00.000Z'),
    price: 2500000
  },
  {
    _id: ObjectId('69a046b2bae8b961fd7c290f'),
    vehicle_name: 'Ashok Leyland Bus',
    model: 'commercial',
    category: 'bus',
    variants: [ 'diesel' ],
    manufacturer: 'Ashok Leyland',
    performance: 6,
    timestamp: ISODate('2020-04-14T00:00:00.000Z'),
    price: 3000000
  }
]
```

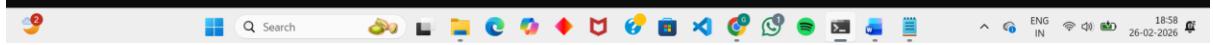


```
{
  _id: ObjectId('69a046b2bae8b961fd7c290f'),
  vehicle_name: 'Ashok Leyland Bus',
  model: 'commercial',
  category: 'bus',
  variants: [ 'diesel' ],
  manufacturer: 'Ashok Leyland',
  performance: 6,
  timestamp: ISODate('2020-04-14T00:00:00.000Z'),
  price: 3000000
},
{
  _id: ObjectId('69a046b2bae8b961fd7c2910'),
  vehicle_name: 'Bolero',
  model: 'own',
  category: 'mini truck',
  variants: [ 'diesel' ],
  manufacturer: 'Mahindra',
  performance: 7,
  timestamp: ISODate('2022-12-01T00:00:00.000Z'),
  price: 950000
}
]
vehicles> |
```



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

```
vehicles> db.two_wheelers.find({}, { bike_name: 1, price: 1, _id: 0 })
[ { bike_name: 'Shine', price: 60000 },
  { bike_name: 'Activa', price: 80000 },
  { bike_name: 'Pulsar 150', price: 120000 },
  { bike_name: 'Apache RTR 160', price: 125000 },
  { bike_name: 'Dupiter', price: 90000 }
]
vehicles> db.four_wheelers.find({}, { vehicle_name: 1, price: 1, _id: 0 })
[ { vehicle_name: 'Swift', price: 700000 },
  { vehicle_name: 'Innova', price: 1800000 },
  { vehicle_name: 'Eicher Truck', price: 2500000 },
  { vehicle_name: 'Ashok Leyland Bus', price: 3000000 },
  { vehicle_name: 'Bolero', price: 950000 }
]
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: M.V.S.K GANESH

Reg. no.: 23BCE8536

8. Write a MongoDB query to display two_wheelers from a particular company

```
vehicles> db.two_wheelers.find({ manufacturer: "Honda" })
[ {
    _id: ObjectId('69a04581bae8b961fd7c2907'),
    bike_name: 'Shine',
    model: 'gear',
    category: '125cc',
    colors_available: [ 'red', 'black', 'blue' ],
    manufacturer: 'Honda',
    performance: 9,
    timestamp: ISODate('2022-03-10T00:00:00.000Z'),
    price: 60000
},
{
    _id: ObjectId('69a04581bae8b961fd7c2908'),
    bike_name: 'Activa',
    model: 'gearless',
    category: '110cc',
    colors_available: [ 'white', 'grey' ],
    manufacturer: 'Honda',
    performance: 8,
    timestamp: ISODate('2021-06-15T00:00:00.000Z'),
    price: 80000
}
]
vehicles> |
```



