

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

Date: 26-02-2026

Faculty Name: Prof. S.Gopikrishnan

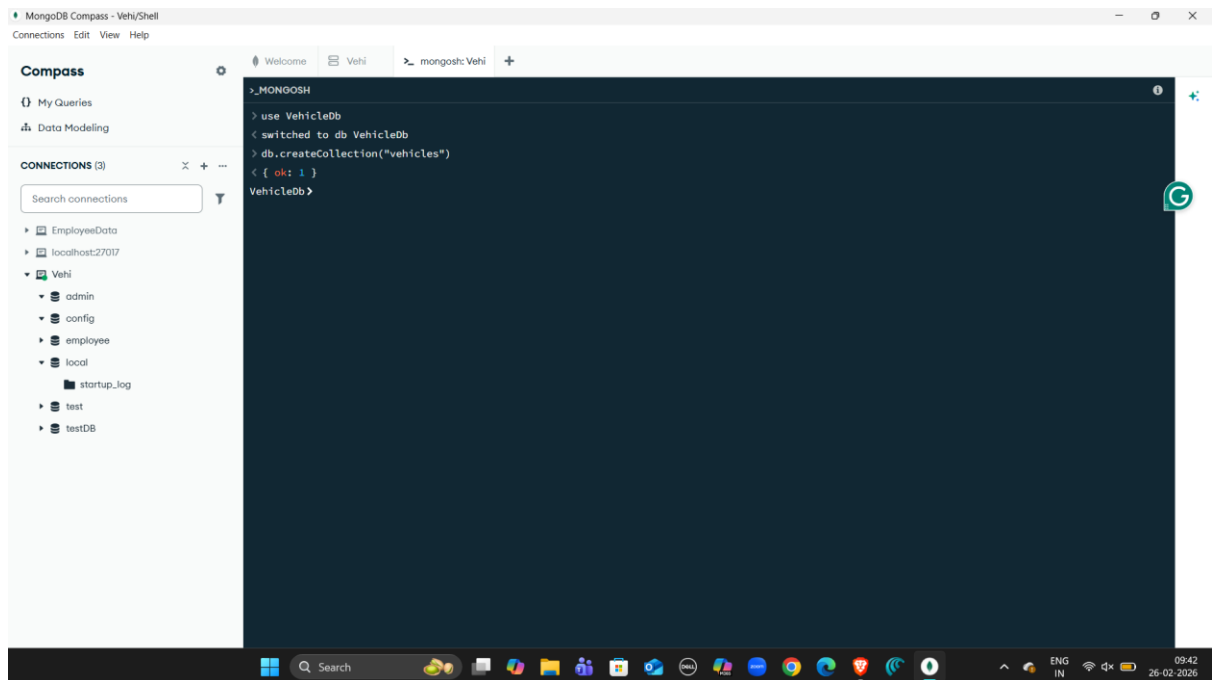
School: SCOPE

Student name: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

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".

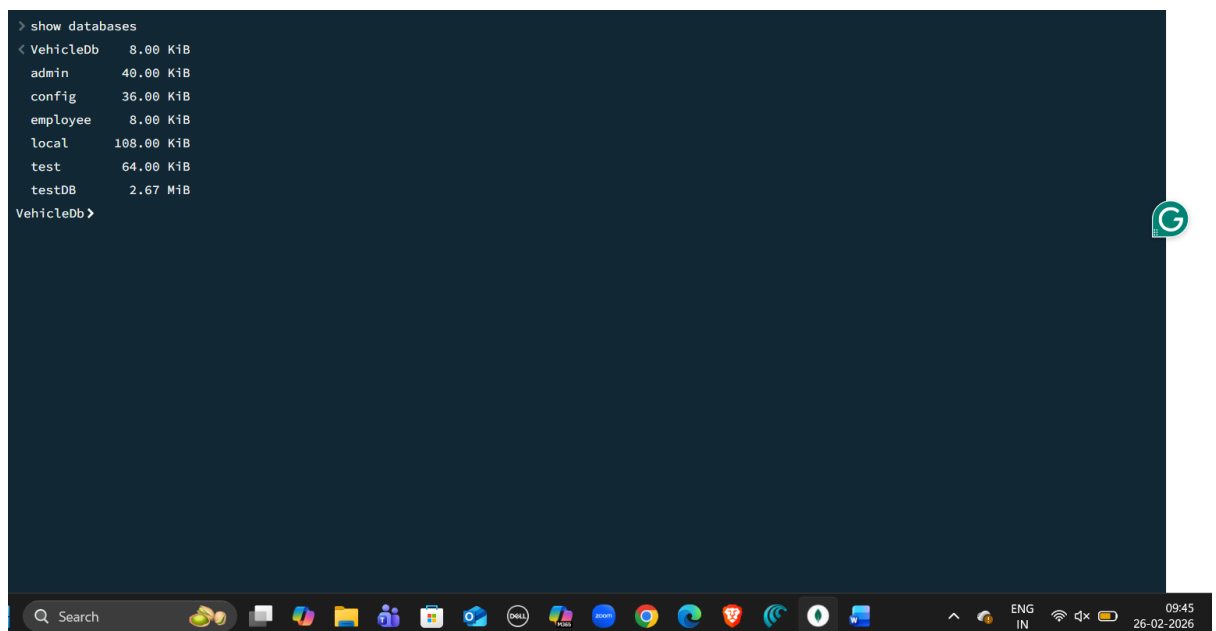


The screenshot shows the MongoDB Compass application window. The left sidebar displays the 'CONNECTIONS (3)' list with a search bar. Below it, a tree view shows the 'Vehi' database selected, with sub-items for 'admin', 'config', 'employee', 'local', 'startup_log', 'test', and 'testDB'. The main panel on the right is titled 'MongoDB Compass - Vehi/Shell' and contains a terminal window. The terminal shows the following commands and output:

```
> use VehicleDb
< switched to db VehicleDb
> db.createCollection("vehicles")
< { ok: 1 }
VehicleDb>
```

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right indicates the language is 'ENG IN', signal strength, and the time is 09:42 on 26-02-2026.

