

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

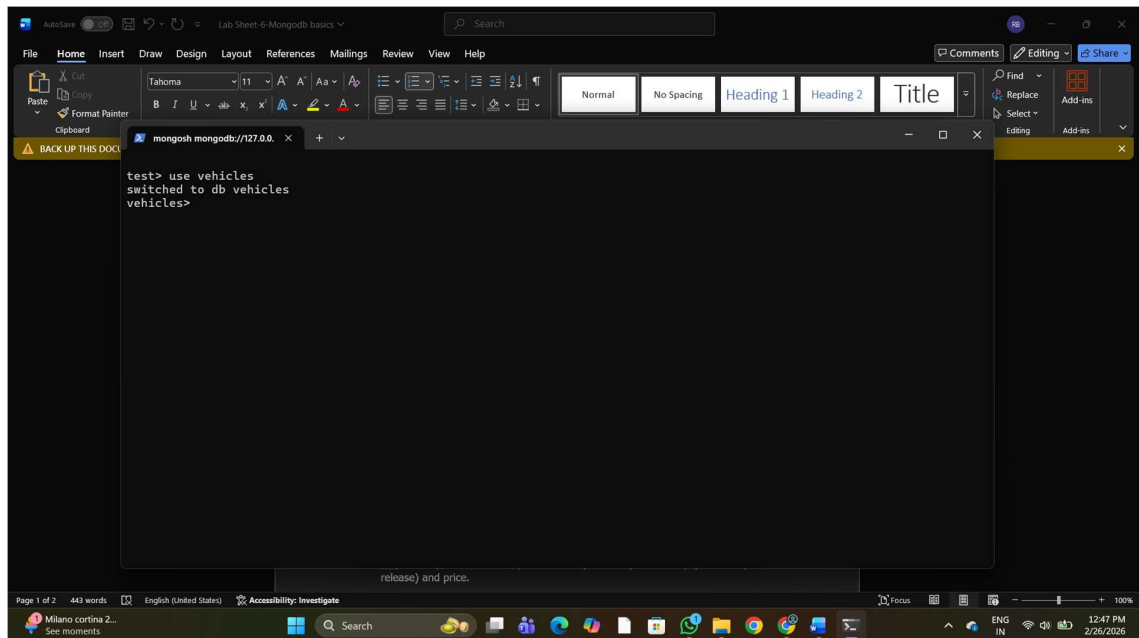
School: SCOPE

Student name: V.Raj Bharath

Reg. no.: 23BCE8517

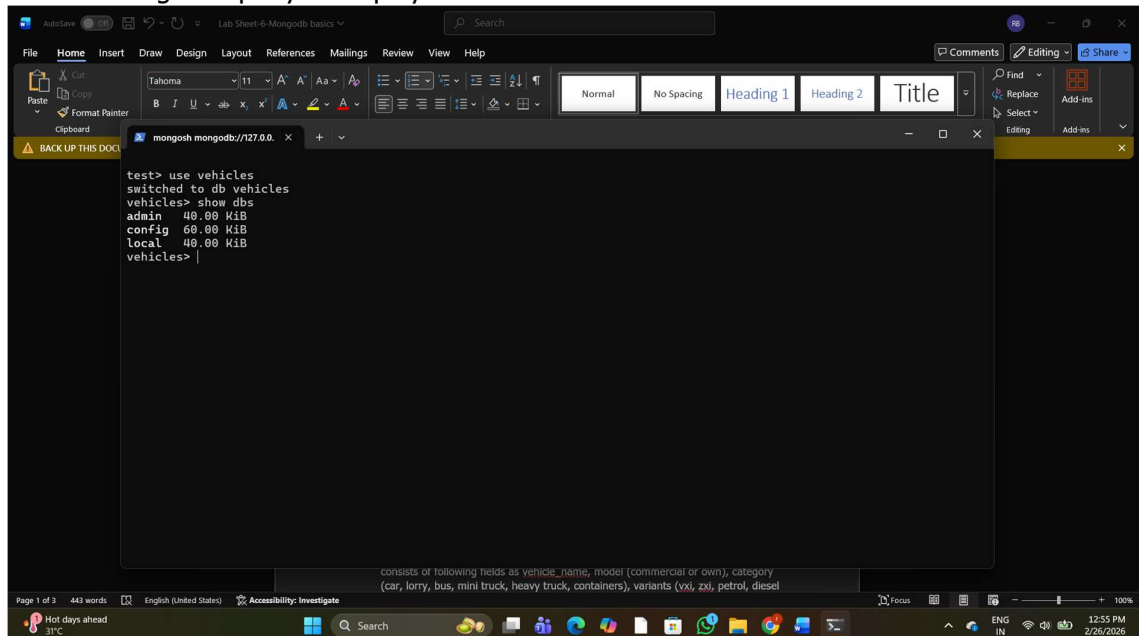
1. Use MongoDB to implement the following DB operations