9. Write a MongoDB query to display four_wheelers available in diesel variants

```
vehicles> db.four_wheelers.find({ variants: "diesel" })
[ {
    _id: ObjectId('69a046b2bae8b961fd7c290c'),
    vehicle_name: 'Swift',
    model: 'own',
    category: 'car',
    variants: [ 'vxi', 'xsi', 'petrol', 'diesel' ],
    manufacturer: 'Maruti',
    performance: 8,
    timestamp: ISODate('2023-02-12T00:00:00.000Z'),
    price: 70000
},
{
    _id: ObjectId('69a046b2bae8b961fd7c290d'),
    vehicle_name: 'Innova',
    model: 'commercial',
    category: 'car',
    variants: [ 'petrol', 'diesel' ],
    manufacturer: 'Toyota',
    performance: 9,
    timestamp: ISODate('2022-05-22T00:00:00.000Z'),
    price: 1800000
},
{
    _id: ObjectId('69a046b2bae8b961fd7c290e'),
    vehicle_name: 'Eicher Truck',
    model: 'commercial',
    category: 'heavy truck',
    variants: [ 'diesel' ],
    manufacturer: 'Eicher',
    performance: 7,
    timestamp: ISODate('2021-07-19T00:00:00.000Z'),
    price: 2500000
},
{
    _id: ObjectId('69a046b2bae8b961fd7c290f'),
    vehicle_name: 'Ashok Leyland Bus',
    model: 'commercial',
    category: 'bus',
    variants: [ 'diesel' ],
    manufacturer: 'Ashok Leyland',
    performance: 10,
    timestamp: ISODate('2022-06-15T00:00:00.000Z'),
    price: 1500000
}
]
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: M.V.S.K GANESH

Reg. no.: 23BCE8536

```
{
  "_id": ObjectId('69a046b2bae8b961fd7c290f'),
  "vehicle_name": 'Ashok Leyland Bus',
  "model": 'commercial',
  "category": 'bus',
  "variants": [ 'diesel' ],
  "manufacturer": 'Ashok Leyland',
  "performance": 6,
  "timestamp": ISODate('2020-04-14T00:00:00.000Z'),
  "price": 300000
},
{
  "_id": ObjectId('69a046b2bae8b961fd7c2910'),
  "vehicle_name": 'Bolero',
  "model": 'own',
  "category": 'mini truck',
  "variants": [ 'diesel' ],
  "manufacturer": 'Mahindra',
  "performance": 7,
  "timestamp": ISODate('2022-12-01T00:00:00.000Z'),
  "price": 950000
}
| vehicles> |
```



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

```
vehicles> db.two_wheelers.find(
|   { performance: { $gt: 5 } },
|   { bike_name: 1, category: 1, manufacturer: 1, _id: 0 }
| )
[
  { bike_name: 'Shine', category: '125cc', manufacturer: 'Honda' },
  { bike_name: 'Activa', category: '110cc', manufacturer: 'Honda' },
  { bike_name: 'Pulsar 150', category: '150cc', manufacturer: 'Bajaj' },
  { bike_name: 'Apache RTR 160', category: '160cc', manufacturer: 'TVS' },
  { bike_name: 'Jupiter', category: '110cc', manufacturer: 'TVS' }
]
vehicles> db.four_wheelers.find(
|   { performance: { $gt: 5 } },
|   { vehicle_name: 1, category: 1, manufacturer: 1, _id: 0 }
| )
[
  { vehicle_name: 'Swift', category: 'car', manufacturer: 'Maruti' },
  { vehicle_name: 'Innova', category: 'car', manufacturer: 'Toyota' },
  { vehicle_name: 'Eicher Truck', category: 'heavy truck', manufacturer: 'Eicher' },
  {
    vehicle_name: 'Ashok Leyland Bus',
    category: 'bus',
    manufacturer: 'Ashok Leyland'
  },
  {
    vehicle_name: 'Bolero',
    category: 'mini truck',
    manufacturer: 'Mahindra'
  }
]
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: M.V.S.K GANESH

Reg. no.: 23BCE8536

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'.

```
vehicles> use animal
switched to db animal
animal> |
```



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

```
vehicles> use animal
switched to db animal
animal> show dbs
admin   40.00 KiB
config  72.00 KiB
local   72.00 KiB
vehicles 80.00 KiB
animal> |
```



3. Create a collection called 'wild_animals'.(use capping) and Create a collection called 'domestic_animals'.

```
vehicles> use animal
switched to db animal
animal> show dbs
admin   40.00 KiB
config  72.00 KiB
local   72.00 KiB
vehicles 80.00 KiB
animal> db.createCollection("wild_animals", { capped: true, size: 100000 })
{ ok: 1 }
animal> db.createCollection("domestic_animals")
{ ok: 1 }
animal> |
```