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



The screenshot shows the MongoDB Compass application window with the terminal panel. The terminal displays the output of the 'show databases' command:

```
> show databases
< VehicleDb 8.00 KiB
  admin 40.00 KiB
  config 36.00 KiB
  employee 8.00 KiB
  local 108.00 KiB
  test 64.00 KiB
  testDB 2.67 MiB
VehicleDb>
```

The Windows taskbar at the bottom shows the search bar, task view button, and several application icons. The system tray on the right indicates the language is 'ENG IN', signal strength, and the time is 09:45 on 26-02-2026.

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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

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

```
> db.createCollection("two_wheelers", { capped: true, size: 102400 })
< { ok: 1 }
> db.createCollection("four_wheelers")
< { ok: 1 }
> show collections
four_wheelers
two_wheelers
vehicles
VehicleDb>
```

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.

```
> db.two_wheelers.insertMany([
  {
    bike_name: "Yamaha R15",
    model: "gear",
    category: "150cc",
    colors_available: ["red", "black", "blue"],
    manufacturer: "Yamaha",
    performance: 8,
    timestamp: new Date("2022-03-15"),
    price: 180000
  },
  {
    bike_name: "Honda Activa",
    model: "gearless",
    category: "125cc",
    colors_available: ["white", "black", "red"],
    manufacturer: "Honda",
    performance: 7,
    timestamp: new Date("2021-06-10"),
    price: 90000
  },
  {
    bike_name: "Royal Enfield Classic 350",
    model: "gear",
    category: "200cc",
    colors_available: ["black", "blue", "red"],
    manufacturer: "Royal Enfield",
    performance: 9,
    timestamp: new Date("2020-09-20")
  }
])
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

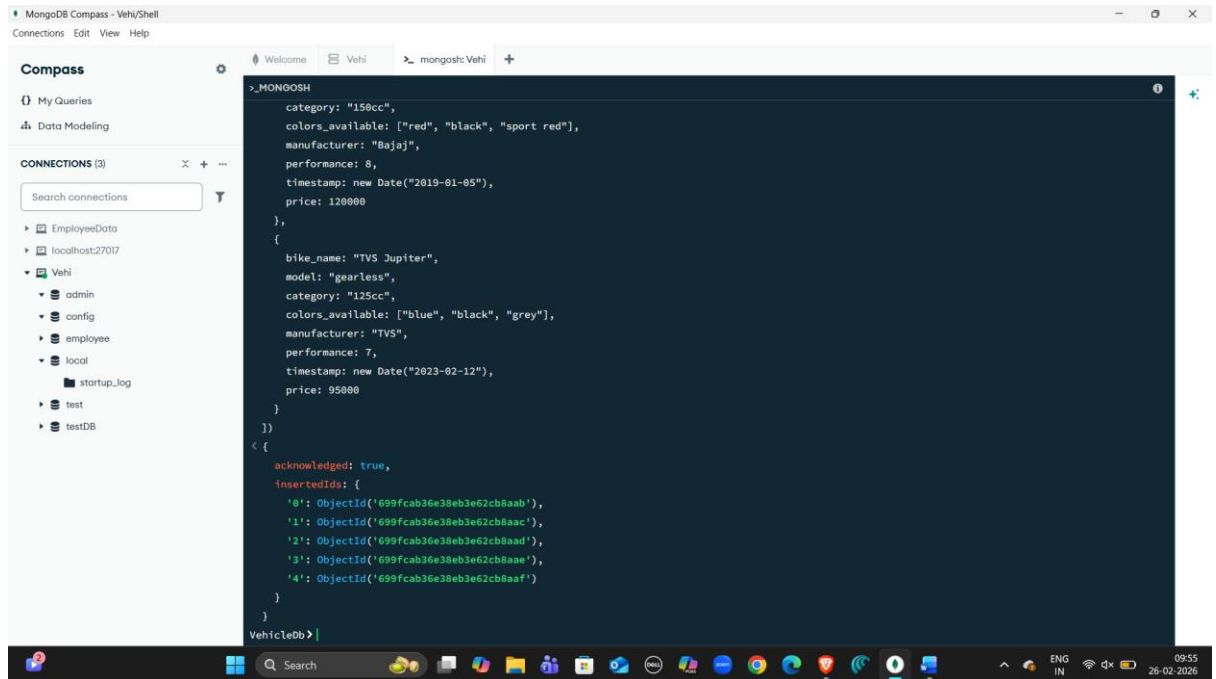
Date: 26-02-2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

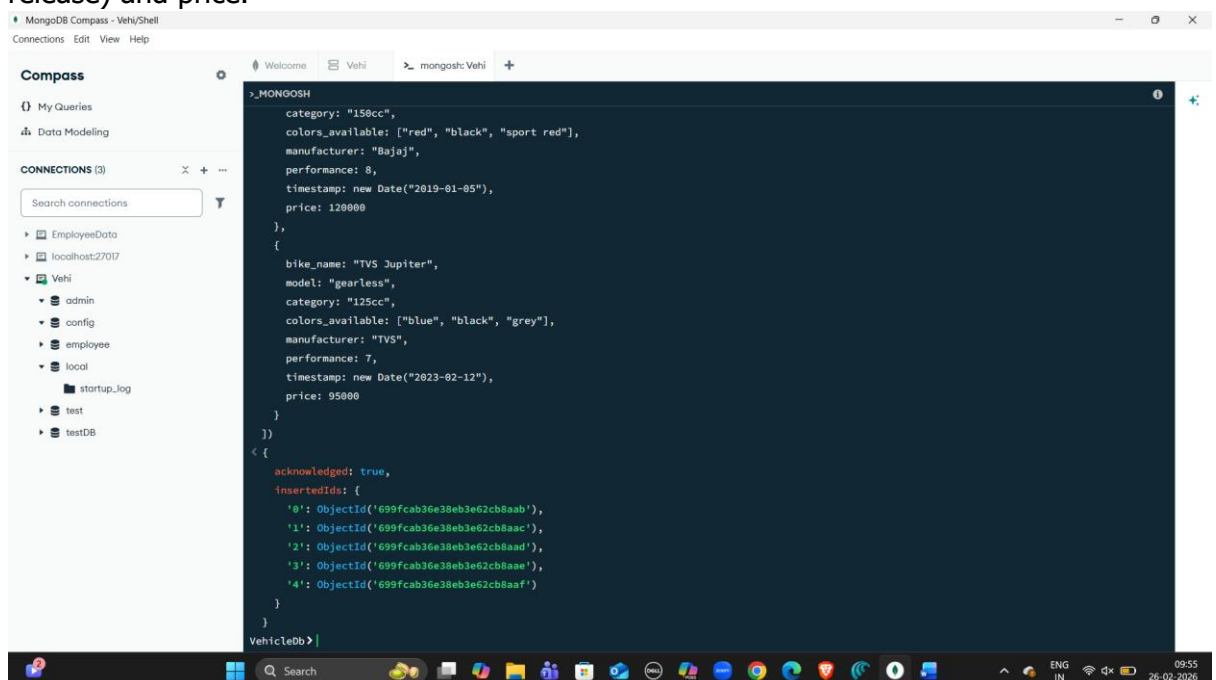
Student name: Samika Sanjeev Naidu

Reg. no.: 23BCE8855



