Name	: Kuruva Gayathri
Registration Number	: 23BCE20212
Slot	: L14+L15
Course Name	: Web Technologies
Course Code	: CSE4004
Faculty Name	: Prof. Gopikrishnan

Assignment-4 : MongoDB Basic commands

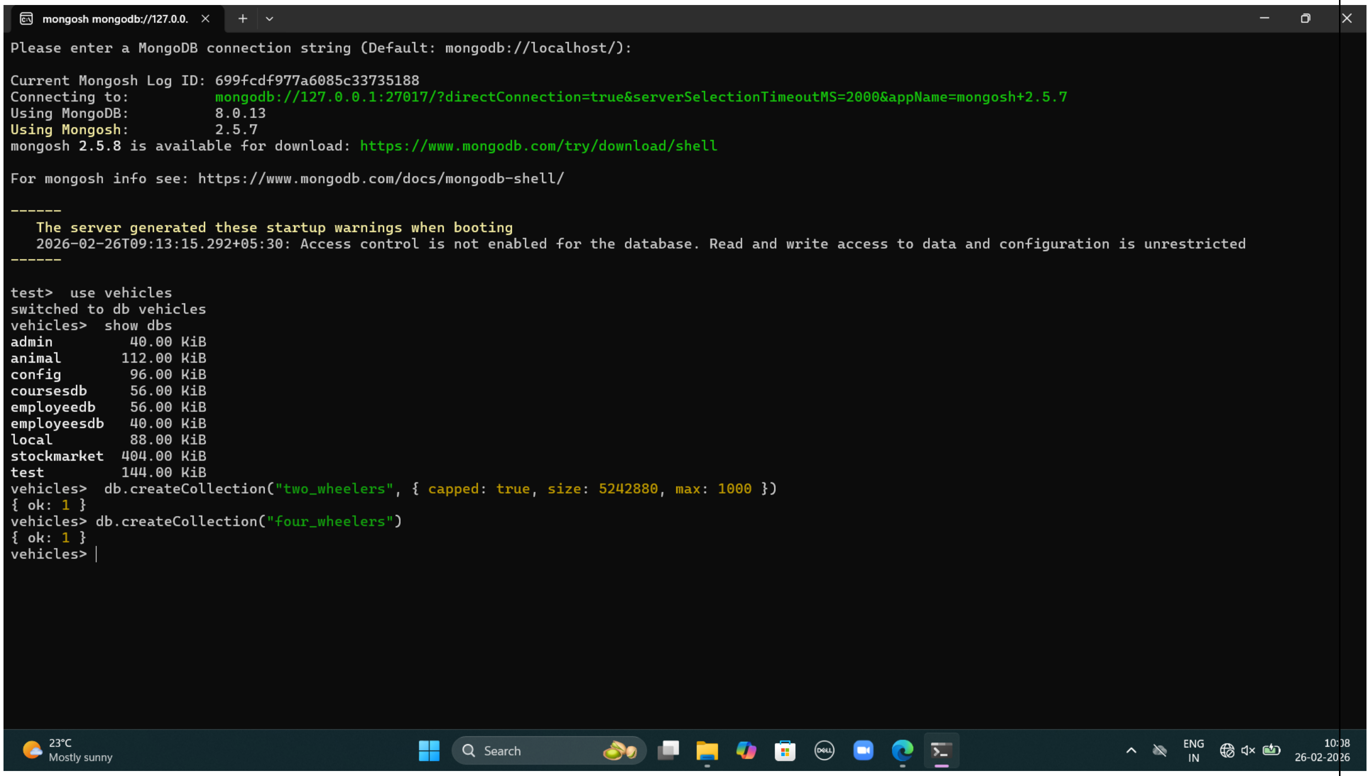
1. **Use MongoDB to implement the following DB operations**
   1. **Create a database called 'vehicles' and write a MongoDB query to select database as "vehicles".**

      use vehicles
   2. **Write a MongoDB query to display all the databases.**

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

      db.createCollection("two_wheelers", { capped: true, size: 5242880, max: 1000 })

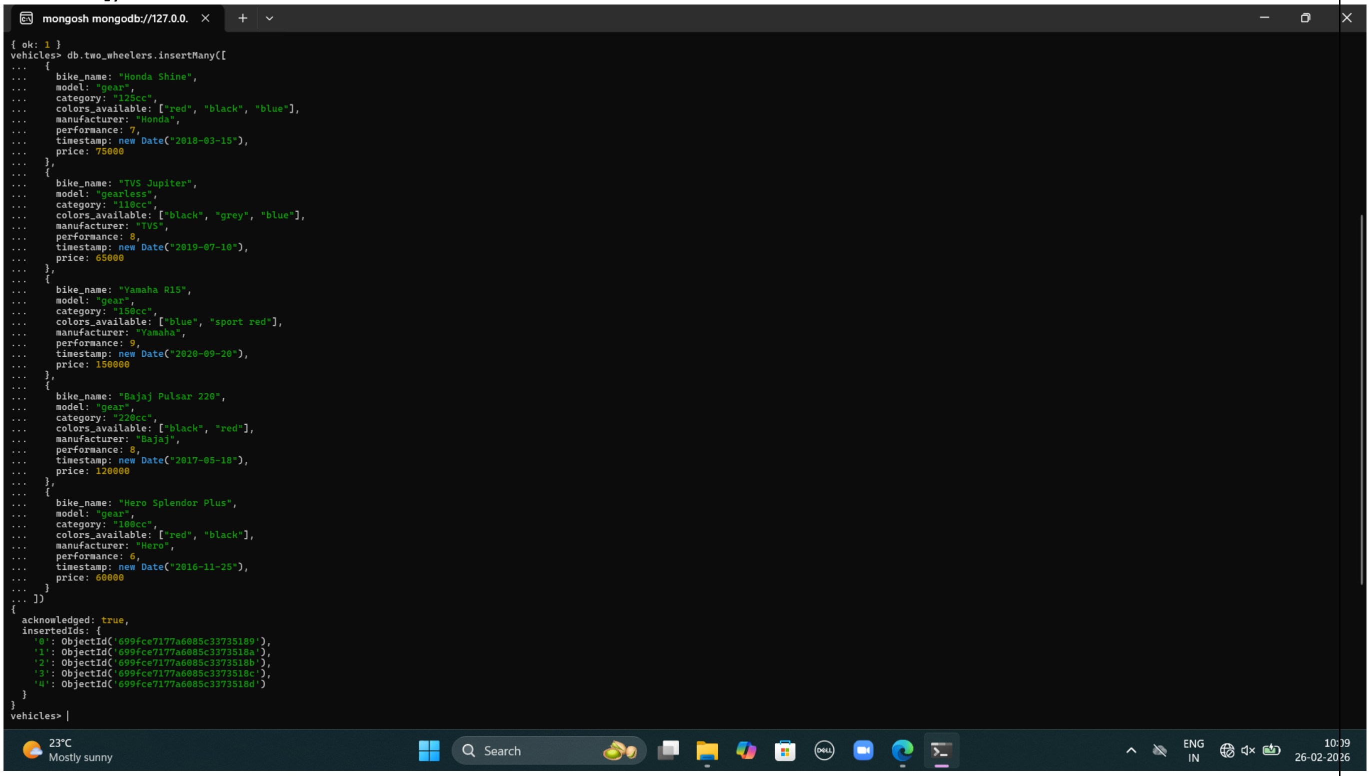
      db.createCollection("four_wheelers")



```
Please enter a MongoDB connection string (Default: mongodb://localhost/):

Current Mongosh Log ID: 699fcdf977a6085c33735188
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.5.7
Using MongoDB:          8.0.13
Using Mongosh:          2.5.7
mongosh 2.5.8 is available for download: https://www.mongodb.com/try/download/shell

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

------
   The server generated these startup warnings when booting
   2026-02-26T09:13:15.292+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
animal       112.00 KiB
config        96.00 KiB
coursesdb     56.00 KiB
employeedb    56.00 KiB
employeesdb   40.00 KiB
local         88.00 KiB
stockmarket  404.00 KiB
test         144.00 KiB
vehicles>  db.createCollection("two_wheelers", { capped: true, size: 5242880, max: 1000 })
{ ok: 1 }
vehicles> db.createCollection("four_wheelers")
{ ok: 1 }
vehicles> |
```