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: M.V.S.K GANESH

Reg. no.: 23BCE8536

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.

```
animal> db.wild_animals.insertMany([
  {
    animal_name: "Lion",
    nature: "harm",
    favorite_foods: ["meat", "deer"],
    care_taker_name: "Ramesh",
    life_span: 10,
    timestamp: new Date("2022-01-01"),
    expenses: 50000
  },
  {
    animal_name: "Tiger",
    nature: "harm",
    favorite_foods: ["meat", "rabbits"],
    care_taker_name: "Suresh",
    life_span: 10,
    timestamp: new Date("2021-02-10"),
    expenses: 60000
  },
  {
    animal_name: "Elephant",
    nature: "harmless",
    favorite_foods: ["grass", "fruits"],
    care_taker_name: "Mahesh",
    life_span: 60,
    timestamp: new Date("2020-05-15"),
    expenses: 80000
  },
  {
    animal_name: "Bear",
    nature: "harm",
    favorite_foods: ["fish", "honey"],
    care_taker_name: "Ramesh",
    life_span: 20,
    timestamp: new Date("2019-08-20"),
    expenses: 45000
  },
  {
    animal_name: "Deer",
    nature: "harmless",
    favorite_foods: ["grass"]
  }
])

20:04
ENG IN 26-02-2026
```

```
{
  animal_name: "Deer",
  nature: "harmless",
  favorite_foods: ["grass"],
  care_taker_name: "Suresh",
  life_span: 12,
  timestamp: new Date("2023-03-03"),
  expenses: 20000
}
])
{
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('69a059eabae8b961fd7c2911'),
    '1': ObjectId('69a059eabae8b961fd7c2912'),
    '2': ObjectId('69a059eabae8b961fd7c2913'),
    '3': ObjectId('69a059eabae8b961fd7c2914'),
    '4': ObjectId('69a059eabae8b961fd7c2915')
  ]
}
animal> |
```

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: M.V.S.K GANESH

Reg. no.: 23BCE8536

5. Add 5 domestic-animal details to the collection named 'domestic_animals'. Each document consists of following fields as animal_name, gender (male or female), favorite_foods (meat, rabbits, deer etc) as array, animal_petname, life span (in years), timestamp (when the animal registered at the Zoo) and expenses.

```
animal> db.domestic_animals.insertMany([
  {
    animal_name: "Dog",
    gender: "male",
    favorite_foods: ["meat"],
    animal_petname: "Tommy",
    life_span: 13,
    timestamp: new Date("2022-07-07"),
    expenses: 10000
  },
  {
    animal_name: "Cat",
    gender: "female",
    favorite_foods: ["fish"],
    animal_petname: "Kitty",
    life_span: 15,
    timestamp: new Date("2021-06-06"),
    expenses: 8000
  },
  {
    animal_name: "Cow",
    gender: "female",
    favorite_foods: ["grass"],
    animal_petname: "Lakshmi",
    life_span: 20,
    timestamp: new Date("2020-09-09"),
    expenses: 15000
  },
  {
    animal_name: "Goat",
    gender: "male",
    favorite_foods: ["leaves"],
    animal_petname: "Raju",
    life_span: 10,
    timestamp: new Date("2023-01-15"),
    expenses: 5000
  },
  {
    animal_name: "Horse",
    gender: "male",
    favorite_foods: ["grass"],
    animal_petname: "Rani",
    life_span: 25,
    timestamp: new Date("2019-12-12"),
    expenses: 30000
  }
])
```



```
{
  "0": ObjectId('69a05a76bae8b961fd7c2916'),
  "1": ObjectId('69a05a76bae8b961fd7c2917'),
  "2": ObjectId('69a05a76bae8b961fd7c2918'),
  "3": ObjectId('69a05a76bae8b961fd7c2919'),
  "4": ObjectId('69a05a76bae8b961fd7c291a')
}
animal> |
```