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



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

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



```
test> use vehicles
switched to db vehicles
vehicles> show dbs
admin 48.00 KiB
config 60.00 KiB
local 40.00 KiB
vehicles> |
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Date: 26/02/2026

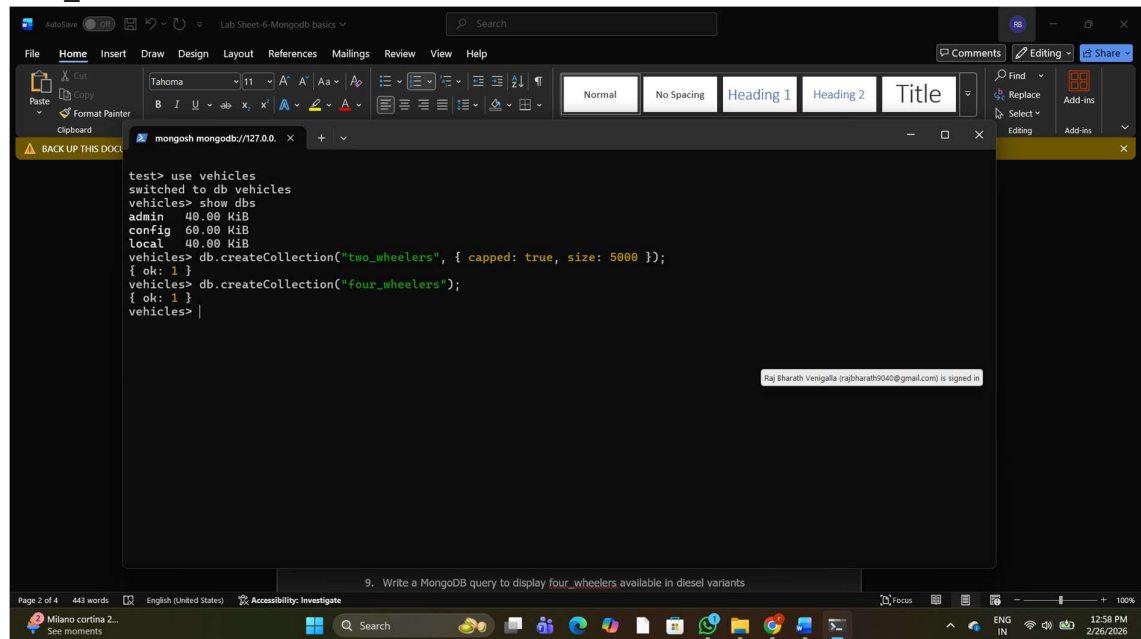
Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: V.Raj Bharath

Reg. no.: 23BCE8517

3. Create a collection called 'two_wheelers'. (use capping) and Create a collection called 'four_wheelers'.



```
test> use vehicles
switched to db vehicles
vehicles> show dbs
admin 48.00 KiB
config 60.00 KiB
local 48.00 KiB
vehicles> db.createCollection("two_wheelers", { capped: true, size: 5000 });
{ ok: 1 }
vehicles> db.createCollection("four_wheelers");
{ ok: 1 }
vehicles>
```

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

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.

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech
Faculty Name: Prof. S.Gopikrishnan
Student name: V.Raj Bharath

Date: 26/02/2026
School: SCOPE
Reg. no.: 23BCE8517

```
vehicles> db.two_wheelers.insertMany([
  {
    bike_name: "MT-15 V2",
    model: "gear",
    category: "155cc",
    colors_available: ["Cyan Storm", "Ice Fluo-Vermillion"],
    manufacturer: "Yamaha",
    performance: 9,
    timestamp: new Date("2024-01-10"),
    price: 172000
  },
  {
    bike_name: "Continental GT 650",
    model: "gear",
    category: "650cc",
    colors_available: ["British Racing Green", "Mr Clean"],
    manufacturer: "Royal Enfield",
    performance: 8,
    timestamp: new Date("2023-12-05"),
    price: 345000
  },
  {
    bike_name: "R15 V4",
    model: "gear",
    category: "155cc",
    colors_available: ["Racing Blue", "Dark Knight"],
    manufacturer: "Yamaha",
    performance: 10,
    timestamp: new Date("2024-02-15"),
    price: 185000
  },
  {
    bike_name: "Ola S1 Pro",
    model: "gearless",
    category: "Electric",
    colors_available: ["Matte Black", "Stellar Blue"],
    manufacturer: "Ola Electric",
    performance: 8,
    timestamp: new Date("2023-10-20"),
    price: 140000
  },
  {
    bike_name: "Duke 250",
    model: "gear",
    category: "250cc",
    colors_available: ["Electronic Orange", "Ceramic White"],
    manufacturer: "KTM",
    performance: 9,
    timestamp: new Date("2024-01-25"),
    price: 240000
  }
]);

acknowledged: true,
insertedIds: {
  '0': ObjectId('699ff66e63ac447d667c2987'),
  '1': ObjectId('699ff66e63ac447d667c2988'),
  '2': ObjectId('699ff66e63ac447d667c2989'),
  '3': ObjectId('699ff66e63ac447d667c298a'),
  '4': ObjectId('699ff66e63ac447d667c298b')
}
}
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.

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech
Faculty Name: Prof. S.Gopikrishnan
Student name: V.Raj Bharath

Date: 26/02/2026
School: SCOPE
Reg. no.: 23BCE8517

```
vehicles> db.four_wheelers.insertMany([
  {
    vehicle_name: "Thar",
    model: "own",
    category: "SUV",
    variants: ["Earth Editon", "4x4 Diesel", "LX"],
    manufacturer: "Mahindra",
    performance: 9,
    timestamp: new Date("2024-01-15"),
    price: 1750000
  },
  {
    vehicle_name: "Verna",
    model: "own",
    category: "Sedan",
    variants: ["Turbo DCT", "SX(0)"],
    manufacturer: "Hyundai",
    performance: 10,
    timestamp: new Date("2023-11-20"),
    price: 1850000
  },
  {
    vehicle_name: "Scorpio-N",
    model: "own",
    category: "SUV",
    variants: ["Z8L", "4WD Diesel"],
    manufacturer: "Mahindra",
    performance: 9,
    timestamp: new Date("2024-02-05"),
    price: 2300000
  },
  {
    vehicle_name: "Virtus",
    model: "own",
    category: "Sedan",
    variants: ["GT Line", "1.5 TSI"],
    manufacturer: "Volkswagen",
    performance: 10,
    timestamp: new Date("2023-09-12"),
    price: 1950000
  }
]);

vehicles>

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699ff72963ac447d667c290c'),
    '1': ObjectId('699ff72963ac447d667c290d'),
    '2': ObjectId('699ff72963ac447d667c290e'),
    '3': ObjectId('699ff72963ac447d667c290f'),
    '4': ObjectId('699ff72963ac447d667c2910')
  }
}
```