```
>_MONGOOSH
category: "150cc",
colors_available: ["red", "black", "sport red"],
manufacturer: "Bajaj",
performance: 8,
timestamp: new Date("2019-01-05"),
price: 120000
},
{
  bike_name: "TVS Jupiter",
  model: "gearless",
  category: "125cc",
  colors_available: ["blue", "black", "grey"],
  manufacturer: "TVS",
  performance: 7,
  timestamp: new Date("2023-02-12"),
  price: 95000
}
}
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fcab36e38eb3e62cb8aab'),
    '1': ObjectId('699fcab36e38eb3e62cb8aac'),
    '2': ObjectId('699fcab36e38eb3e62cb8aad'),
    '3': ObjectId('699fcab36e38eb3e62cb8aae'),
    '4': ObjectId('699fcab36e38eb3e62cb8aaf')
  }
}
VehicleDb>
```

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.



```
>_MONGOOSH
category: "150cc",
colors_available: ["red", "black", "sport red"],
manufacturer: "Bajaj",
performance: 8,
timestamp: new Date("2019-01-05"),
price: 120000
},
{
  bike_name: "TVS Jupiter",
  model: "gearless",
  category: "125cc",
  colors_available: ["blue", "black", "grey"],
  manufacturer: "TVS",
  performance: 7,
  timestamp: new Date("2023-02-12"),
  price: 95000
}
}
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fcab36e38eb3e62cb8aab'),
    '1': ObjectId('699fcab36e38eb3e62cb8aac'),
    '2': ObjectId('699fcab36e38eb3e62cb8aad'),
    '3': ObjectId('699fcab36e38eb3e62cb8aae'),
    '4': ObjectId('699fcab36e38eb3e62cb8aaf')
  }
}
VehicleDb>
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

