VIT-AP UNIVERSITY, ANDHRA PRADESH

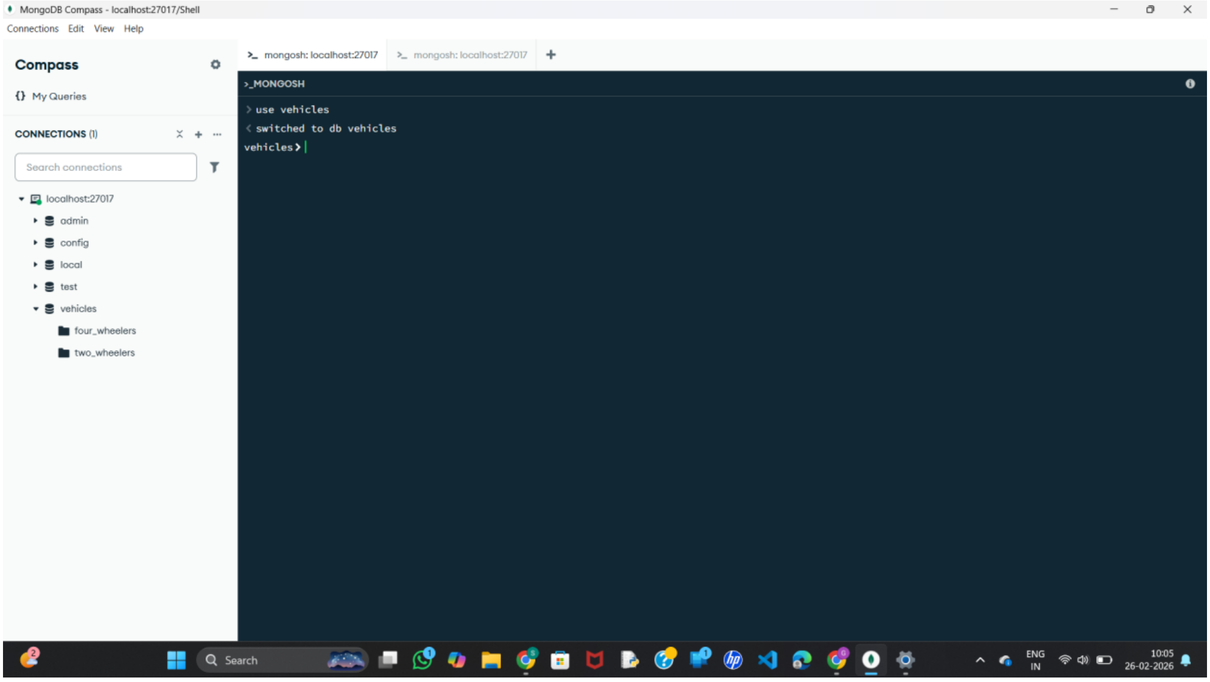
Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech/M.Tech **Date:26-02-2026**
Faculty Name: Prof. S.Gopikrishnan **School:** SCOPE
Student name: G Sai Siva Mani **Reg. no.: 23BCE9808**

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".

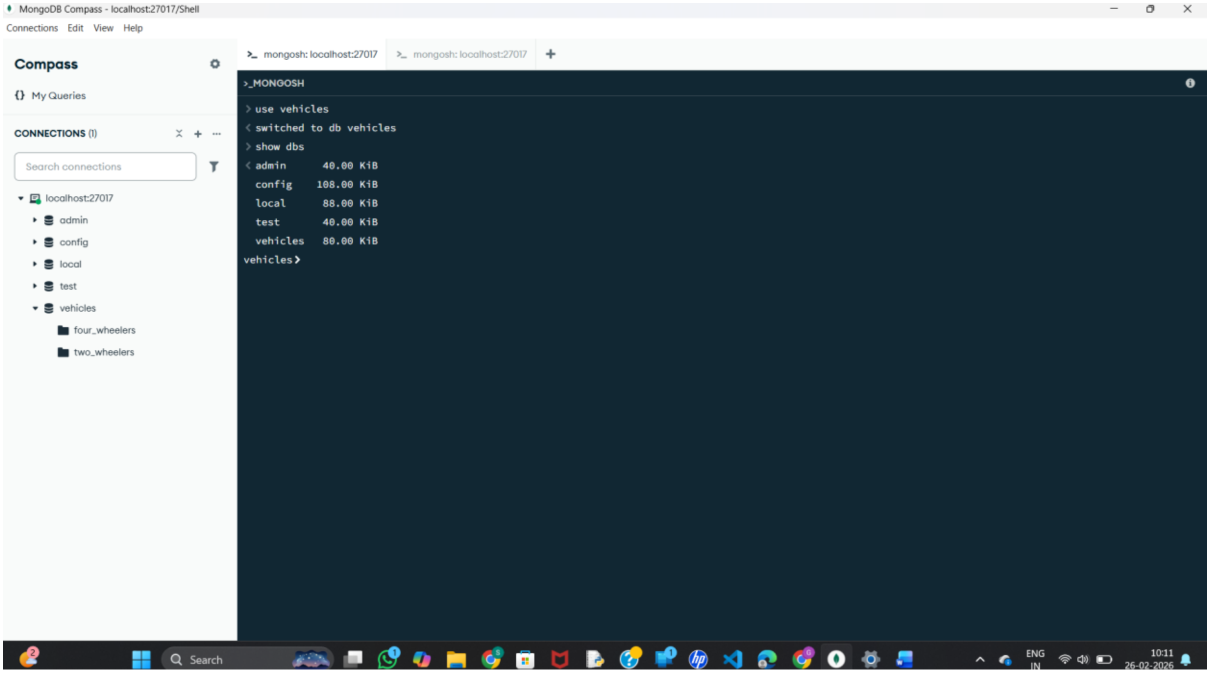
      Code:



```
> use vehicles
< switched to db vehicles
vehicles>
```

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

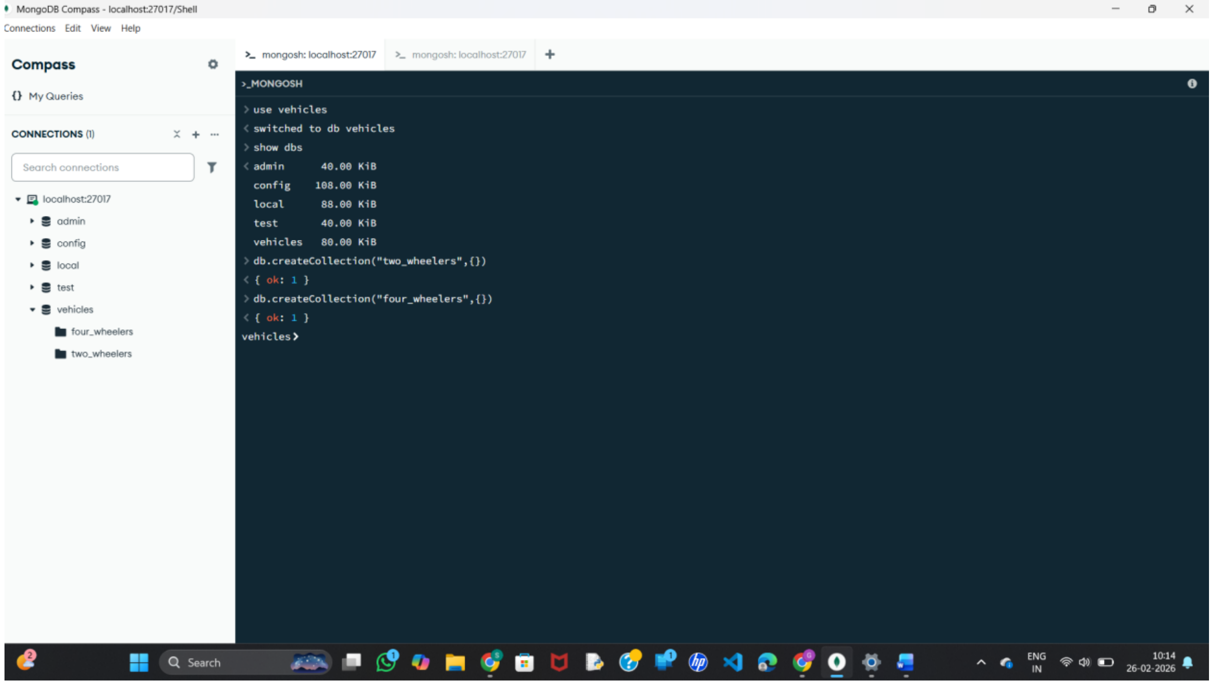
      Code:



```
> use vehicles
< switched to db vehicles
> show dbs
< admin      40.00 KiB
  config    108.00 KiB
  local      88.00 KiB
  test       40.00 KiB
  vehicles   80.00 KiB
vehicles>
```

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

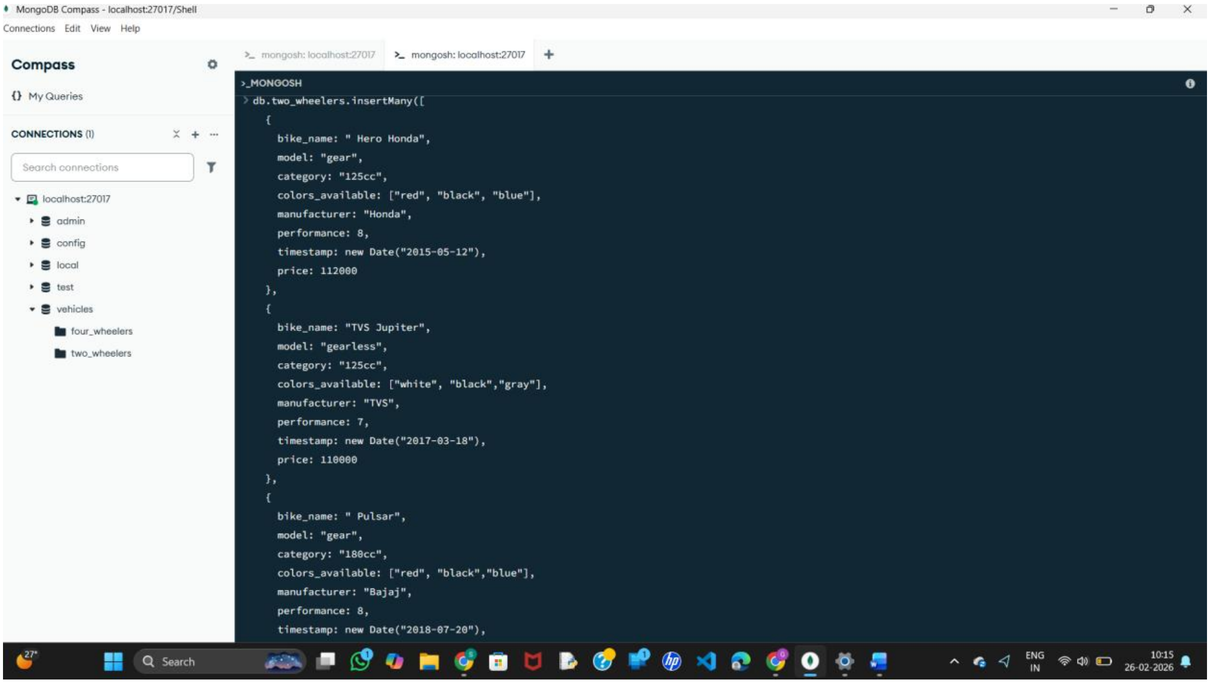
   Code:



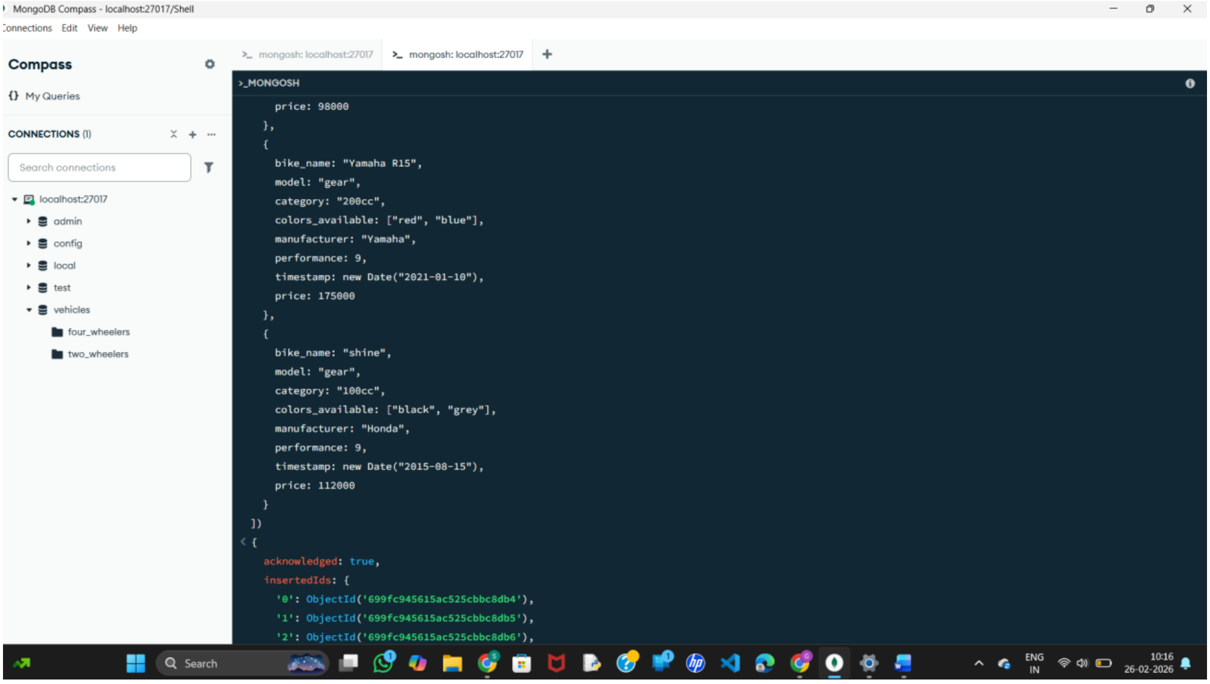
```
> use vehicles
< switched to db vehicles
> show dbs
< admin      40.00 KiB
  config    108.00 KiB
  local      88.00 KiB
  test       40.00 KiB
  vehicles   80.00 KiB
> db.createCollection("two_wheelers",{})
< { ok: 1 }
> db.createCollection("four_wheelers",{})
< { ok: 1 }
vehicles>
```

4. 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.