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

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: V.Raj Bharath

Reg. no.: 23BCE8517

```
vehicles> db.two_wheelers.find();
[
  {
    _id: ObjectId('699ff66e63ac447d667c2987'),
    bike_name: 'MT-15 V2',
    model: 'gear',
    category: '155cc',
    colors_available: [ 'Cyan Storm', 'Ice Fluo-Vermillion' ],
    manufacturer: 'Yamaha',
    performance: 9,
    timestamp: ISODate('2024-01-10T00:00:00.000Z'),
    price: 172000
  },
  {
    _id: ObjectId('699ff66e63ac447d667c2988'),
    bike_name: 'Continental GT 650',
    model: 'gear',
    category: '650cc',
    colors_available: [ 'British Racing Green', 'Mr Clean' ],
    manufacturer: 'Royal Enfield',
    performance: 8,
    timestamp: ISODate('2023-12-05T00:00:00.000Z'),
    price: 345000
  },
  {
    _id: ObjectId('699ff66e63ac447d667c2989'),
    bike_name: 'R15 V4',
    model: 'gear',
    category: '155cc',
    colors_available: [ 'Racing Blue', 'Dark Knight' ],
    manufacturer: 'Yamaha',
    performance: 10,
    timestamp: ISODate('2024-02-15T00:00:00.000Z'),
    price: 185000
  },
  {
    _id: ObjectId('699ff66e63ac447d667c298a'),
    bike_name: 'Ola S1 Pro',
    model: 'gearless',
    category: 'Electric',
    colors_available: [ 'Matte Black', 'Stellar Blue' ],
    manufacturer: 'Ola Electric',
    performance: 8,
    timestamp: ISODate('2023-10-20T00:00:00.000Z'),
    price: 148000
  },
  {
    _id: ObjectId('699ff66e63ac447d667c298b'),
    bike_name: 'Duke 250',
    model: 'gear',
    category: '250cc',
    colors_available: [ 'Electronic Orange', 'Ceramic White' ],
    manufacturer: 'KTM',
    performance: 9,
    timestamp: ISODate('2024-01-25T00:00:00.000Z'),
    price: 248000
  }
]
vehicles> db.four_wheelers.find();
[
  {
    _id: ObjectId('699ff72963ac447d667c298c'),
    vehicle_name: 'Thar',
    model: 'own',
    category: 'SUV',
    variants: [ 'Earth Edition', '4x4 Diesel', 'LX' ],
    manufacturer: 'Mahindra',
    performance: 9,
    timestamp: ISODate('2024-01-15T00:00:00.000Z'),
    price: 1750000
  },
  {
    _id: ObjectId('699ff72963ac447d667c298d'),
    vehicle_name: 'Verna',
    model: 'own',
    category: 'Sedan',
    manufacturer: 'Maruti Suzuki',
    performance: 8,
    timestamp: ISODate('2023-11-10T00:00:00.000Z'),
    price: 1250000
  }
]
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech
Faculty Name: Prof. S.Gopikrishnan
Student name: V.Raj Bharath

Date: 26/02/2026
School: SCOPE
Reg. no.: 23BCE8517

```
mongo> use vehicles
vehicles> find()
{
  "_id": ObjectId("699ff72963ac447d667c290e"),
  "vehicle_name": "Scorpio-N",
  "model": "own",
  "category": "SUV",
  "variants": [ "28L", "4WD Diesel" ],
  "manufacturer": "Mahindra",
  "performance": 9,
  "timestamp": ISODate("2024-02-05T00:00:00.000Z"),
  "price": 2300000
},
{
  "_id": ObjectId("699ff72963ac447d667c290f"),
  "vehicle_name": "Virtus",
  "model": "own",
  "category": "Sedan",
  "variants": [ "GT Line", "1.5 TSI" ],
  "manufacturer": "Volkswagen",
  "performance": 10,
  "timestamp": ISODate("2023-09-12T00:00:00.000Z"),
  "price": 1950000
},
{
  "_id": ObjectId("699ff72963ac447d667c2910"),
  "vehicle_name": "Fronx",
  "model": "own",
  "category": "Crossover",
  "variants": [ "Alpha Turbo", "Zeta" ],
  "manufacturer": "Maruti Suzuki",
  "performance": 8,
  "timestamp": ISODate("2024-01-30T00:00:00.000Z"),
  "price": 1250000
}
]
vehicles>
```

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

```
mongo> use vehicles
vehicles> find()
{
  "_id": ObjectId("699ff72963ac447d667c290f"),
  "vehicle_name": "Virtus",
  "model": "own",
  "category": "Sedan",
  "variants": [ "GT Line", "1.5 TSI" ],
  "manufacturer": "Volkswagen",
  "performance": 10,
  "timestamp": ISODate("2023-09-12T00:00:00.000Z"),
  "price": 1950000
},
{
  "_id": ObjectId("699ff72963ac447d667c2910"),
  "vehicle_name": "Fronx",
  "model": "own",
  "category": "Crossover",
  "variants": [ "Alpha Turbo", "Zeta" ],
  "manufacturer": "Maruti Suzuki",
  "performance": 8,
  "timestamp": ISODate("2024-01-30T00:00:00.000Z"),
  "price": 1250000
}
]
vehicles> db.two_wheelers.find({}, { bike_name: 1, price: 1, _id: 0 });
[
  { bike_name: 'MT-15 V2', price: 172000 },
  { bike_name: 'Continental GT 650', price: 345000 },
  { bike_name: 'R15 V4', price: 185000 },
  { bike_name: 'Ola S1 Pro', price: 140000 },
  { bike_name: 'Duke 250', price: 240000 }
]
vehicles> db.four_wheelers.find({}, { vehicle_name: 1, price: 1, _id: 0 });
[
  { vehicle_name: 'Thar', price: 1750000 },
  { vehicle_name: 'Verano', price: 1850000 },
  { vehicle_name: 'Scorpio-N', price: 2300000 },
  { vehicle_name: 'Virtus', price: 1950000 },
  { vehicle_name: 'Fronx', price: 1250000 }
]
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: V.Raj Bharath