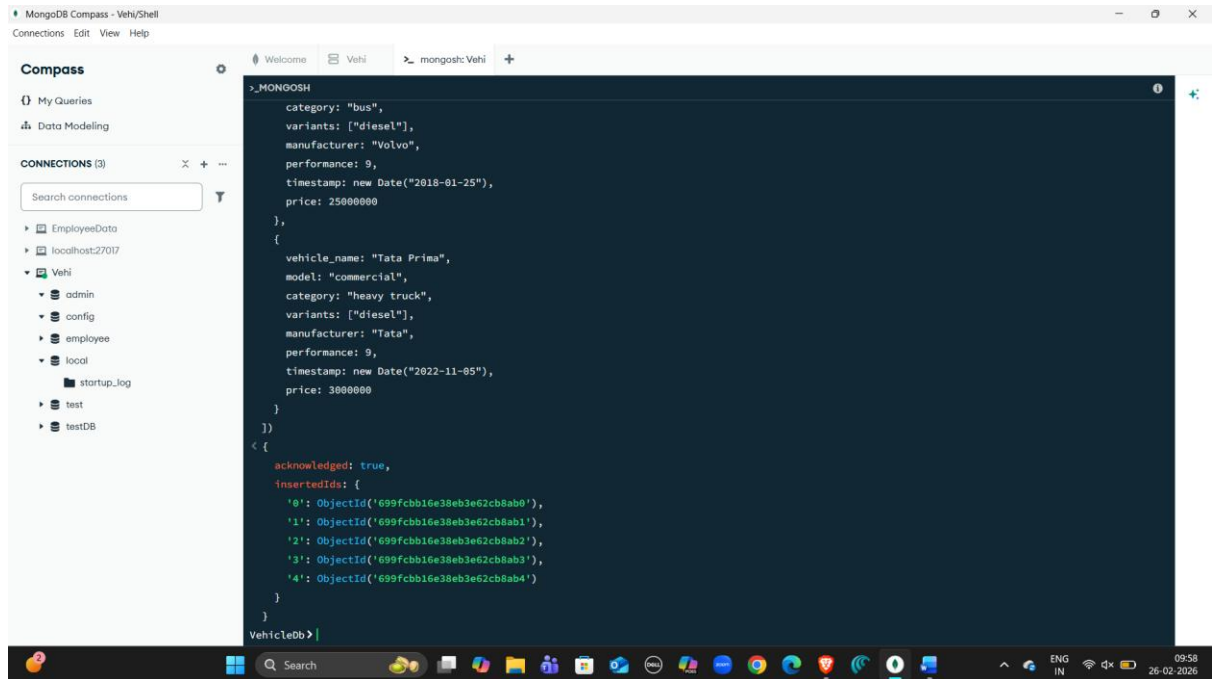
Date: 26-02-2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Samika Sanjeev Naidu

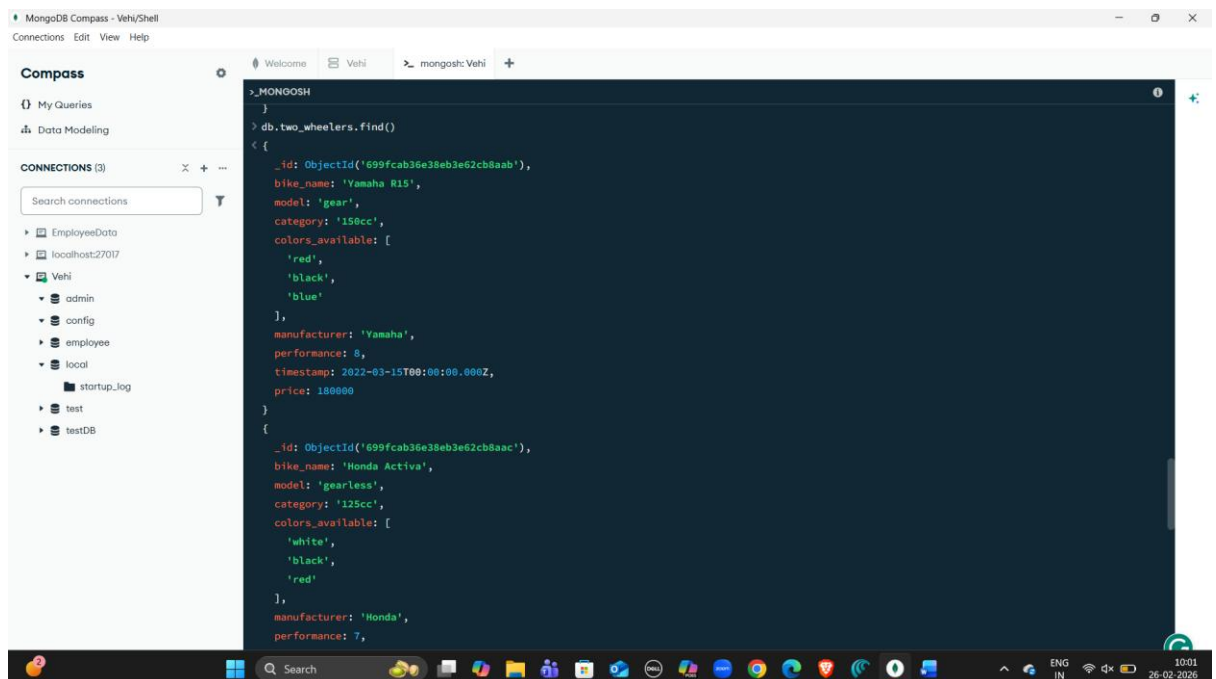
Reg. no.: 23BCE8855



The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS' panel lists several databases, including 'Vehi'. The main window displays the 'mongoosh' shell with a successful insert command. The command inserts two documents into the 'Vehi' database. The first document is a 'bus' with a price of 25000000. The second document is a 'heavy truck' with a price of 3000000. The output shows the 'acknowledged' status as true and the 'insertedIds' for both documents.

```
>_MONGOOSH
category: "bus",
variants: ["diesel"],
manufacturer: "Volvo",
performance: 9,
timestamp: new Date("2018-01-25"),
price: 25000000
},
{
  vehicle_name: "Tata Prima",
  model: "commercial",
  category: "heavy truck",
  variants: ["diesel"],
  manufacturer: "Tata",
  performance: 9,
  timestamp: new Date("2022-11-05"),
  price: 3000000
}
}
}
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fcb16e38eb3e62cb8ab0'),
    '1': ObjectId('699fcb16e38eb3e62cb8ab1'),
    '2': ObjectId('699fcb16e38eb3e62cb8ab2'),
    '3': ObjectId('699fcb16e38eb3e62cb8ab3'),
    '4': ObjectId('699fcb16e38eb3e62cb8ab4')
  }
}
```