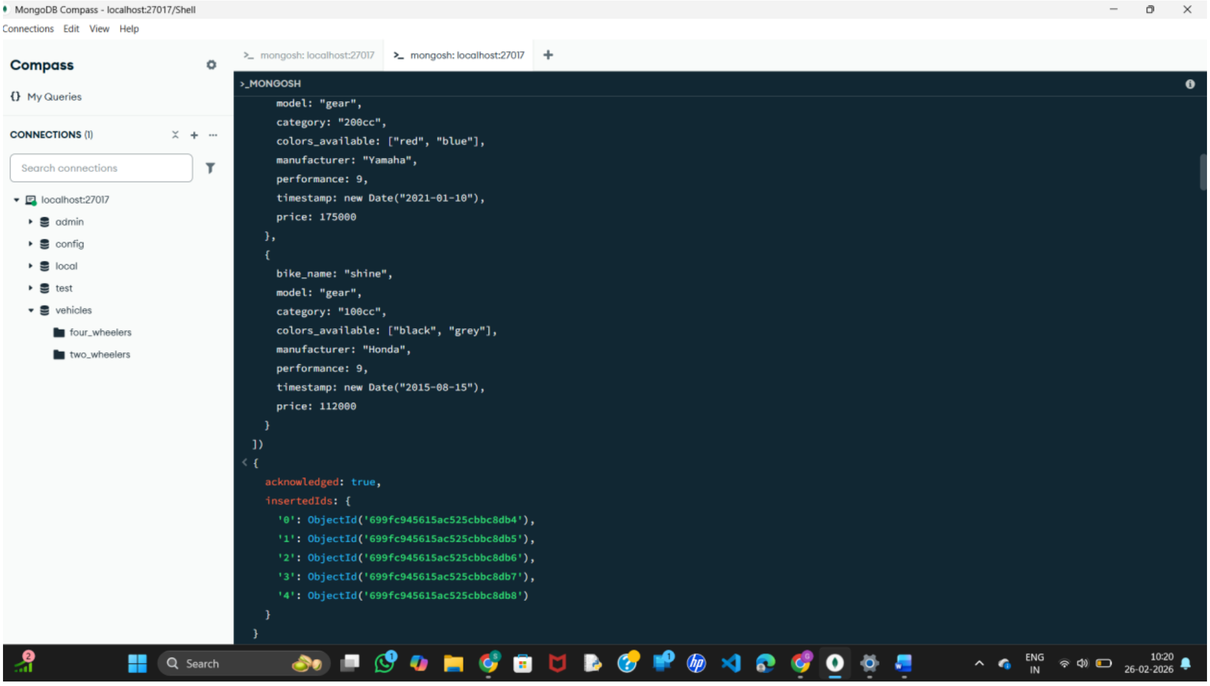
   Code:



```
> db.two_wheelers.insertMany([
    {
      bike_name: " Hero Honda",
      model: "gear",
      category: "125cc",
      colors_available: ["red", "black", "blue"],
      manufacturer: "Honda",
      performance: 8,
      timestamp: new Date("2015-05-12"),
      price: 112000
    },
    {
      bike_name: "TVS Jupiter",
      model: "gearless",
      category: "125cc",
      colors_available: ["white", "black","gray"],
      manufacturer: "TVS",
      performance: 7,
      timestamp: new Date("2017-03-18"),
      price: 110000
    },
    {
      bike_name: " Pulsar",
      model: "gear",
      category: "180cc",
      colors_available: ["red", "black","blue"],
      manufacturer: "Bajaj",
      performance: 8,
      timestamp: new Date("2018-07-20"),
```



```
      price: 98000
    },
    {
      bike_name: "Yamaha R15",
      model: "gear",
      category: "200cc",
      colors_available: ["red", "blue"],
      manufacturer: "Yamaha",
      performance: 9,
      timestamp: new Date("2021-01-10"),
      price: 175000
    },
    {
      bike_name: "shine",
      model: "gear",
      category: "100cc",
      colors_available: ["black", "grey"],
      manufacturer: "Honda",
      performance: 9,
      timestamp: new Date("2015-08-15"),
      price: 112000
    }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('699fc945615ac525cbbc8db4'),
      '1': ObjectId('699fc945615ac525cbbc8db5'),
      '2': ObjectId('699fc945615ac525cbbc8db6'),
```

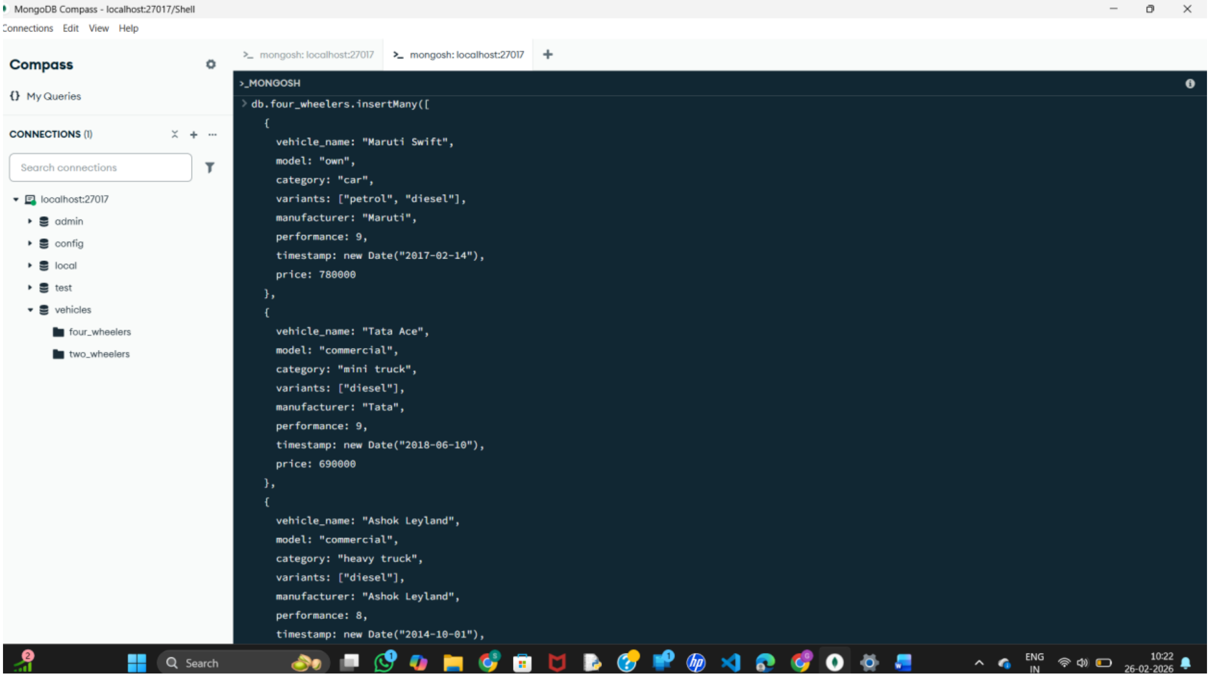


```
    model: "gear",
    category: "200cc",
    colors_available: ["red", "blue"],
    manufacturer: "Yamaha",
    performance: 9,
    timestamp: new Date("2021-01-10"),
    price: 175000
  },
  {
    bike_name: "shine",
    model: "gear",
    category: "100cc",
    colors_available: ["black", "grey"],
    manufacturer: "Honda",
    performance: 9,
    timestamp: new Date("2015-08-15"),
    price: 112000
  }
])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fc945615ac525cbbc8db4'),
    '1': ObjectId('699fc945615ac525cbbc8db5'),
    '2': ObjectId('699fc945615ac525cbbc8db6'),
    '3': ObjectId('699fc945615ac525cbbc8db7'),
    '4': ObjectId('699fc945615ac525cbbc8db8')
  }
}
```

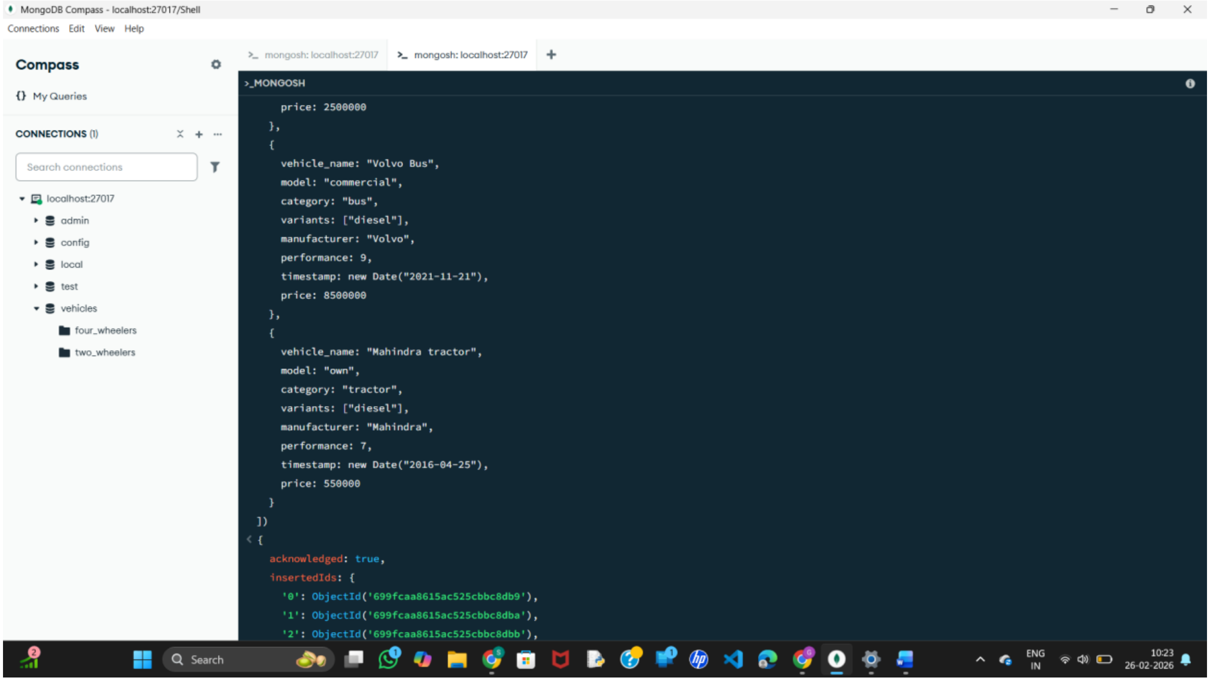
5. Add 5 four-wheeler details to the collection named 'four_wheelers'. Each document consists of following fields as vehicle_name, model (commercial or own), category (car, lorry, bus, mini truck, heavy truck, containers), variants (vxi, zxi, petrol, diesel etc) as array, manufacturer, performance (out of 10), timestamp (date and year release) and price.

   Code:

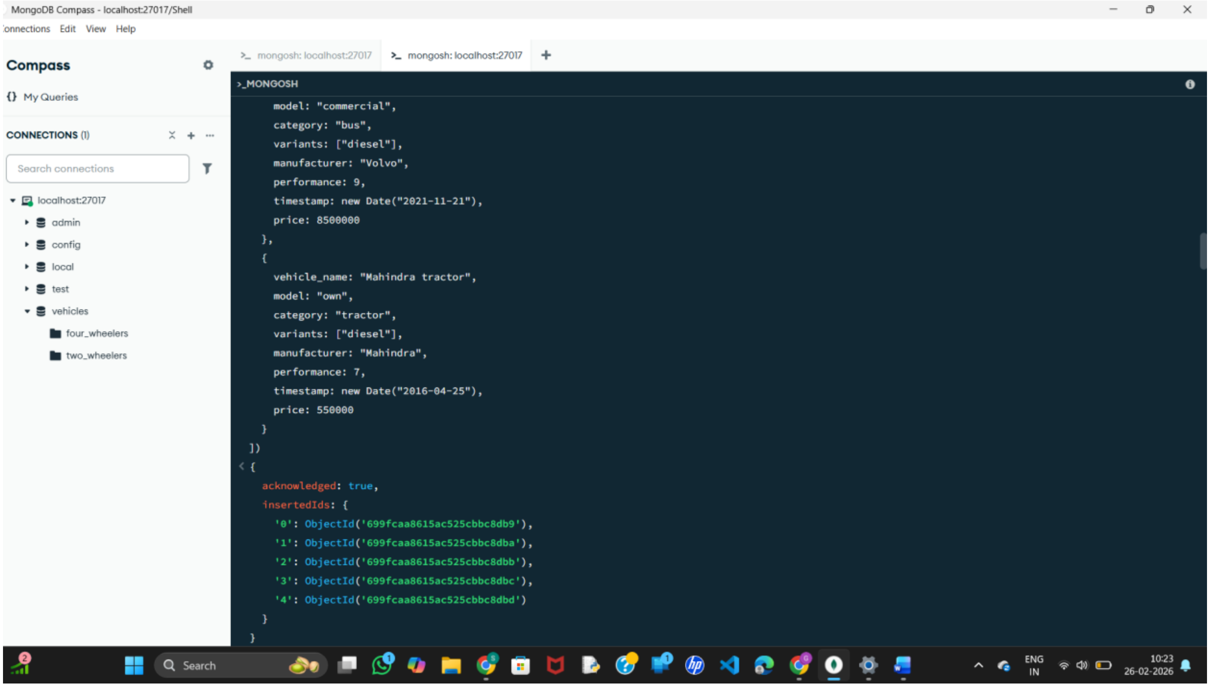
   Code:



```
> db.four_wheelers.insertMany([
  {
    vehicle_name: "Maruti Swift",
    model: "own",
    category: "car",
    variants: ["petrol", "diesel"],
    manufacturer: "Maruti",
    performance: 9,
    timestamp: new Date("2017-02-14"),
    price: 780000
  },
  {
    vehicle_name: "Tata Ace",
    model: "commercial",
    category: "mini truck",
    variants: ["diesel"],
    manufacturer: "Tata",
    performance: 9,
    timestamp: new Date("2018-06-10"),
    price: 690000
  },
  {
    vehicle_name: "Ashok Leyland",
    model: "commercial",
    category: "heavy truck",
    variants: ["diesel"],
    manufacturer: "Ashok Leyland",
    performance: 8,
    timestamp: new Date("2014-10-01"),
```



