VIT-AP UNIVERSITY, ANDHRA PRADESH

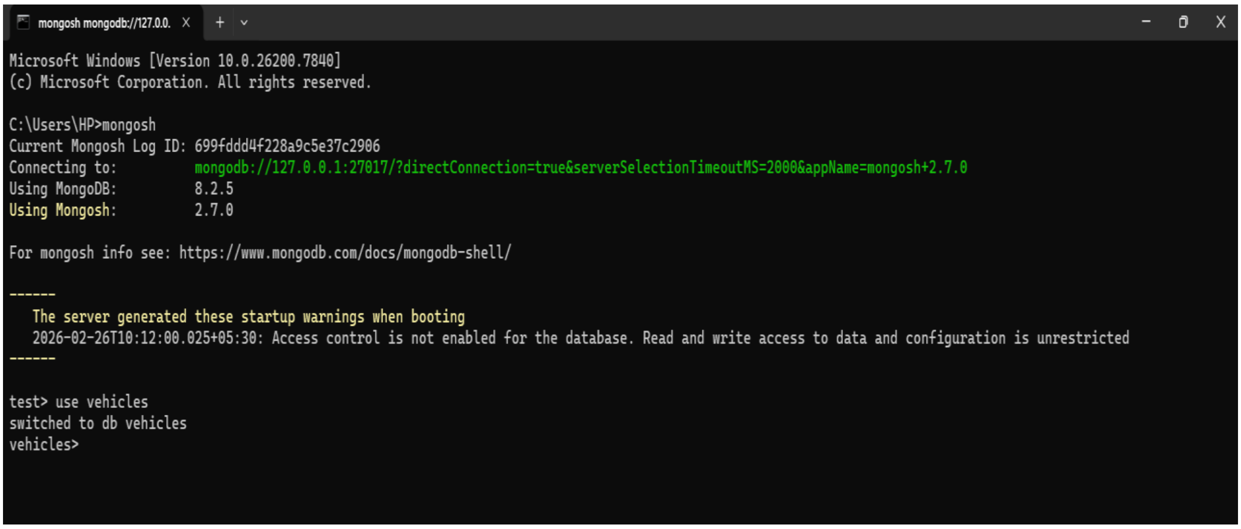
Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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".

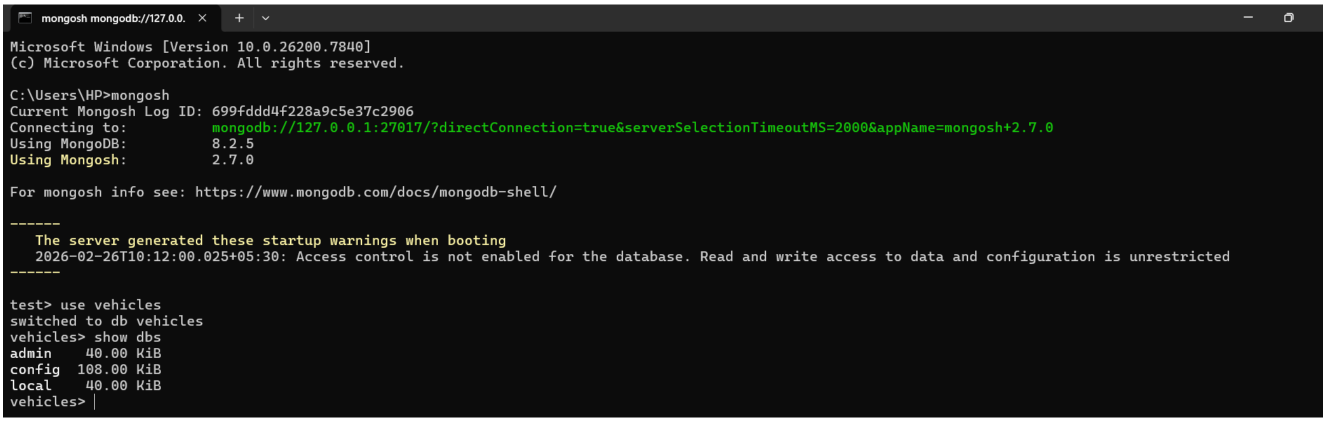
Output:



```
Microsoft Windows [Version 10.0.26200.7840]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>mongosh
Current Mongosh Log ID: 699fddd4f228a9c5e37c2906
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
   2026-02-26T10:12:00.025+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.

Output:



```
Microsoft Windows [Version 10.0.26200.7840]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>mongosh
Current Mongosh Log ID: 699fddd4f228a9c5e37c2906
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
   2026-02-26T10:12:00.025+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  108.00 KiB
local    40.00 KiB
vehicles> |
```

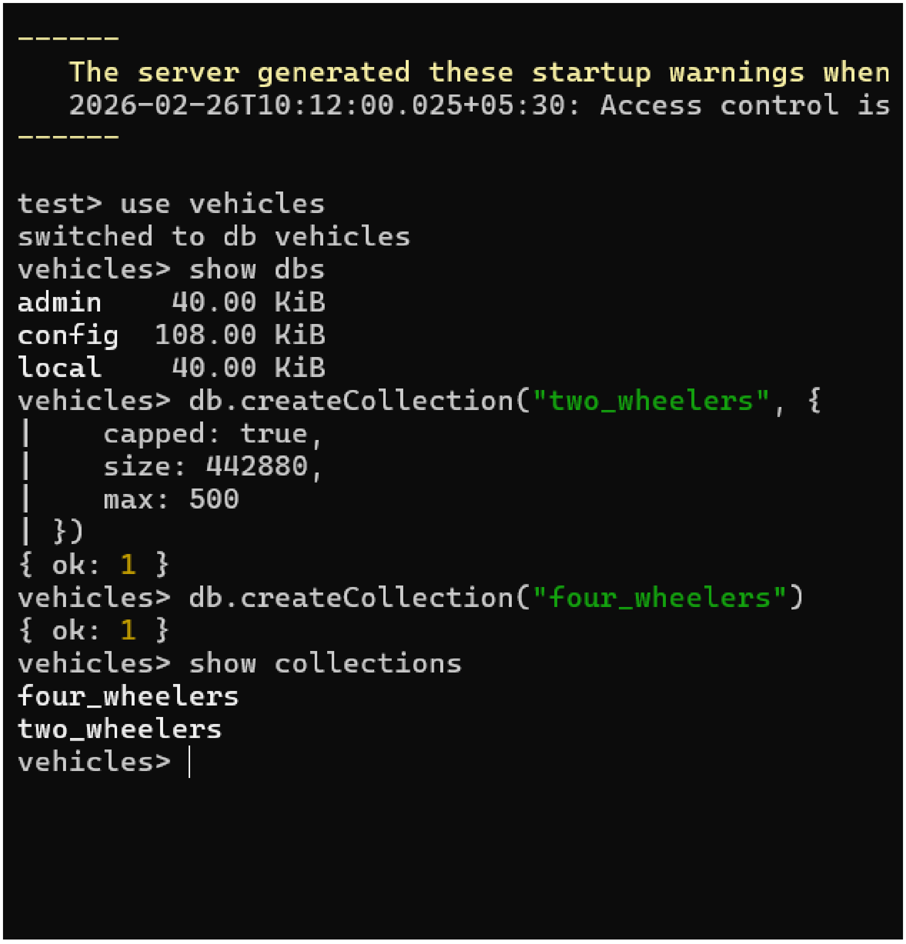
VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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

Output:



```
------
   The server generated these startup warnings when
   2026-02-26T10:12:00.025+05:30: Access control is
------

test> use vehicles
switched to db vehicles
vehicles> show dbs
admin     40.00 KiB
config   108.00 KiB
local     40.00 KiB
vehicles> db.createCollection("two_wheelers", {
|     capped: true,
|     size: 442880,
|     max: 500
| })
{ ok: 1 }
vehicles> db.createCollection("four_wheelers")
{ ok: 1 }
vehicles> show collections
four_wheelers
two_wheelers
vehicles>
```

VIT-AP UNIVERSITY, ANDHRA PRADESH

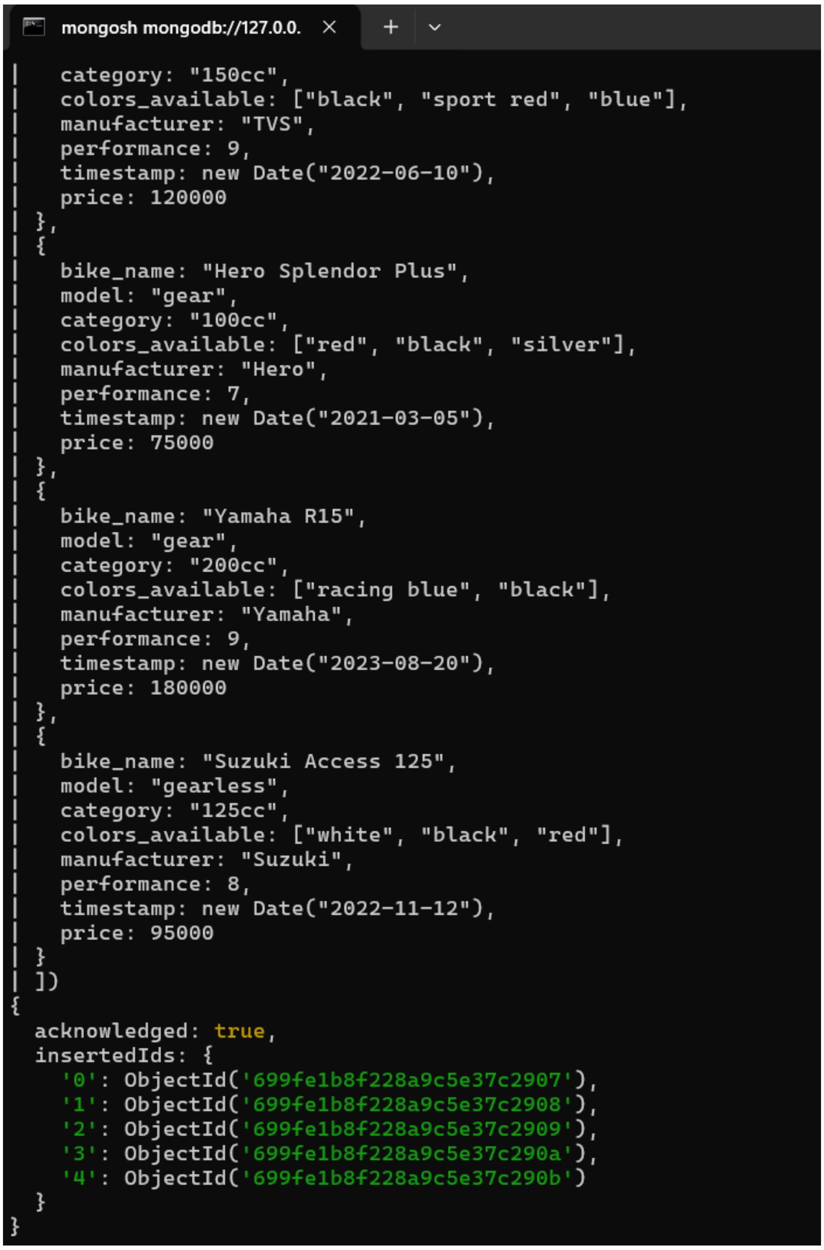
Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

---

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.

Output:



```
|    category: "150cc",
|    colors_available: ["black", "sport red", "blue"],
|    manufacturer: "TVS",
|    performance: 9,
|    timestamp: new Date("2022-06-10"),
|    price: 120000
|  },
|  {
|    bike_name: "Hero Splendor Plus",
|    model: "gear",
|    category: "100cc",
|    colors_available: ["red", "black", "silver"],
|    manufacturer: "Hero",
|    performance: 7,
|    timestamp: new Date("2021-03-05"),
|    price: 75000
|  },
|  {
|    bike_name: "Yamaha R15",
|    model: "gear",
|    category: "200cc",
|    colors_available: ["racing blue", "black"],
|    manufacturer: "Yamaha",
|    performance: 9,
|    timestamp: new Date("2023-08-20"),
|    price: 180000
|  },
|  {
|    bike_name: "Suzuki Access 125",
|    model: "gearless",
|    category: "125cc",
|    colors_available: ["white", "black", "red"],
|    manufacturer: "Suzuki",
|    performance: 8,
|    timestamp: new Date("2022-11-12"),
|    price: 95000
|  }
|  ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fe1b8f228a9c5e37c2907'),
    '1': ObjectId('699fe1b8f228a9c5e37c2908'),
    '2': ObjectId('699fe1b8f228a9c5e37c2909'),
    '3': ObjectId('699fe1b8f228a9c5e37c290a'),
    '4': ObjectId('699fe1b8f228a9c5e37c290b')
  }
}
```

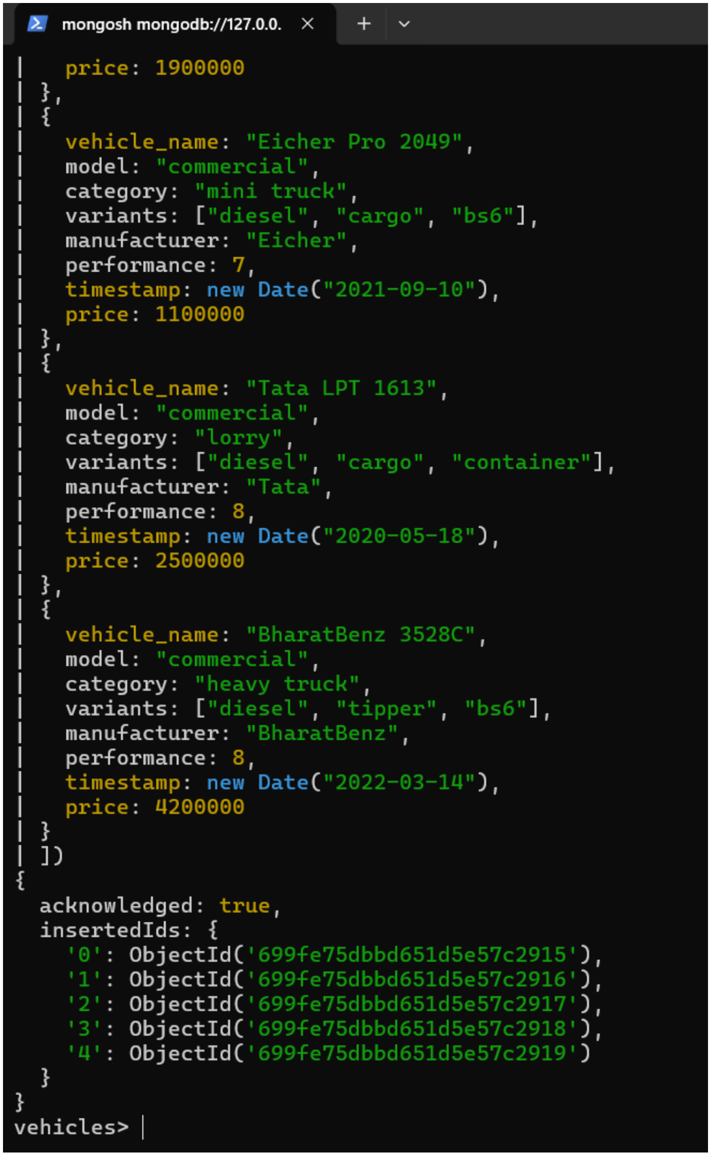
VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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.

Output:



```
|   price: 1900000
| },
| {
|   vehicle_name: "Eicher Pro 2049",
|   model: "commercial",
|   category: "mini truck",
|   variants: ["diesel", "cargo", "bs6"],
|   manufacturer: "Eicher",
|   performance: 7,
|   timestamp: new Date("2021-09-10"),
|   price: 1100000
| },
| {
|   vehicle_name: "Tata LPT 1613",
|   model: "commercial",
|   category: "lorry",
|   variants: ["diesel", "cargo", "container"],
|   manufacturer: "Tata",
|   performance: 8,
|   timestamp: new Date("2020-05-18"),
|   price: 2500000
| },
| {
|   vehicle_name: "BharatBenz 3528C",
|   model: "commercial",
|   category: "heavy truck",
|   variants: ["diesel", "tipper", "bs6"],
|   manufacturer: "BharatBenz",
|   performance: 8,
|   timestamp: new Date("2022-03-14"),
|   price: 4200000
| }
| ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fe75dbbd651d5e57c2915'),
    '1': ObjectId('699fe75dbbd651d5e57c2916'),
    '2': ObjectId('699fe75dbbd651d5e57c2917'),
    '3': ObjectId('699fe75dbbd651d5e57c2918'),
    '4': ObjectId('699fe75dbbd651d5e57c2919')
  }
}
vehicles> |
```

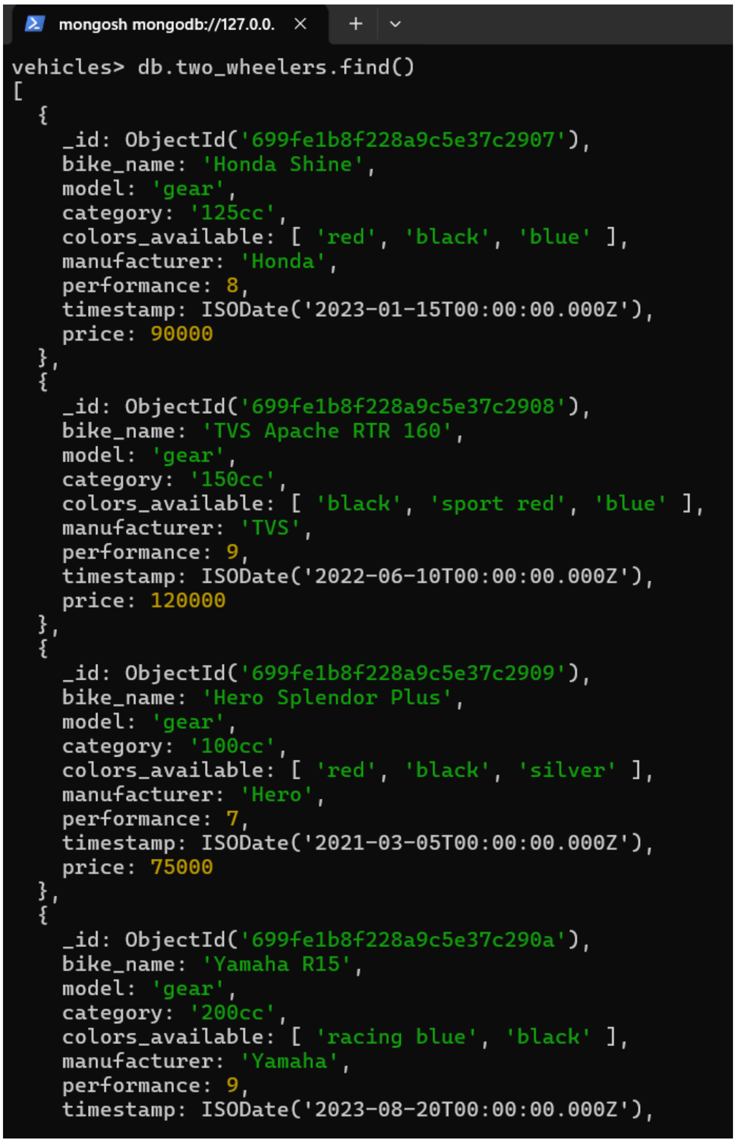
VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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

Output:

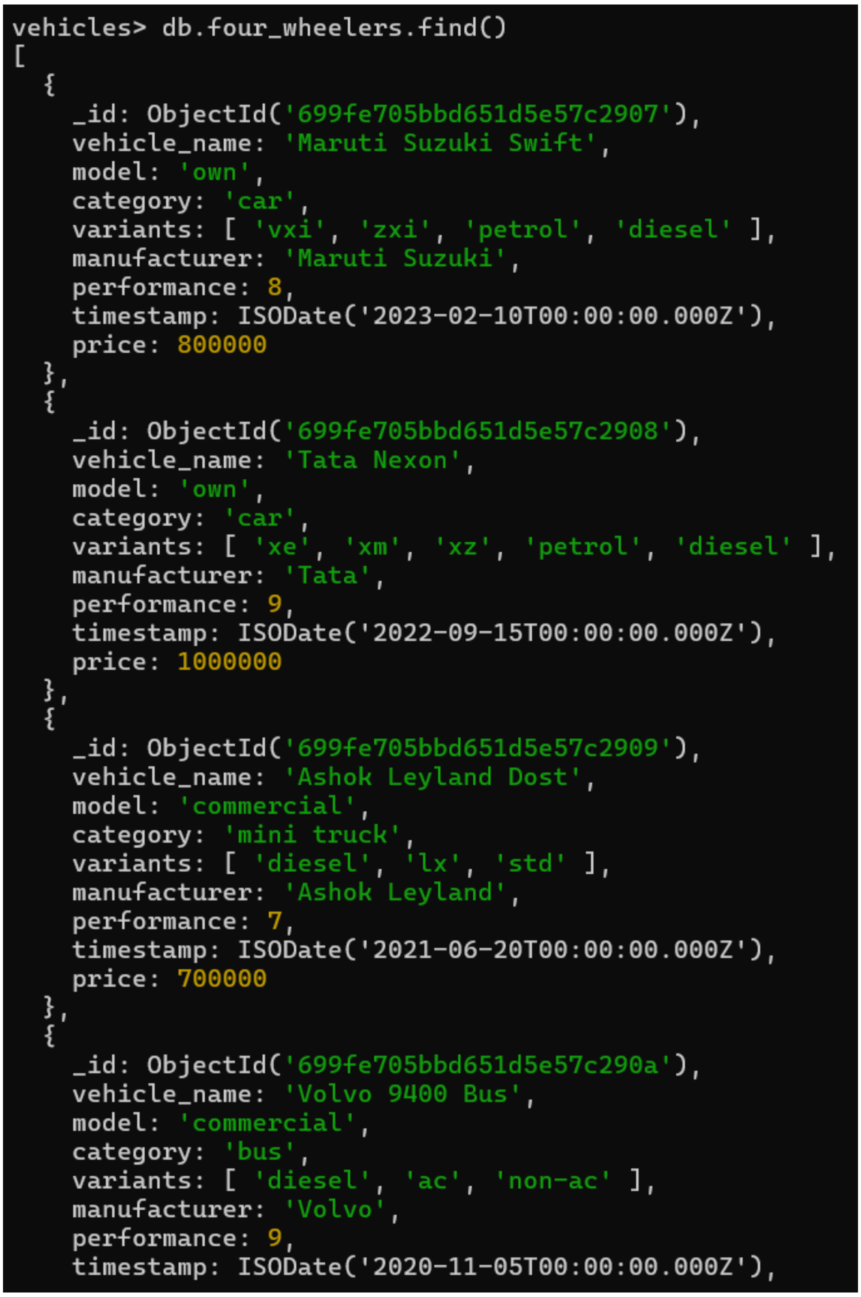