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: M.V.S.K GANESH

Reg. no.: 23BCE8536

6. Write a MongoDB query to display all documents available in wild_animals and domestic_animals.

```
animal> db.wild_animals.find()
[{"_id": ObjectId('69a059eabae8b961fd7c2911'), "animal_name": "Lion", "nature": "harm", "favorite_foods": ["meat", "deer"], "care_taker_name": "Ramesh", "life_span": 14, "timestamp": ISODate('2022-01-01T00:00:00Z'), "expenses": 50000}, {"_id": ObjectId('69a059eabae8b961fd7c2912'), "animal_name": "Tiger", "nature": "harm", "favorite_foods": ["meat", "rabbits"], "care_taker_name": "Suresh", "life_span": 16, "timestamp": ISODate('2021-02-10T00:00:00Z'), "expenses": 60000}, {"_id": ObjectId('69a059eabae8b961fd7c2913'), "animal_name": "Elephant", "nature": "harmless", "favorite_foods": ["grass", "fruits"], "care_taker_name": "Mahesh", "life_span": 60, "timestamp": ISODate('2020-05-15T00:00:00Z'), "expenses": 80000}, {"_id": ObjectId('69a059eabae8b961fd7c2914'), "animal_name": "Bear", "nature": "harm", "favorite_foods": ["fish", "honey"], "care_taker_name": "Ramesh", "life_span": 20, "timestamp": ISODate('2019-08-20T00:00:00Z'), "expenses": 45000}], animal> |
```

```
{}, {"_id": ObjectId('69a059eabae8b961fd7c2914'), "animal_name": "Bear", "nature": "harm", "favorite_foods": ["fish", "honey"], "care_taker_name": "Ramesh", "life_span": 20, "timestamp": ISODate('2019-08-20T00:00:00Z'), "expenses": 45000}, {"_id": ObjectId('69a059eabae8b961fd7c2915'), "animal_name": "Deer", "nature": "harmless", "favorite_foods": ["grass"], "care_taker_name": "Suresh", "life_span": 12, "timestamp": ISODate('2023-03-03T00:00:00Z'), "expenses": 20000}], animal> |
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Date: 26-02-2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

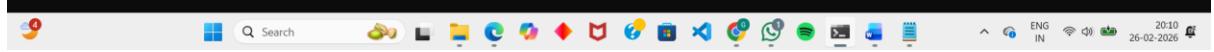
Student name: M.V.S.K GANESH

Reg. no.: 23BCE8536

```
animal> db.domestic_animals.find()
[
  {
    _id: ObjectId('69a05a76bae8b961fd7c2916'),
    animal_name: 'Dog',
    gender: 'male',
    favorite_foods: [ 'meat' ],
    animal_petsname: 'Tommy',
    life_span: 13,
    timestamp: ISODate('2022-07-07T00:00:00.000Z'),
    expenses: 10000
  },
  {
    _id: ObjectId('69a05a76bae8b961fd7c2917'),
    animal_name: 'Cat',
    gender: 'female',
    favorite_foods: [ 'fish' ],
    animal_petsname: 'Kitty',
    life_span: 15,
    timestamp: ISODate('2021-06-06T00:00:00.000Z'),
    expenses: 8000
  },
  {
    _id: ObjectId('69a05a76bae8b961fd7c2918'),
    animal_name: 'Cow',
    gender: 'female',
    favorite_foods: [ 'grass' ],
    animal_petsname: 'Lakshmi',
    life_span: 20,
    timestamp: ISODate('2020-09-09T00:00:00.000Z'),
    expenses: 15000
  },
  {
    _id: ObjectId('69a05a76bae8b961fd7c2919'),
    animal_name: 'Goat',
    gender: 'male',
    favorite_foods: [ 'leaves' ],
    animal_petsname: 'Raju',
    life_span: 10,
    timestamp: ISODate('2023-01-15T00:00:00.000Z'),
    expenses: 5000
  }
]
```

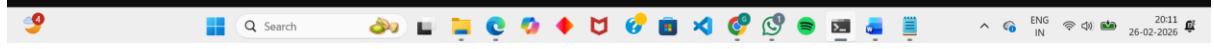