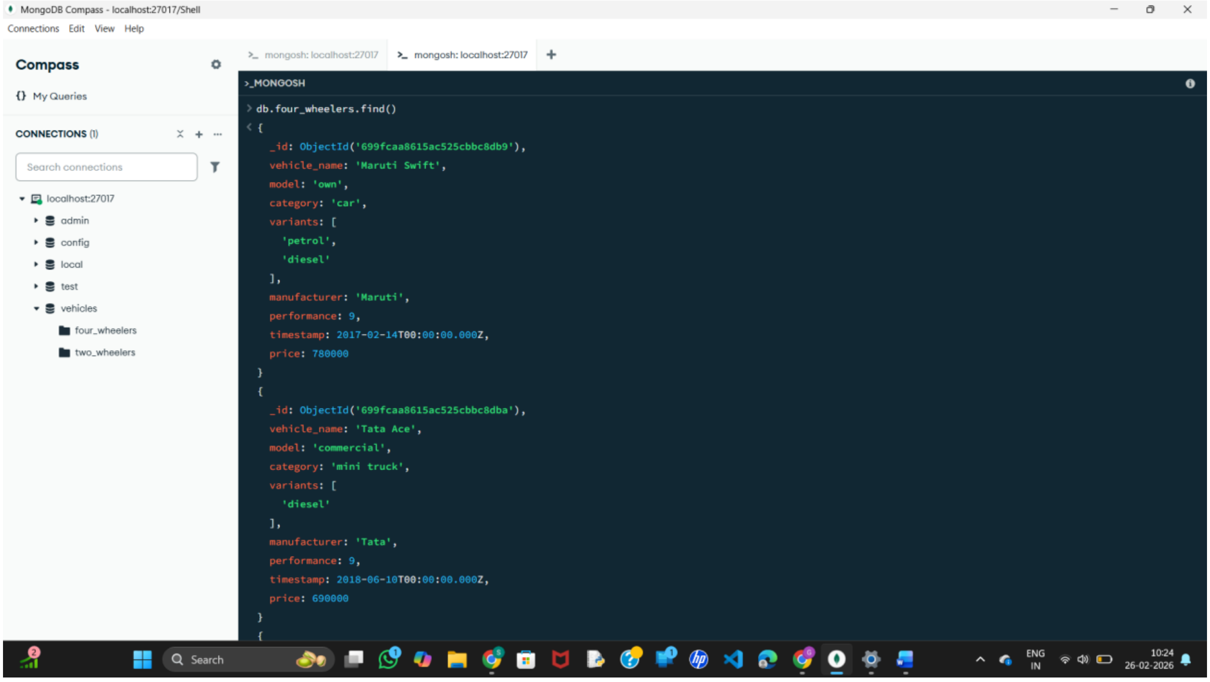
```
      price: 2500000
    },
    {
      vehicle_name: "Volvo Bus",
      model: "commercial",
      category: "bus",
      variants: ["diesel"],
      manufacturer: "Volvo",
      performance: 9,
      timestamp: new Date("2021-11-21"),
      price: 8500000
    },
    {
      vehicle_name: "Mahindra tractor",
      model: "own",
      category: "tractor",
      variants: ["diesel"],
      manufacturer: "Mahindra",
      performance: 7,
      timestamp: new Date("2016-04-25"),
      price: 550000
    }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('699fcaa8615ac525cbbc8db9'),
      '1': ObjectId('699fcaa8615ac525cbbc8dba'),
      '2': ObjectId('699fcaa8615ac525cbbc8dbb'),
```



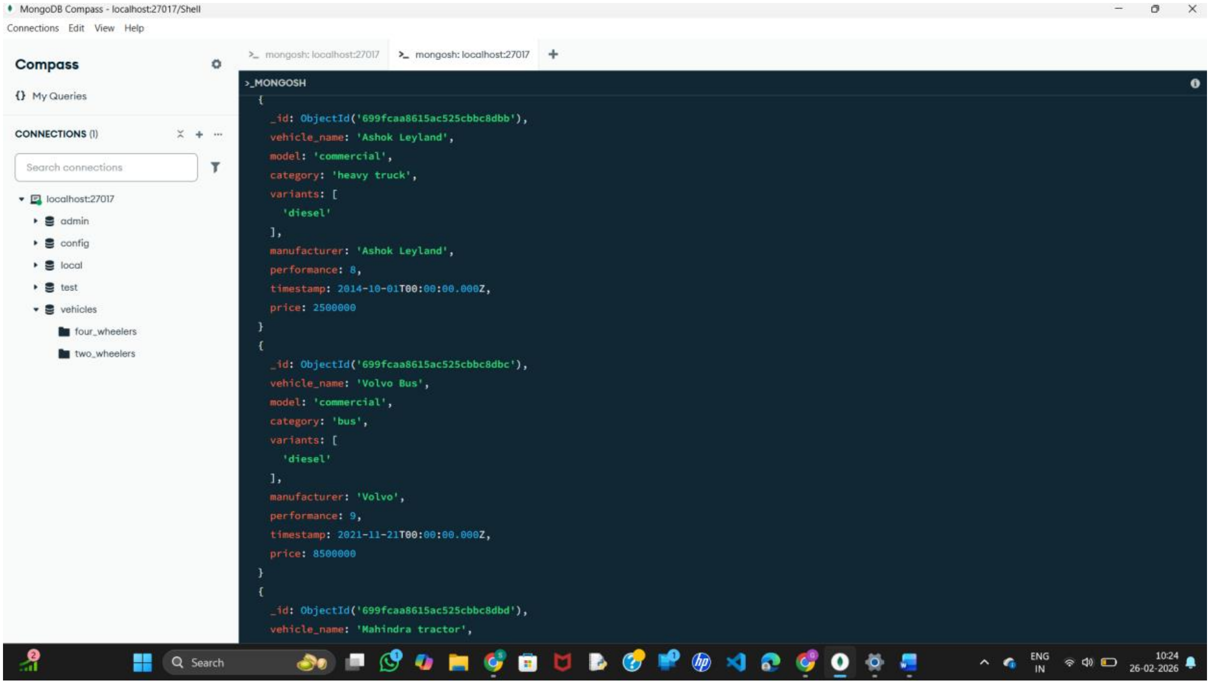
```
      model: "commercial",
      category: "bus",
      variants: ["diesel"],
      manufacturer: "Volvo",
      performance: 9,
      timestamp: new Date("2021-11-21"),
      price: 8500000
    },
    {
      vehicle_name: "Mahindra tractor",
      model: "own",
      category: "tractor",
      variants: ["diesel"],
      manufacturer: "Mahindra",
      performance: 7,
      timestamp: new Date("2016-04-25"),
      price: 550000
    }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('699fcaa8615ac525cbbc8db9'),
      '1': ObjectId('699fcaa8615ac525cbbc8dba'),
      '2': ObjectId('699fcaa8615ac525cbbc8dbb'),
      '3': ObjectId('699fcaa8615ac525cbbc8dbc'),
      '4': ObjectId('699fcaa8615ac525cbbc8dbd')
    }
  }
```

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

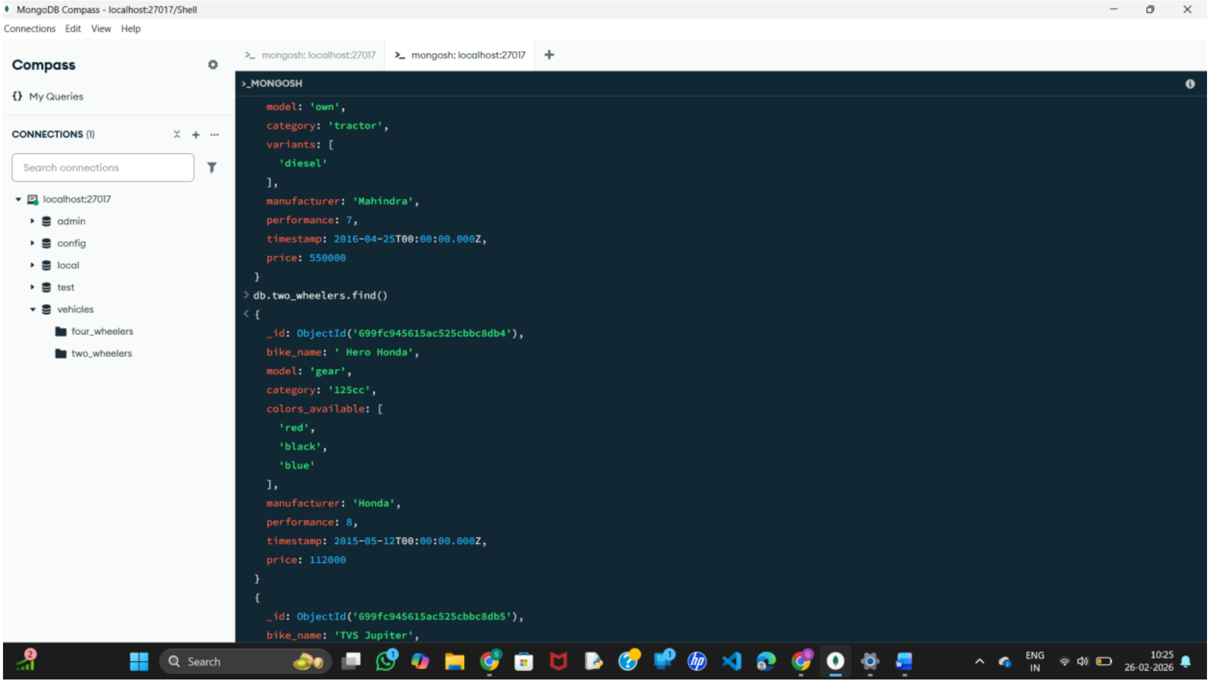
   Code:



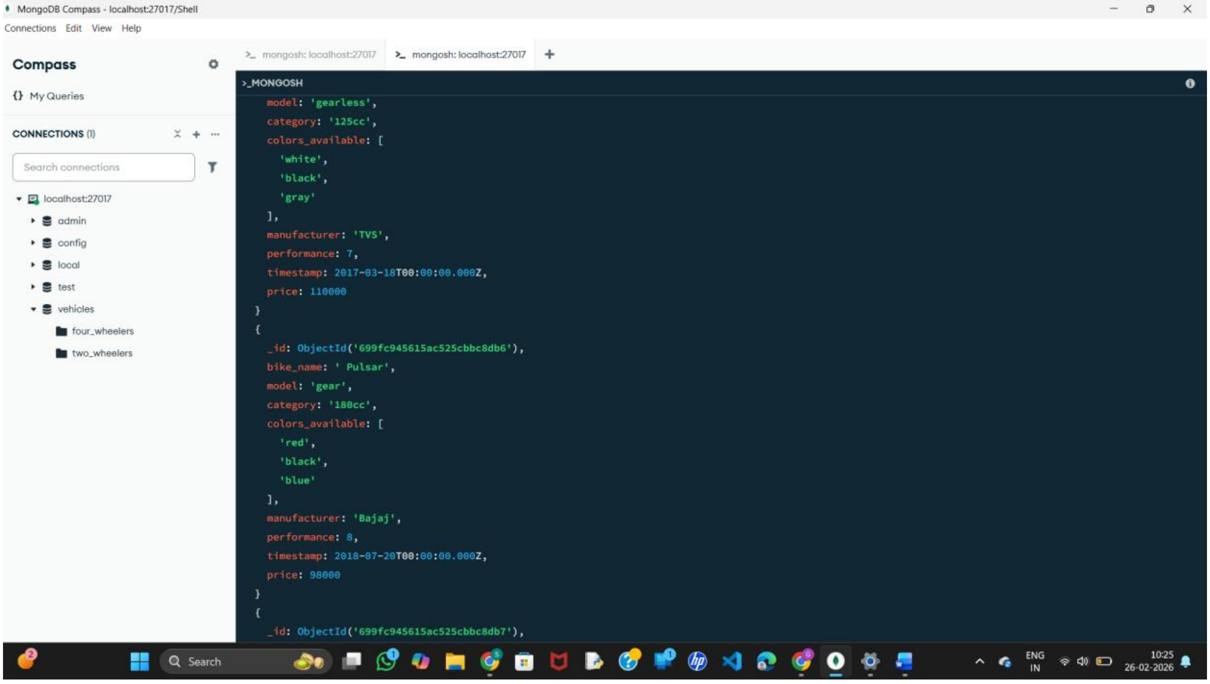
```
> db.four_wheelers.find()
< {
    _id: ObjectId('699fcaa8615ac525cbbc8db9'),
    vehicle_name: 'Maruti Swift',
    model: 'own',
    category: 'car',
    variants: [
      'petrol',
      'diesel'
    ],
    manufacturer: 'Maruti',
    performance: 9,
    timestamp: 2017-02-14T00:00:00.000Z,
    price: 780000
  }
  {
    _id: ObjectId('699fcaa8615ac525cbbc8dba'),
    vehicle_name: 'Tata Ace',
    model: 'commercial',
    category: 'mini truck',
    variants: [
      'diesel'
    ],
    manufacturer: 'Tata',
    performance: 9,
    timestamp: 2018-06-10T00:00:00.000Z,
    price: 690000
  }
  {
```