```
vehicles> db.two_wheelers.find()
[
  {
    _id: ObjectId('699fe1b8f228a9c5e37c2907'),
    bike_name: 'Honda Shine',
    model: 'gear',
    category: '125cc',
    colors_available: [ 'red', 'black', 'blue' ],
    manufacturer: 'Honda',
    performance: 8,
    timestamp: ISODate('2023-01-15T00:00:00.000Z'),
    price: 90000
  },
  {
    _id: ObjectId('699fe1b8f228a9c5e37c2908'),
    bike_name: 'TVS Apache RTR 160',
    model: 'gear',
    category: '150cc',
    colors_available: [ 'black', 'sport red', 'blue' ],
    manufacturer: 'TVS',
    performance: 9,
    timestamp: ISODate('2022-06-10T00:00:00.000Z'),
    price: 120000
  },
  {
    _id: ObjectId('699fe1b8f228a9c5e37c2909'),
    bike_name: 'Hero Splendor Plus',
    model: 'gear',
    category: '100cc',
    colors_available: [ 'red', 'black', 'silver' ],
    manufacturer: 'Hero',
    performance: 7,
    timestamp: ISODate('2021-03-05T00:00:00.000Z'),
    price: 75000
  },
  {
    _id: ObjectId('699fe1b8f228a9c5e37c290a'),
    bike_name: 'Yamaha R15',
    model: 'gear',
    category: '200cc',
    colors_available: [ 'racing blue', 'black' ],
    manufacturer: 'Yamaha',
    performance: 9,
    timestamp: ISODate('2023-08-20T00:00:00.000Z'),
```

VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

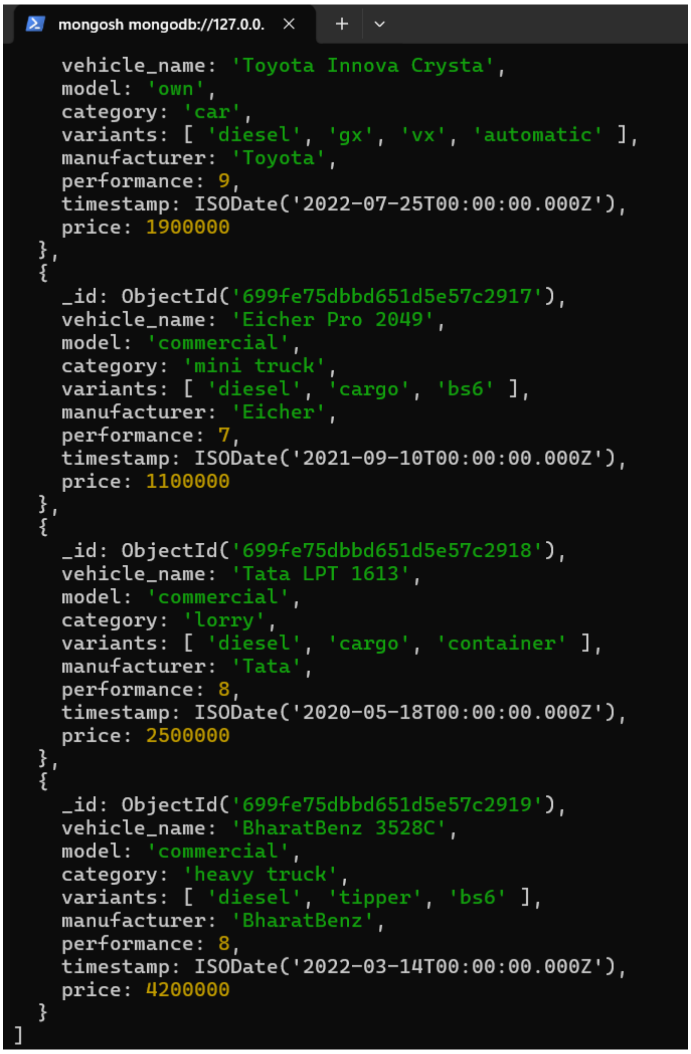


```
vehicles> db.four_wheelers.find()
[
  {
    _id: ObjectId('699fe705bbd651d5e57c2907'),
    vehicle_name: 'Maruti Suzuki Swift',
    model: 'own',
    category: 'car',
    variants: [ 'vxi', 'zxi', 'petrol', 'diesel' ],
    manufacturer: 'Maruti Suzuki',
    performance: 8,
    timestamp: ISODate('2023-02-10T00:00:00.000Z'),
    price: 800000
  },
  {
    _id: ObjectId('699fe705bbd651d5e57c2908'),
    vehicle_name: 'Tata Nexon',
    model: 'own',
    category: 'car',
    variants: [ 'xe', 'xm', 'xz', 'petrol', 'diesel' ],
    manufacturer: 'Tata',
    performance: 9,
    timestamp: ISODate('2022-09-15T00:00:00.000Z'),
    price: 1000000
  },
  {
    _id: ObjectId('699fe705bbd651d5e57c2909'),
    vehicle_name: 'Ashok Leyland Dost',
    model: 'commercial',
    category: 'mini truck',
    variants: [ 'diesel', 'lx', 'std' ],
    manufacturer: 'Ashok Leyland',
    performance: 7,
    timestamp: ISODate('2021-06-20T00:00:00.000Z'),
    price: 700000
  },
  {
    _id: ObjectId('699fe705bbd651d5e57c290a'),
    vehicle_name: 'Volvo 9400 Bus',
    model: 'commercial',
    category: 'bus',
    variants: [ 'diesel', 'ac', 'non-ac' ],
    manufacturer: 'Volvo',
    performance: 9,
    timestamp: ISODate('2020-11-05T00:00:00.000Z'),
```

VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304



```
    vehicle_name: 'Toyota Innova Crysta',
    model: 'own',
    category: 'car',
    variants: [ 'diesel', 'gx', 'vx', 'automatic' ],
    manufacturer: 'Toyota',
    performance: 9,
    timestamp: ISODate('2022-07-25T00:00:00.000Z'),
    price: 1900000
  },
  {
    _id: ObjectId('699fe75dbbd651d5e57c2917'),
    vehicle_name: 'Eicher Pro 2049',
    model: 'commercial',
    category: 'mini truck',
    variants: [ 'diesel', 'cargo', 'bs6' ],
    manufacturer: 'Eicher',
    performance: 7,
    timestamp: ISODate('2021-09-10T00:00:00.000Z'),
    price: 1100000
  },
  {
    _id: ObjectId('699fe75dbbd651d5e57c2918'),
    vehicle_name: 'Tata LPT 1613',
    model: 'commercial',
    category: 'lorry',
    variants: [ 'diesel', 'cargo', 'container' ],
    manufacturer: 'Tata',
    performance: 8,
    timestamp: ISODate('2020-05-18T00:00:00.000Z'),
    price: 2500000
  },
  {
    _id: ObjectId('699fe75dbbd651d5e57c2919'),
    vehicle_name: 'BharatBenz 3528C',
    model: 'commercial',
    category: 'heavy truck',
    variants: [ 'diesel', 'tipper', 'bs6' ],
    manufacturer: 'BharatBenz',
    performance: 8,
    timestamp: ISODate('2022-03-14T00:00:00.000Z'),
    price: 4200000
  }
]
```

VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

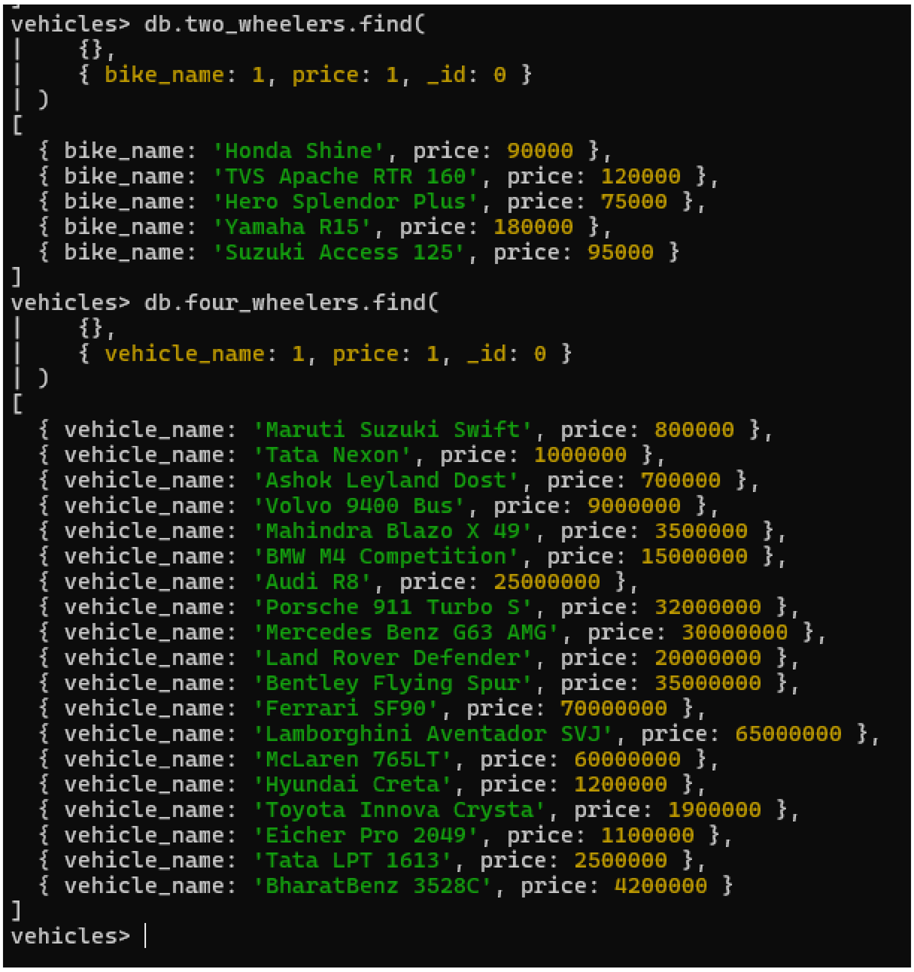
Branch/ Class: B.Tech/M.Tech **Date:**

Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE

Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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

Output:



```
vehicles> db.two_wheelers.find(
|     {},
|     { bike_name: 1, price: 1, _id: 0 }
| )
[
  { bike_name: 'Honda Shine', price: 90000 },
  { bike_name: 'TVS Apache RTR 160', price: 120000 },
  { bike_name: 'Hero Splendor Plus', price: 75000 },
  { bike_name: 'Yamaha R15', price: 180000 },
  { bike_name: 'Suzuki Access 125', price: 95000 }
]
vehicles> db.four_wheelers.find(
|     {},
|     { vehicle_name: 1, price: 1, _id: 0 }
| )
[
  { vehicle_name: 'Maruti Suzuki Swift', price: 800000 },
  { vehicle_name: 'Tata Nexon', price: 1000000 },
  { vehicle_name: 'Ashok Leyland Dost', price: 700000 },
  { vehicle_name: 'Volvo 9400 Bus', price: 9000000 },
  { vehicle_name: 'Mahindra Blazo X 49', price: 3500000 },
  { vehicle_name: 'BMW M4 Competition', price: 15000000 },
  { vehicle_name: 'Audi R8', price: 25000000 },
  { vehicle_name: 'Porsche 911 Turbo S', price: 32000000 },
  { vehicle_name: 'Mercedes Benz G63 AMG', price: 30000000 },
  { vehicle_name: 'Land Rover Defender', price: 20000000 },
  { vehicle_name: 'Bentley Flying Spur', price: 35000000 },
  { vehicle_name: 'Ferrari SF90', price: 70000000 },
  { vehicle_name: 'Lamborghini Aventador SVJ', price: 65000000 },
  { vehicle_name: 'McLaren 765LT', price: 60000000 },
  { vehicle_name: 'Hyundai Creta', price: 1200000 },
  { vehicle_name: 'Toyota Innova Crysta', price: 1900000 },
  { vehicle_name: 'Eicher Pro 2049', price: 1100000 },
  { vehicle_name: 'Tata LPT 1613', price: 2500000 },
  { vehicle_name: 'BharatBenz 3528C', price: 4200000 }
]
vehicles> |
```

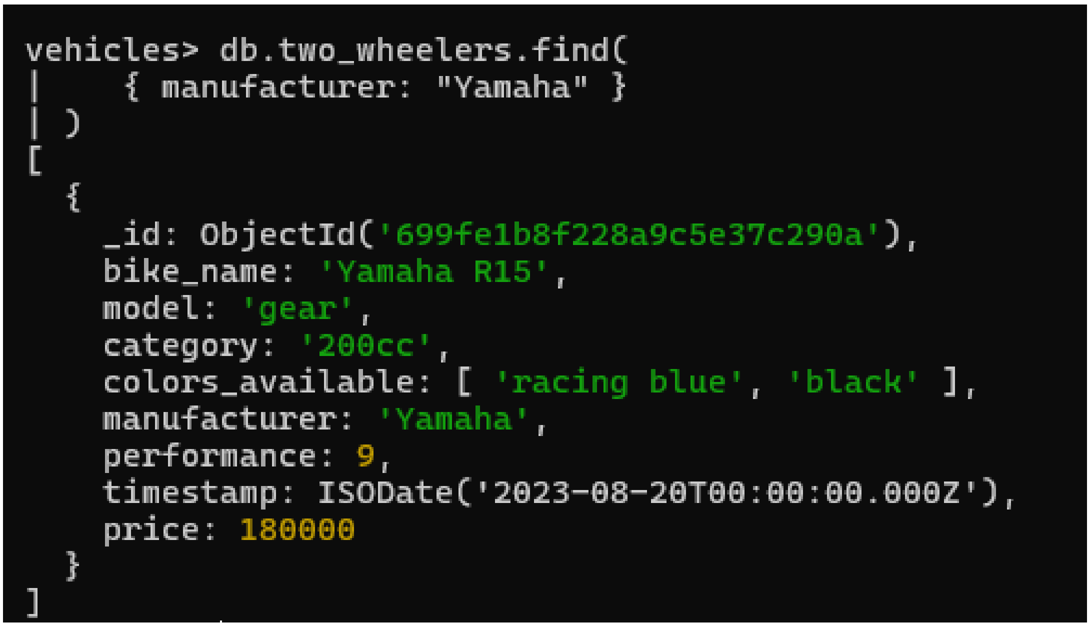
VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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

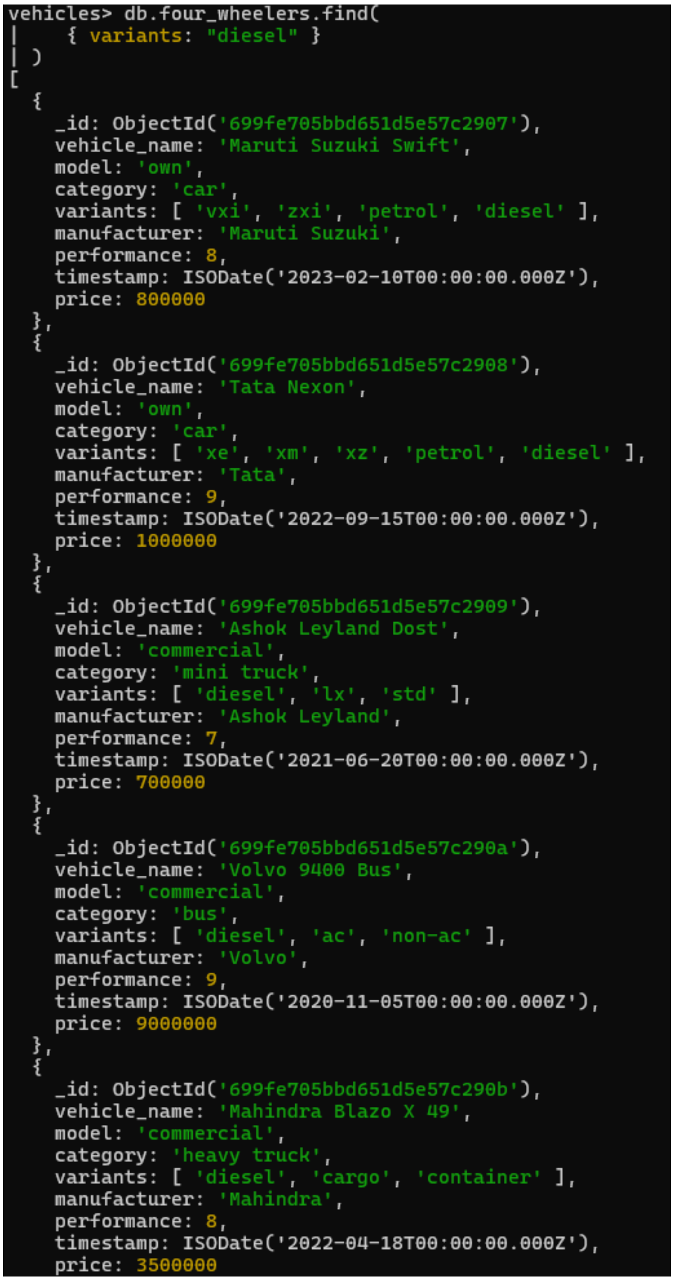
Output:



```
vehicles> db.two_wheelers.find(
|     { manufacturer: "Yamaha" }
| )
[
  {
    _id: ObjectId('699fe1b8f228a9c5e37c290a'),
    bike_name: 'Yamaha R15',
    model: 'gear',
    category: '200cc',
    colors_available: [ 'racing blue', 'black' ],
    manufacturer: 'Yamaha',
    performance: 9,
    timestamp: ISODate('2023-08-20T00:00:00.000Z'),
    price: 180000
  }
]
```

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

Output:



```
vehicles> db.four_wheelers.find(
|     { variants: "diesel" }
| )
[
  {
    _id: ObjectId('699fe705bbd651d5e57c2907'),
    vehicle_name: 'Maruti Suzuki Swift',
    model: 'own',
    category: 'car',
    variants: [ 'vxi', 'zxi', 'petrol', 'diesel' ],
    manufacturer: 'Maruti Suzuki',
    performance: 8,
    timestamp: ISODate('2023-02-10T00:00:00.000Z'),
    price: 800000
  },
  {
    _id: ObjectId('699fe705bbd651d5e57c2908'),
    vehicle_name: 'Tata Nexon',
    model: 'own',
    category: 'car',
    variants: [ 'xe', 'xm', 'xz', 'petrol', 'diesel' ],
    manufacturer: 'Tata',
    performance: 9,
    timestamp: ISODate('2022-09-15T00:00:00.000Z'),
    price: 1000000
  },
  {
    _id: ObjectId('699fe705bbd651d5e57c2909'),
    vehicle_name: 'Ashok Leyland Dost',
    model: 'commercial',
    category: 'mini truck',
    variants: [ 'diesel', 'lx', 'std' ],
    manufacturer: 'Ashok Leyland',
    performance: 7,
    timestamp: ISODate('2021-06-20T00:00:00.000Z'),
    price: 700000
  },
  {
    _id: ObjectId('699fe705bbd651d5e57c290a'),
    vehicle_name: 'Volvo 9400 Bus',
    model: 'commercial',
    category: 'bus',
    variants: [ 'diesel', 'ac', 'non-ac' ],
    manufacturer: 'Volvo',
    performance: 9,
    timestamp: ISODate('2020-11-05T00:00:00.000Z'),
    price: 9000000
  },
  {
    _id: ObjectId('699fe705bbd651d5e57c290b'),
    vehicle_name: 'Mahindra Blazo X 49',
    model: 'commercial',
    category: 'heavy truck',
    variants: [ 'diesel', 'cargo', 'container' ],
    manufacturer: 'Mahindra',
    performance: 8,
    timestamp: ISODate('2022-04-18T00:00:00.000Z'),
    price: 3500000
```

VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

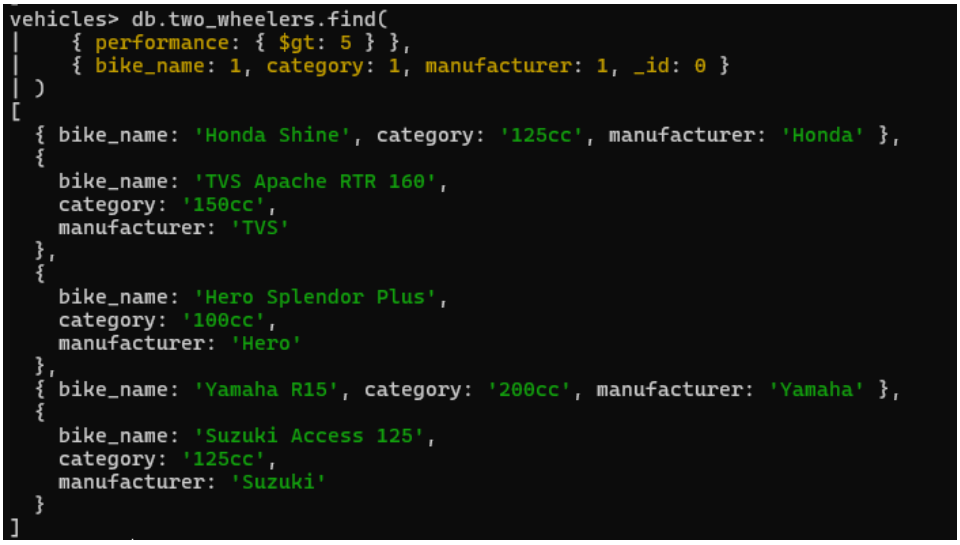
Branch/ Class: B.Tech/M.Tech **Date:**

Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE

Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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

Output:

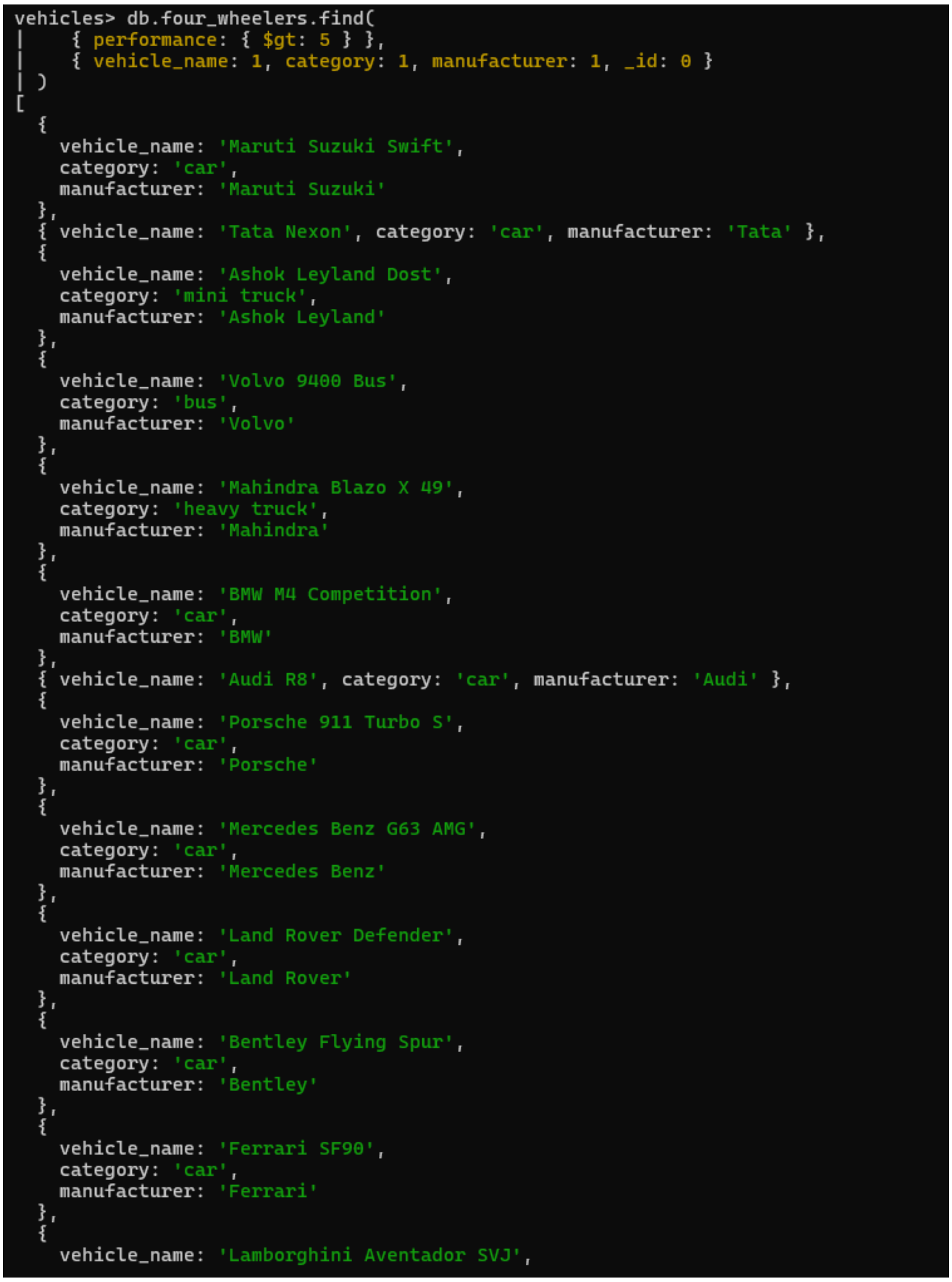


```
vehicles> db.two_wheelers.find(
|     { performance: { $gt: 5 } },
|     { bike_name: 1, category: 1, manufacturer: 1, _id: 0 }
| )
[
  { bike_name: 'Honda Shine', category: '125cc', manufacturer: 'Honda' },
  {
    bike_name: 'TVS Apache RTR 160',
    category: '150cc',
    manufacturer: 'TVS'
  },
  {
    bike_name: 'Hero Splendor Plus',
    category: '100cc',
    manufacturer: 'Hero'
  },
  { bike_name: 'Yamaha R15', category: '200cc', manufacturer: 'Yamaha' },
  {
    bike_name: 'Suzuki Access 125',
    category: '125cc',
    manufacturer: 'Suzuki'
  }
]
```

VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304



```
vehicles> db.four_wheelers.find(
|    { performance: { $gt: 5 } },
|    { vehicle_name: 1, category: 1, manufacturer: 1, _id: 0 }
| )
[
  {
    vehicle_name: 'Maruti Suzuki Swift',
    category: 'car',
    manufacturer: 'Maruti Suzuki'
  },
  { vehicle_name: 'Tata Nexon', category: 'car', manufacturer: 'Tata' },
  {
    vehicle_name: 'Ashok Leyland Dost',
    category: 'mini truck',
    manufacturer: 'Ashok Leyland'
  },
  {
    vehicle_name: 'Volvo 9400 Bus',
    category: 'bus',
    manufacturer: 'Volvo'
  },
  {
    vehicle_name: 'Mahindra Blazo X 49',
    category: 'heavy truck',
    manufacturer: 'Mahindra'
  },
  {
    vehicle_name: 'BMW M4 Competition',
    category: 'car',
    manufacturer: 'BMW'
  },
  { vehicle_name: 'Audi R8', category: 'car', manufacturer: 'Audi' },
  {
    vehicle_name: 'Porsche 911 Turbo S',
    category: 'car',
    manufacturer: 'Porsche'
  },
  {
    vehicle_name: 'Mercedes Benz G63 AMG',
    category: 'car',
    manufacturer: 'Mercedes Benz'
  },
  {
    vehicle_name: 'Land Rover Defender',
    category: 'car',
    manufacturer: 'Land Rover'
  },
  {
    vehicle_name: 'Bentley Flying Spur',
    category: 'car',
    manufacturer: 'Bentley'
  },
  {
    vehicle_name: 'Ferrari SF90',
    category: 'car',
    manufacturer: 'Ferrari'
  },
  {
    vehicle_name: 'Lamborghini Aventador SVJ',
```

VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**

Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE

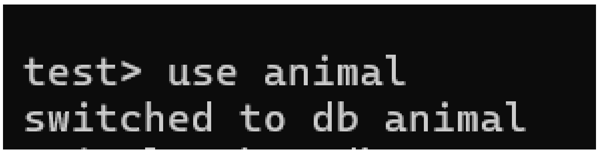
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

---

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'.

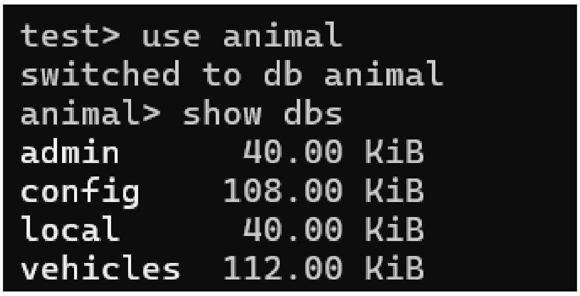
Output:



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

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

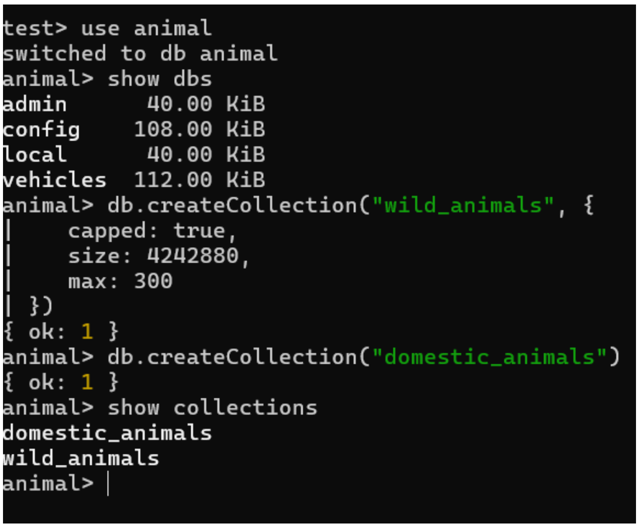
Output:



```
test> use animal
switched to db animal
animal> show dbs
admin       40.00 KiB
config     108.00 KiB
local       40.00 KiB
vehicles   112.00 KiB
```

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

Output:



```
test> use animal
switched to db animal
animal> show dbs
admin      40.00 KiB
config    108.00 KiB
local      40.00 KiB
vehicles  112.00 KiB
animal> db.createCollection("wild_animals", {
|     capped: true,
|     size: 4242880,
|     max: 300
| })
{ ok: 1 }
animal> db.createCollection("domestic_animals")
{ ok: 1 }
animal> show collections
domestic_animals
wild_animals
animal> |
```

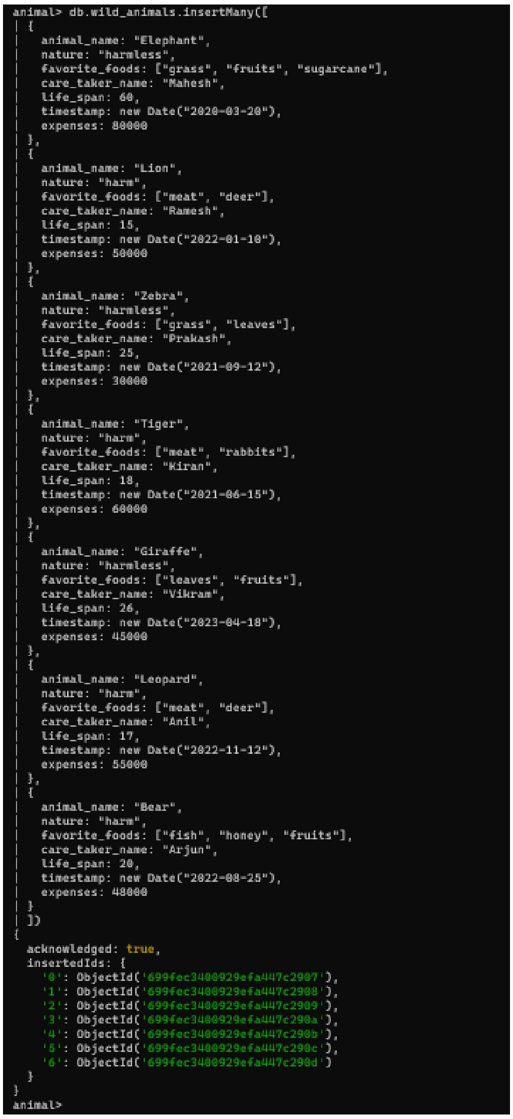
VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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.

Output:



```
animal> db.wild_animals.insertMany([
| {
|   animal_name: "Elephant",
|   nature: "harmless",
|   favorite_foods: ["grass", "fruits", "sugarcane"],
|   care_taker_name: "Mahesh",
|   life_span: 60,
|   timestamp: new Date("2020-03-20"),
|   expenses: 80000
| },
| {
|   animal_name: "Lion",
|   nature: "harm",
|   favorite_foods: ["meat", "deer"],
|   care_taker_name: "Ramesh",
|   life_span: 15,
|   timestamp: new Date("2022-01-10"),
|   expenses: 50000
| },
| {
|   animal_name: "Zebra",
|   nature: "harmless",
|   favorite_foods: ["grass", "leaves"],
|   care_taker_name: "Prakash",
|   life_span: 25,
|   timestamp: new Date("2021-09-12"),
|   expenses: 30000
| },
| {
|   animal_name: "Tiger",
|   nature: "harm",
|   favorite_foods: ["meat", "rabbits"],
|   care_taker_name: "Kiran",
|   life_span: 18,
|   timestamp: new Date("2021-06-15"),
|   expenses: 60000
| },
| {
|   animal_name: "Giraffe",
|   nature: "harmless",
|   favorite_foods: ["leaves", "fruits"],
|   care_taker_name: "Vikram",
|   life_span: 26,
|   timestamp: new Date("2023-04-18"),
|   expenses: 45000
| },
| {
|   animal_name: "Leopard",
|   nature: "harm",
|   favorite_foods: ["meat", "deer"],
|   care_taker_name: "Anil",
|   life_span: 17,
|   timestamp: new Date("2022-11-12"),
|   expenses: 55000
| },
| {
|   animal_name: "Bear",
|   nature: "harm",
|   favorite_foods: ["fish", "honey", "fruits"],
|   care_taker_name: "Arjun",
|   life_span: 20,
|   timestamp: new Date("2022-08-25"),
|   expenses: 48000
| }
| ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fec3400929efa447c2907'),
    '1': ObjectId('699fec3400929efa447c2908'),
    '2': ObjectId('699fec3400929efa447c2909'),
    '3': ObjectId('699fec3400929efa447c290a'),
    '4': ObjectId('699fec3400929efa447c290b'),
    '5': ObjectId('699fec3400929efa447c290c'),
    '6': ObjectId('699fec3400929efa447c290d')
  }
}
animal>
```

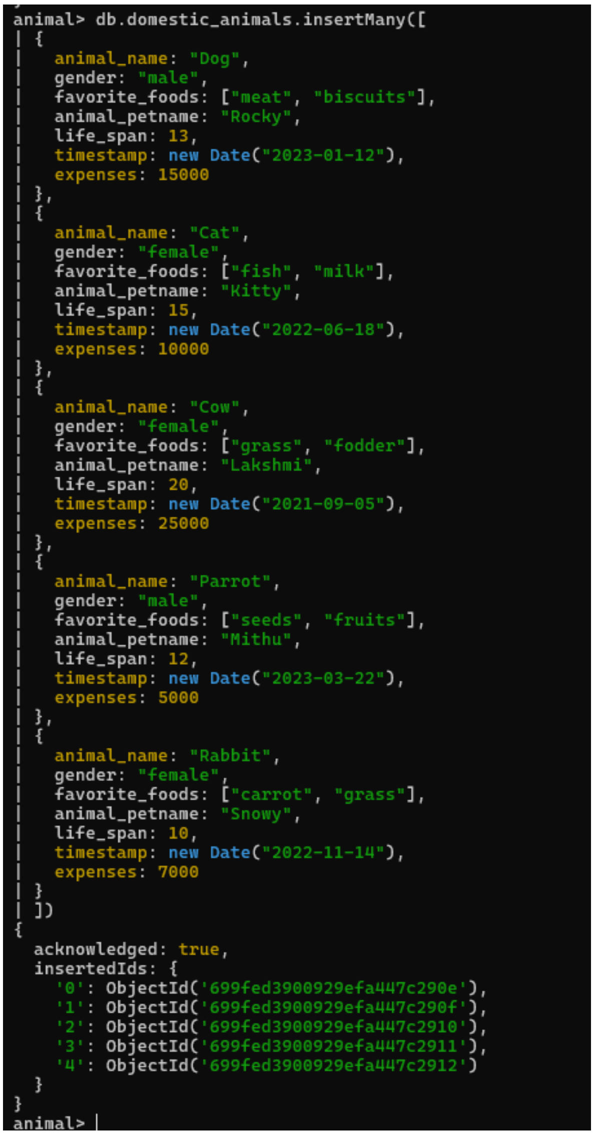
VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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.

Output:



```
animal> db.domestic_animals.insertMany([
| {
|   animal_name: "Dog",
|   gender: "male",
|   favorite_foods: ["meat", "biscuits"],
|   animal_petname: "Rocky",
|   life_span: 13,
|   timestamp: new Date("2023-01-12"),
|   expenses: 15000
| },
| {
|   animal_name: "Cat",
|   gender: "female",
|   favorite_foods: ["fish", "milk"],
|   animal_petname: "Kitty",
|   life_span: 15,
|   timestamp: new Date("2022-06-18"),
|   expenses: 10000
| },
| {
|   animal_name: "Cow",
|   gender: "female",
|   favorite_foods: ["grass", "fodder"],
|   animal_petname: "Lakshmi",
|   life_span: 20,
|   timestamp: new Date("2021-09-05"),
|   expenses: 25000
| },
| {
|   animal_name: "Parrot",
|   gender: "male",
|   favorite_foods: ["seeds", "fruits"],
|   animal_petname: "Mithu",
|   life_span: 12,
|   timestamp: new Date("2023-03-22"),
|   expenses: 5000
| },
| {
|   animal_name: "Rabbit",
|   gender: "female",
|   favorite_foods: ["carrot", "grass"],
|   animal_petname: "Snowy",
|   life_span: 10,
|   timestamp: new Date("2022-11-14"),
|   expenses: 7000
| }
| ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fed3900929efa447c290e'),
    '1': ObjectId('699fed3900929efa447c290f'),
    '2': ObjectId('699fed3900929efa447c2910'),
    '3': ObjectId('699fed3900929efa447c2911'),
    '4': ObjectId('699fed3900929efa447c2912')
  }
}
animal> |
```

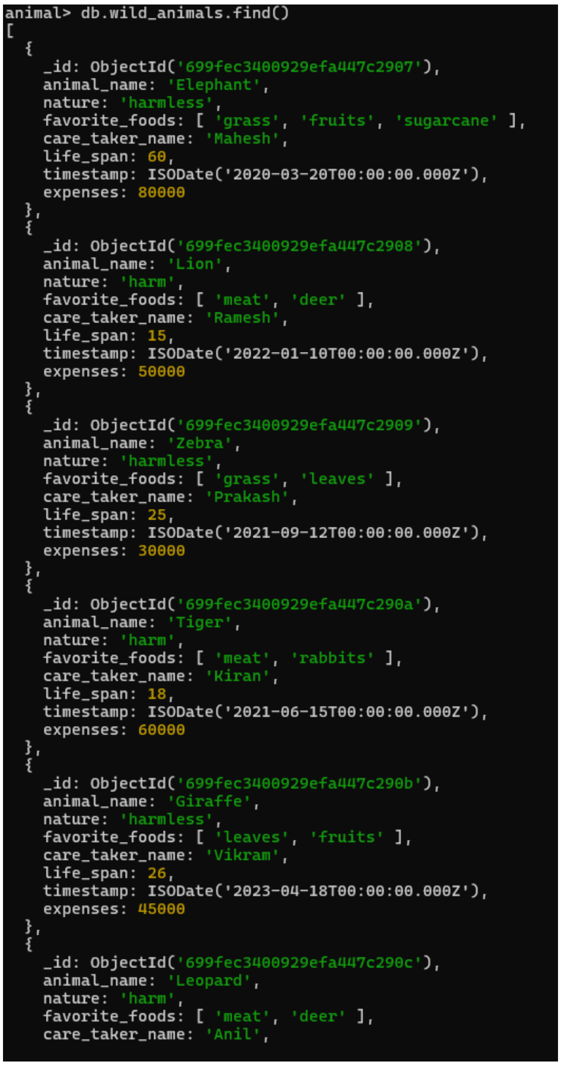
VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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

Output:

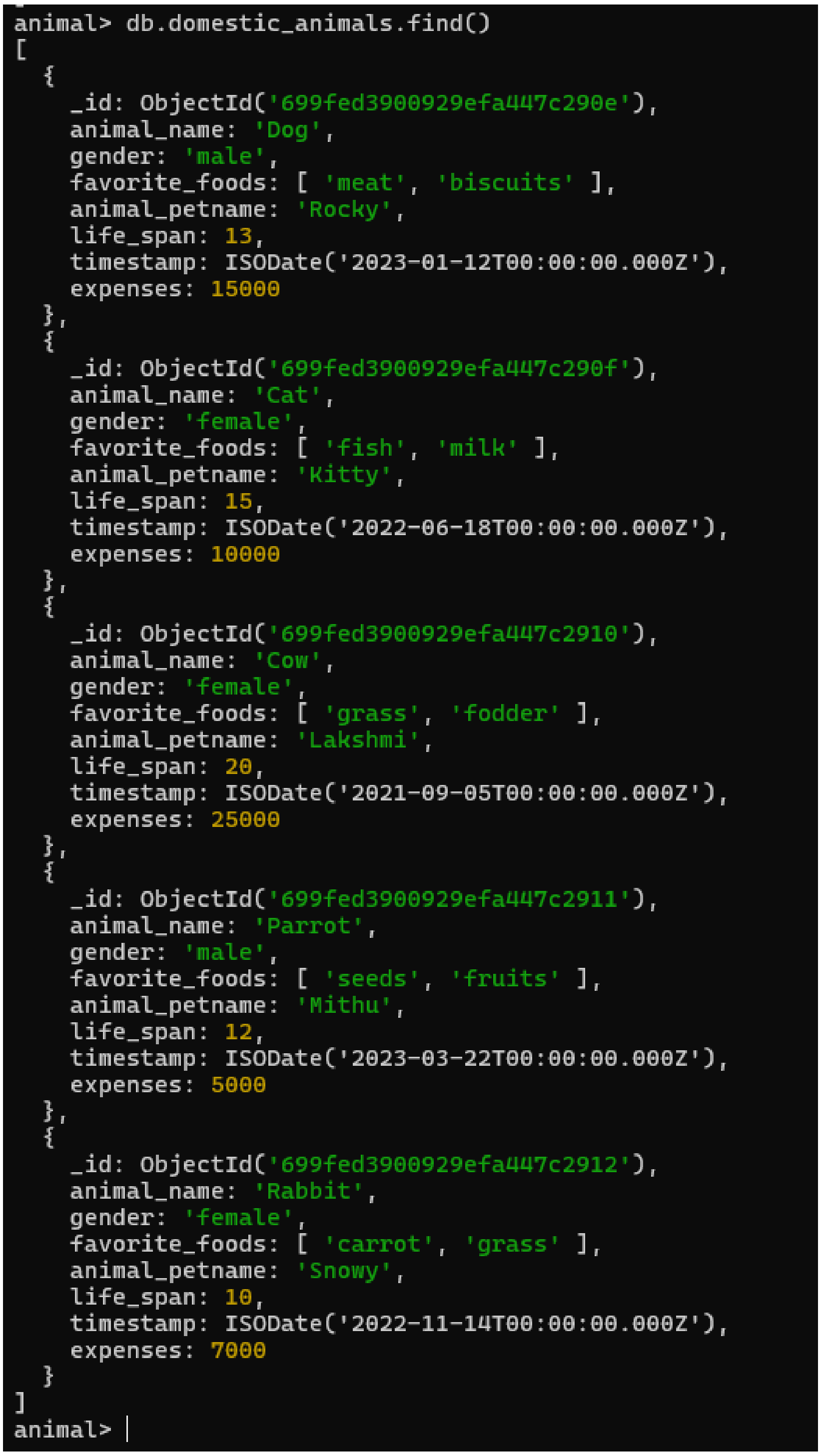


```
animal> db.wild_animals.find()
[
  {
    _id: ObjectId('699fec3400929efa447c2907'),
    animal_name: 'Elephant',
    nature: 'harmless',
    favorite_foods: [ 'grass', 'fruits', 'sugarcane' ],
    care_taker_name: 'Mahesh',
    life_span: 60,
    timestamp: ISODate('2020-03-20T00:00:00.000Z'),
    expenses: 80000
  },
  {
    _id: ObjectId('699fec3400929efa447c2908'),
    animal_name: 'Lion',
    nature: 'harm',
    favorite_foods: [ 'meat', 'deer' ],
    care_taker_name: 'Ramesh',
    life_span: 15,
    timestamp: ISODate('2022-01-10T00:00:00.000Z'),
    expenses: 50000
  },
  {
    _id: ObjectId('699fec3400929efa447c2909'),
    animal_name: 'Zebra',
    nature: 'harmless',
    favorite_foods: [ 'grass', 'leaves' ],
    care_taker_name: 'Prakash',
    life_span: 25,
    timestamp: ISODate('2021-09-12T00:00:00.000Z'),
    expenses: 30000
  },
  {
    _id: ObjectId('699fec3400929efa447c290a'),
    animal_name: 'Tiger',
    nature: 'harm',
    favorite_foods: [ 'meat', 'rabbits' ],
    care_taker_name: 'Kiran',
    life_span: 18,
    timestamp: ISODate('2021-06-15T00:00:00.000Z'),
    expenses: 60000
  },
  {
    _id: ObjectId('699fec3400929efa447c290b'),
    animal_name: 'Giraffe',
    nature: 'harmless',
    favorite_foods: [ 'leaves', 'fruits' ],
    care_taker_name: 'Vikram',
    life_span: 26,
    timestamp: ISODate('2023-04-18T00:00:00.000Z'),
    expenses: 45000
  },
  {
    _id: ObjectId('699fec3400929efa447c290c'),
    animal_name: 'Leopard',
    nature: 'harm',
    favorite_foods: [ 'meat', 'deer' ],
    care_taker_name: 'Anil',
```

VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304



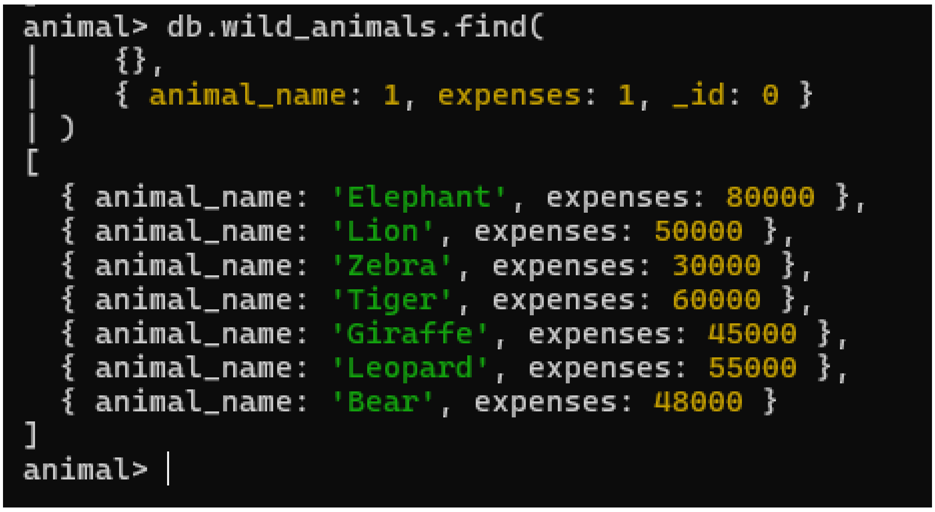
```
animal> db.domestic_animals.find()
[
  {
    _id: ObjectId('699fed3900929efa447c290e'),
    animal_name: 'Dog',
    gender: 'male',
    favorite_foods: [ 'meat', 'biscuits' ],
    animal_petname: 'Rocky',
    life_span: 13,
    timestamp: ISODate('2023-01-12T00:00:00.000Z'),
    expenses: 15000
  },
  {
    _id: ObjectId('699fed3900929efa447c290f'),
    animal_name: 'Cat',
    gender: 'female',
    favorite_foods: [ 'fish', 'milk' ],
    animal_petname: 'Kitty',
    life_span: 15,
    timestamp: ISODate('2022-06-18T00:00:00.000Z'),
    expenses: 10000
  },
  {
    _id: ObjectId('699fed3900929efa447c2910'),
    animal_name: 'Cow',
    gender: 'female',
    favorite_foods: [ 'grass', 'fodder' ],
    animal_petname: 'Lakshmi',
    life_span: 20,
    timestamp: ISODate('2021-09-05T00:00:00.000Z'),
    expenses: 25000
  },
  {
    _id: ObjectId('699fed3900929efa447c2911'),
    animal_name: 'Parrot',
    gender: 'male',
    favorite_foods: [ 'seeds', 'fruits' ],
    animal_petname: 'Mithu',
    life_span: 12,
    timestamp: ISODate('2023-03-22T00:00:00.000Z'),
    expenses: 5000
  },
  {
    _id: ObjectId('699fed3900929efa447c2912'),
    animal_name: 'Rabbit',
    gender: 'female',
    favorite_foods: [ 'carrot', 'grass' ],
    animal_petname: 'Snowy',
    life_span: 10,
    timestamp: ISODate('2022-11-14T00:00:00.000Z'),
    expenses: 7000
  }
]
animal> |
```

Lab Sheet 6: MongoDB Basic commands

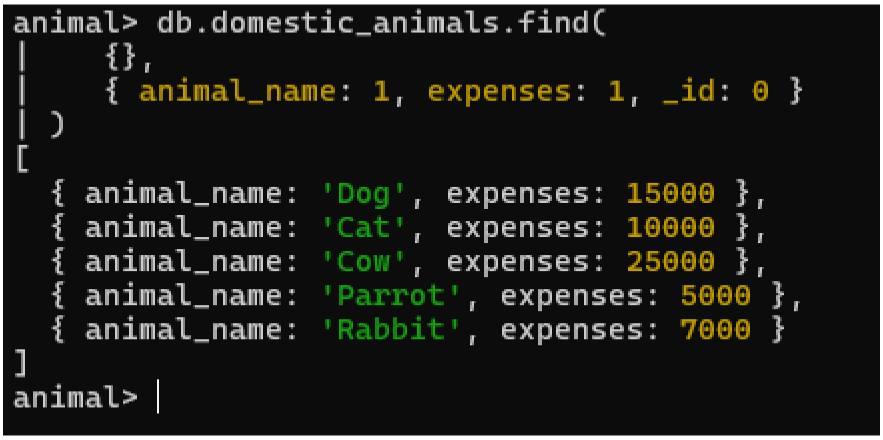
Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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

Output:



```
animal> db.wild_animals.find(
|     {},
|     { animal_name: 1, expenses: 1, _id: 0 }
| )
[
  { animal_name: 'Elephant', expenses: 80000 },
  { animal_name: 'Lion', expenses: 50000 },
  { animal_name: 'Zebra', expenses: 30000 },
  { animal_name: 'Tiger', expenses: 60000 },
  { animal_name: 'Giraffe', expenses: 45000 },
  { animal_name: 'Leopard', expenses: 55000 },
  { animal_name: 'Bear', expenses: 48000 }
]
animal> 
```



```
animal> 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: 'Parrot', expenses: 5000 },
  { animal_name: 'Rabbit', expenses: 7000 }
]
animal> 
```

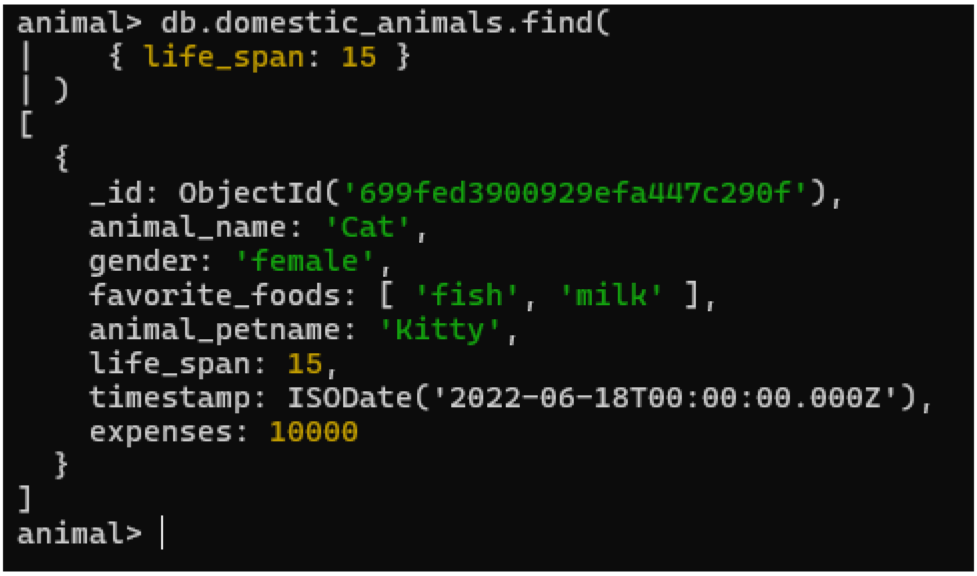
VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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

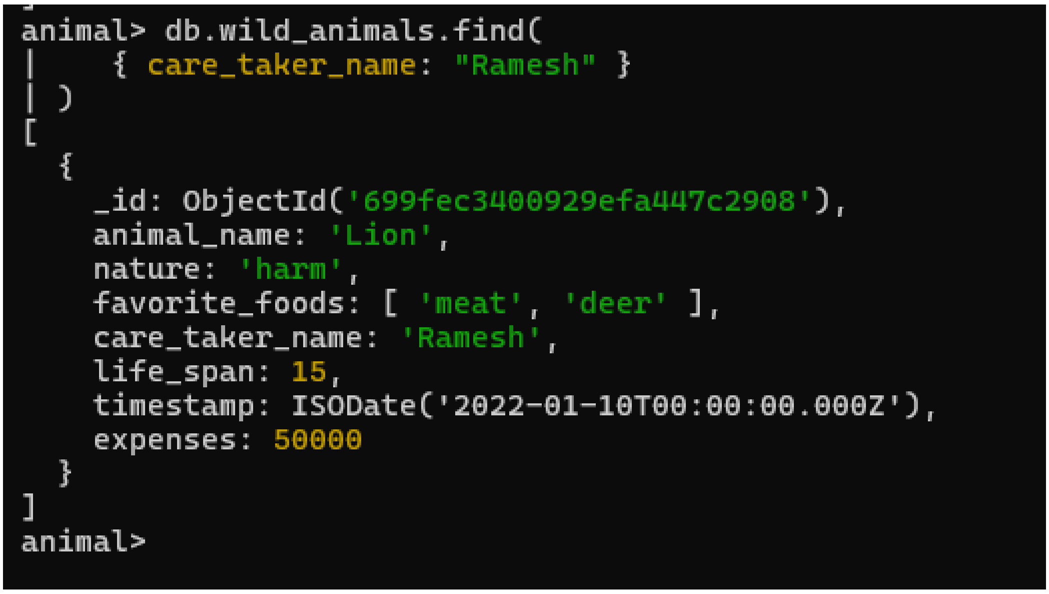
Output:



```
animal> db.domestic_animals.find(
|     { life_span: 15 }
| )
[
  {
    _id: ObjectId('699fed3900929efa447c290f'),
    animal_name: 'Cat',
    gender: 'female',
    favorite_foods: [ 'fish', 'milk' ],
    animal_petname: 'Kitty',
    life_span: 15,
    timestamp: ISODate('2022-06-18T00:00:00.000Z'),
    expenses: 10000
  }
]
animal> |
```

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

Output:



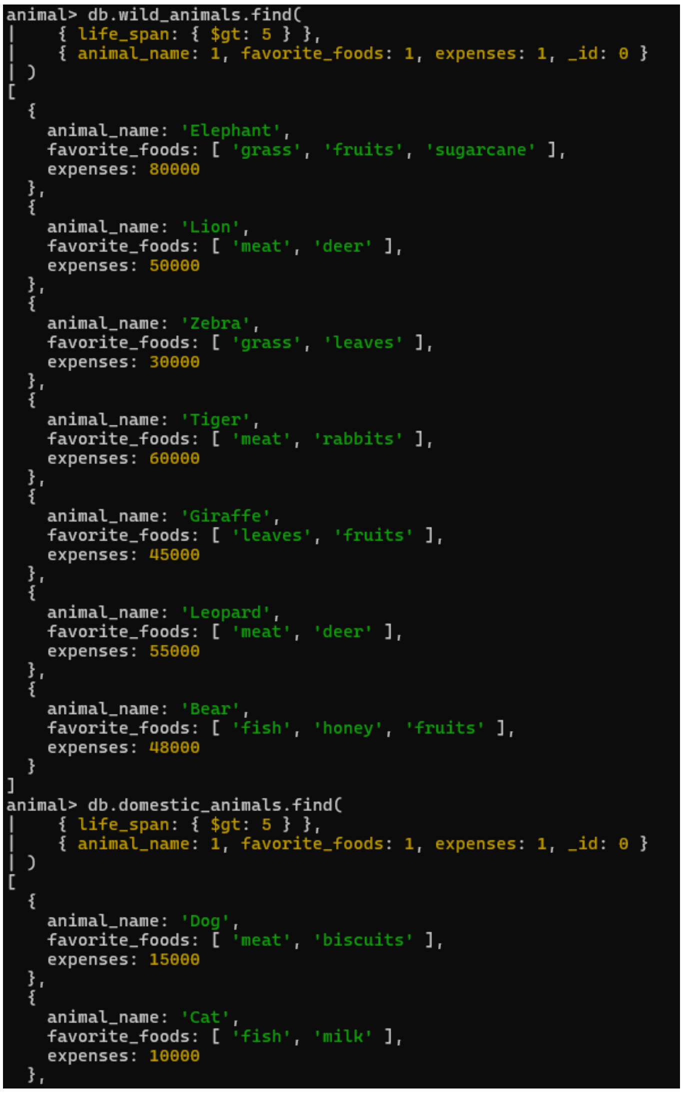
```
animal> db.wild_animals.find(
|     { care_taker_name: "Ramesh" }
| )
[
  {
    _id: ObjectId('699fec3400929efa447c2908'),
    animal_name: 'Lion',
    nature: 'harm',
    favorite_foods: [ 'meat', 'deer' ],
    care_taker_name: 'Ramesh',
    life_span: 15,
    timestamp: ISODate('2022-01-10T00:00:00.000Z'),
    expenses: 50000
  }
]
animal>
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: Ponnoju Sreeram Siddardha **Reg. no.:** 23BCE9304

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

Output:



```
animal> db.wild_animals.find(
|     { life_span: { $gt: 5 } },
|     { animal_name: 1, favorite_foods: 1, expenses: 1, _id: 0 }
| )
[
  {
    animal_name: 'Elephant',
    favorite_foods: [ 'grass', 'fruits', 'sugarcane' ],
    expenses: 80000
  },
  {
    animal_name: 'Lion',
    favorite_foods: [ 'meat', 'deer' ],
    expenses: 50000
  },
  {
    animal_name: 'Zebra',
    favorite_foods: [ 'grass', 'leaves' ],
    expenses: 30000
  },
  {
    animal_name: 'Tiger',
    favorite_foods: [ 'meat', 'rabbits' ],
    expenses: 60000
  },
  {
    animal_name: 'Giraffe',
    favorite_foods: [ 'leaves', 'fruits' ],
    expenses: 45000
  },
  {
    animal_name: 'Leopard',
    favorite_foods: [ 'meat', 'deer' ],
    expenses: 55000
  },
  {
    animal_name: 'Bear',
    favorite_foods: [ 'fish', 'honey', 'fruits' ],
    expenses: 48000
  }
]
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', 'biscuits' ],
    expenses: 15000
  },
  {
    animal_name: 'Cat',
    favorite_foods: [ 'fish', 'milk' ],
    expenses: 10000
  },
```