   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.**

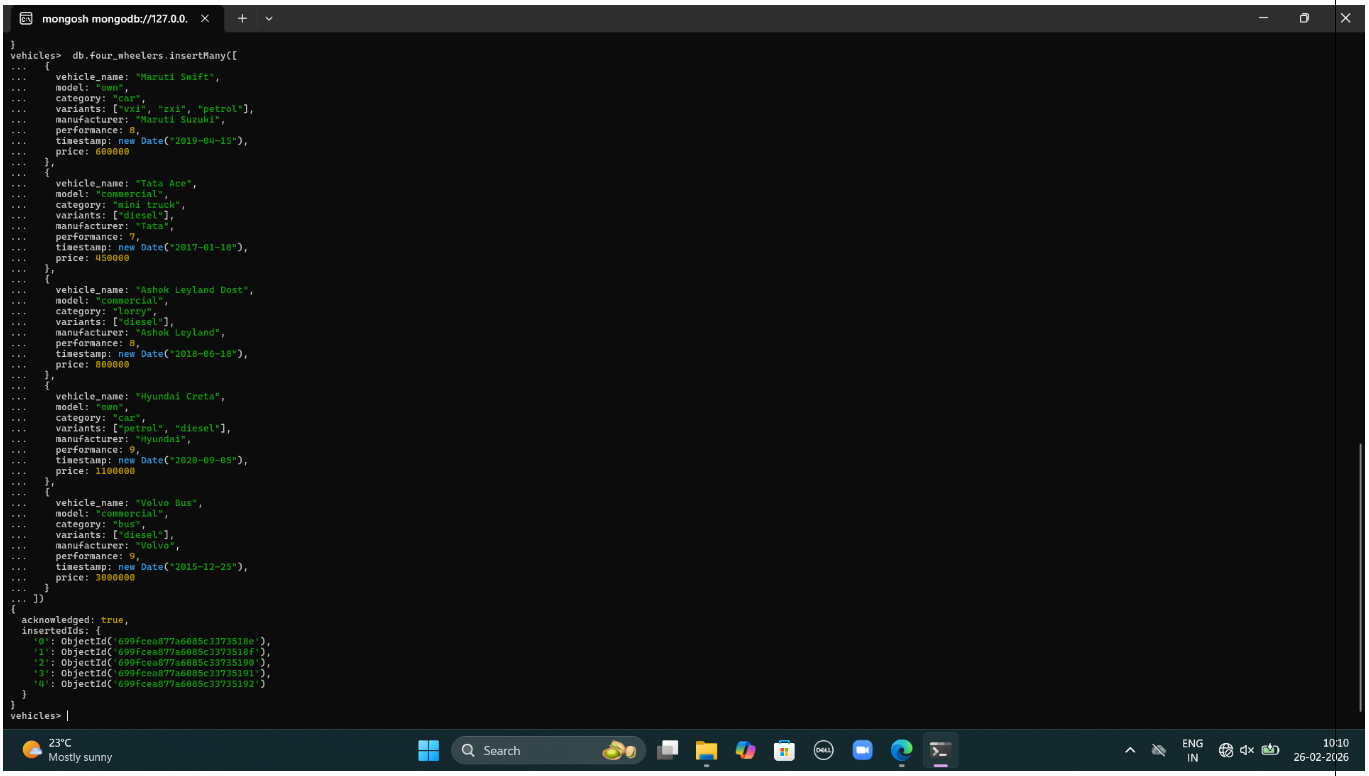
```
db.two_wheelers.insertMany([
 {
   bike_name: "Honda Shine",
   model: "gear",
   category: "125cc",
   colors_available: ["red", "black", "blue"],
    manufacturer: "Honda",
    performance: 7,
    timestamp: new Date("2018-03-15"),
    price: 75000
  },
  {
    bike_name: "TVS Jupiter",
    model: "gearless",
    category: "110cc",
    colors_available: ["black", "grey", "blue"],
    manufacturer: "TVS",
    performance: 8,
    timestamp: new Date("2019-07-10"),
    price: 65000
  },
  {
    bike_name: "Yamaha R15",
    model: "gear",
    category: "150cc",
    colors_available: ["blue", "sport red"],
    manufacturer: "Yamaha",
    performance: 9,
    timestamp: new Date("2020-09-20"),
    price: 150000
  },
  {
    bike_name: "Bajaj Pulsar 220",
    model: "gear",
    category: "220cc",
    colors_available: ["black", "red"],
    manufacturer: "Bajaj",
    performance: 8,
    timestamp: new Date("2017-05-18"),
    price: 120000
  },
  {
    bike_name: "Hero Splendor Plus",
    model: "gear",
    category: "100cc",
    colors_available: ["red", "black"],
    manufacturer: "Hero",
  performance: 6,
  timestamp: new Date("2016-11-25"),
  price: 60000
 }
])
```



```
{ ok: 1 }
vehicles> db.two_wheelers.insertMany([
...   {
...     bike_name: "Honda Shine",
...     model: "gear",
...     category: "125cc",
...     colors_available: ["red", "black", "blue"],
...     manufacturer: "Honda",
...     performance: 7,
...     timestamp: new Date("2018-03-15"),
...     price: 75000
...   },
...   {
...     bike_name: "TVS Jupiter",
...     model: "gearless",
...     category: "110cc",
...     colors_available: ["black", "grey", "blue"],
...     manufacturer: "TVS",
...     performance: 8,
...     timestamp: new Date("2019-07-10"),
...     price: 65000
...   },
...   {
...     bike_name: "Yamaha R15",
...     model: "gear",
...     category: "150cc",
...     colors_available: ["blue", "sport red"],
...     manufacturer: "Yamaha",
...     performance: 9,
...     timestamp: new Date("2020-09-20"),
...     price: 150000
...   },
...   {
...     bike_name: "Bajaj Pulsar 220",
...     model: "gear",
...     category: "220cc",
...     colors_available: ["black", "red"],
...     manufacturer: "Bajaj",
...     performance: 8,
...     timestamp: new Date("2017-05-18"),
...     price: 120000
...   },
...   {
...     bike_name: "Hero Splendor Plus",
...     model: "gear",
...     category: "100cc",
...     colors_available: ["red", "black"],
...     manufacturer: "Hero",
...     performance: 6,
...     timestamp: new Date("2016-11-25"),
...     price: 60000
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fce7177a6085c33735189'),
    '1': ObjectId('699fce7177a6085c3373518a'),
    '2': ObjectId('699fce7177a6085c3373518b'),
    '3': ObjectId('699fce7177a6085c3373518c'),
    '4': ObjectId('699fce7177a6085c3373518d')
  }
}
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.**

```
db.four_wheelers.insertMany([
{
  vehicle_name: "Maruti Swift",
  model: "own",
  category: "car",
  variants: ["vxi", "zxi", "petrol"],
  manufacturer: "Maruti Suzuki",
  performance: 8,
  timestamp: new Date("2019-04-15"),
  price: 600000
},
{
    vehicle_name: "Tata Ace",
    model: "commercial",
    category: "mini truck",
    variants: ["diesel"],
    manufacturer: "Tata",
    performance: 7,
    timestamp: new Date("2017-01-10"),
    price: 450000
  },
  {
    vehicle_name: "Ashok Leyland Dost",
    model: "commercial",
    category: "lorry",
    variants: ["diesel"],
    manufacturer: "Ashok Leyland",
    performance: 8,
    timestamp: new Date("2018-06-18"),
    price: 800000
  },
  {
    vehicle_name: "Hyundai Creta",
    model: "own",
    category: "car",
    variants: ["petrol", "diesel"],
    manufacturer: "Hyundai",
    performance: 9,
    timestamp: new Date("2020-09-05"),
    price: 1100000
  },
  {
    vehicle_name: "Volvo Bus",
    model: "commercial",
    category: "bus",
    variants: ["diesel"],
    manufacturer: "Volvo",
    performance: 9,
    timestamp: new Date("2015-12-25"),
    price: 3000000
  }
])
```

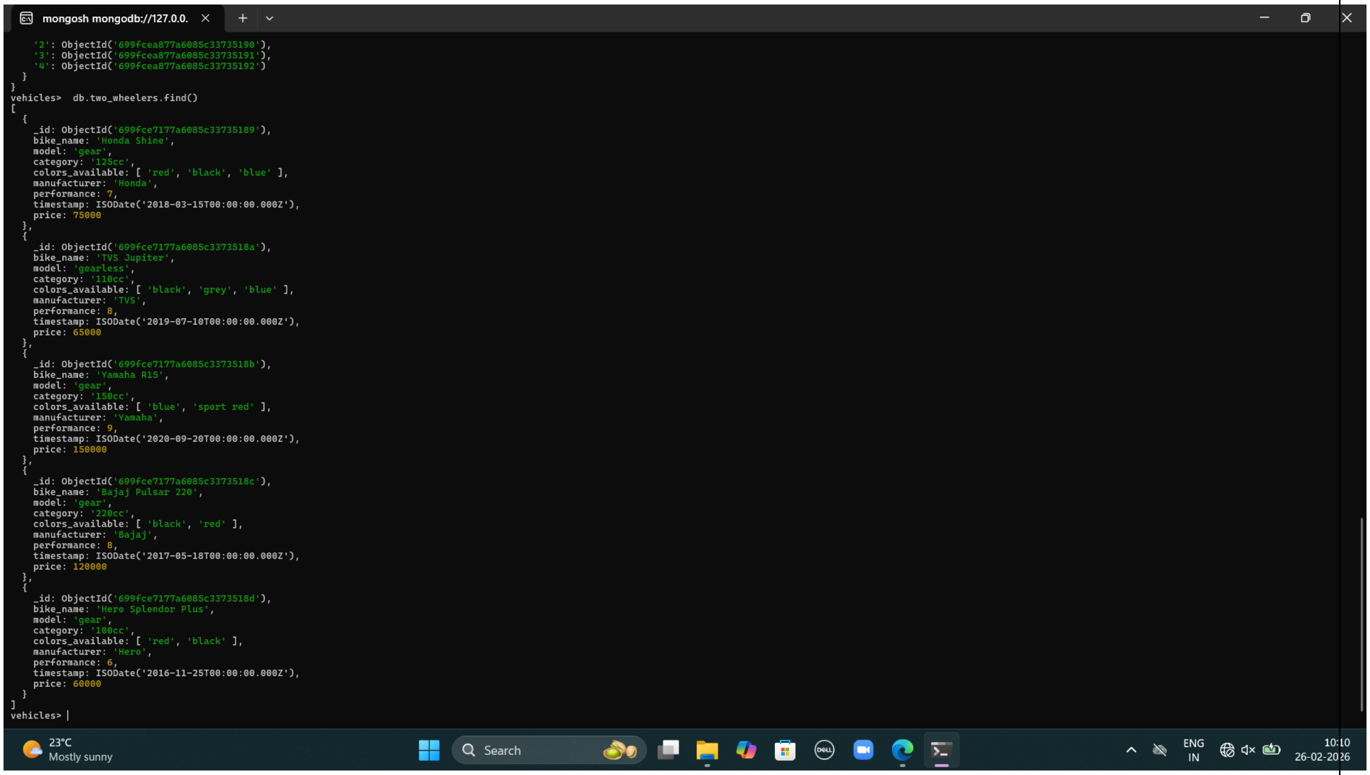