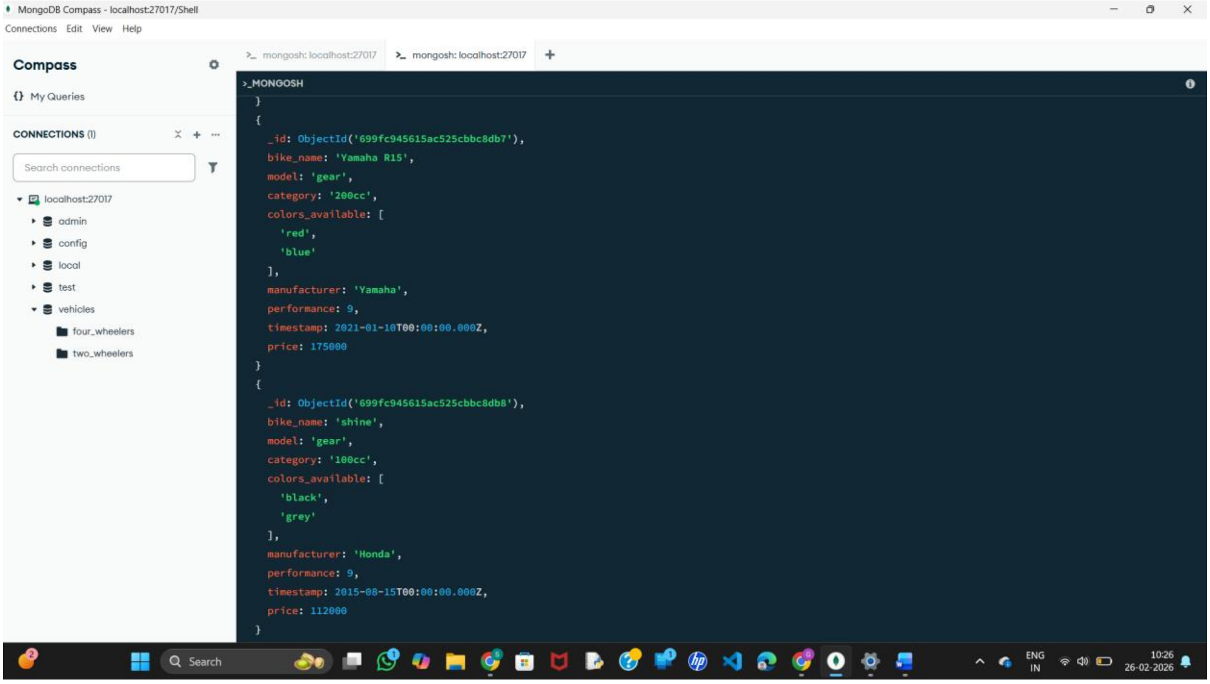
```
  {
    _id: ObjectId('699fcaa8615ac525cbbc8dbb'),
    vehicle_name: 'Ashok Leyland',
    model: 'commercial',
    category: 'heavy truck',
    variants: [
      'diesel'
    ],
    manufacturer: 'Ashok Leyland',
    performance: 8,
    timestamp: 2014-10-01T00:00:00.000Z,
    price: 2500000
  }
  {
    _id: ObjectId('699fcaa8615ac525cbbc8dbc'),
    vehicle_name: 'Volvo Bus',
    model: 'commercial',
    category: 'bus',
    variants: [
      'diesel'
    ],
    manufacturer: 'Volvo',
    performance: 9,
    timestamp: 2021-11-21T00:00:00.000Z,
    price: 8500000
  }
  {
    _id: ObjectId('699fcaa8615ac525cbbc8dbd'),
    vehicle_name: 'Mahindra tractor',
```



```
    model: 'own',
    category: 'tractor',
    variants: [
      'diesel'
    ],
    manufacturer: 'Mahindra',
    performance: 7,
    timestamp: 2016-04-25T00:00:00.000Z,
    price: 550000
  }
> db.two_wheelers.find()
< {
    _id: ObjectId('699fc945615ac525cbbc8db4'),
    bike_name: ' Hero Honda',
    model: 'gear',
    category: '125cc',
    colors_available: [
      'red',
      'black',
      'blue'
    ],
    manufacturer: 'Honda',
    performance: 8,
    timestamp: 2015-05-12T00:00:00.000Z,
    price: 112000
  }
  {
    _id: ObjectId('699fc945615ac525cbbc8db5'),
    bike_name: 'TVS Jupiter',
```



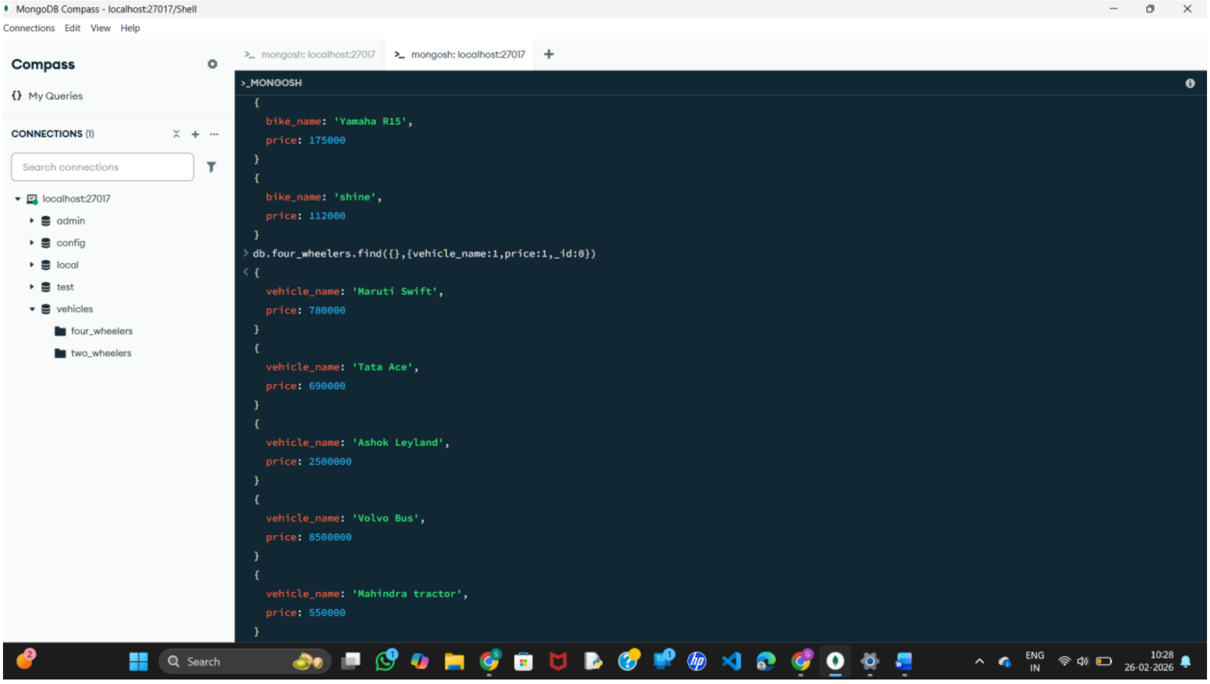
```
    model: 'gearless',
    category: '125cc',
    colors_available: [
      'white',
      'black',
      'gray'
    ],
    manufacturer: 'TVS',
    performance: 7,
    timestamp: 2017-03-18T00:00:00.000Z,
    price: 110000
  }
  {
    _id: ObjectId('699fc945615ac525cbbc8db6'),
    bike_name: ' Pulsar',
    model: 'gear',
    category: '180cc',
    colors_available: [
      'red',
      'black',
      'blue'
    ],
    manufacturer: 'Bajaj',
    performance: 8,
    timestamp: 2018-07-20T00:00:00.000Z,
    price: 98000
  }
  {
    _id: ObjectId('699fc945615ac525cbbc8db7'),
```



```
  }
  {
    _id: ObjectId('699fc945615ac525cbbc8db7'),
    bike_name: 'Yamaha R15',
    model: 'gear',
    category: '200cc',
    colors_available: [
      'red',
      'blue'
    ],
    manufacturer: 'Yamaha',
    performance: 9,
    timestamp: 2021-01-10T00:00:00.000Z,
    price: 175000
  }
  {
    _id: ObjectId('699fc945615ac525cbbc8db8'),
    bike_name: 'shine',
    model: 'gear',
    category: '100cc',
    colors_available: [
      'black',
      'grey'
    ],
    manufacturer: 'Honda',
    performance: 9,
    timestamp: 2015-08-15T00:00:00.000Z,
    price: 112000
  }
```

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

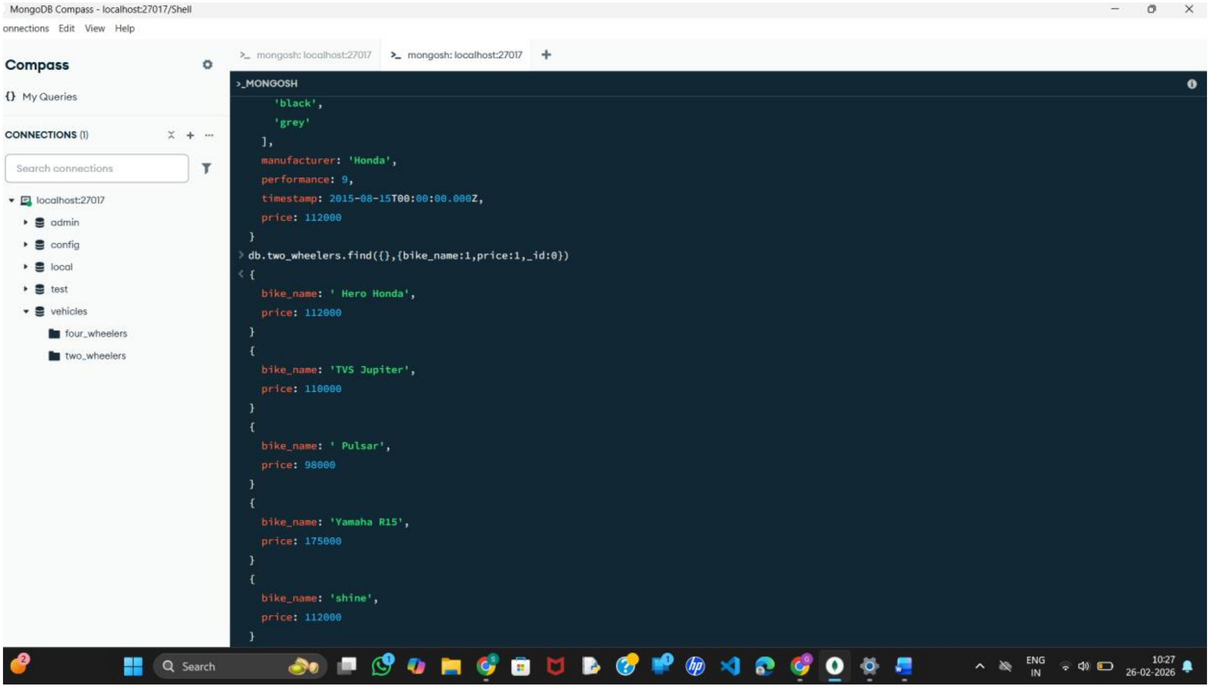
   Code:



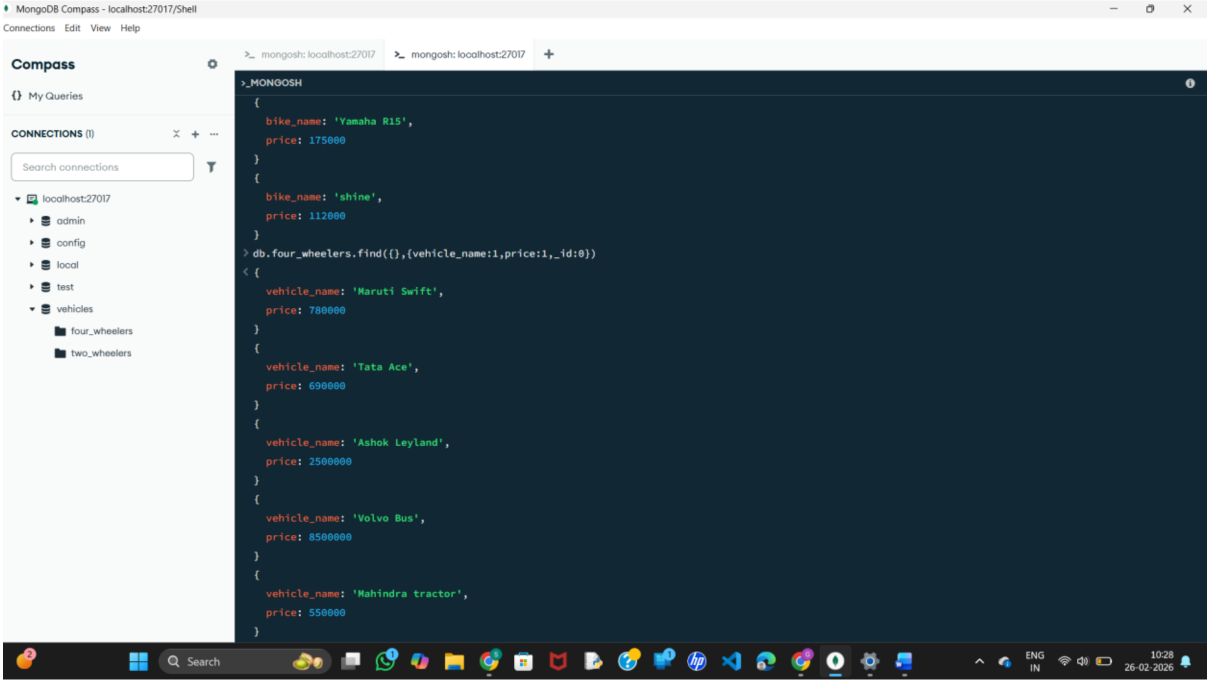
```
  {
    bike_name: 'Yamaha R15',
    price: 175000
  }
  {
    bike_name: 'shine',
    price: 112000
  }
> db.four_wheelers.find({},{vehicle_name:1,price:1,_id:0})
< {
    vehicle_name: 'Maruti Swift',
    price: 780000
  }
  {
    vehicle_name: 'Tata Ace',
    price: 690000
  }
  {
    vehicle_name: 'Ashok Leyland',
    price: 2500000
  }
  {
    vehicle_name: 'Volvo Bus',
    price: 8500000
  }
  {
    vehicle_name: 'Mahindra tractor',
    price: 550000
  }
```

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

   Code:



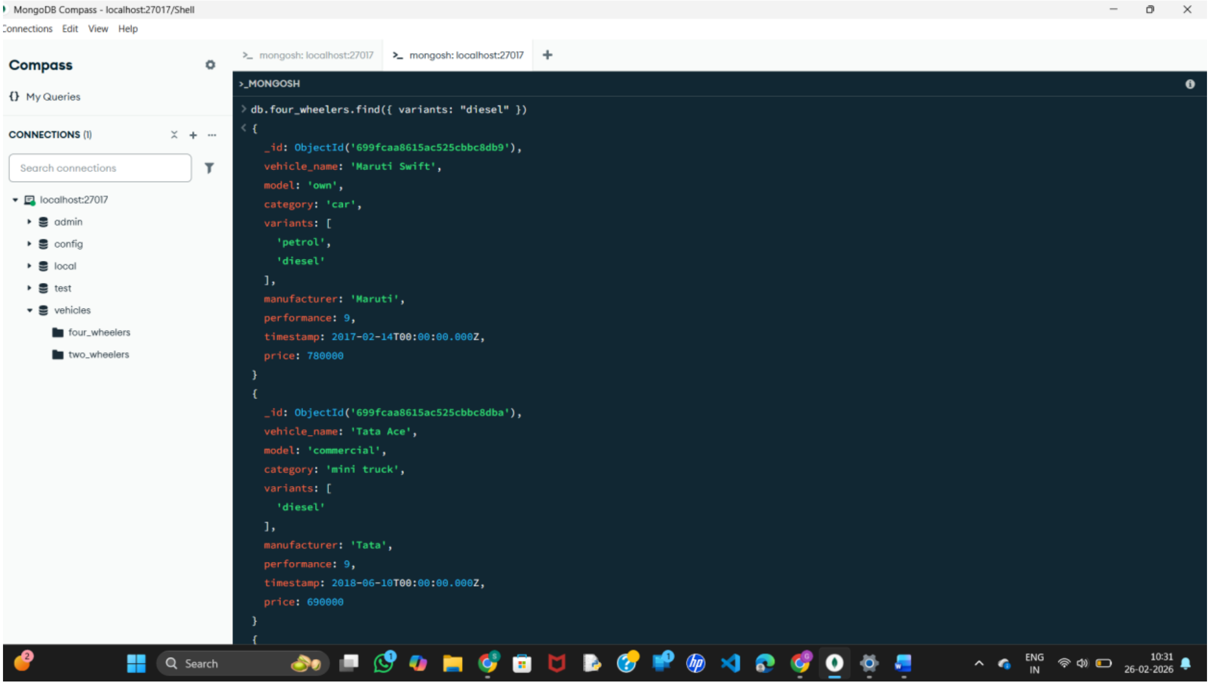
```
      'black',
      'grey'
    ],
    manufacturer: 'Honda',
    performance: 9,
    timestamp: 2015-08-15T00:00:00.000Z,
    price: 112000
  }
> db.two_wheelers.find({},{bike_name:1,price:1,_id:0})
< {
    bike_name: ' Hero Honda',
    price: 112000
  }
  {
    bike_name: 'TVS Jupiter',
    price: 110000
  }
  {
    bike_name: ' Pulsar',
    price: 98000
  }
  {
    bike_name: 'Yamaha R15',
    price: 175000
  }
  {
    bike_name: 'shine',
    price: 112000
  }
```



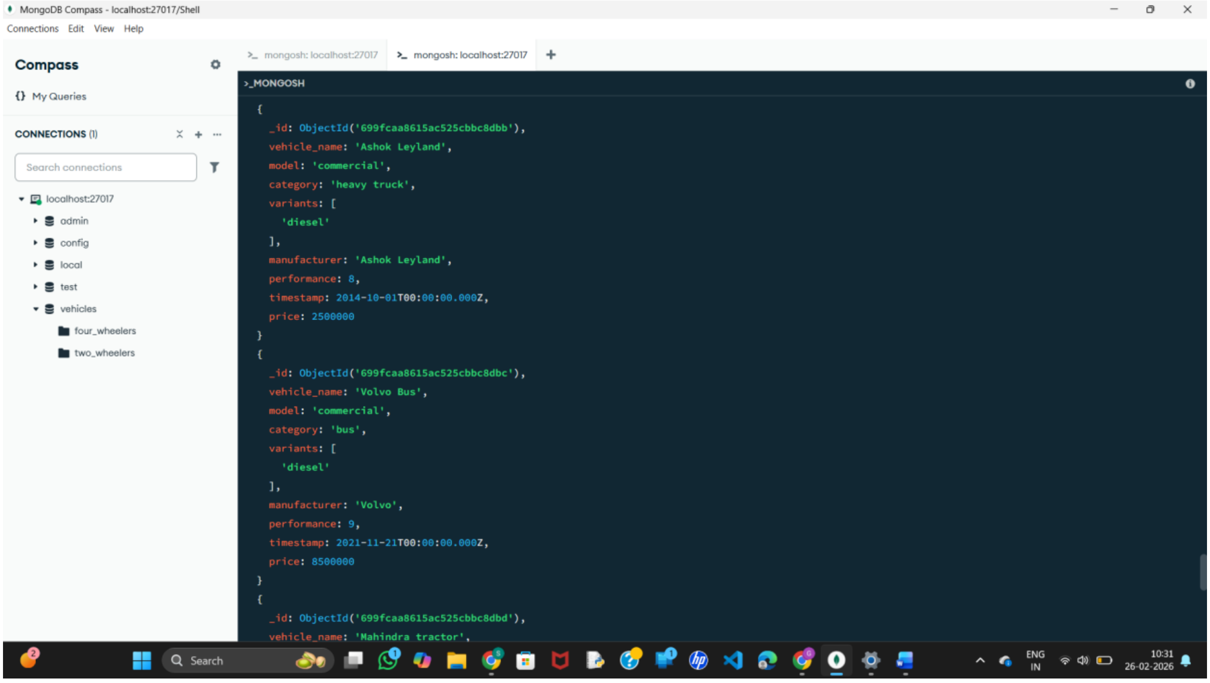
```
  {
    bike_name: 'Yamaha R15',
    price: 175000
  }
  {
    bike_name: 'shine',
    price: 112000
  }
> db.four_wheelers.find({},{vehicle_name:1,price:1,_id:0})
< {
    vehicle_name: 'Maruti Swift',
    price: 780000
  }
  {
    vehicle_name: 'Tata Ace',
    price: 690000
  }
  {
    vehicle_name: 'Ashok Leyland',
    price: 2500000
  }
  {
    vehicle_name: 'Volvo Bus',
    price: 8500000
  }
  {
    vehicle_name: 'Mahindra tractor',
    price: 550000
  }
```

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

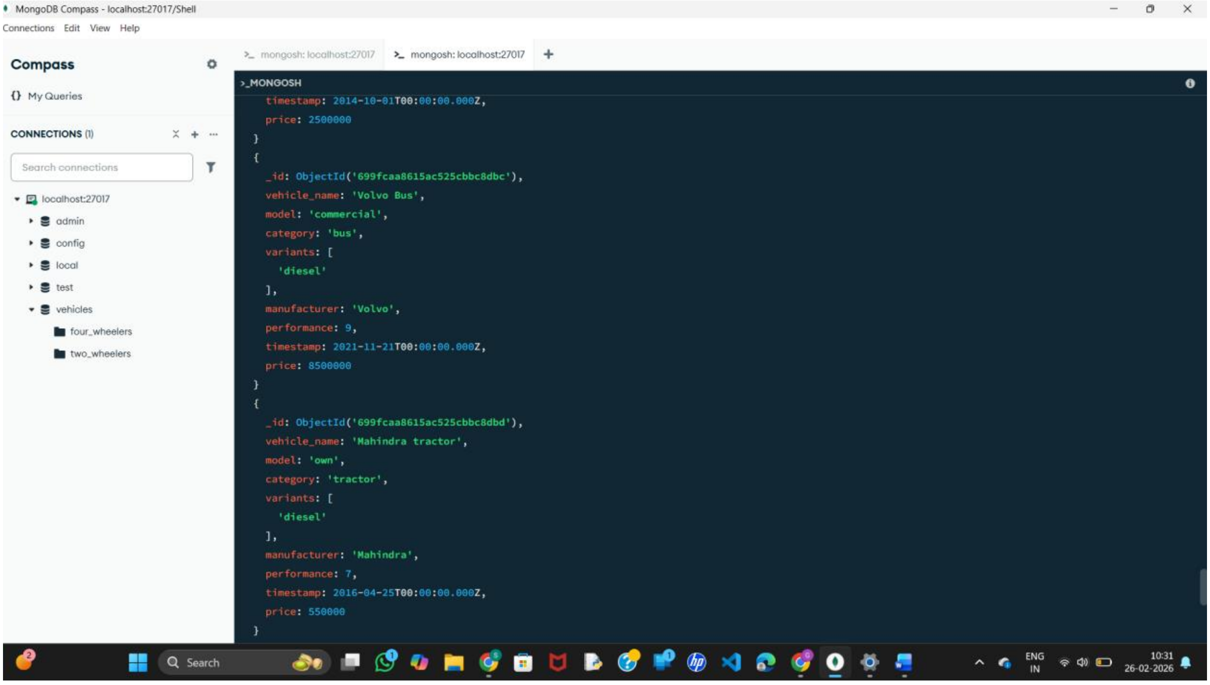
Code:



```
> db.four_wheelers.find({ variants: "diesel" })
< {
    _id: ObjectId('699fcaa8615ac525cbbc8db9'),
    vehicle_name: 'Maruti Swift',
    model: 'own',
    category: 'car',
    variants: [
      'petrol',
      'diesel'
    ],
    manufacturer: 'Maruti',
    performance: 9,
    timestamp: 2017-02-14T00:00:00.000Z,
    price: 780000
  }
  {
    _id: ObjectId('699fcaa8615ac525cbbc8dba'),
    vehicle_name: 'Tata Ace',
    model: 'commercial',
    category: 'mini truck',
    variants: [
      'diesel'
    ],
    manufacturer: 'Tata',
    performance: 9,
    timestamp: 2018-06-10T00:00:00.000Z,
    price: 690000
  }
  {
```



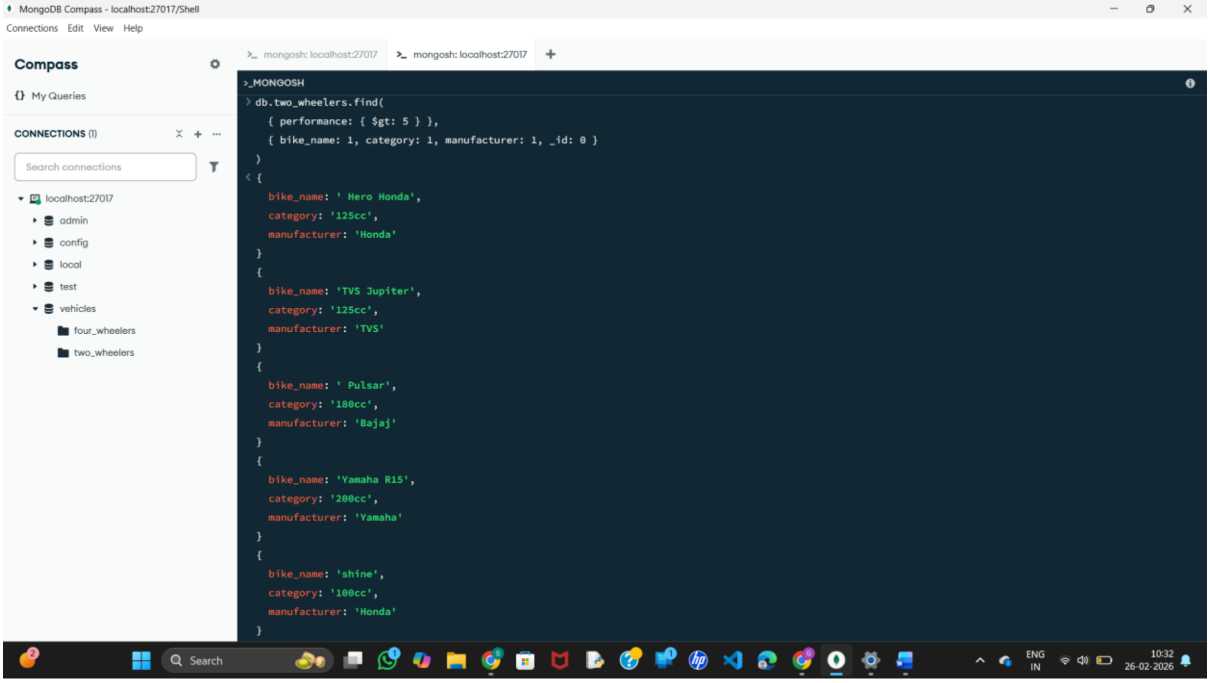
```
  {
    _id: ObjectId('699fcaa8615ac525cbbc8dbb'),
    vehicle_name: 'Ashok Leyland',
    model: 'commercial',
    category: 'heavy truck',
    variants: [
      'diesel'
    ],
    manufacturer: 'Ashok Leyland',
    performance: 8,
    timestamp: 2014-10-01T00:00:00.000Z,
    price: 2500000
  }
  {
    _id: ObjectId('699fcaa8615ac525cbbc8dbc'),
    vehicle_name: 'Volvo Bus',
    model: 'commercial',
    category: 'bus',
    variants: [
      'diesel'
    ],
    manufacturer: 'Volvo',
    performance: 9,
    timestamp: 2021-11-21T00:00:00.000Z,
    price: 8500000
  }
  {
    _id: ObjectId('699fcaa8615ac525cbbc8dbd'),
    vehicle_name: 'Mahindra tractor',
```



```
    timestamp: 2014-10-01T00:00:00.000Z,
    price: 2500000
  }
  {
    _id: ObjectId('699fcaa8615ac525cbbc8dbc'),
    vehicle_name: 'Volvo Bus',
    model: 'commercial',
    category: 'bus',
    variants: [
      'diesel'
    ],
    manufacturer: 'Volvo',
    performance: 9,
    timestamp: 2021-11-21T00:00:00.000Z,
    price: 8500000
  }
  {
    _id: ObjectId('699fcaa8615ac525cbbc8dbd'),
    vehicle_name: 'Mahindra tractor',
    model: 'own',
    category: 'tractor',
    variants: [
      'diesel'
    ],
    manufacturer: 'Mahindra',
    performance: 7,
    timestamp: 2016-04-25T00:00:00.000Z,
    price: 550000
  }
```