Reg. no.: 23BCE8517

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

```
mongosh mongodb://127.0.0.1:27017
> use vehicles
> db.two_wheelers.find({ manufacturer: "Maruti Suzuki" })
{
  category: 'Crossover',
  variants: [ 'Alpha Turbo', 'Zeta' ],
  manufacturer: 'Maruti Suzuki',
  performance: 8,
  timestamp: ISODate('2024-01-30T00:00:00.000Z'),
  price: 1250000
}
vehicles> db.two_wheelers.find({ bike_name: 1, price: 1, _id: 0 });
[
  { bike_name: 'MT-15 V2', price: 172000 },
  { bike_name: 'Continental GT 650', price: 345000 },
  { bike_name: 'R15 V4', price: 165000 },
  { bike_name: 'Ola S1 Pro', price: 140000 },
  { bike_name: 'Duke 250', price: 240000 }
]
vehicles> db.two_wheelers.find({ vehicle_name: 1, price: 1, _id: 0 });
[
  { vehicle_name: 'Thar', price: 1750000 },
  { vehicle_name: 'Verna', price: 1850000 },
  { vehicle_name: 'Scorpio-N', price: 2300000 },
  { vehicle_name: 'Virtus', price: 1950000 },
  { vehicle_name: 'Fronx', price: 1250000 }
]
vehicles> db.two_wheelers.find({ manufacturer: "Honda" });
vehicles> db.two_wheelers.find({ manufacturer: "KTM" });
[
  {
    _id: ObjectId('699ff66e63ac447d667c290b'),
    bike_name: 'Duke 250',
    model: 'gear',
    category: '250cc',
    colors_available: [ 'Electronic Orange', 'Ceramic White' ],
    manufacturer: 'KTM',
    performance: 9,
    timestamp: ISODate('2024-01-25T00:00:00.000Z'),
    price: 240000
  }
]
vehicles>
```

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

```
mongosh mongodb://127.0.0.1:27017
> use vehicles
> db.four_wheelers.find({ vehicle_name: 1, price: 1, _id: 0 });
[
  { vehicle_name: 'Thar', price: 1750000 },
  { vehicle_name: 'Verna', price: 1850000 },
  { vehicle_name: 'Scorpio-N', price: 2300000 },
  { vehicle_name: 'Virtus', price: 1950000 },
  { vehicle_name: 'Fronx', price: 1250000 }
]
vehicles> db.two_wheelers.find({ manufacturer: "Honda" });
vehicles> db.two_wheelers.find({ manufacturer: "KTM" });
[
  {
    _id: ObjectId('699ff66e63ac447d667c290b'),
    bike_name: 'Duke 250',
    model: 'gear',
    category: '250cc',
    colors_available: [ 'Electronic Orange', 'Ceramic White' ],
    manufacturer: 'KTM',
    performance: 9,
    timestamp: ISODate('2024-01-25T00:00:00.000Z'),
    price: 240000
  }
]
vehicles> db.four_wheelers.find({ variants: "diesel" });
vehicles> db.four_wheelers.find({ variants: "4x4 Diesel" });
[
  {
    _id: ObjectId('699ff72963ac447d667c290c'),
    vehicle_name: 'Thar',
    model: 'own',
    category: 'SUV',
    variants: [ 'Earth Edition', '4x4 Diesel', 'LX' ],
    manufacturer: 'Mahindra',
    performance: 9,
    timestamp: ISODate('2024-01-15T00:00:00.000Z'),
    price: 1750000
  }
]
vehicles>
```


Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Date: 26/02/2026

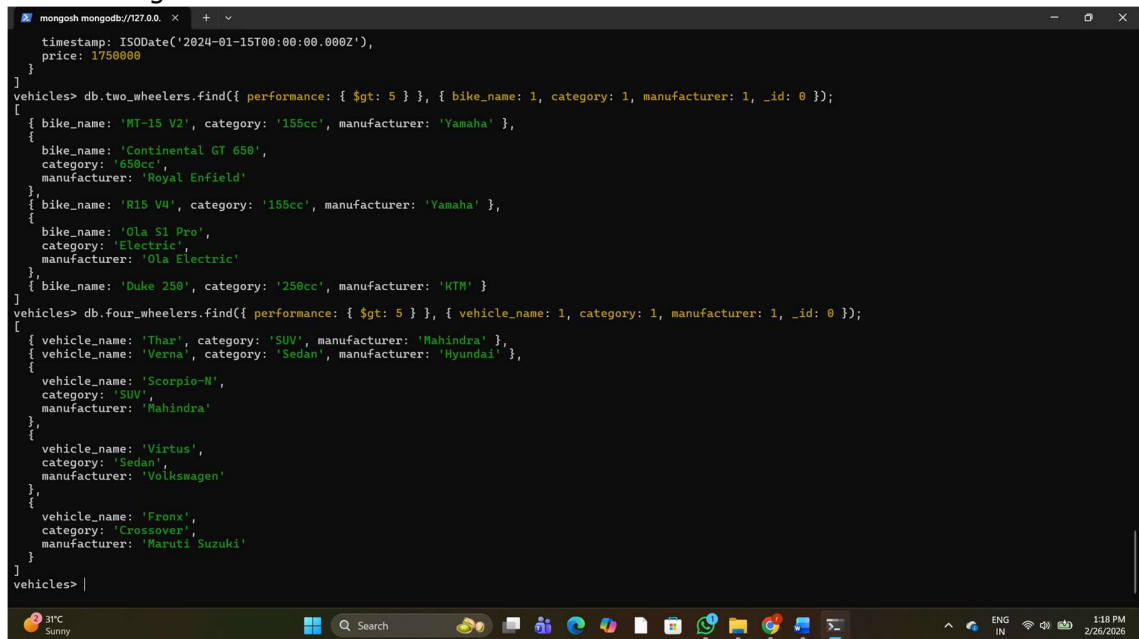
Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: V.Raj Bharath