```
}
vehicles>  db.four_wheelers.insertMany([
...    {
...      vehicle_name: "Maruti Swift",
...      model: "own",
...      category: "car",
...      variants: ["vxi", "zxi", "petrol"],
...      manufacturer: "Maruti Suzuki",
...      performance: 8,
...      timestamp: new Date("2019-04-15"),
...      price: 600000
...    },
...    {
...      vehicle_name: "Tata Ace",
...      model: "commercial",
...      category: "mini truck",
...      variants: ["diesel"],
...      manufacturer: "Tata",
...      performance: 7,
...      timestamp: new Date("2017-01-10"),
...      price: 450000
...    },
...    {
...      vehicle_name: "Ashok Leyland Dost",
...      model: "commercial",
...      category: "lorry",
...      variants: ["diesel"],
...      manufacturer: "Ashok Leyland",
...      performance: 8,
...      timestamp: new Date("2018-06-18"),
...      price: 800000
...    },
...    {
...      vehicle_name: "Hyundai Creta",
...      model: "own",
...      category: "car",
...      variants: ["petrol", "diesel"],
...      manufacturer: "Hyundai",
...      performance: 9,
...      timestamp: new Date("2020-09-05"),
...      price: 1100000
...    },
...    {
...      vehicle_name: "Volvo Bus",
...      model: "commercial",
...      category: "bus",
...      variants: ["diesel"],
...      manufacturer: "Volvo",
...      performance: 9,
...      timestamp: new Date("2015-12-25"),
...      price: 3000000
...    }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fcea877a6085c3373518e'),
    '1': ObjectId('699fcea877a6085c3373518f'),
    '2': ObjectId('699fcea877a6085c33735190'),
    '3': ObjectId('699fcea877a6085c33735191'),
    '4': ObjectId('699fcea877a6085c33735192')
  }
}
vehicles> |
```

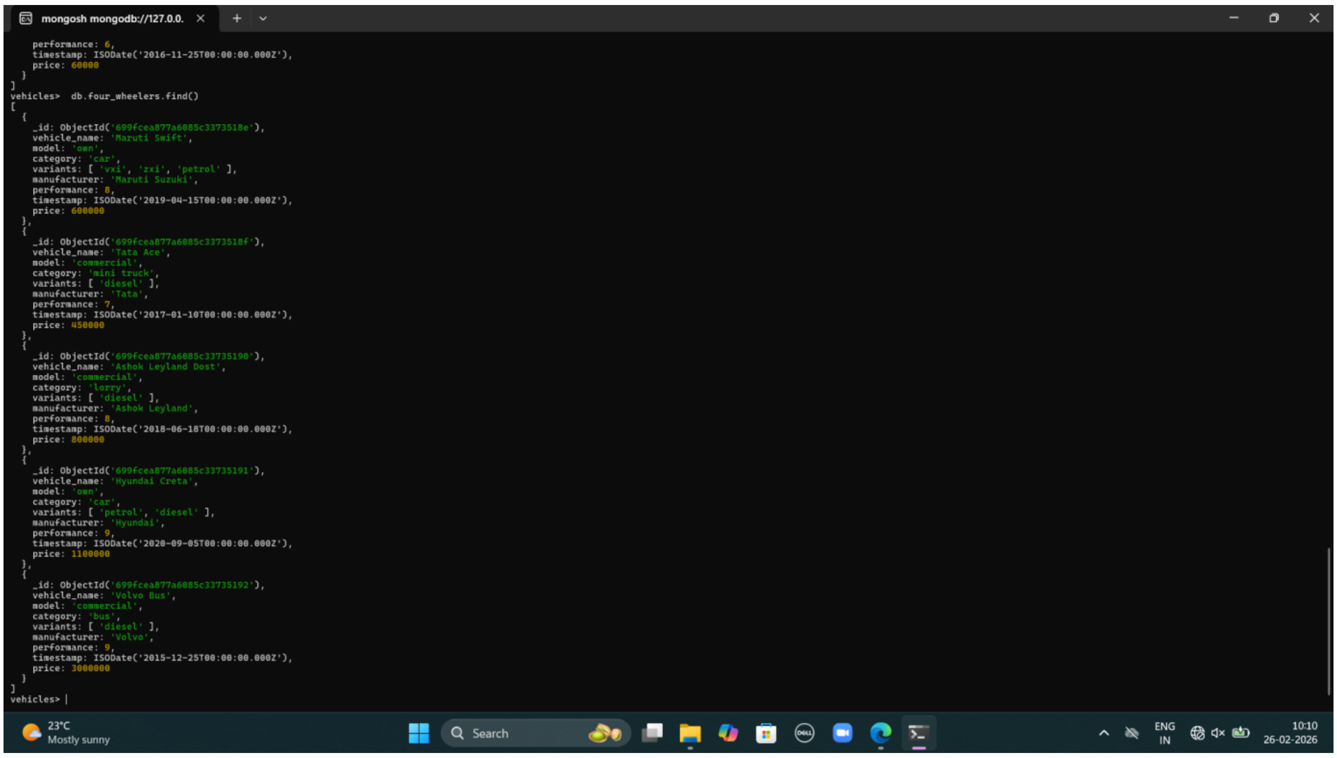
6. **Write a MongoDB query to display all documents available in two_wheelers and four_wheelers.**

   db.two_wheelers.find()

   db.four_wheelers.find()



```
    '2': ObjectId('699fcea877a6085c33735190'),
    '3': ObjectId('699fcea877a6085c33735191'),
    '4': ObjectId('699fcea877a6085c33735192')
  }
}
vehicles>  db.two_wheelers.find()
[
  {
    _id: ObjectId('699fce7177a6085c33735189'),
    bike_name: 'Honda Shine',
    model: 'gear',
    category: '125cc',
    colors_available: [ 'red', 'black', 'blue' ],
    manufacturer: 'Honda',
    performance: 7,
    timestamp: ISODate('2018-03-15T00:00:00.000Z'),
    price: 75000
  },
  {
    _id: ObjectId('699fce7177a6085c3373518a'),
    bike_name: 'TVS Jupiter',
    model: 'gearless',
    category: '110cc',
    colors_available: [ 'black', 'grey', 'blue' ],
    manufacturer: 'TVS',
    performance: 8,
    timestamp: ISODate('2019-07-10T00:00:00.000Z'),
    price: 65000
  },
  {
    _id: ObjectId('699fce7177a6085c3373518b'),
    bike_name: 'Yamaha R15',
    model: 'gear',
    category: '150cc',
    colors_available: [ 'blue', 'sport red' ],
    manufacturer: 'Yamaha',
    performance: 9,
    timestamp: ISODate('2020-09-20T00:00:00.000Z'),
    price: 150000
  },
  {
    _id: ObjectId('699fce7177a6085c3373518c'),
    bike_name: 'Bajaj Pulsar 220',
    model: 'gear',
    category: '220cc',
    colors_available: [ 'black', 'red' ],
    manufacturer: 'Bajaj',
    performance: 8,
    timestamp: ISODate('2017-05-18T00:00:00.000Z'),
    price: 120000
  },
  {
    _id: ObjectId('699fce7177a6085c3373518d'),
    bike_name: 'Hero Splendor Plus',
    model: 'gear',
    category: '100cc',
    colors_available: [ 'red', 'black' ],
    manufacturer: 'Hero',
    performance: 6,
    timestamp: ISODate('2016-11-25T00:00:00.000Z'),
    price: 60000
  }
]
vehicles> |
```



```
    performance: 6,
    timestamp: ISODate('2016-11-25T00:00:00.000Z'),
    price: 60000
  }
]
vehicles>  db.four_wheelers.find()
[
  {
    _id: ObjectId('699fcea877a6085c3373518e'),
    vehicle_name: 'Maruti Swift',
    model: 'own',
    category: 'car',
    variants: [ 'vxi', 'zxi', 'petrol' ],
    manufacturer: 'Maruti Suzuki',
    performance: 8,
    timestamp: ISODate('2019-04-15T00:00:00.000Z'),
    price: 600000
  },
  {
    _id: ObjectId('699fcea877a6085c3373518f'),
    vehicle_name: 'Tata Ace',
    model: 'commercial',
    category: 'mini truck',
    variants: [ 'diesel' ],
    manufacturer: 'Tata',
    performance: 7,
    timestamp: ISODate('2017-01-10T00:00:00.000Z'),
    price: 450000
  },
  {
    _id: ObjectId('699fcea877a6085c33735190'),
    vehicle_name: 'Ashok Leyland Dost',
    model: 'commercial',
    category: 'lorry',
    variants: [ 'diesel' ],
    manufacturer: 'Ashok Leyland',
    performance: 8,
    timestamp: ISODate('2018-06-18T00:00:00.000Z'),
    price: 800000
  },
  {
    _id: ObjectId('699fcea877a6085c33735191'),
    vehicle_name: 'Hyundai Creta',
    model: 'own',
    category: 'car',
    variants: [ 'petrol', 'diesel' ],
    manufacturer: 'Hyundai',
    performance: 9,
    timestamp: ISODate('2020-09-05T00:00:00.000Z'),
    price: 1100000
  },
  {
    _id: ObjectId('699fcea877a6085c33735192'),
    vehicle_name: 'Volvo Bus',
    model: 'commercial',
    category: 'bus',
    variants: [ 'diesel' ],
    manufacturer: 'Volvo',
    performance: 9,
    timestamp: ISODate('2015-12-25T00:00:00.000Z'),
    price: 3000000
  }
]
vehicles> |
```

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

of the database

- For two_wheelers:

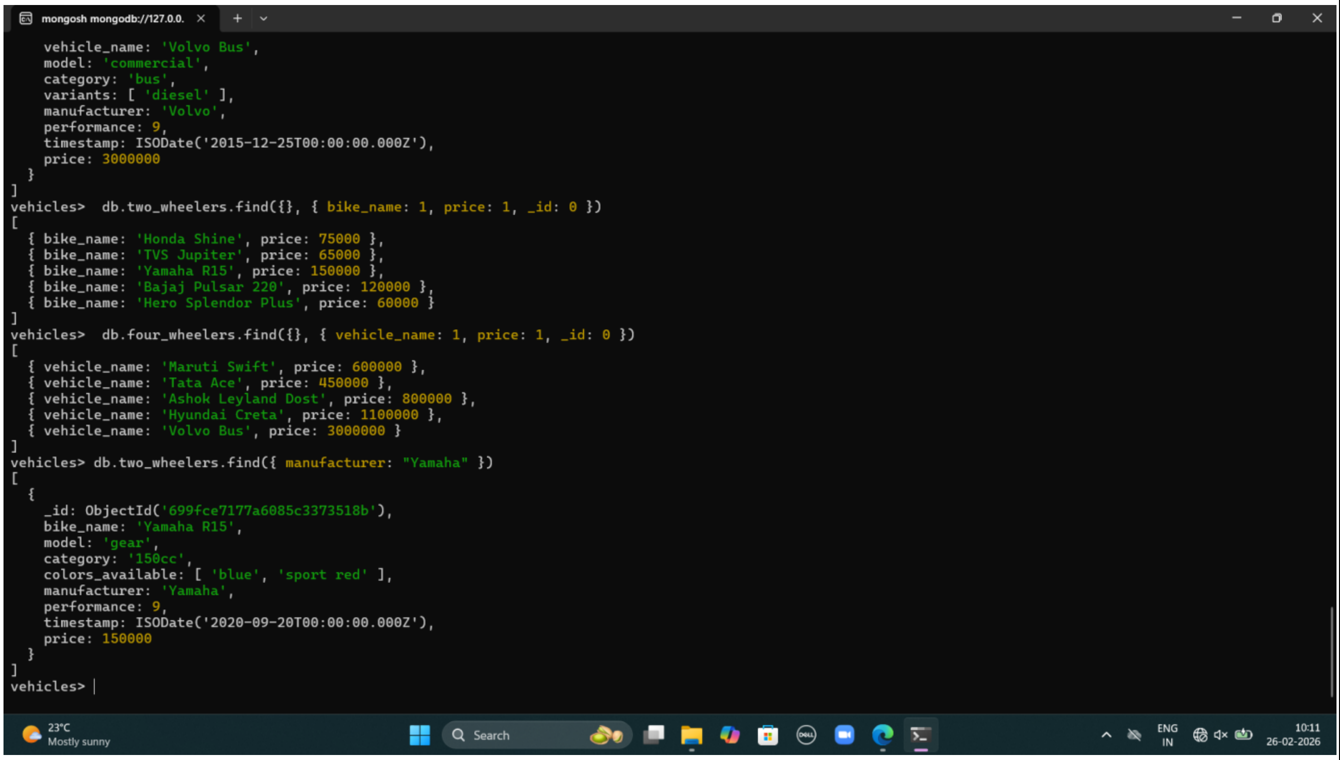
db.two_wheelers.find({}, { bike_name: 1, price: 1, _id: 0 })

- For four_wheelers:

db.four_wheelers.find({}, { vehicle_name: 1, price: 1, _id: 0 }

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

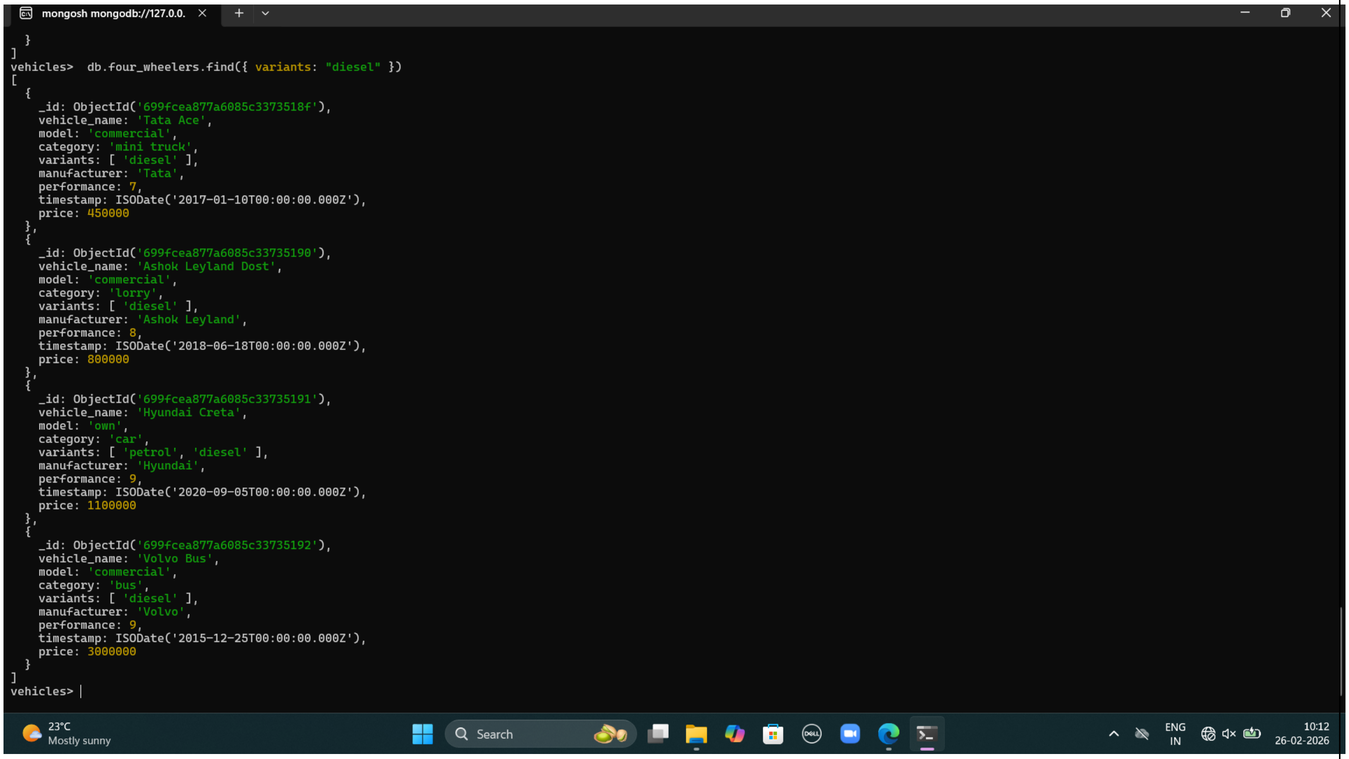
db.two_wheelers.find({ manufacturer: "Yamaha" })