```
  },
  {
    _id: ObjectId('69a05a76bae8b961fd7c291a'),
    animal_name: 'Horse',
    gender: 'male',
    favorite_foods: [ 'grass' ],
    animal_petsname: 'Rani',
    life_span: 25,
    timestamp: ISODate('2019-12-12T00:00:00.000Z'),
    expenses: 30000
  }
]
animal> |
```



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

```
animal> db.wild_animals.find({}, { animal_name: 1, expenses: 1, _id: 0 })
[
  { animal_name: 'Lion', expenses: 50000 },
  { animal_name: 'Tiger', expenses: 60000 },
  { animal_name: 'Elephant', expenses: 80000 },
  { animal_name: 'Bear', expenses: 45000 },
  { animal_name: 'Deer', expenses: 20000 }
]
animal> db.domestic_animals.find({}, { animal_name: 1, expenses: 1, _id: 0 })
[
  { animal_name: 'Dog', expenses: 10000 },
  { animal_name: 'Cat', expenses: 8000 },
  { animal_name: 'Cow', expenses: 15000 },
  { animal_name: 'Goat', expenses: 5000 },
  { animal_name: 'Horse', expenses: 30000 }
]
animal> |
```



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: M.V.S.K GANESH

Reg. no.: 23BCE8536

8. Write a MongoDB query to display domestic_animals whose life is a particular year

```
animal> db.domestic_animals.find({ life_span: 15 })
[ {
    _id: ObjectId('69a05a76bae8b961fd7c2917'),
    animal_name: 'Cat',
    gender: 'female',
    favorite_foods: [ 'fish' ],
    animal_pname: 'Kitty',
    life_span: 15,
    timestamp: ISODate('2021-06-06T00:00:00.000Z'),
    expenses: 8000
}
]
animal> |
```

9. Write a MongoDB query to display wild_animals available under a particular care_taker

```
animal> db.wild_animals.find({ care_taker_name: "Ramesh" })
[ {
    _id: ObjectId('69a059eabae8b961fd7c2911'),
    animal_name: 'Lion',
    nature: 'harm',
    favorite_foods: [ 'meat', 'deer' ],
    care_taker_name: 'Ramesh',
    life_span: 14,
    timestamp: ISODate('2022-01-01T00:00:00.000Z'),
    expenses: 50000
},
{
    _id: ObjectId('69a059eabae8b961fd7c2914'),
    animal_name: 'Bear',
    nature: 'harm',
    favorite_foods: [ 'fish', 'honey' ],
    care_taker_name: 'Ramesh',
    life_span: 20,
    timestamp: ISODate('2019-08-20T00:00:00.000Z'),
    expenses: 45000
}
]
animal> |
```

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

```
animal> db.wild_animals.find(
| { life_span: { $gt: 5 } },
| { animal_name: 1, favorite_foods: 1, expenses: 1, _id: 0 }
| )
[ {
    animal_name: 'Lion',
    favorite_foods: [ 'meat', 'deer' ],
    expenses: 50000
},
{
    animal_name: 'Tiger',
    favorite_foods: [ 'meat', 'rabbits' ],
    expenses: 60000
},
{
    animal_name: 'Elephant',
    favorite_foods: [ 'grass', 'fruits' ],
    expenses: 80000
},
{
    animal_name: 'Bear',
    favorite_foods: [ 'fish', 'honey' ],
    expenses: 45000
},
{
    animal_name: 'Deer', favorite_foods: [ 'grass' ], expenses: 20000
}
]
animal> db.domestic_animals.find(
| { life_span: { $gt: 5 } },
| { animal_name: 1, favorite_foods: 1, expenses: 1, _id: 0 }
| )
[ {
    animal_name: 'Dog', favorite_foods: [ 'meat' ], expenses: 10000 },
    { animal_name: 'Cat', favorite_foods: [ 'fish' ], expenses: 8000 },
    { animal_name: 'Cow', favorite_foods: [ 'grass' ], expenses: 15000 },
    { animal_name: 'Goat', favorite_foods: [ 'leaves' ], expenses: 5000 },
    { animal_name: 'Horse', favorite_foods: [ 'grass' ], expenses: 30000 }
]
animal> |
```