Reg. no.: 23BCE8517

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



```
mongosh mongodb://127.0.0.1:27020
timestamp: ISODate('2024-01-15T00:00:00.000Z'),
price: 1750000
}
vehicles> db.two_wheelers.find({ performance: { $gt: 5 } }, { bike_name: 1, category: 1, manufacturer: 1, _id: 0 });
[
  { bike_name: 'MT-15 V2', category: '155cc', manufacturer: 'Yamaha' },
  { bike_name: 'Continental GT 650', category: '650cc', manufacturer: 'Royal Enfield' },
  { bike_name: 'R15 V4', category: '155cc', manufacturer: 'Yamaha' },
  { bike_name: 'Ola S1 Pro', category: 'Electric', manufacturer: 'Ola Electric' },
  { bike_name: 'Duke 250', category: '250cc', manufacturer: 'KTM' }
]
vehicles> db.four_wheelers.find({ performance: { $gt: 5 } }, { vehicle_name: 1, category: 1, manufacturer: 1, _id: 0 });
[
  { vehicle_name: 'Thar', category: 'SUV', manufacturer: 'Mahindra' },
  { vehicle_name: 'Verna', category: 'Sedan', manufacturer: 'Hyundai' },
  { vehicle_name: 'Scorpio-N', category: 'SUV', manufacturer: 'Mahindra' },
  { vehicle_name: 'Virtus', category: 'Sedan', manufacturer: 'Volkswagen' },
  { vehicle_name: 'Fronx', category: 'Crossover', manufacturer: 'Maruti Suzuki' }
]
vehicles> |
```

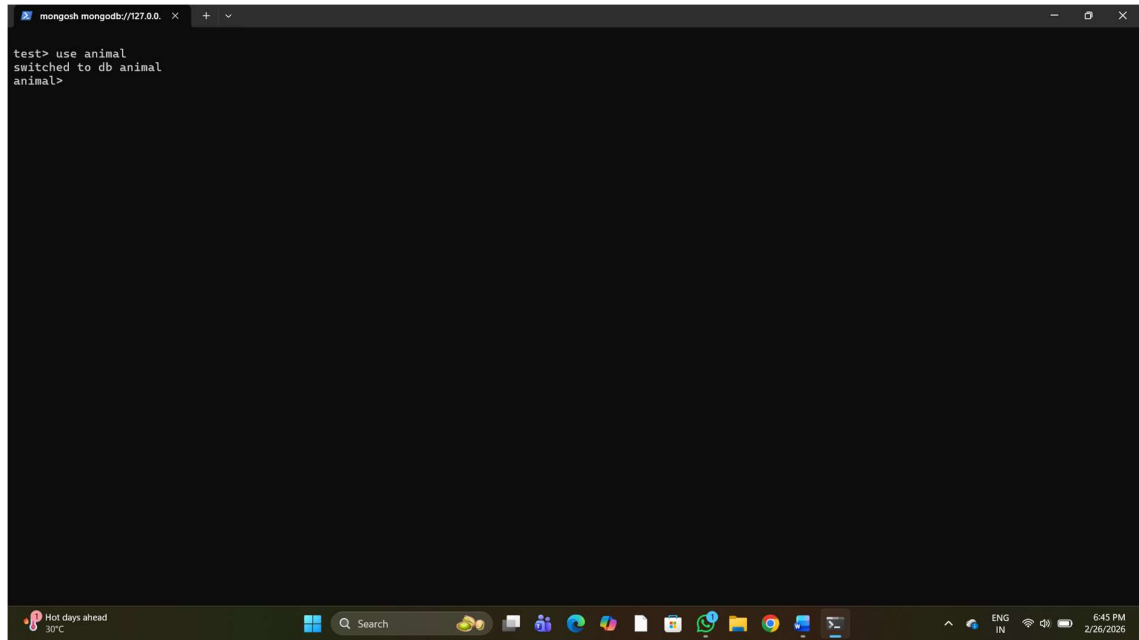
2. Use MongoDB to implement the following DB operations for a Zoo

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech
Faculty Name: Prof. S.Gopikrishnan
Student name: V.Raj Bharath

Date: 26/02/2026
School: SCOPE
Reg. no.: 23BCE8517

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

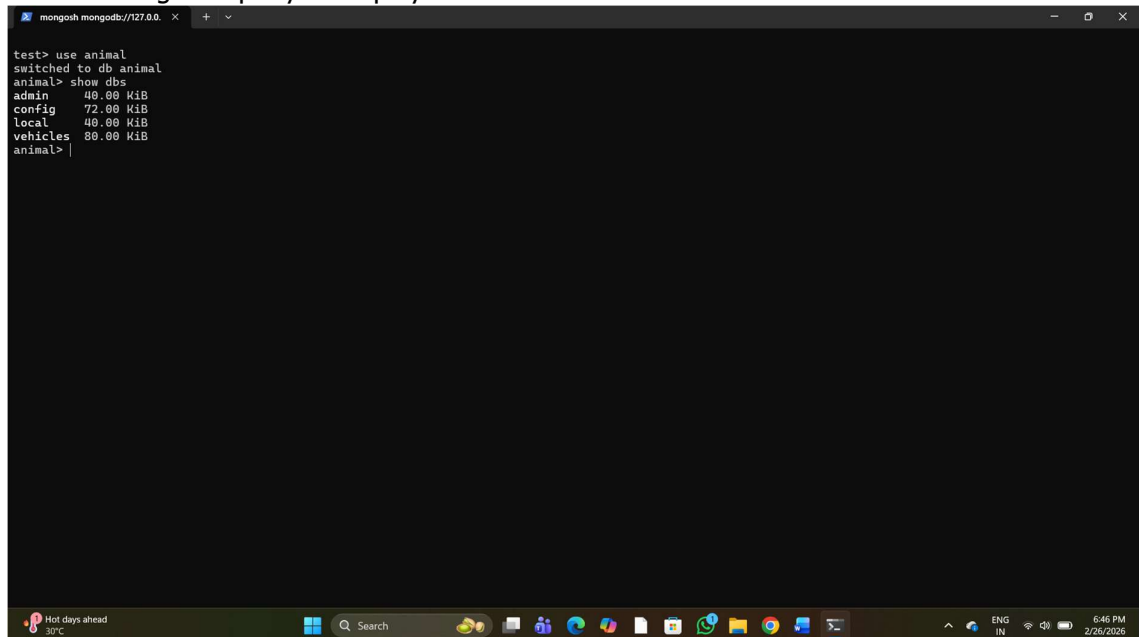


A screenshot of a MongoDB command prompt window titled 'mongosh mongodb://127.0.0.1:27020'. The prompt shows the following commands and output:

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

The window is running on a Windows operating system, as evidenced by the taskbar at the bottom showing the Start button, search bar, and various application icons. The system tray shows the date and time as 6:45 PM on 2/26/2026.