```
    vehicle_name: 'Volvo Bus',
    model: 'commercial',
    category: 'bus',
    variants: [ 'diesel' ],
    manufacturer: 'Volvo',
    performance: 9,
    timestamp: ISODate('2015-12-25T00:00:00.000Z'),
    price: 3000000
  }
]
vehicles>  db.two_wheelers.find({}, { bike_name: 1, price: 1, _id: 0 })
[
  { bike_name: 'Honda Shine', price: 75000 },
  { bike_name: 'TVS Jupiter', price: 65000 },
  { bike_name: 'Yamaha R15', price: 150000 },
  { bike_name: 'Bajaj Pulsar 220', price: 120000 },
  { bike_name: 'Hero Splendor Plus', price: 60000 }
]
vehicles>  db.four_wheelers.find({}, { vehicle_name: 1, price: 1, _id: 0 })
[
  { vehicle_name: 'Maruti Swift', price: 600000 },
  { vehicle_name: 'Tata Ace', price: 450000 },
  { vehicle_name: 'Ashok Leyland Dost', price: 800000 },
  { vehicle_name: 'Hyundai Creta', price: 1100000 },
  { vehicle_name: 'Volvo Bus', price: 3000000 }
]
vehicles> db.two_wheelers.find({ manufacturer: "Yamaha" })
[
  {
    _id: ObjectId('699fce7177a6085c3373518b'),
    bike_name: 'Yamaha R15',
    model: 'gear',
    category: '150cc',
    colors_available: [ 'blue', 'sport red' ],
    manufacturer: 'Yamaha',
    performance: 9,
    timestamp: ISODate('2020-09-20T00:00:00.000Z'),
    price: 150000
  }
]
vehicles> |
```

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

db.four_wheelers.find({ variants: "diesel" })



```
  }
]
vehicles>  db.four_wheelers.find({ variants: "diesel" })
[
  {
    _id: ObjectId('699fcea877a6085c3373518f'),
    vehicle_name: 'Tata Ace',
    model: 'commercial',
    category: 'mini truck',
    variants: [ 'diesel' ],
    manufacturer: 'Tata',
    performance: 7,
    timestamp: ISODate('2017-01-10T00:00:00.000Z'),
    price: 450000
  },
  {
    _id: ObjectId('699fcea877a6085c33735190'),
    vehicle_name: 'Ashok Leyland Dost',
    model: 'commercial',
    category: 'lorry',
    variants: [ 'diesel' ],
    manufacturer: 'Ashok Leyland',
    performance: 8,
    timestamp: ISODate('2018-06-18T00:00:00.000Z'),
    price: 800000
  },
  {
    _id: ObjectId('699fcea877a6085c33735191'),
    vehicle_name: 'Hyundai Creta',
    model: 'own',
    category: 'car',
    variants: [ 'petrol', 'diesel' ],
    manufacturer: 'Hyundai',
    performance: 9,
    timestamp: ISODate('2020-09-05T00:00:00.000Z'),
    price: 1100000
  },
  {
    _id: ObjectId('699fcea877a6085c33735192'),
    vehicle_name: 'Volvo Bus',
    model: 'commercial',
    category: 'bus',
    variants: [ 'diesel' ],
    manufacturer: 'Volvo',
    performance: 9,
    timestamp: ISODate('2015-12-25T00:00:00.000Z'),
    price: 3000000
  }
]
vehicles> |
```

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

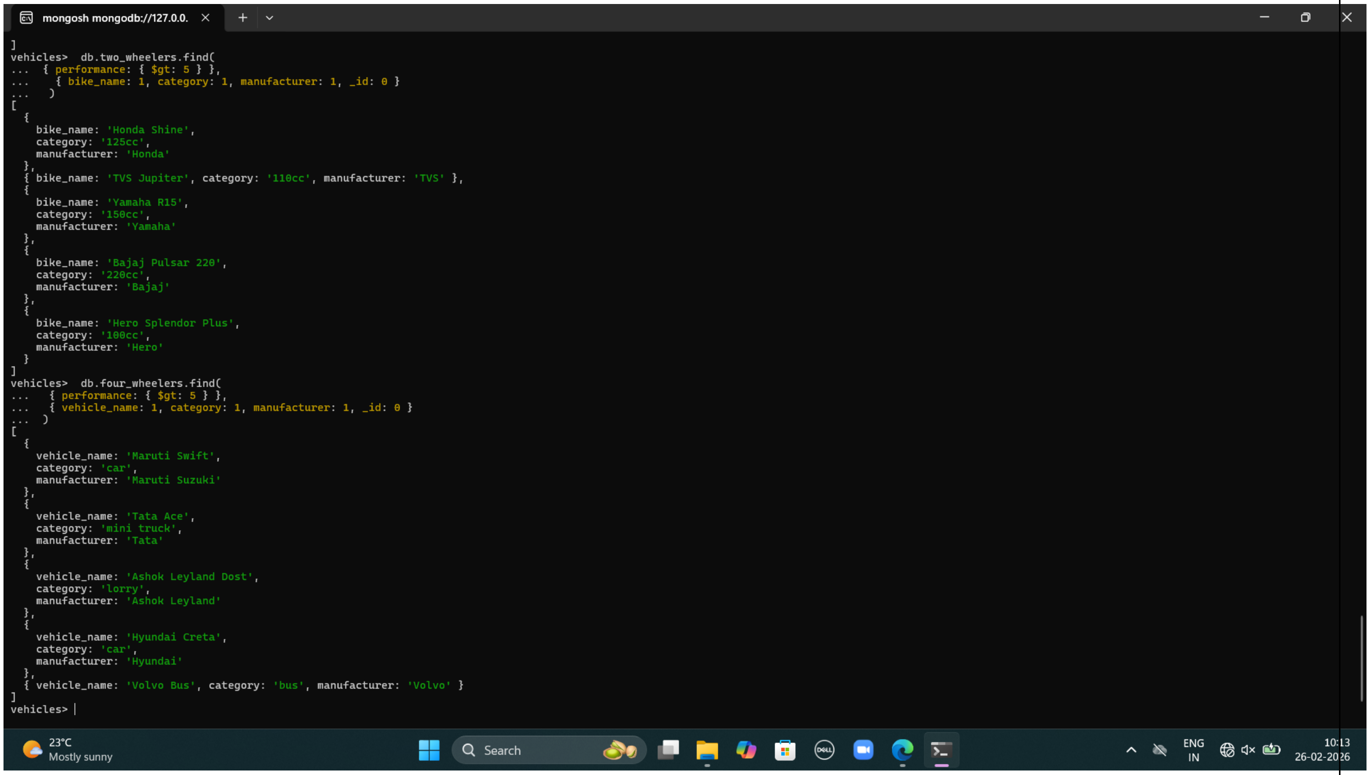
db.two_wheelers.find({ performance: { $gt: 5 } },

{ bike_name: 1, category: 1, manufacturer: 1, _id: 0 })

db.four_wheelers.find(

{ performance: { $gt: 5 } },

{ vehicle_name: 1, category: 1, manufacturer: 1, _id: 0 })