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

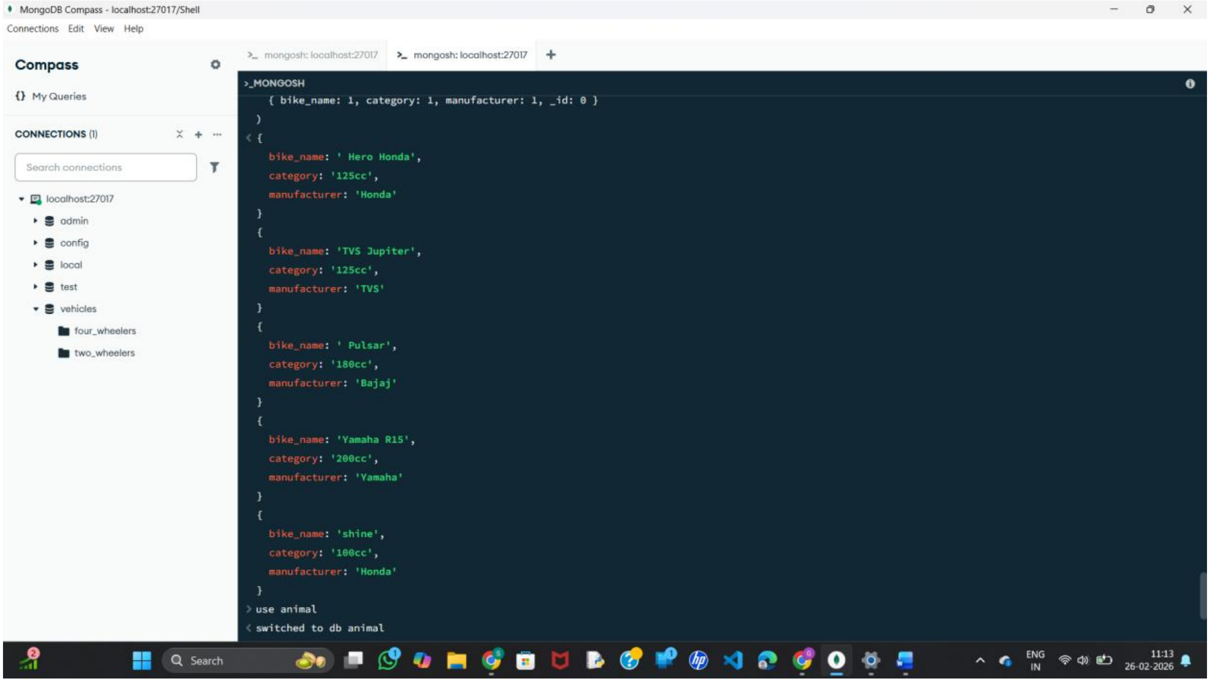
Code:



```
> db.two_wheelers.find(
    { performance: { $gt: 5 } },
    { bike_name: 1, category: 1, manufacturer: 1, _id: 0 }
  )
< {
    bike_name: ' Hero Honda',
    category: '125cc',
    manufacturer: 'Honda'
  }
  {
    bike_name: 'TVS Jupiter',
    category: '125cc',
    manufacturer: 'TVS'
  }
  {
    bike_name: ' Pulsar',
    category: '180cc',
    manufacturer: 'Bajaj'
  }
  {
    bike_name: 'Yamaha R15',
    category: '200cc',
    manufacturer: 'Yamaha'
  }
  {
    bike_name: 'shine',
    category: '100cc',
    manufacturer: 'Honda'
  }
```

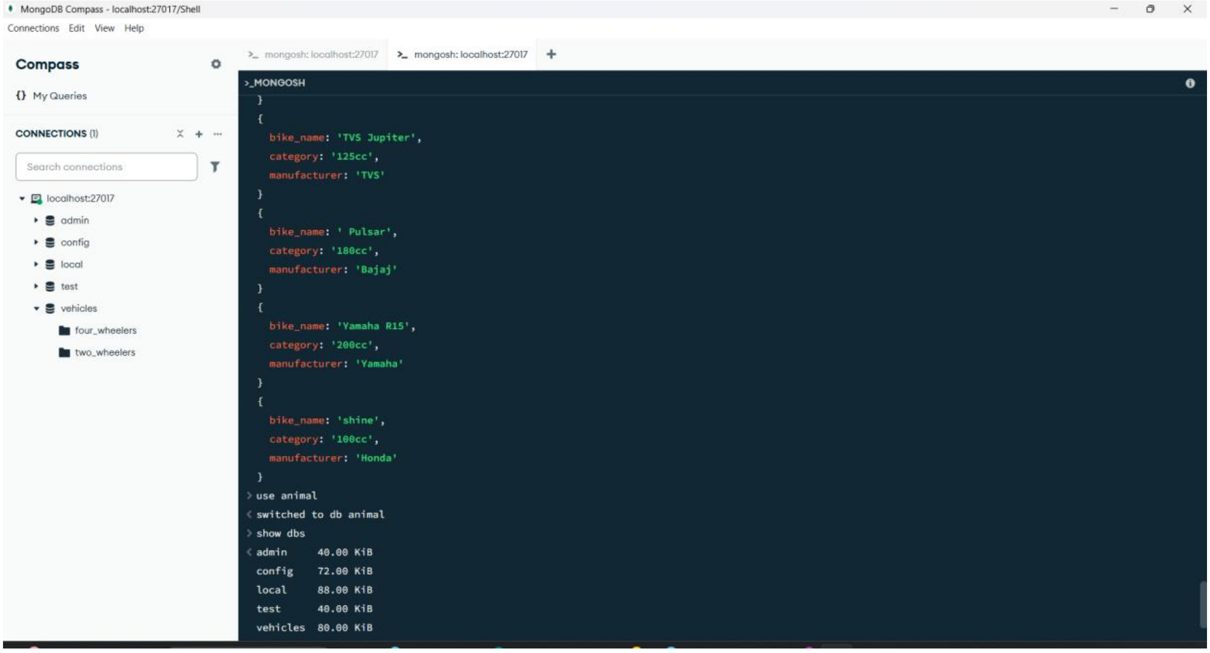
2. Use MongoDB to implement the following DB operations for a Zoo

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



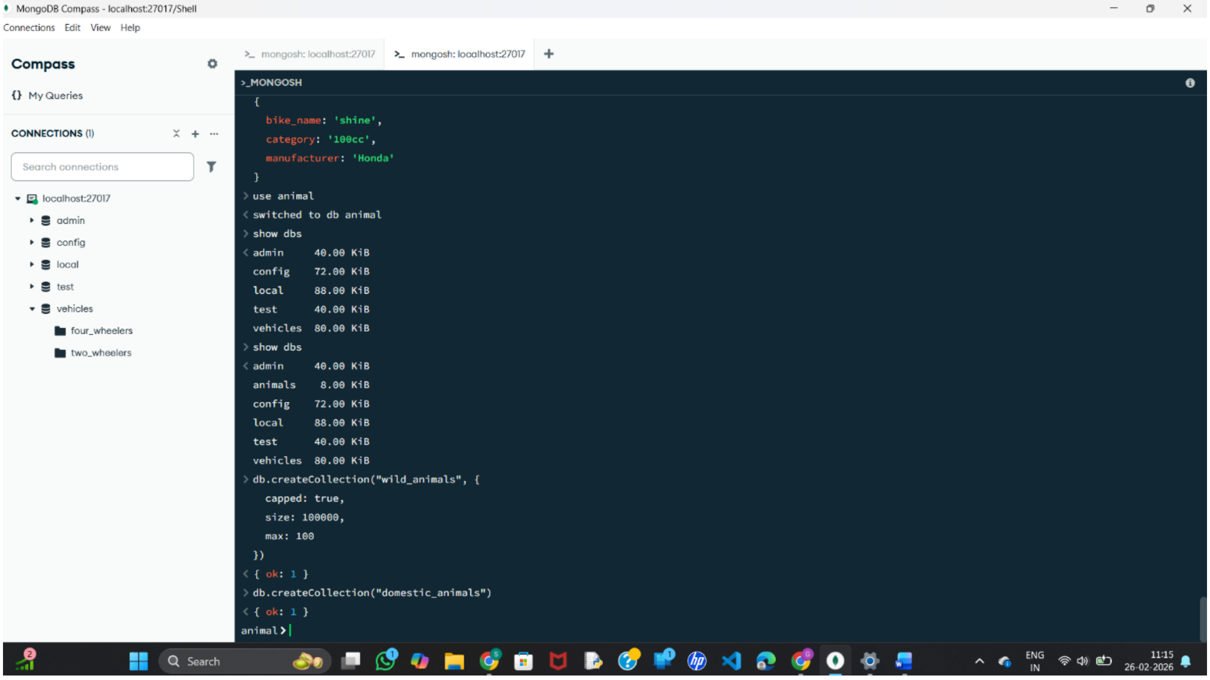
```
    { bike_name: 1, category: 1, manufacturer: 1, _id: 0 }
  )
< {
    bike_name: ' Hero Honda',
    category: '125cc',
    manufacturer: 'Honda'
  }
  {
    bike_name: 'TVS Jupiter',
    category: '125cc',
    manufacturer: 'TVS'
  }
  {
    bike_name: ' Pulsar',
    category: '180cc',
    manufacturer: 'Bajaj'
  }
  {
    bike_name: 'Yamaha R15',
    category: '200cc',
    manufacturer: 'Yamaha'
  }
  {
    bike_name: 'shine',
    category: '100cc',
    manufacturer: 'Honda'
  }
> use animal
< switched to db animal
```

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



```
  }
  {
    bike_name: 'TVS Jupiter',
    category: '125cc',
    manufacturer: 'TVS'
  }
  {
    bike_name: ' Pulsar',
    category: '180cc',
    manufacturer: 'Bajaj'
  }
  {
    bike_name: 'Yamaha R15',
    category: '200cc',
    manufacturer: 'Yamaha'
  }
  {
    bike_name: 'shine',
    category: '100cc',
    manufacturer: 'Honda'
  }
> use animal
< switched to db animal
> show dbs
< admin     40.00 KiB
  config    72.00 KiB
  local     88.00 KiB
  test      40.00 KiB
  vehicles  80.00 KiB
```

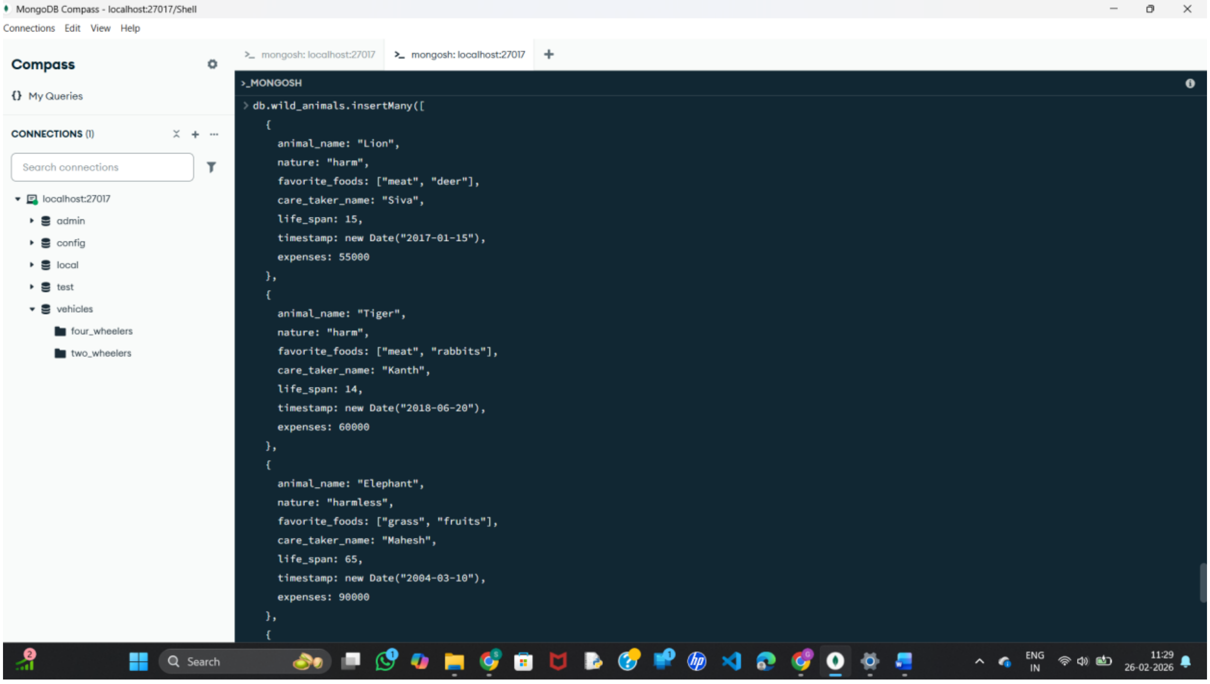
3. Create a collection called ‘wild_animals’.(use capping) and Create a collection called ‘domestic_animals’.