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



The screenshot shows the MongoDB Compass interface. The main window displays the 'mongoosh' shell with a query to find all documents in the 'two_wheelers' collection. The query is 'db.two_wheelers.find()'. The output shows two documents. The first document is a 'Yamaha R15' with a price of 180000. The second document is a 'Honda Activa' with a price of 70000.

```
>_MONGOOSH
}
> db.two_wheelers.find()
< {
  _id: ObjectId('699fcab36e38eb3e62cb8aab'),
  bike_name: 'Yamaha R15',
  model: 'gear',
  category: '150cc',
  colors_available: [
    'red',
    'black',
    'blue'
  ],
  manufacturer: 'Yamaha',
  performance: 8,
  timestamp: 2022-03-15T00:00:00.000Z,
  price: 180000
}
{
  _id: ObjectId('699fcab36e38eb3e62cb8aac'),
  bike_name: 'Honda Activa',
  model: 'gearless',
  category: '125cc',
  colors_available: [
    'white',
    'black',
    'red'
  ],
  manufacturer: 'Honda',
  performance: 7,
}
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech

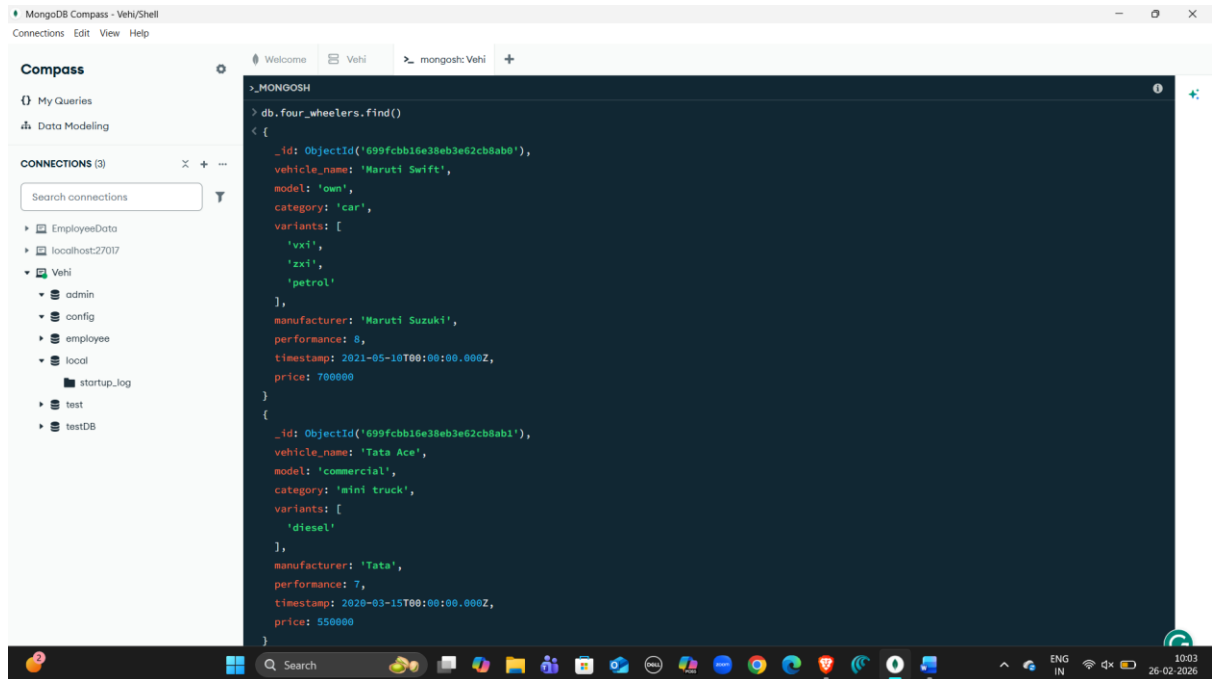
Date: 26-02-2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Samika Sanjeev Naidu

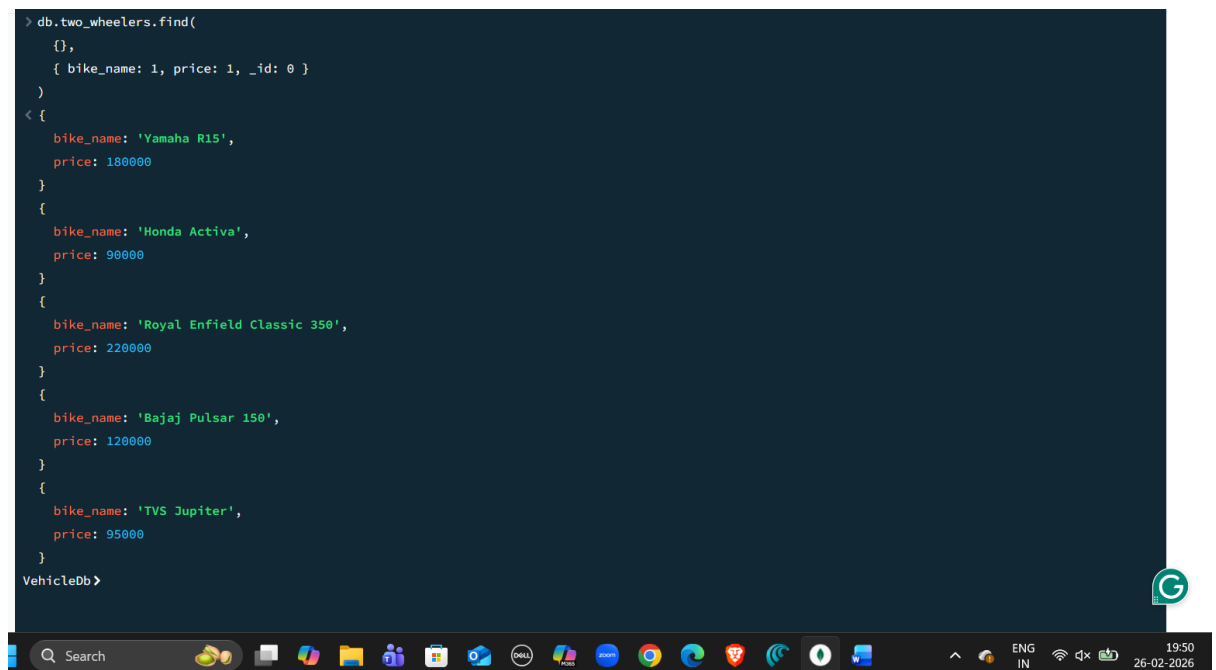
Reg. no.: 23BCE8855



The screenshot shows the MongoDB Compass application. On the left, the 'CONNECTIONS' panel lists several databases, including 'Vehi'. The main window displays a MongoDB shell with the following query and results:

```
> db.four_wheelers.find()
< [
  {
    "_id": ObjectId("699fcb16e38eb3e62cb8ab0"),
    "vehicle_name": "Maruti Swift",
    "model": "own",
    "category": "car",
    "variants": [
      "vxi",
      "zxi",
      "petrol"
    ],
    "manufacturer": "Maruti Suzuki",
    "performance": 8,
    "timestamp": "2021-05-10T00:00:00.000Z",
    "price": 700000
  },
  {
    "_id": ObjectId("699fcb16e38eb3e62cb8ab1"),
    "vehicle_name": "Tata Ace",
    "model": "commercial",
    "category": "mini truck",
    "variants": [
      "diesel"
    ],
    "manufacturer": "Tata",
    "performance": 7,
    "timestamp": "2020-03-15T00:00:00.000Z",
    "price": 550000
  }
]
```

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



The screenshot shows the MongoDB Compass application. The main window displays a MongoDB shell with the following query and results:

```
> db.two_wheelers.find(
  {},
  { bike_name: 1, price: 1, _id: 0 }
)
< [
  {
    "bike_name": "Yamaha R15",
    "price": 180000
  },
  {
    "bike_name": "Honda Activa",
    "price": 90000
  },
  {
    "bike_name": "Royal Enfield Classic 350",
    "price": 220000
  },
  {
    "bike_name": "Bajaj Pulsar 150",
    "price": 120000
  },
  {
    "bike_name": "TVS Jupiter",
    "price": 95000
  }
]
```

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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

```
> db.four_wheelers.find(
  {},
  { vehicle_name: 1, price: 1, _id: 0 }
)
< {
  vehicle_name: 'Maruti Swift',
  price: 700000
}
{
  vehicle_name: 'Tata Ace',
  price: 550000
}
{
  vehicle_name: 'Ashok Leyland Dost',
  price: 900000
}
{
  vehicle_name: 'Volvo Bus',
  price: 25000000
}
{
  vehicle_name: 'Tata Prima',
  price: 3000000
}
VehicleDb>
```

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

```
> db.two_wheelers.find({ manufacturer: "Yamaha" })
< {
  _id: ObjectId('699fcab36e38eb3e62cb8aab'),
  bike_name: 'Yamaha R15',
  model: 'gear',
  category: '150cc',
  colors_available: [
    'red',
    'black',
    'blue'
  ],
  manufacturer: 'Yamaha',
  performance: 8,
  timestamp: 2022-03-15T00:00:00.000Z,
  price: 180000
}
VehicleDb>
```

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

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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

```
>_MONGOSH
> db.four_wheelers.find({ variants: "diesel" })
< {
  _id: ObjectId('699fcbb16e38eb3e62cb8ab1'),
  vehicle_name: 'Tata Ace',
  model: 'commercial',
  category: 'mini truck',
  variants: [
    'diesel'
  ],
  manufacturer: 'Tata',
  performance: 7,
  timestamp: 2020-03-15T00:00:00.000Z,
  price: 550000
}
{
  _id: ObjectId('699fcbb16e38eb3e62cb8ab2'),
  vehicle_name: 'Ashok Leyland Dost',
  model: 'commercial',
  category: 'lorry',
  variants: [
    'diesel'
  ],
  manufacturer: 'Ashok Leyland',
  performance: 8,
  timestamp: 2019-08-20T00:00:00.000Z,
  price: 900000
}
{
  _id: ObjectId('699fcbb16e38eb3e62cb8ab3'),
```

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

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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

```
>_MONGOSH
> db.two_wheelers.find(
  { performance: { $gt: 5 } },
  { bike_name: 1, category: 1, manufacturer: 1, _id: 0 }
)
< {
  bike_name: 'Yamaha R15',
  category: '150cc',
  manufacturer: 'Yamaha'
}
{
  bike_name: 'Honda Activa',
  category: '125cc',
  manufacturer: 'Honda'
}
{
  bike_name: 'Royal Enfield Classic 350',
  category: '200cc',
  manufacturer: 'Royal Enfield'
}
{
  bike_name: 'Bajaj Pulsar 150',
  category: '150cc',
  manufacturer: 'Bajaj'
}
{
  bike_name: 'TVS Jupiter',
  category: '125cc',
  manufacturer: 'TVS'
}
}
```