```
]
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 Jupiter', category: '110cc', manufacturer: 'TVS' },
  {
    bike_name: 'Yamaha R15',
    category: '150cc',
    manufacturer: 'Yamaha'
  },
  {
    bike_name: 'Bajaj Pulsar 220',
    category: '220cc',
    manufacturer: 'Bajaj'
  },
  {
    bike_name: 'Hero Splendor Plus',
    category: '100cc',
    manufacturer: 'Hero'
  }
]
vehicles>  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: 'Hyundai Creta',
    category: 'car',
    manufacturer: 'Hyundai'
  },
  { vehicle_name: 'Volvo Bus', category: 'bus', manufacturer: 'Volvo' }
]
vehicles> |
```

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 animal

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

show dbs

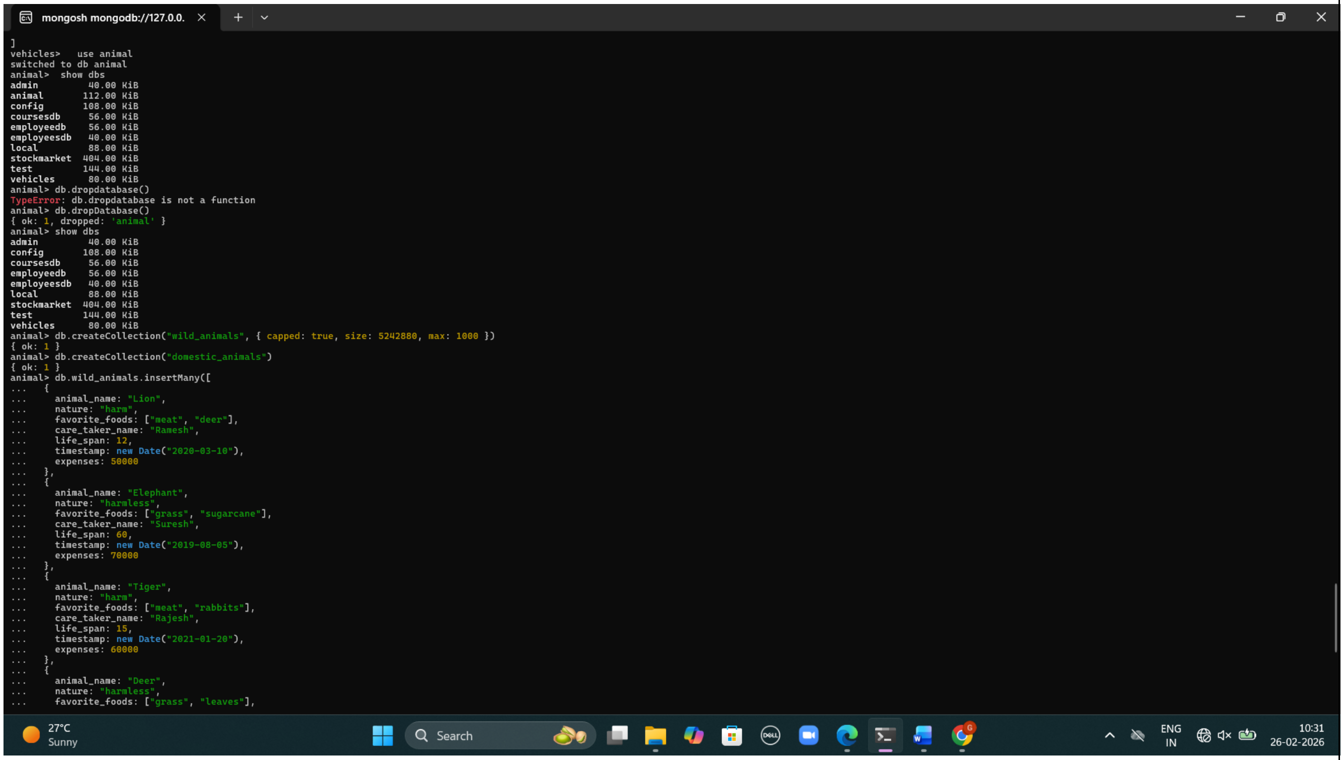
3. Create a collection called 'wild_animals'.(use capping) and Create a collection called 'domestic_animals'.

- wild_animals (capped collection):

db.createCollection("wild_animals", { capped: true, size: 5242880, max: 1000 })

- domestic_animals (normal collection):

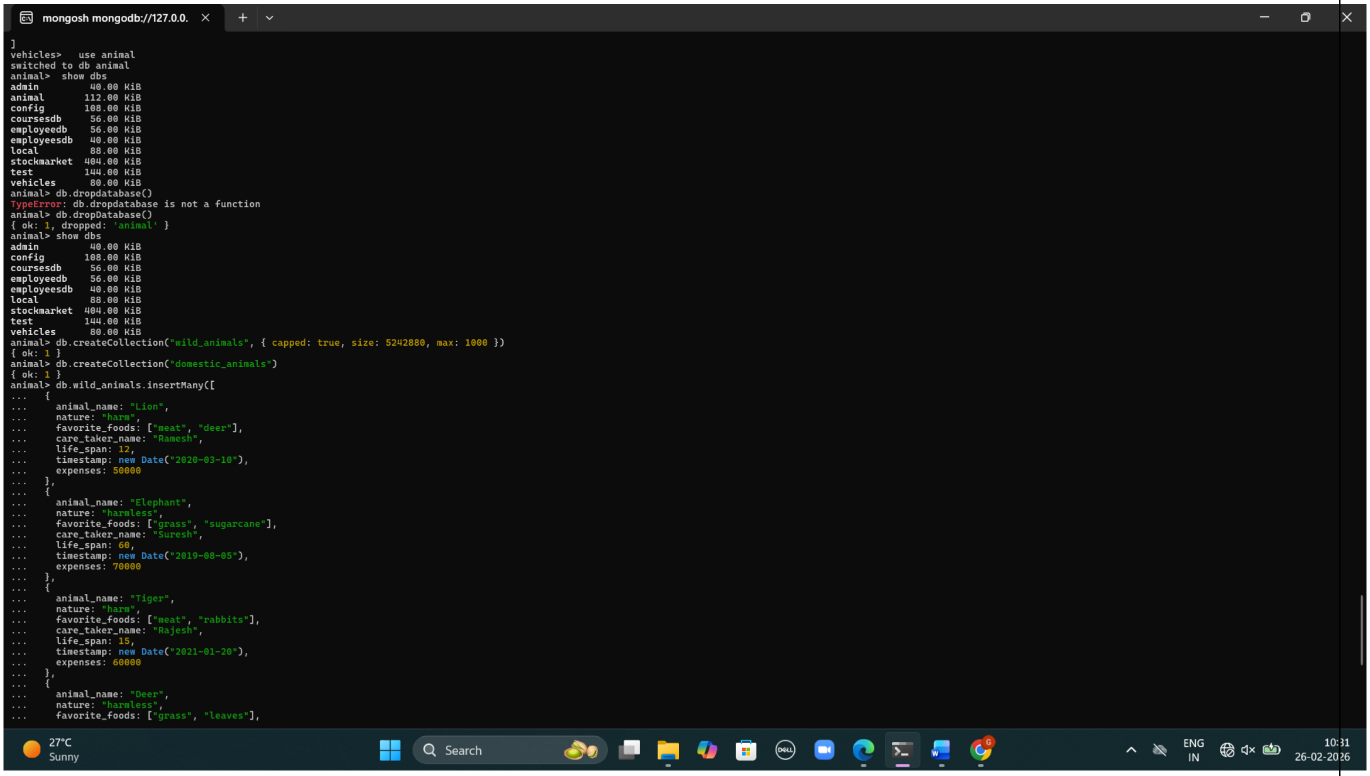
db.createCollection("domestic_animals")



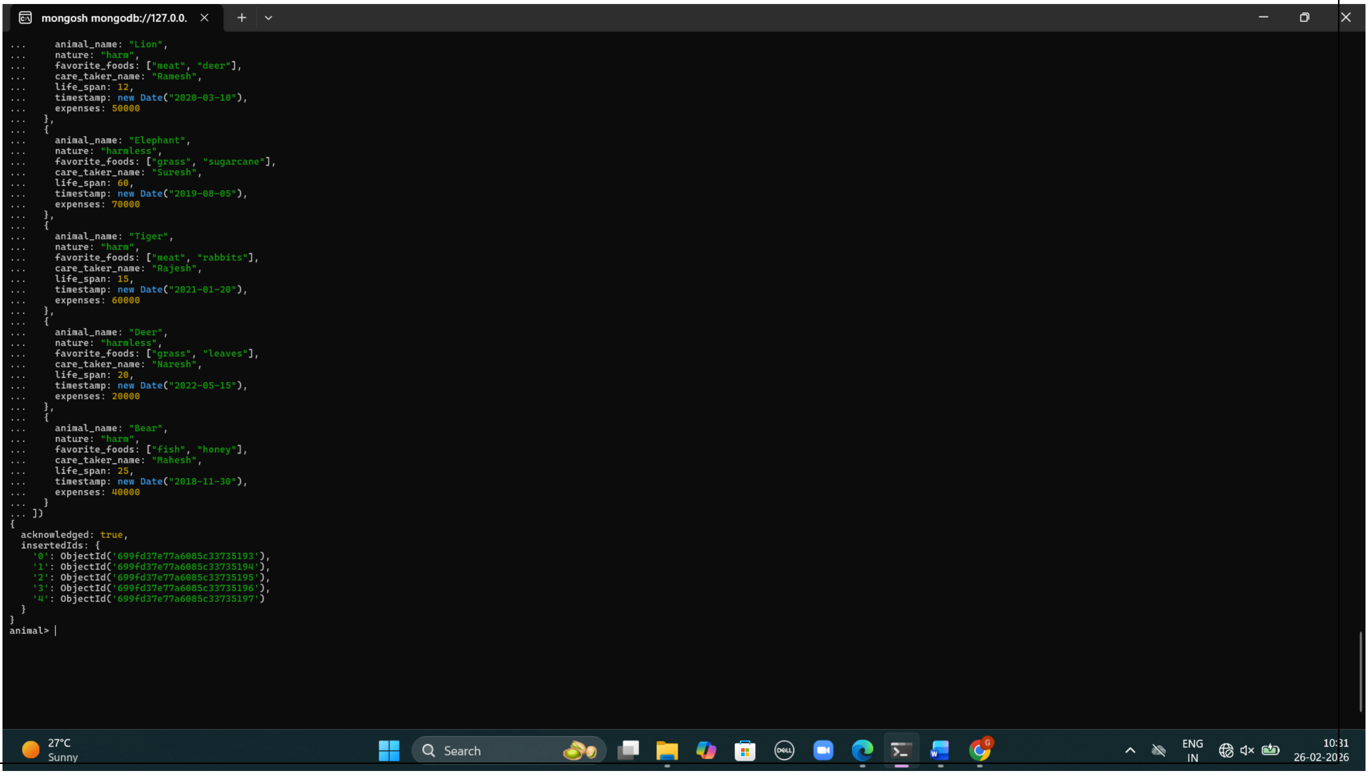
```
]
vehicles>   use animal
switched to db animal
animal>  show dbs
admin         40.00 KiB
animal       112.00 KiB
config       108.00 KiB
coursesdb     56.00 KiB
employeedb    56.00 KiB
employeesdb   40.00 KiB
local         88.00 KiB
stockmarket  404.00 KiB
test         144.00 KiB
vehicles      80.00 KiB
animal> db.dropdatabase()
TypeError: db.dropdatabase is not a function
animal> db.dropDatabase()
{ ok: 1, dropped: 'animal' }
animal> show dbs
admin         40.00 KiB
config       108.00 KiB
coursesdb     56.00 KiB
employeedb    56.00 KiB
employeesdb   40.00 KiB
local         88.00 KiB
stockmarket  404.00 KiB
test         144.00 KiB
vehicles      80.00 KiB
animal> db.createCollection("wild_animals", { capped: true, size: 5242880, max: 1000 })
{ ok: 1 }
animal> db.createCollection("domestic_animals")
{ ok: 1 }
animal> db.wild_animals.insertMany([
...   {
...     animal_name: "Lion",
...     nature: "harm",
...     favorite_foods: ["meat", "deer"],
...     care_taker_name: "Ramesh",
...     life_span: 12,
...     timestamp: new Date("2020-03-10"),
...     expenses: 50000
...   },
...   {
...     animal_name: "Elephant",
...     nature: "harmless",
...     favorite_foods: ["grass", "sugarcane"],
...     care_taker_name: "Suresh",
...     life_span: 60,
...     timestamp: new Date("2019-08-05"),
...     expenses: 70000
...   },
...   {
...     animal_name: "Tiger",
...     nature: "harm",
...     favorite_foods: ["meat", "rabbits"],
...     care_taker_name: "Rajesh",
...     life_span: 15,
...     timestamp: new Date("2021-01-20"),
...     expenses: 60000
...   },
...   {
...     animal_name: "Deer",
...     nature: "harmless",
...     favorite_foods: ["grass", "leaves"],
```

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.**