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



A screenshot of a MongoDB command prompt window titled 'mongosh mongodb://127.0.0.1:27020'. The prompt shows the following commands and output:

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

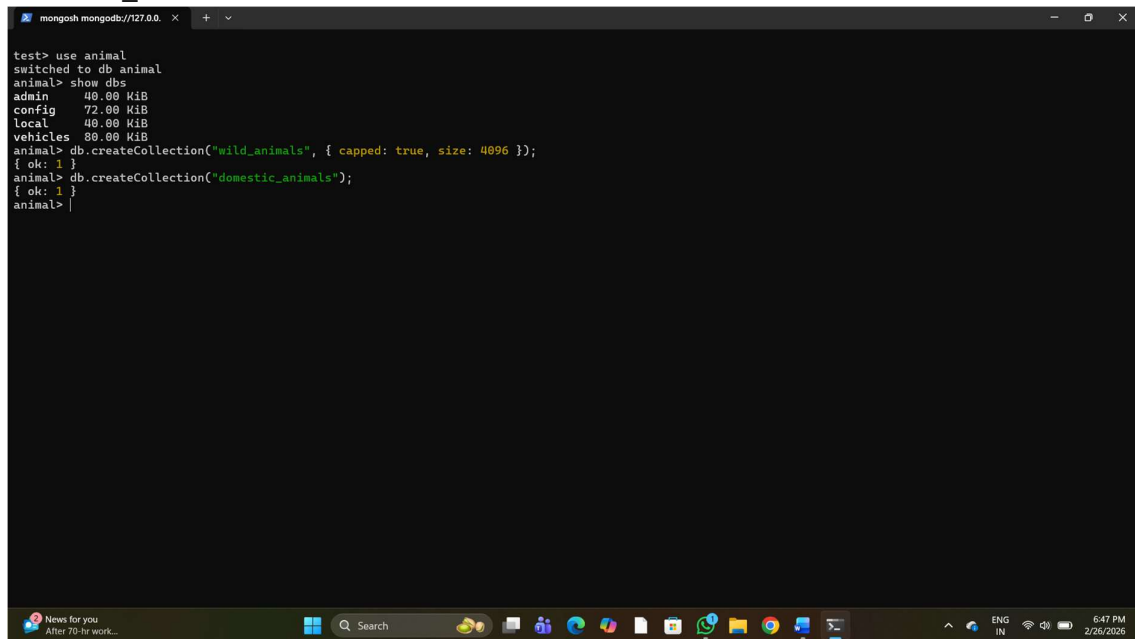
The window is running on a Windows operating system, as evidenced by the taskbar at the bottom showing the Start button, search bar, and various application icons. The system tray shows the date and time as 6:46 PM on 2/26/2026.

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech
Faculty Name: Prof. S.Gopikrishnan
Student name: V.Raj Bharath

Date: 26/02/2026
School: SCOPE
Reg. no.: 23BCE8517

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



```
test> use animal
switched to db animal
animal> show dbs
admin      40.00 KiB
config     72.00 KiB
local      40.00 KiB
vehicles   80.00 KiB
animal> db.createCollection("wild_animals", { capped: true, size: 4096 });
{ ok: 1 }
animal> db.createCollection("domestic_animals");
{ ok: 1 }
animal> |
```

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.

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech
Faculty Name: Prof. S.Gopikrishnan
Student name: V.Raj Bharath

Date: 26/02/2026
School: SCOPE
Reg. no.: 23BCE8517

```
mongosh mongodb://127.0.0.1:27017
> use zoo
> db.createCollection("domestic_animals");
{ ok: 1 }
> db.wild_animals.insertMany([
  {
    animal_name: "Lion",
    nature: "harm",
    favorite_foods: ["meat", "wildebeest"],
    care_taker_name: "John",
    life_span: 15,
    timestamp: new Date(),
    expenses: 5200
  },
  {
    animal_name: "Tiger",
    nature: "harm",
    favorite_foods: ["meat", "deer"],
    care_taker_name: "Robert",
    life_span: 12,
    timestamp: new Date(),
    expenses: 4800
  },
  {
    animal_name: "Zebra",
    nature: "harmless",
    favorite_foods: ["grass", "bark"],
    care_taker_name: "Alice",
    life_span: 25,
    timestamp: new Date(),
    expenses: 2200
  },
  {
    animal_name: "Hyena",
    nature: "harm",
    favorite_foods: ["meat", "bones"],
    care_taker_name: "Robert",
    life_span: 12,
    timestamp: new Date(),
    expenses: 3100
  },
  {
    animal_name: "Panda",
    nature: "harmless",
    favorite_foods: ["bamboo", "shoots"],
    care_taker_name: "John",
    life_span: 20,
    timestamp: new Date(),
    expenses: 7500
  }
]);
acknowledged: true,
insertedIds: {
  '0': ObjectId('69a048e58d562f65ca7c2907'),
  '1': ObjectId('69a048e58d562f65ca7c2908'),
  '2': ObjectId('69a048e58d562f65ca7c2909'),
  '3': ObjectId('69a048e58d562f65ca7c290a'),
  '4': ObjectId('69a048e58d562f65ca7c290b')
}
>
```

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.