```
>_MONGOSH
> db.four_wheelers.find(
  { performance: { $gt: 5 } },
  { vehicle_name: 1, category: 1, manufacturer: 1, _id: 0 }
)
< {
  vehicle_name: 'Maruti Swift',
  category: 'car',
  manufacturer: 'Maruti Suzuki'
}
{
  vehicle_name: 'Tata Ace',
  category: 'mini truck',
  manufacturer: 'Tata'
}
{
  vehicle_name: 'Ashok Leyland Dost',
  category: 'lorry',
  manufacturer: 'Ashok Leyland'
}
{
  vehicle_name: 'Volvo Bus',
  category: 'bus',
  manufacturer: 'Volvo'
}
{
  vehicle_name: 'Tata Prima',
  category: 'heavy truck',
  manufacturer: 'Tata'
}
}
```


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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

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'.



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

The screenshot shows a terminal window with a dark blue background. The text is white. It shows the MongoDB command prompt where the user has switched to the 'test' database and then to the 'animal' database. The prompt is now 'animal>'. A small green circular logo with a white 'G' is visible in the top right corner of the terminal window. At the bottom of the image, there is a Windows taskbar with various icons and a system tray showing the date and time as 20:53 on 26-02-2026.

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

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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

```
> db.createCollection("Animals")
< { ok: 1 }
> show databases
< VehicleDb 88.00 KiB
  admin      40.00 KiB
  animal     8.00 KiB
  config     96.00 KiB
  employee   8.00 KiB
  local     108.00 KiB
  test       64.00 KiB
  testDB    2.67 MiB
animal>
```

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

```
> db.createCollection("wild_animals", { capped: true, size: 102400 })
< { ok: 1 }
> 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

Date: 26-02-2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

```
>_MONGOSH

> db.wild_animals.insertMany([
  {
    animal_name: "Lion",
    nature: "harm",
    favorite_foods: ["meat", "deer"],
    care_taker_name: "Ramesh",
    life_span: 14,
    timestamp: new Date("2021-05-10"),
    expenses: 50000
  },
  {
    animal_name: "Tiger",
    nature: "harm",
    favorite_foods: ["meat", "deer"],
    care_taker_name: "Suresh",
    life_span: 16,
    timestamp: new Date("2020-03-12"),
    expenses: 60000
  },
  {
    animal_name: "Elephant",
    nature: "harmless",
    favorite_foods: ["grass", "fruits"],
    care_taker_name: "Ramesh",
    life_span: 60,
    timestamp: new Date("2019-01-20"),
    expenses: 80000
  },
  {
```

```
    animal_name: "Deer",
    nature: "harmless",
    favorite_foods: ["grass", "leaves"],
    care_taker_name: "Anil",
    life_span: 12,
    timestamp: new Date("2022-07-18"),
    expenses: 20000
  },
  {
    animal_name: "Wolf",
    nature: "harm",
    favorite_foods: ["meat", "rabbits"],
    care_taker_name: "Suresh",
    life_span: 13,
    timestamp: new Date("2023-02-05"),
    expenses: 30000
  }
])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('69a0679474ce2fa6e958bbb7'),
    '1': ObjectId('69a0679474ce2fa6e958bbb8'),
    '2': ObjectId('69a0679474ce2fa6e958bbb9'),
    '3': ObjectId('69a0679474ce2fa6e958bbba'),
    '4': ObjectId('69a0679474ce2fa6e958bbb')
  }
}
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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

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.

```
>_MONGOSH
> db.domestic_animals.insertMany([
  {
    animal_name: "Dog",
    gender: "male",
    favorite_foods: ["meat", "rice"],
    animal_petname: "Buddy",
    life_span: 12,
    timestamp: new Date("2022-01-10"),
    expenses: 15000
  },
  {
    animal_name: "Cat",
    gender: "female",
    favorite_foods: ["milk", "fish"],
    animal_petname: "Kitty",
    life_span: 14,
    timestamp: new Date("2021-06-15"),
    expenses: 10000
  },
  {
    animal_name: "Cow",
    gender: "female",
    favorite_foods: ["grass"],
    animal_petname: "Ganga",
    life_span: 20,
    timestamp: new Date("2020-03-20"),
    expenses: 25000
  },
  {
    animal_name: "Rabbit",
    gender: "male",
    favorite_foods: ["carrots", "hay"],
    animal_petname: "Bunny",
    life_span: 8,
    timestamp: new Date("2023-05-01"),
    expenses: 8000
  },
  {
    animal_name: "Deer",
    gender: "female",
    favorite_foods: ["grass", "leaves"],
    animal_petname: "Nisha",
    life_span: 15,
    timestamp: new Date("2021-12-01"),
    expenses: 12000
  }
])
```

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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

```
>_MONGOSH
{
  animal_name: "Goat",
  gender: "male",
  favorite_foods: ["grass", "leaves"],
  animal_petname: "Chintu",
  life_span: 10,
  timestamp: new Date("2023-04-12"),
  expenses: 8000
},
{
  animal_name: "Rabbit",
  gender: "female",
  favorite_foods: ["carrot", "grass"],
  animal_petname: "Snowy",
  life_span: 9,
  timestamp: new Date("2024-01-05"),
  expenses: 5000
}
})
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('69a0688874ce2fa6e958bbc1'),
    '1': ObjectId('69a0688874ce2fa6e958bbc2'),
    '2': ObjectId('69a0688874ce2fa6e958bbc3'),
    '3': ObjectId('69a0688874ce2fa6e958bbc4'),
    '4': ObjectId('69a0688874ce2fa6e958bbc5')
  }
}
animal>
```

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

Date: 26-02-2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