```
db.wild_animals.insertMany([
 {
   animal_name: "Lion",
   nature: "harm",
   favorite_foods: ["meat", "deer"],
   care_taker_name: "Ramesh",
   life_span: 12,
   timestamp: new Date("2020-03-10"),
   expenses: 50000
 },
 {
   animal_name: "Elephant",
   nature: "harmless",
   favorite_foods: ["grass", "sugarcane"],
   care_taker_name: "Suresh",
   life_span: 60,
   timestamp: new Date("2019-08-05"),
   expenses: 70000
 },
 {
   animal_name: "Tiger",
   nature: "harm",
   favorite_foods: ["meat", "rabbits"],
   care_taker_name: "Rajesh",
   life_span: 15,
   timestamp: new Date("2021-01-20"),
   expenses: 60000
 },
 {
   animal_name: "Deer",
   nature: "harmless",
   favorite_foods: ["grass", "leaves"],
   care_taker_name: "Naresh",
   life_span: 20,
   timestamp: new Date("2022-05-15"),
   expenses: 20000
 },
    {
      animal_name: "Bear",
      nature: "harm",
      favorite_foods: ["fish", "honey"],
      care_taker_name: "Mahesh",
      life_span: 25,
      timestamp: new Date("2018-11-30"),
      expenses: 40000
    }
  ])
```



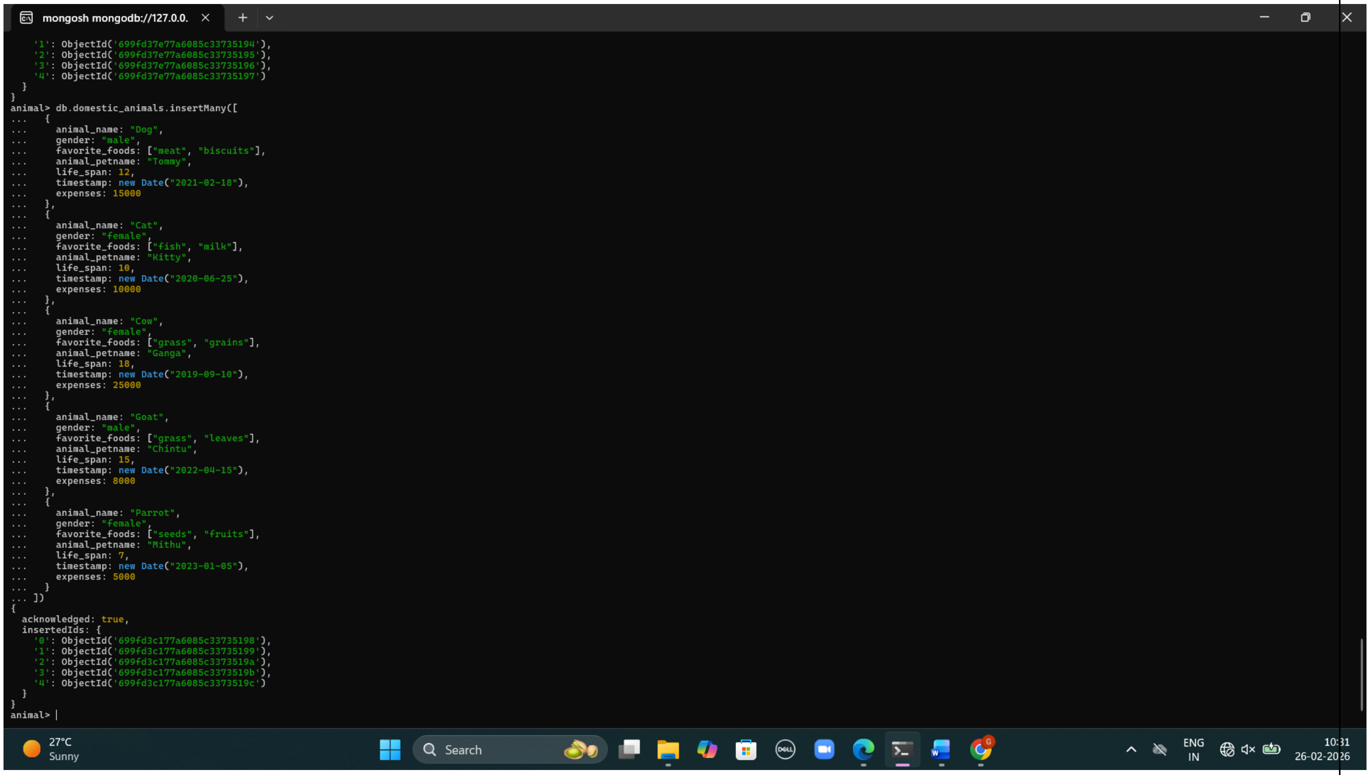
```
]
vehicles>   use animal
switched to db animal
animal>  show dbs
admin          40.00 KiB
animal        112.00 KiB
config        108.00 KiB
coursesdb      56.00 KiB
employeedb     56.00 KiB
employeesdb    40.00 KiB
local          88.00 KiB
stockmarket   404.00 KiB
test          144.00 KiB
vehicles       80.00 KiB
animal> db.dropdatabase()
TypeError: db.dropdatabase is not a function
animal> db.dropDatabase()
{ ok: 1, dropped: 'animal' }
animal> show dbs
admin          40.00 KiB
config        108.00 KiB
coursesdb      56.00 KiB
employeedb     56.00 KiB
employeesdb    40.00 KiB
local          88.00 KiB
stockmarket   404.00 KiB
test          144.00 KiB
vehicles       80.00 KiB
animal> db.createCollection("wild_animals", { capped: true, size: 5242880, max: 1000 })
{ ok: 1 }
animal> db.createCollection("domestic_animals")
{ ok: 1 }
animal> db.wild_animals.insertMany([
...     {
...       animal_name: "Lion",
...       nature: "harm",
...       favorite_foods: ["meat", "deer"],
...       care_taker_name: "Ramesh",
...       life_span: 12,
...       timestamp: new Date("2020-03-10"),
...       expenses: 50000
...     },
...     {
...       animal_name: "Elephant",
...       nature: "harmless",
...       favorite_foods: ["grass", "sugarcane"],
...       care_taker_name: "Suresh",
...       life_span: 60,
...       timestamp: new Date("2019-08-05"),
...       expenses: 70000
...     },
...     {
...       animal_name: "Tiger",
...       nature: "harm",
...       favorite_foods: ["meat", "rabbits"],
...       care_taker_name: "Rajesh",
...       life_span: 15,
...       timestamp: new Date("2021-01-20"),
...       expenses: 60000
...     },
...     {
...       animal_name: "Deer",
...       nature: "harmless",
...       favorite_foods: ["grass", "leaves"],
```



```
...       animal_name: "Lion",
...       nature: "harm",
...       favorite_foods: ["meat", "deer"],
...       care_taker_name: "Ramesh",
...       life_span: 12,
...       timestamp: new Date("2020-03-10"),
...       expenses: 50000
...     },
...     {
...       animal_name: "Elephant",
...       nature: "harmless",
...       favorite_foods: ["grass", "sugarcane"],
...       care_taker_name: "Suresh",
...       life_span: 60,
...       timestamp: new Date("2019-08-05"),
...       expenses: 70000
...     },
...     {
...       animal_name: "Tiger",
...       nature: "harm",
...       favorite_foods: ["meat", "rabbits"],
...       care_taker_name: "Rajesh",
...       life_span: 15,
...       timestamp: new Date("2021-01-20"),
...       expenses: 60000
...     },
...     {
...       animal_name: "Deer",
...       nature: "harmless",
...       favorite_foods: ["grass", "leaves"],
...       care_taker_name: "Naresh",
...       life_span: 20,
...       timestamp: new Date("2022-05-15"),
...       expenses: 20000
...     },
...     {
...       animal_name: "Bear",
...       nature: "harm",
...       favorite_foods: ["fish", "honey"],
...       care_taker_name: "Mahesh",
...       life_span: 25,
...       timestamp: new Date("2018-11-30"),
...       expenses: 40000
...     }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fd37e77a6085c33735193'),
    '1': ObjectId('699fd37e77a6085c33735194'),
    '2': ObjectId('699fd37e77a6085c33735195'),
    '3': ObjectId('699fd37e77a6085c33735196'),
    '4': ObjectId('699fd37e77a6085c33735197')
  }
}
animal> |
```

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.

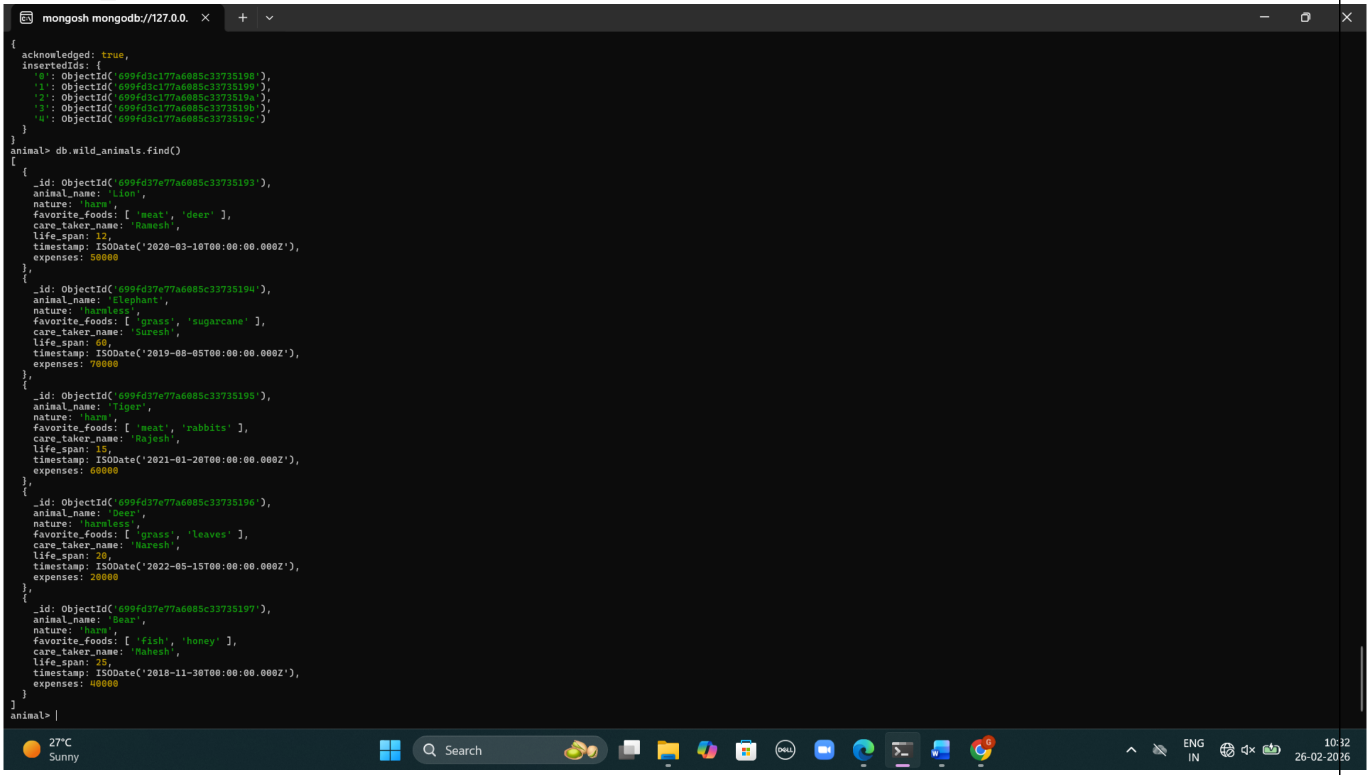
```
db.domestic_animals.insertMany([
{ animal_name: "Dog",
gender: "male",
 favorite_foods: ["meat", "biscuits"],
animal_petname: "Tommy",
life_span: 12,
 timestamp: new Date("2021-02-18"),
expenses: 15000
 },
 { animal_name: "Cat",
 gender: "female",
 favorite_foods: ["fish", "milk"],
 animal_petname: "Kitty",
 life_span: 10,
 timestamp: new Date("2020-06-25"),
 expenses: 10000
 },
{ animal_name: "Cow",
 gender: "female",
favorite_foods: ["grass", "grains"],
 animal_petname: "Ganga",
 life_span: 18,
timestamp: new Date("2019-09-10"),
 expenses: 25000
},
 { animal_name: "Goat",
gender: "male",
 favorite_foods: ["grass", "leaves"],
 animal_petname: "Chintu",
 life_span: 15,
 timestamp: new Date("2022-04-15"),
 expenses: 8000
},
{ animal_name: "Parrot",
 gender: "female",
 favorite_foods: ["seeds", "fruits"],
 animal_petname: "Mithu",
life_span: 7,
timestamp: new Date("2023-01-05"),
expenses: 5000
} ])
```