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech
Faculty Name: Prof. S.Gopikrishnan
Student name: V.Raj Bharath

Date: 26/02/2026
School: SCOPE
Reg. no.: 23BCE8517

```
mongo> use domestic_animals
switched to db domestic_animals
mongo> db.domestic_animals.insertMany([
  {
    animal_name: "Cow",
    gender: "female",
    favorite_foods: ["hay", "grass"],
    animal_petname: "Daisy",
    life_span: 20,
    timestamp: new Date(),
    expenses: 1500
  },
  {
    animal_name: "Buffalo",
    gender: "male",
    favorite_foods: ["green fodder", "water plants"],
    animal_petname: "Shera",
    life_span: 25,
    timestamp: new Date(),
    expenses: 1800
  },
  {
    animal_name: "Goat",
    gender: "female",
    favorite_foods: ["shrubs", "grains"],
    animal_petname: "Giddy",
    life_span: 12,
    timestamp: new Date(),
    expenses: 400
  },
  {
    animal_name: "Dog",
    gender: "male",
    favorite_foods: ["kibble", "chicken"],
    animal_petname: "Max",
    life_span: 13,
    timestamp: new Date(),
    expenses: 1200
  },
  {
    animal_name: "Cat",
    gender: "female",
    favorite_foods: ["tuna", "dry food"],
    animal_petname: "Misty",
    life_span: 16,
    timestamp: new Date(),
    expenses: 600
  }
]);
acknowledged: true,
insertedIds: {
  '0': ObjectId('69a04a5d6dd1ab6d317c2987'),
  '1': ObjectId('69a04a5d6dd1ab6d317c2988'),
  '2': ObjectId('69a04a5d6dd1ab6d317c2989'),
  '3': ObjectId('69a04a5d6dd1ab6d317c298a'),
  '4': ObjectId('69a04a5d6dd1ab6d317c298b')
}
mongo>
```

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

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech
Faculty Name: Prof. S.Gopikrishnan
Student name: V.Raj Bharath

Date: 26/02/2026
School: SCOPE
Reg. no.: 23BCE8517

```
mongosh mongodb://127.0.0.1:27020/
> use animal
> db.wild_animals.find();
[
  {
    _id: ObjectId('69a048e58d562f65ca7c2907'),
    animal_name: 'Lion',
    nature: 'harm',
    favorite_foods: [ 'meat', 'wildebeest' ],
    care_taker_name: 'John',
    life_span: 15,
    timestamp: ISODate('2026-02-26T13:21:41.243Z'),
    expenses: 5200
  },
  {
    _id: ObjectId('69a048e58d562f65ca7c2908'),
    animal_name: 'Tiger',
    nature: 'harm',
    favorite_foods: [ 'meat', 'deer' ],
    care_taker_name: 'Robert',
    life_span: 12,
    timestamp: ISODate('2026-02-26T13:21:41.243Z'),
    expenses: 4800
  },
  {
    _id: ObjectId('69a048e58d562f65ca7c2909'),
    animal_name: 'Zebra',
    nature: 'harmless',
    favorite_foods: [ 'grass', 'bark' ],
    care_taker_name: 'Alice',
    life_span: 25,
    timestamp: ISODate('2026-02-26T13:21:41.243Z'),
    expenses: 2200
  },
  {
    _id: ObjectId('69a048e58d562f65ca7c290a'),
    animal_name: 'Hyena',
    nature: 'harm',
    favorite_foods: [ 'meat', 'bones' ],
    care_taker_name: 'Robert',
    life_span: 12,
    timestamp: ISODate('2026-02-26T13:21:41.243Z'),
    expenses: 3100
  },
  {
    _id: ObjectId('69a048e58d562f65ca7c290b'),
    animal_name: 'Panda',
    nature: 'harmless',
    favorite_foods: [ 'bamboo', 'shoots' ],
    care_taker_name: 'John',
    life_span: 20,
    timestamp: ISODate('2026-02-26T13:21:41.243Z'),
    expenses: 7500
  }
]
> db.domestic_animals.find();
[
  {
    _id: ObjectId('69a049e18d562f65ca7c290c'),
    animal_name: 'Cow',
    gender: 'female',
    favorite_foods: [ 'hay', 'grass' ],
    animal_petname: 'Daisy',
    life_span: 20,
    timestamp: ISODate('2026-02-26T13:25:53.503Z'),
    expenses: 1500
  },
  {
    _id: ObjectId('69a049e18d562f65ca7c290d'),
    animal_name: 'Buffalo',
    gender: 'male',
    favorite_foods: [ 'green fodder', 'water plants' ],
    animal_petname: 'Shera',
    life_span: 25,
    timestamp: ISODate('2026-02-26T13:25:53.503Z'),
    expenses: 1800
  },
  {
    _id: ObjectId('69a049e18d562f65ca7c290e'),
    animal_name: 'Goat',
    gender: 'female',
    life_span: 15,
    timestamp: ISODate('2026-02-26T13:25:53.503Z'),
    expenses: 1200
  }
]
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech
Faculty Name: Prof. S.Gopikrishnan
Student name: V.Raj Bharath

Date: 26/02/2026
School: SCOPE
Reg. no.: 23BCE8517

```
mongosh mongodb://127.0.0.1:27017
> use animals
> find()
{
  "_id": ObjectId("69a04a5d6dd1ab6d317c2988"),
  "animal_name": "Buffalo",
  "gender": "male",
  "favorite_foods": [ "green fodder", "water plants" ],
  "animal_petname": "Shera",
  "life_span": 25,
  "timestamp": ISODate("2026-02-26T13:27:57.672Z"),
  "expenses": 1800
},
{
  "_id": ObjectId("69a04a5d6dd1ab6d317c2989"),
  "animal_name": "Goat",
  "gender": "female",
  "favorite_foods": [ "shrubs", "grains" ],
  "animal_petname": "Giddy",
  "life_span": 12,
  "timestamp": ISODate("2026-02-26T13:27:57.672Z"),
  "expenses": 400
},
{
  "_id": ObjectId("69a04a5d6dd1ab6d317c298a"),
  "animal_name": "Dog",
  "gender": "male",
  "favorite_foods": [ "kibble", "chicken" ],
  "animal_petname": "Max",
  "life_span": 13,
  "timestamp": ISODate("2026-02-26T13:27:57.672Z"),
  "expenses": 1200
},
{
  "_id": ObjectId("69a04a5d6dd1ab6d317c298b"),
  "animal_name": "Cat",
  "gender": "female",
  "favorite_foods": [ "tuna", "dry food" ],
  "animal_petname": "Misty",
  "life_span": 16,
  "timestamp": ISODate("2026-02-26T13:27:57.672Z"),
  "expenses": 600
}
]
animal>
```

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