```
>_MONGOSH
> db.wild_animals.find()
< {
  _id: ObjectId('69a0679474ce2fa6e958bbb7'),
  animal_name: 'Lion',
  nature: 'harm',
  favorite_foods: [
    'meat',
    'deer'
  ],
  care_taker_name: 'Ramesh',
  life_span: 14,
  timestamp: 2021-05-10T00:00:00.000Z,
  expenses: 50000
}
{
  _id: ObjectId('69a0679474ce2fa6e958bbb8'),
  animal_name: 'Tiger',
  nature: 'harm',
  favorite_foods: [
    'meat',
    'deer'
  ],
  care_taker_name: 'Suresh',
  life_span: 16,
  timestamp: 2020-03-12T00:00:00.000Z,
  expenses: 60000
}
{
  _id: ObjectId('69a0679474ce2fa6e958bbb9'),
```

```
>_MONGOSH
> db.domestic_animals.find()
< {
  _id: ObjectId('69a0686674ce2fa6e958bbbc'),
  animal_name: 'Dog',
  gender: 'male',
  favorite_foods: [
    'meat',
    'rice'
  ],
  animal_petname: 'Buddy',
  life_span: 12,
  timestamp: 2022-01-10T00:00:00.000Z,
  expenses: 15000
}
{
  _id: ObjectId('69a0686674ce2fa6e958bbbd'),
  animal_name: 'Cat',
  gender: 'female',
  favorite_foods: [
    'milk',
    'fish'
  ],
  animal_petname: 'Kitty',
  life_span: 14,
  timestamp: 2021-06-15T00:00:00.000Z,
  expenses: 10000
}
{
  _id: ObjectId('69a0686674ce2fa6e958bbbe'),
```

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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

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

```
> 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: 'Deer',
  expenses: 20000
}
{
  animal_name: 'Wolf',
  expenses: 30000
}
}
animal>
```

```
> MONGOSH
> db.domestic_animals.find({}, { animal_name: 1, expenses: 1, _id: 0 })
< {
  animal_name: 'Dog',
  expenses: 15000
}
{
  animal_name: 'Cat',
  expenses: 10000
}
{
  animal_name: 'Cow',
  expenses: 25000
}
{
  animal_name: 'Goat',
  expenses: 8000
}
{
  animal_name: 'Rabbit',
  expenses: 5000
}
{
  animal_name: 'Dog',
  expenses: 15000
}
{
  animal_name: 'Cat',
  expenses: 10000
}
}
```

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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

```
> db.domestic_animals.find({ life_span: 10 })
< {
  _id: ObjectId('69a068674ce2fa6e958bbbf'),
  animal_name: 'Goat',
  gender: 'male',
  favorite_foods: [
    'grass',
    'leaves'
  ],
  animal_petname: 'Chintu',
  life_span: 10,
  timestamp: 2023-04-12T00:00:00.000Z,
  expenses: 8000
}
{
  _id: ObjectId('69a0688874ce2fa6e958bbc4'),
  animal_name: 'Goat',
  gender: 'male',
  favorite_foods: [
    'grass',
    'leaves'
  ],
  animal_petname: 'Chintu',
  life_span: 10,
  timestamp: 2023-04-12T00:00:00.000Z,
  expenses: 8000
}
animal>
```

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

```
> db.wild_animals.find({ care_taker_name: "Ramesh" })
< {
  _id: ObjectId('69a0679474ce2fa6e958bbb7'),
  animal_name: 'Lion',
  nature: 'harm',
  favorite_foods: [
    'meat',
    'deer'
  ],
  care_taker_name: 'Ramesh',
  life_span: 14,
  timestamp: 2021-05-10T00:00:00.000Z,
  expenses: 50000
}
{
  _id: ObjectId('69a0679474ce2fa6e958bbb9'),
  animal_name: 'Elephant',
  nature: 'harmless',
  favorite_foods: [
    'grass',
    'fruits'
  ],
  care_taker_name: 'Ramesh',
  life_span: 60,
  timestamp: 2019-01-20T00:00:00.000Z,
  expenses: 80000
}
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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

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

```
>_MONGOSH
> 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',
    'deer'
  ],
  expenses: 60000
}
{
  animal_name: 'Elephant',
  favorite_foods: [
    'grass',
    'fruits'
  ],
  expenses: 80000
}
{
```

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: Samika Sanjeev Naidu

Reg. no.: 23BCE8855

```
>_MONGOSH
> db.domestic_animals.find(
  { life_span: { $gt: 5 } },
  { animal_name: 1, favorite_foods: 1, expenses: 1, _id: 0 }
)
< {
  animal_name: 'Dog',
  favorite_foods: [
    'meat',
    'rice'
  ],
  expenses: 15000
}
{
  animal_name: 'Cat',
  favorite_foods: [
    'milk',
    'fish'
  ],
  expenses: 10000
}
{
  animal_name: 'Cow',
  favorite_foods: [
    'grass'
  ],
  expenses: 25000
}
{
  animal_name: 'Goat',
```