```
    '1': ObjectId('699fd37e77a6085c33735194'),
    '2': ObjectId('699fd37e77a6085c33735195'),
    '3': ObjectId('699fd37e77a6085c33735196'),
    '4': ObjectId('699fd37e77a6085c33735197')
  }
}
animal> db.domestic_animals.insertMany([
...   {
...     animal_name: "Dog",
...     gender: "male",
...     favorite_foods: ["meat", "biscuits"],
...     animal_petname: "Tommy",
...     life_span: 12,
...     timestamp: new Date("2021-02-18"),
...     expenses: 15000
...   },
...   {
...     animal_name: "Cat",
...     gender: "female",
...     favorite_foods: ["fish", "milk"],
...     animal_petname: "Kitty",
...     life_span: 10,
...     timestamp: new Date("2020-06-25"),
...     expenses: 10000
...   },
...   {
...     animal_name: "Cow",
...     gender: "female",
...     favorite_foods: ["grass", "grains"],
...     animal_petname: "Ganga",
...     life_span: 18,
...     timestamp: new Date("2019-09-10"),
...     expenses: 25000
...   },
...   {
...     animal_name: "Goat",
...     gender: "male",
...     favorite_foods: ["grass", "leaves"],
...     animal_petname: "Chintu",
...     life_span: 15,
...     timestamp: new Date("2022-04-15"),
...     expenses: 8000
...   },
...   {
...     animal_name: "Parrot",
...     gender: "female",
...     favorite_foods: ["seeds", "fruits"],
...     animal_petname: "Mithu",
...     life_span: 7,
...     timestamp: new Date("2023-01-05"),
...     expenses: 5000
...   }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fd3c177a6085c33735198'),
    '1': ObjectId('699fd3c177a6085c33735199'),
    '2': ObjectId('699fd3c177a6085c3373519a'),
    '3': ObjectId('699fd3c177a6085c3373519b'),
    '4': ObjectId('699fd3c177a6085c3373519c')
  }
}
animal> |
```

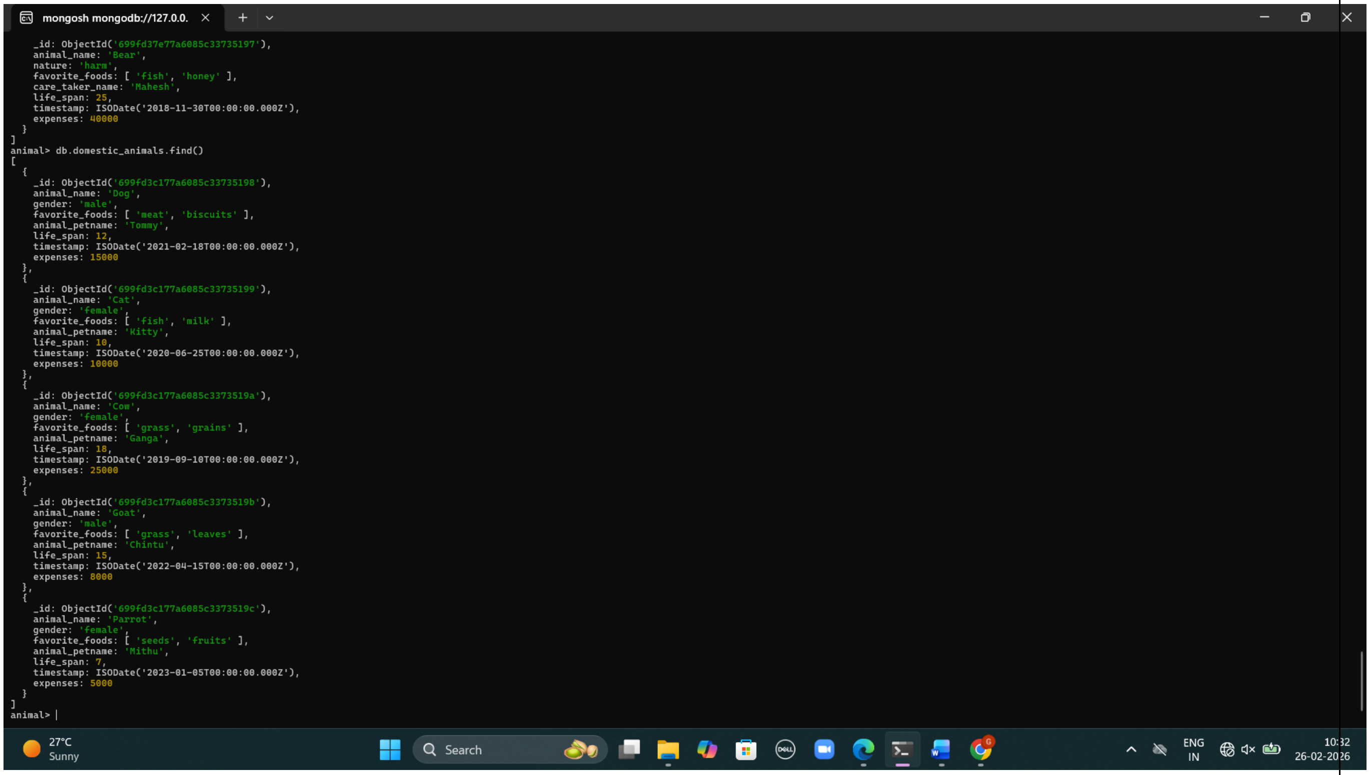
6. Write a MongoDB query to display all documents available in wild_animals and domastic_animals.

db.wild_animals.find()



```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fd3c177a6085c33735198'),
    '1': ObjectId('699fd3c177a6085c33735199'),
    '2': ObjectId('699fd3c177a6085c3373519a'),
    '3': ObjectId('699fd3c177a6085c3373519b'),
    '4': ObjectId('699fd3c177a6085c3373519c')
  }
}
animal> db.wild_animals.find()
[
  {
    _id: ObjectId('699fd37e77a6085c33735193'),
    animal_name: 'Lion',
    nature: 'harm',
    favorite_foods: [ 'meat', 'deer' ],
    care_taker_name: 'Ramesh',
    life_span: 12,
    timestamp: ISODate('2020-03-10T00:00:00.000Z'),
    expenses: 50000
  },
  {
    _id: ObjectId('699fd37e77a6085c33735194'),
    animal_name: 'Elephant',
    nature: 'harmless',
    favorite_foods: [ 'grass', 'sugarcane' ],
    care_taker_name: 'Suresh',
    life_span: 60,
    timestamp: ISODate('2019-08-05T00:00:00.000Z'),
    expenses: 70000
  },
  {
    _id: ObjectId('699fd37e77a6085c33735195'),
    animal_name: 'Tiger',
    nature: 'harm',
    favorite_foods: [ 'meat', 'rabbits' ],
    care_taker_name: 'Rajesh',
    life_span: 15,
    timestamp: ISODate('2021-01-20T00:00:00.000Z'),
    expenses: 60000
  },
  {
    _id: ObjectId('699fd37e77a6085c33735196'),
    animal_name: 'Deer',
    nature: 'harmless',
    favorite_foods: [ 'grass', 'leaves' ],
    care_taker_name: 'Naresh',
    life_span: 20,
    timestamp: ISODate('2022-05-15T00:00:00.000Z'),
    expenses: 20000
  },
  {
    _id: ObjectId('699fd37e77a6085c33735197'),
    animal_name: 'Bear',
    nature: 'harm',
    favorite_foods: [ 'fish', 'honey' ],
    care_taker_name: 'Mahesh',
    life_span: 25,
    timestamp: ISODate('2018-11-30T00:00:00.000Z'),
    expenses: 40000
  }
]
animal> |
```

db.domestic_animals.find()