```
mongosh mongodb://127.0.0.1:27017
> use animals
> find()
{
  "_id": ObjectId("69a04a5d6dd1ab6d317c298a"),
  "animal_name": "Dog",
  "gender": "male",
  "favorite_foods": [ "kibble", "chicken" ],
  "animal_petname": "Max",
  "life_span": 13,
  "timestamp": ISODate("2026-02-26T13:27:57.672Z"),
  "expenses": 1200
},
{
  "_id": ObjectId("69a04a5d6dd1ab6d317c298b"),
  "animal_name": "Cat",
  "gender": "female",
  "favorite_foods": [ "tuna", "dry food" ],
  "animal_petname": "Misty",
  "life_span": 16,
  "timestamp": ISODate("2026-02-26T13:27:57.672Z"),
  "expenses": 600
}
]
animal> db.wild_animals.find({}, { animal_name: 1, expenses: 1, _id: 0 });
[
  { animal_name: 'Lion', expenses: 5200 },
  { animal_name: 'Tiger', expenses: 4800 },
  { animal_name: 'Zebra', expenses: 2200 },
  { animal_name: 'Hyena', expenses: 3100 },
  { animal_name: 'Panda', expenses: 7500 }
]
animal> db.domestic_animals.find({}, { animal_name: 1, expenses: 1, _id: 0 });
[
  { animal_name: 'Cow', expenses: 1500 },
  { animal_name: 'Buffalo', expenses: 1800 },
  { animal_name: 'Goat', expenses: 400 },
  { animal_name: 'Dog', expenses: 1200 },
  { animal_name: 'Cat', expenses: 600 },
  { animal_name: 'Cow', expenses: 1500 },
  { animal_name: 'Buffalo', expenses: 1800 },
  { animal_name: 'Goat', expenses: 400 },
  { animal_name: 'Dog', expenses: 1200 },
  { animal_name: 'Cat', expenses: 600 }
]
animal>
```

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

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: V.Raj Bharath

Reg. no.: 23BCE8517

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

```
mongosh mongodb://127.0.0.1:27020
> use animals
> db.wild_animals.find({})
[ { animal_name: 'Panda', expenses: 7500 } ]
animal> db.domestic_animals.find({}, { animal_name: 1, expenses: 1, _id: 0 });
[ { animal_name: 'Cow', expenses: 1500 },
  { animal_name: 'Buffalo', expenses: 1800 },
  { animal_name: 'Goat', expenses: 400 },
  { animal_name: 'Dog', expenses: 1200 },
  { animal_name: 'Cat', expenses: 600 },
  { animal_name: 'Cow', expenses: 1500 },
  { animal_name: 'Buffalo', expenses: 1800 },
  { animal_name: 'Goat', expenses: 400 },
  { animal_name: 'Dog', expenses: 1200 },
  { animal_name: 'Cat', expenses: 600 } ]
animal> db.domestic_animals.find({ life_span: 15 });
animal> db.domestic_animals.find({ life_span: 16 });
[ { _id: ObjectId('69a049e18d562f65ca7c2910'),
  animal_name: 'Cat',
  gender: 'female',
  favorite_foods: [ 'tuna', 'dry food' ],
  animal_petname: 'Misty',
  life_span: 16,
  timestamp: ISODate('2026-02-26T13:25:53.503Z'),
  expenses: 600 },
  { _id: ObjectId('69a04a5d6dd1ab6d317c290b'),
  animal_name: 'Cat',
  gender: 'female',
  favorite_foods: [ 'tuna', 'dry food' ],
  animal_petname: 'Misty',
  life_span: 16,
  timestamp: ISODate('2026-02-26T13:27:57.672Z'),
  expenses: 600 } ]
animal>
```

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

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Faculty Name: Prof. S.Gopikrishnan

Student name: V.Raj Bharath

Date: 26/02/2026

School: SCOPE

Reg. no.: 23BCE8517

```
mongosh mongodb://127.0.0.1:27020
> use animals
> db.wild_animals.find(
  {
    life_span: { $gt: 5 },
    animal_name: 1, favorite_foods: 1, expenses: 1, _id: 0
  }
);
{
  animal_name: 'Lion',
  favorite_foods: [ 'meat', 'wildebeest' ],
  expenses: 5200
},
{
  animal_name: 'Tiger',
  favorite_foods: [ 'meat', 'deer' ],
  expenses: 4800
},
{
  animal_name: 'Zebra',
  favorite_foods: [ 'grass', 'bark' ],
  expenses: 2200
},
{
  animal_name: 'Hyena',
  favorite_foods: [ 'meat', 'bones' ],
  expenses: 3100
},
{
  animal_name: 'Panda',
  favorite_foods: [ 'bamboo', 'shoots' ],
  expenses: 7500
}
animal> |
```

```
mongosh mongodb://127.0.0.1:27020
> use domestic_animals
> db.domestic_animals.find(
  {
    life_span: { $gt: 5 },
    animal_name: 1, favorite_foods: 1, expenses: 1, _id: 0
  }
);
{
  animal_name: 'Cow',
  favorite_foods: [ 'hay', 'grass' ],
  expenses: 1500
},
{
  animal_name: 'Buffalo',
  favorite_foods: [ 'green fodder', 'water plants' ],
  expenses: 1800
},
{
  animal_name: 'Goat',
  favorite_foods: [ 'shrubs', 'grains' ],
  expenses: 400
},
{
  animal_name: 'Dog',
  favorite_foods: [ 'kibble', 'chicken' ],
  expenses: 1200
},
{
  animal_name: 'Cat',
  favorite_foods: [ 'tuna', 'dry food' ],
  expenses: 600
},
}
```