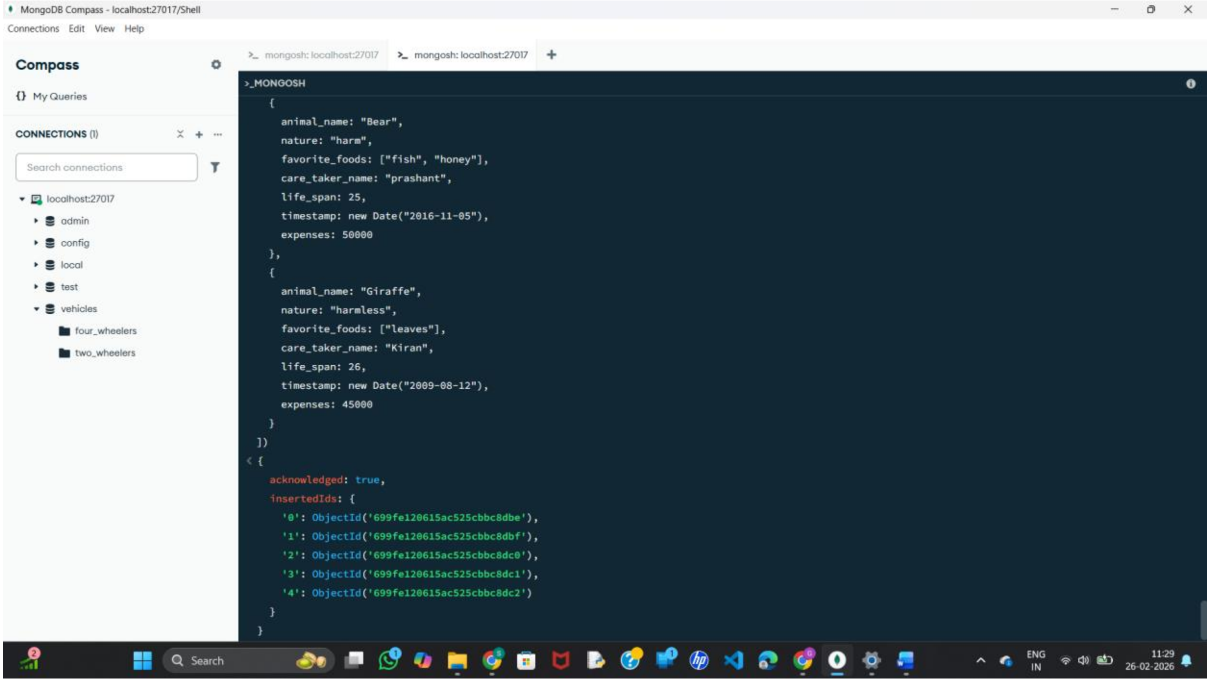
```
  {
    bike_name: 'shine',
    category: '100cc',
    manufacturer: 'Honda'
  }
> use animal
< switched to db animal
> show dbs
< admin     40.00 KiB
  config    72.00 KiB
  local     88.00 KiB
  test      40.00 KiB
  vehicles  80.00 KiB
> show dbs
< admin     40.00 KiB
  animals    8.00 KiB
  config    72.00 KiB
  local     88.00 KiB
  test      40.00 KiB
  vehicles  80.00 KiB
> db.createCollection("wild_animals", {
    capped: true,
    size: 100000,
    max: 100
  })
< { 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.

Code:

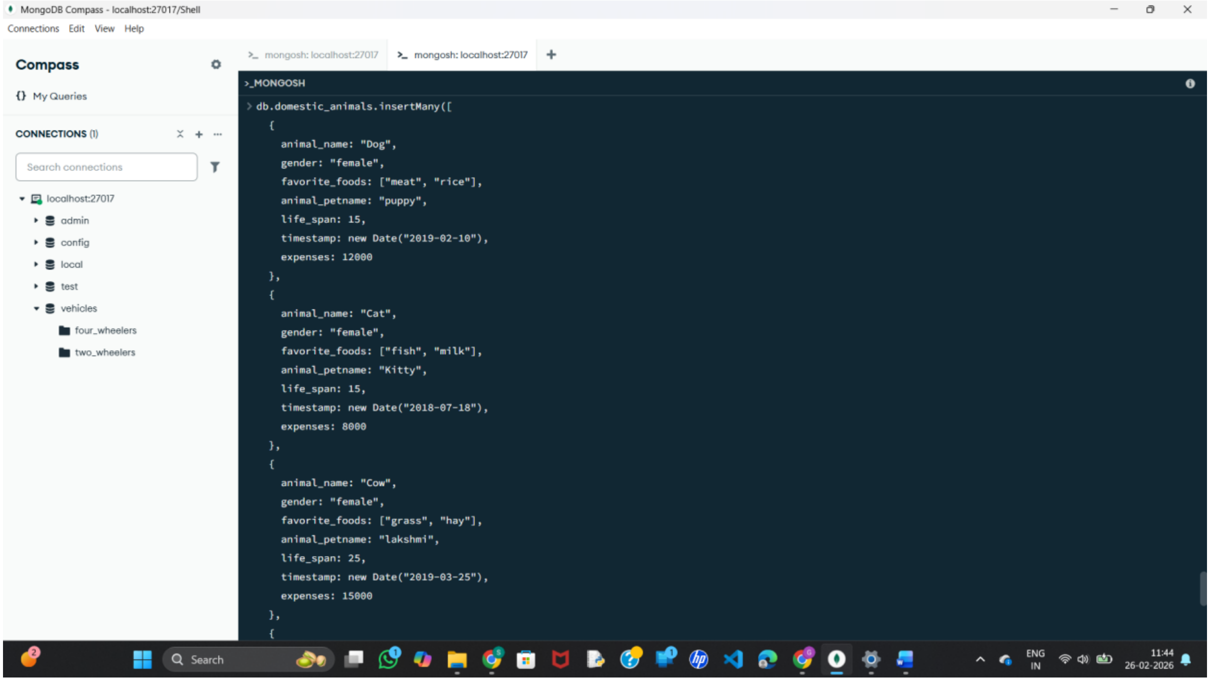


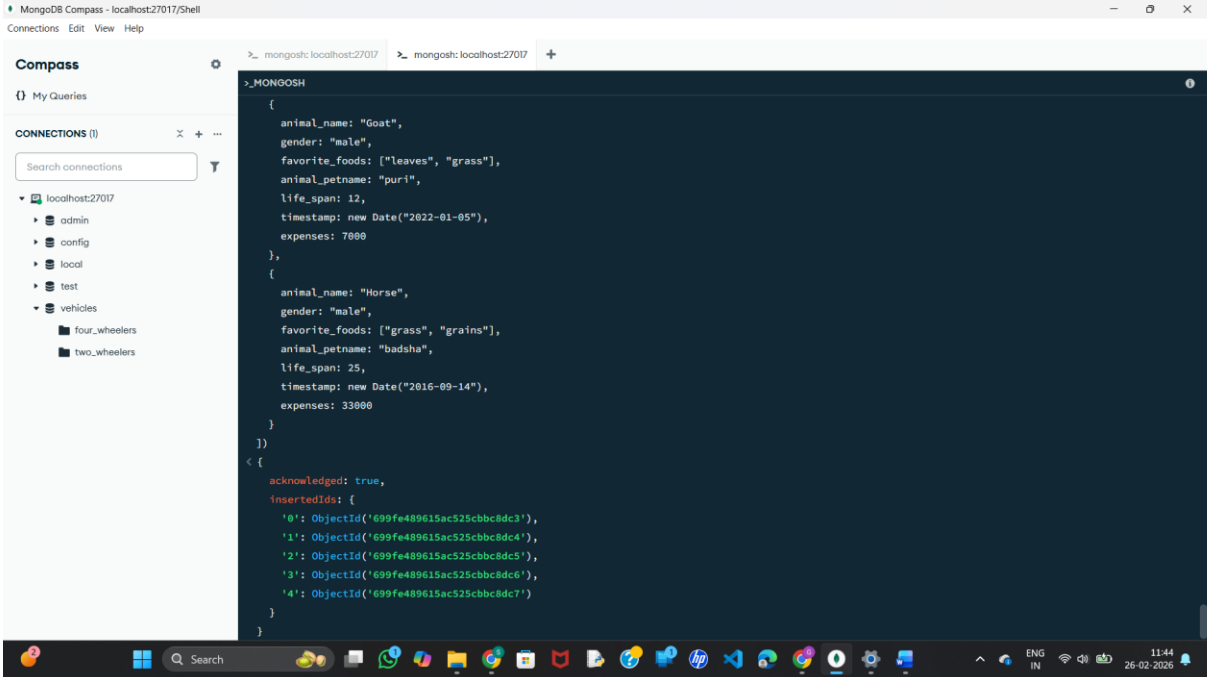
```
> db.wild_animals.insertMany([
    {
      animal_name: "Lion",
      nature: "harm",
      favorite_foods: ["meat", "deer"],
      care_taker_name: "Siva",
      life_span: 15,
      timestamp: new Date("2017-01-15"),
      expenses: 55000
    },
    {
      animal_name: "Tiger",
      nature: "harm",
      favorite_foods: ["meat", "rabbits"],
      care_taker_name: "Kanth",
      life_span: 14,
      timestamp: new Date("2018-06-20"),
      expenses: 60000
    },
    {
      animal_name: "Elephant",
      nature: "harmless",
      favorite_foods: ["grass", "fruits"],
      care_taker_name: "Mahesh",
      life_span: 65,
      timestamp: new Date("2004-03-10"),
      expenses: 90000
    },
    {
```


```
  {
    animal_name: "Bear",
    nature: "harm",
    favorite_foods: ["fish", "honey"],
    care_taker_name: "prashant",
    life_span: 25,
    timestamp: new Date("2016-11-05"),
    expenses: 50000
  },
  {
    animal_name: "Giraffe",
    nature: "harmless",
    favorite_foods: ["leaves"],
    care_taker_name: "Kiran",
    life_span: 26,
    timestamp: new Date("2009-08-12"),
    expenses: 45000
  }
])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('699fe120615ac525cbbc8dbe'),
    '1': ObjectId('699fe120615ac525cbbc8dbf'),
    '2': ObjectId('699fe120615ac525cbbc8dc0'),
    '3': ObjectId('699fe120615ac525cbbc8dc1'),
    '4': ObjectId('699fe120615ac525cbbc8dc2')
  }
}
```

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.

Code:

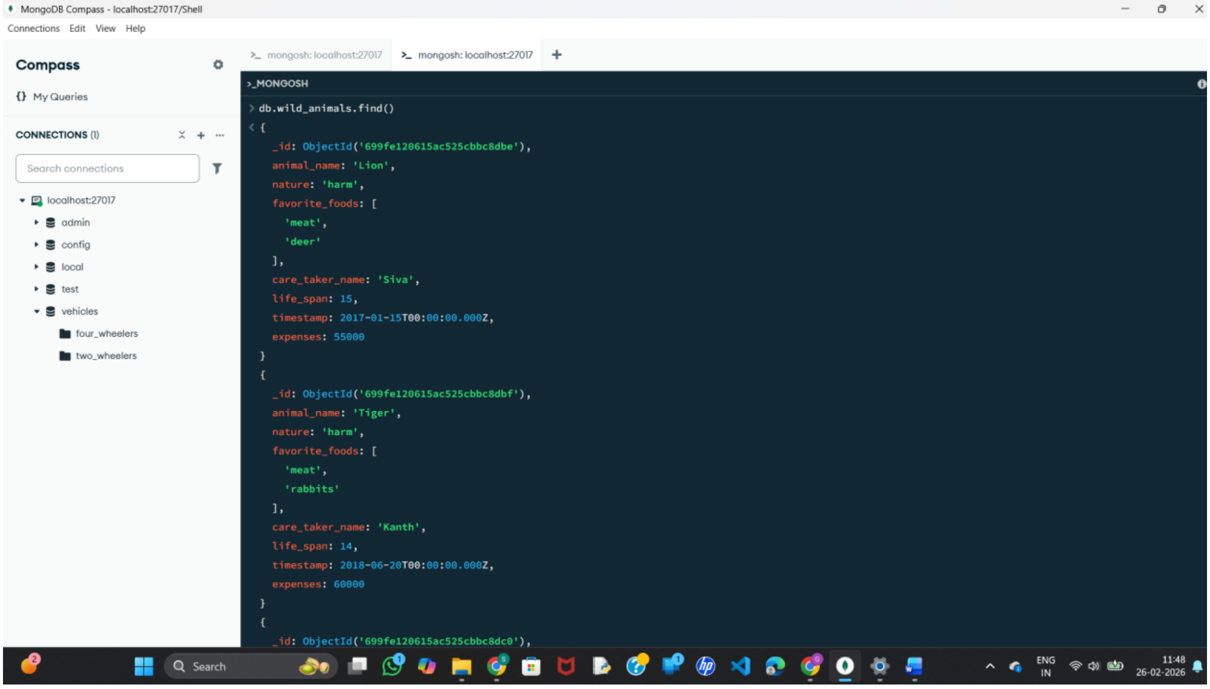