```
    _id: ObjectId('699fd37e77a6085c33735197'),
    animal_name: 'Bear',
    nature: 'harm',
    favorite_foods: [ 'fish', 'honey' ],
    care_taker_name: 'Mahesh',
    life_span: 25,
    timestamp: ISODate('2018-11-30T00:00:00.000Z'),
    expenses: 40000
  }
]
animal> db.domestic_animals.find()
[
  {
    _id: ObjectId('699fd3c177a6085c33735198'),
    animal_name: 'Dog',
    gender: 'male',
    favorite_foods: [ 'meat', 'biscuits' ],
    animal_petname: 'Tommy',
    life_span: 12,
    timestamp: ISODate('2021-02-18T00:00:00.000Z'),
    expenses: 15000
  },
  {
    _id: ObjectId('699fd3c177a6085c33735199'),
    animal_name: 'Cat',
    gender: 'female',
    favorite_foods: [ 'fish', 'milk' ],
    animal_petname: 'Kitty',
    life_span: 10,
    timestamp: ISODate('2020-06-25T00:00:00.000Z'),
    expenses: 10000
  },
  {
    _id: ObjectId('699fd3c177a6085c3373519a'),
    animal_name: 'Cow',
    gender: 'female',
    favorite_foods: [ 'grass', 'grains' ],
    animal_petname: 'Ganga',
    life_span: 18,
    timestamp: ISODate('2019-09-10T00:00:00.000Z'),
    expenses: 25000
  },
  {
    _id: ObjectId('699fd3c177a6085c3373519b'),
    animal_name: 'Goat',
    gender: 'male',
    favorite_foods: [ 'grass', 'leaves' ],
    animal_petname: 'Chintu',
    life_span: 15,
    timestamp: ISODate('2022-04-15T00:00:00.000Z'),
    expenses: 8000
  },
  {
    _id: ObjectId('699fd3c177a6085c3373519c'),
    animal_name: 'Parrot',
    gender: 'female',
    favorite_foods: [ 'seeds', 'fruits' ],
    animal_petname: 'Mithu',
    life_span: 7,
    timestamp: ISODate('2023-01-05T00:00:00.000Z'),
    expenses: 5000
  }
]
animal> |
```

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

• For wild animals:

db.wild_animals.find({}, { animal_name: 1, expenses: 1, _id: 0 })

• For domestic animals:

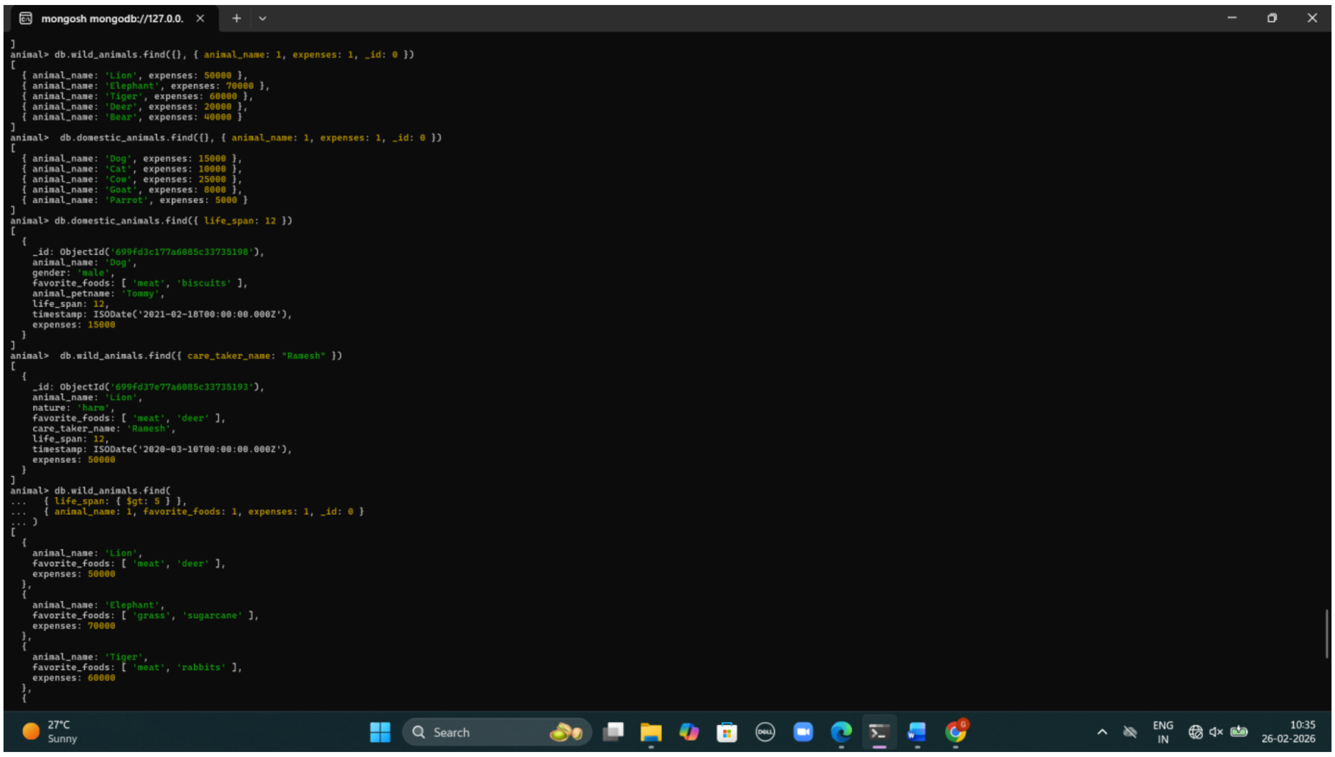
db.domestic_animals.find({}, { animal_name: 1, expenses: 1, _id: 0 })

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

db.domestic_animals.find({ life_span: 12 })

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

db.wild_animals.find({ care_taker_name: "Ramesh" })



```
]
animal> db.wild_animals.find({}, { animal_name: 1, expenses: 1, _id: 0 })
[
  { animal_name: 'Lion', expenses: 50000 },
  { animal_name: 'Elephant', expenses: 70000 },
  { animal_name: 'Tiger', expenses: 60000 },
  { animal_name: 'Deer', expenses: 20000 },
  { animal_name: 'Bear', expenses: 40000 }
]
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: 'Goat', expenses: 8000 },
  { animal_name: 'Parrot', expenses: 5000 }
]
animal> db.domestic_animals.find({ life_span: 12 })
[
  {
    _id: ObjectId('699fd3c177a6085c33735198'),
    animal_name: 'Dog',
    gender: 'male',
    favorite_foods: [ 'meat', 'biscuits' ],
    animal_petname: 'Tommy',
    life_span: 12,
    timestamp: ISODate('2021-02-18T00:00:00.000Z'),
    expenses: 15000
  }
]
animal>  db.wild_animals.find({ care_taker_name: "Ramesh" })
[
  {
    _id: ObjectId('699fd37e77a6085c33735193'),
    animal_name: 'Lion',
    nature: 'harm',
    favorite_foods: [ 'meat', 'deer' ],
    care_taker_name: 'Ramesh',
    life_span: 12,
    timestamp: ISODate('2020-03-10T00:00:00.000Z'),
    expenses: 50000
  }
]
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: 'Elephant',
    favorite_foods: [ 'grass', 'sugarcane' ],
    expenses: 70000
  },
  {
    animal_name: 'Tiger',
    favorite_foods: [ 'meat', 'rabbits' ],
    expenses: 60000
  },
  {
```

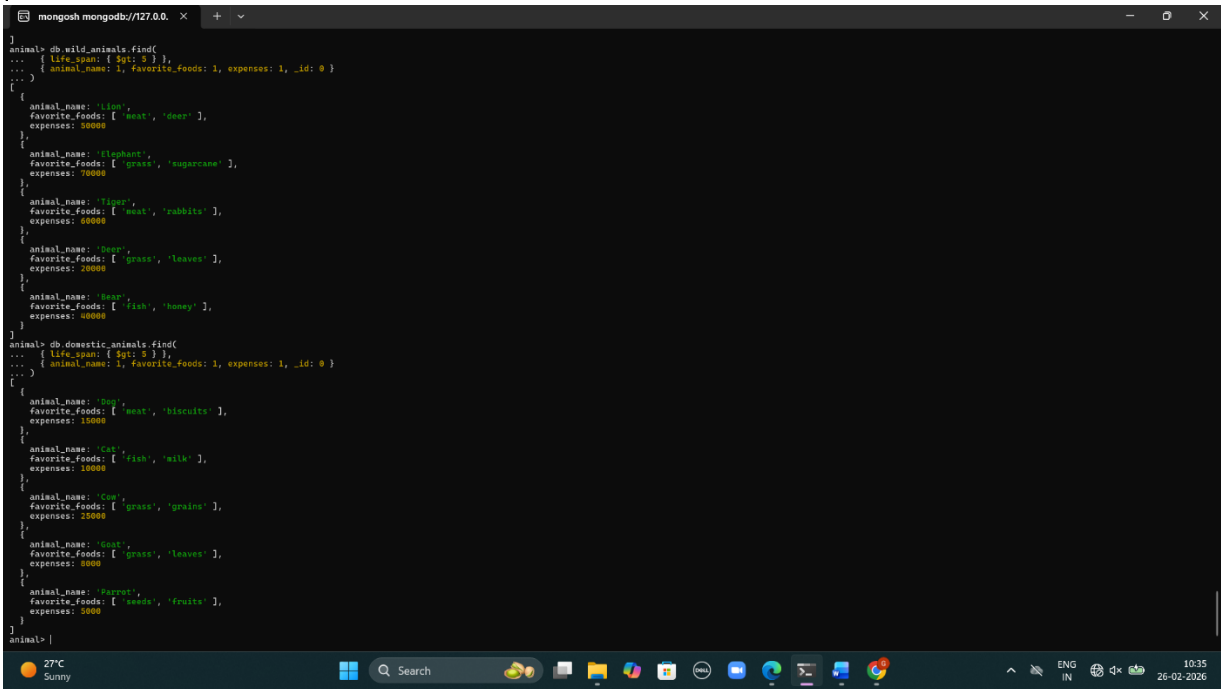
10.Write a MongoDB query to display animal name, favorite_foods and expenses details whose lifespan is more than 5 years.

• For wild animals:

db.wild_animals.find(
{ life_span: { $gt: 5 } },
{ animal_name: 1, favorite_foods: 1, expenses: 1, _id: 0 }
)

• For domestic animals:

db.domestic_animals.find(
{ life_span: { $gt: 5 } },
{ animal_name: 1, favorite_foods: 1, expenses: 1, _id: 0 }
)



```
]
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: 'Elephant',
    favorite_foods: [ 'grass', 'sugarcane' ],
    expenses: 70000
  },
  {
    animal_name: 'Tiger',
    favorite_foods: [ 'meat', 'rabbits' ],
    expenses: 60000
  },
  {
    animal_name: 'Deer',
    favorite_foods: [ 'grass', 'leaves' ],
    expenses: 20000
  },
  {
    animal_name: 'Bear',
    favorite_foods: [ 'fish', 'honey' ],
    expenses: 40000
  }
]
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
  },
  {
    animal_name: 'Cow',
    favorite_foods: [ 'grass', 'grains' ],
    expenses: 25000
  },
  {
    animal_name: 'Goat',
    favorite_foods: [ 'grass', 'leaves' ],
    expenses: 8000
  },
  {
    animal_name: 'Parrot',
    favorite_foods: [ 'seeds', 'fruits' ],
    expenses: 5000
  }
]
animal> |
```