```
> db.domestic_animals.insertMany([
  {
    animal_name: "Dog",
    gender: "female",
    favorite_foods: ["meat", "rice"],
    animal_petname: "puppy",
    life_span: 15,
    timestamp: new Date("2019-02-10"),
    expenses: 12000
  },
  {
    animal_name: "Cat",
    gender: "female",
    favorite_foods: ["fish", "milk"],
    animal_petname: "Kitty",
    life_span: 15,
    timestamp: new Date("2018-07-18"),
    expenses: 8000
  },
  {
    animal_name: "Cow",
    gender: "female",
    favorite_foods: ["grass", "hay"],
    animal_petname: "lakshmi",
    life_span: 25,
    timestamp: new Date("2019-03-25"),
    expenses: 15000
  },
  {
```



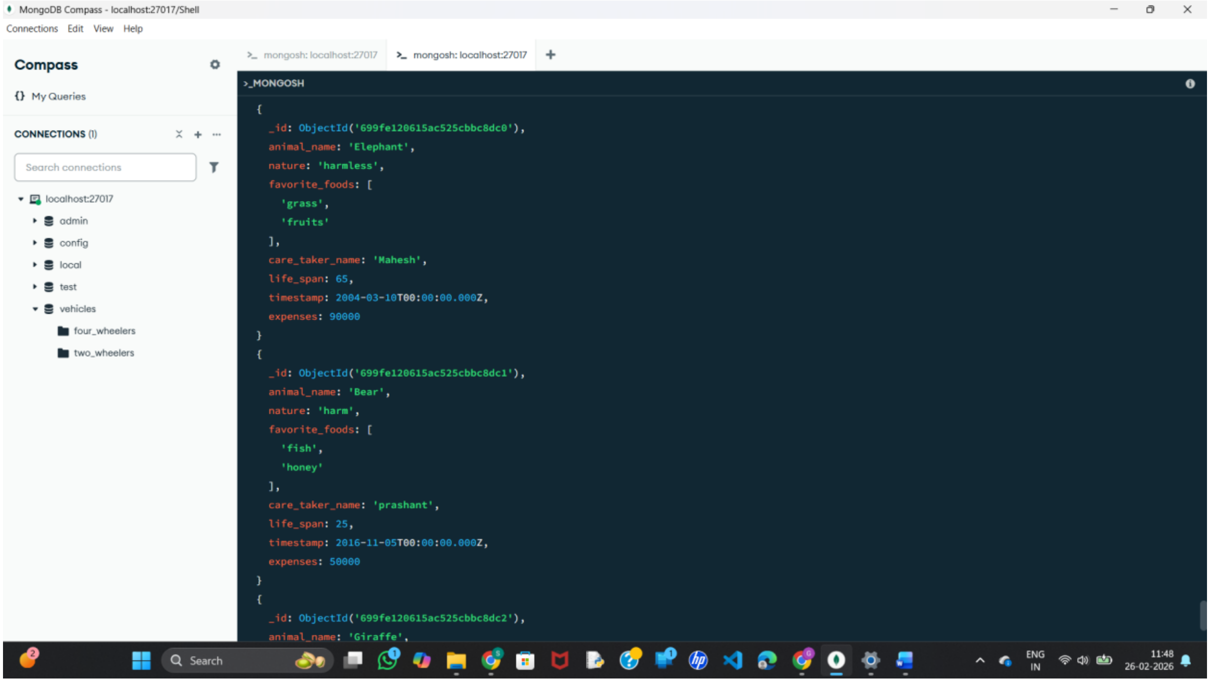
```
    {
      animal_name: "Goat",
      gender: "male",
      favorite_foods: ["leaves", "grass"],
      animal_petname: "puri",
      life_span: 12,
      timestamp: new Date("2022-01-05"),
      expenses: 7000
    },
    {
      animal_name: "Horse",
      gender: "male",
      favorite_foods: ["grass", "grains"],
      animal_petname: "badsha",
      life_span: 25,
      timestamp: new Date("2016-09-14"),
      expenses: 33000
    }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('699fe489615ac525cbbc8dc3'),
      '1': ObjectId('699fe489615ac525cbbc8dc4'),
      '2': ObjectId('699fe489615ac525cbbc8dc5'),
      '3': ObjectId('699fe489615ac525cbbc8dc6'),
      '4': ObjectId('699fe489615ac525cbbc8dc7')
    }
  }
```

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

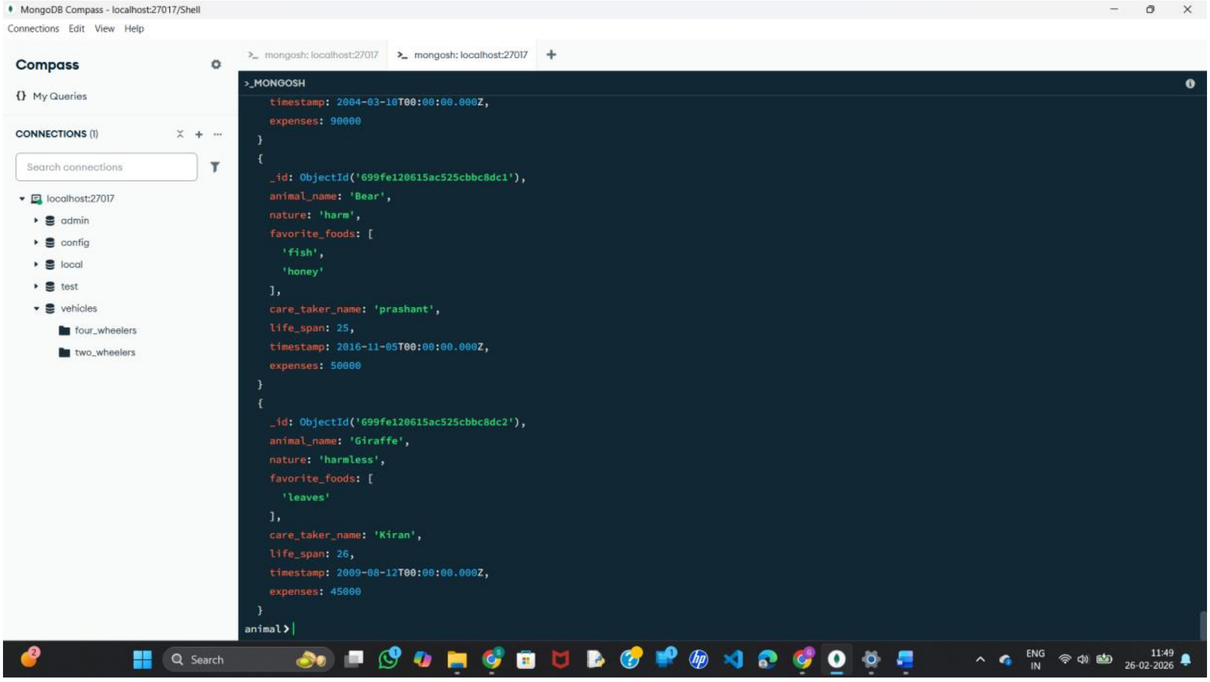
Code:



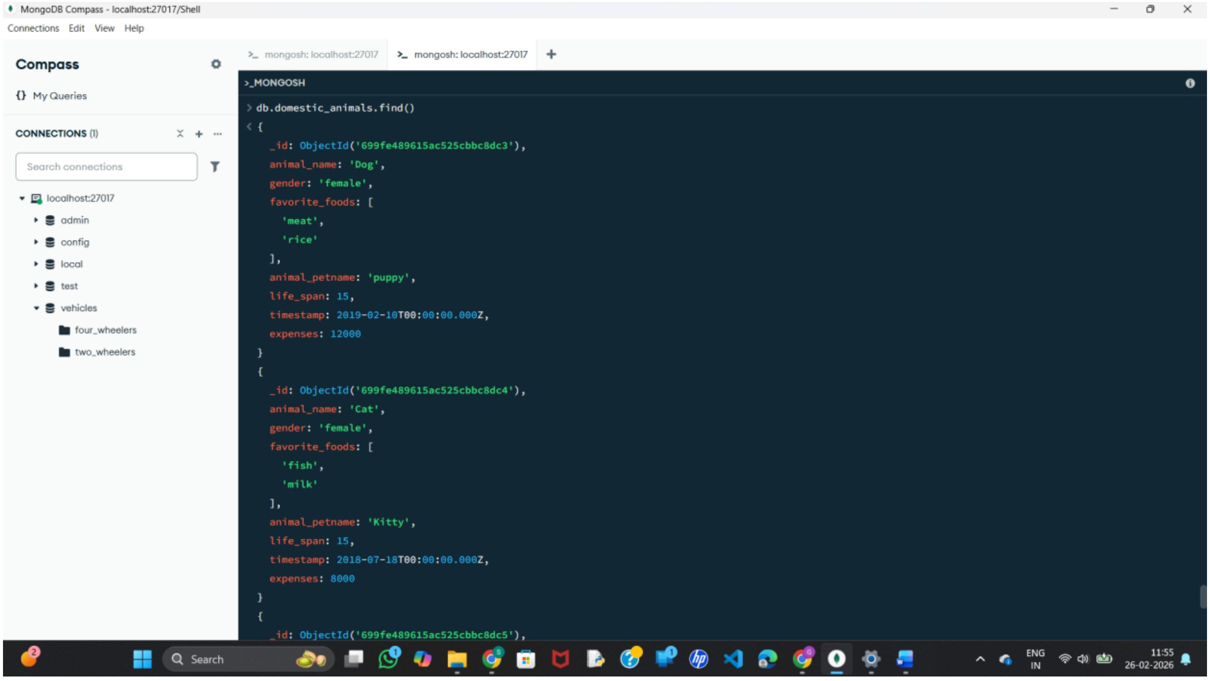
```
> db.wild_animals.find()
< {
    _id: ObjectId('699fe120615ac525cbbc8dbe'),
    animal_name: 'Lion',
    nature: 'harm',
    favorite_foods: [
      'meat',
      'deer'
    ],
    care_taker_name: 'Siva',
    life_span: 15,
    timestamp: 2017-01-15T00:00:00.000Z,
    expenses: 55000
  }
  {
    _id: ObjectId('699fe120615ac525cbbc8dbf'),
    animal_name: 'Tiger',
    nature: 'harm',
    favorite_foods: [
      'meat',
      'rabbits'
    ],
    care_taker_name: 'Kanth',
    life_span: 14,
    timestamp: 2018-06-20T00:00:00.000Z,
    expenses: 60000
  }
  {
    _id: ObjectId('699fe120615ac525cbbc8dc0'),
```



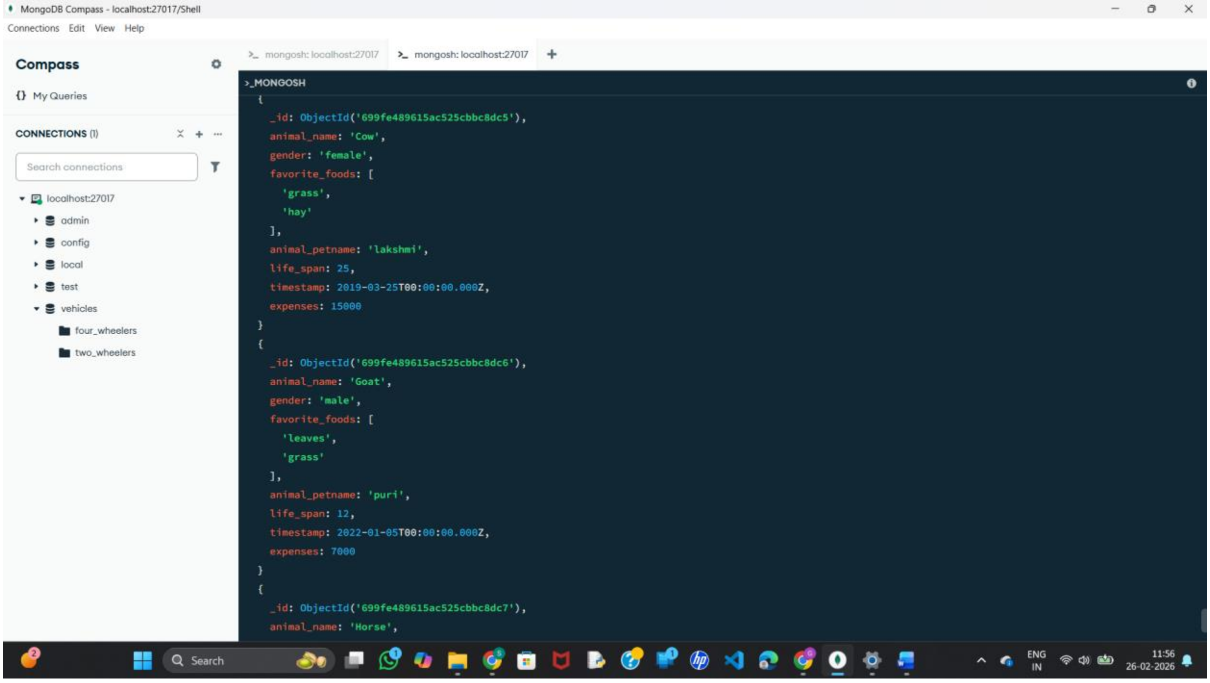
```
  {
    _id: ObjectId('699fe120615ac525cbbc8dc0'),
    animal_name: 'Elephant',
    nature: 'harmless',
    favorite_foods: [
      'grass',
      'fruits'
    ],
    care_taker_name: 'Mahesh',
    life_span: 65,
    timestamp: 2004-03-10T00:00:00.000Z,
    expenses: 90000
  }
  {
    _id: ObjectId('699fe120615ac525cbbc8dc1'),
    animal_name: 'Bear',
    nature: 'harm',
    favorite_foods: [
      'fish',
      'honey'
    ],
    care_taker_name: 'prashant',
    life_span: 25,
    timestamp: 2016-11-05T00:00:00.000Z,
    expenses: 50000
  }
  {
    _id: ObjectId('699fe120615ac525cbbc8dc2'),
    animal_name: 'Giraffe',
```



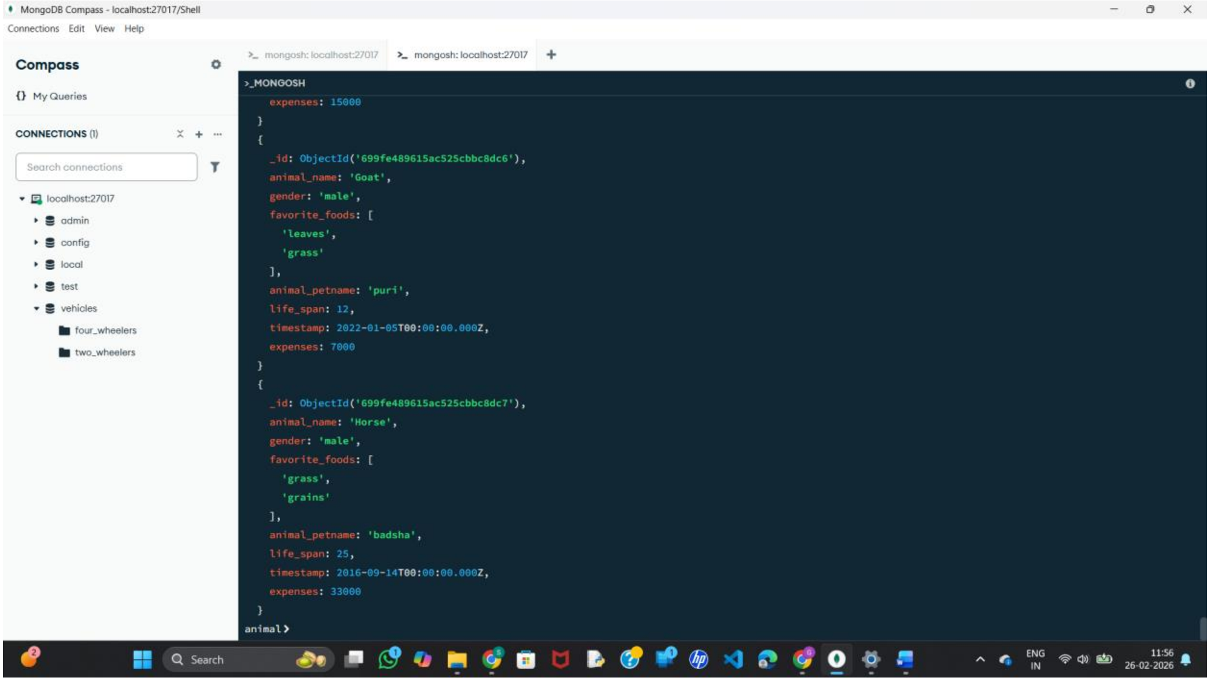
```
    timestamp: 2004-03-10T00:00:00.000Z,
    expenses: 90000
  }
  {
    _id: ObjectId('699fe120615ac525cbbc8dc1'),
    animal_name: 'Bear',
    nature: 'harm',
    favorite_foods: [
      'fish',
      'honey'
    ],
    care_taker_name: 'prashant',
    life_span: 25,
    timestamp: 2016-11-05T00:00:00.000Z,
    expenses: 50000
  }
  {
    _id: ObjectId('699fe120615ac525cbbc8dc2'),
    animal_name: 'Giraffe',
    nature: 'harmless',
    favorite_foods: [
      'leaves'
    ],
    care_taker_name: 'Kiran',
    life_span: 26,
    timestamp: 2009-08-12T00:00:00.000Z,
    expenses: 45000
  }
animal>
```



```
> db.domestic_animals.find()
< {
    _id: ObjectId('699fe489615ac525cbbc8dc3'),
    animal_name: 'Dog',
    gender: 'female',
    favorite_foods: [
      'meat',
      'rice'
    ],
    animal_petname: 'puppy',
    life_span: 15,
    timestamp: 2019-02-10T00:00:00.000Z,
    expenses: 12000
  }
  {
    _id: ObjectId('699fe489615ac525cbbc8dc4'),
    animal_name: 'Cat',
    gender: 'female',
    favorite_foods: [
      'fish',
      'milk'
    ],
    animal_petname: 'Kitty',
    life_span: 15,
    timestamp: 2018-07-18T00:00:00.000Z,
    expenses: 8000
  }
  {
    _id: ObjectId('699fe489615ac525cbbc8dc5'),
```

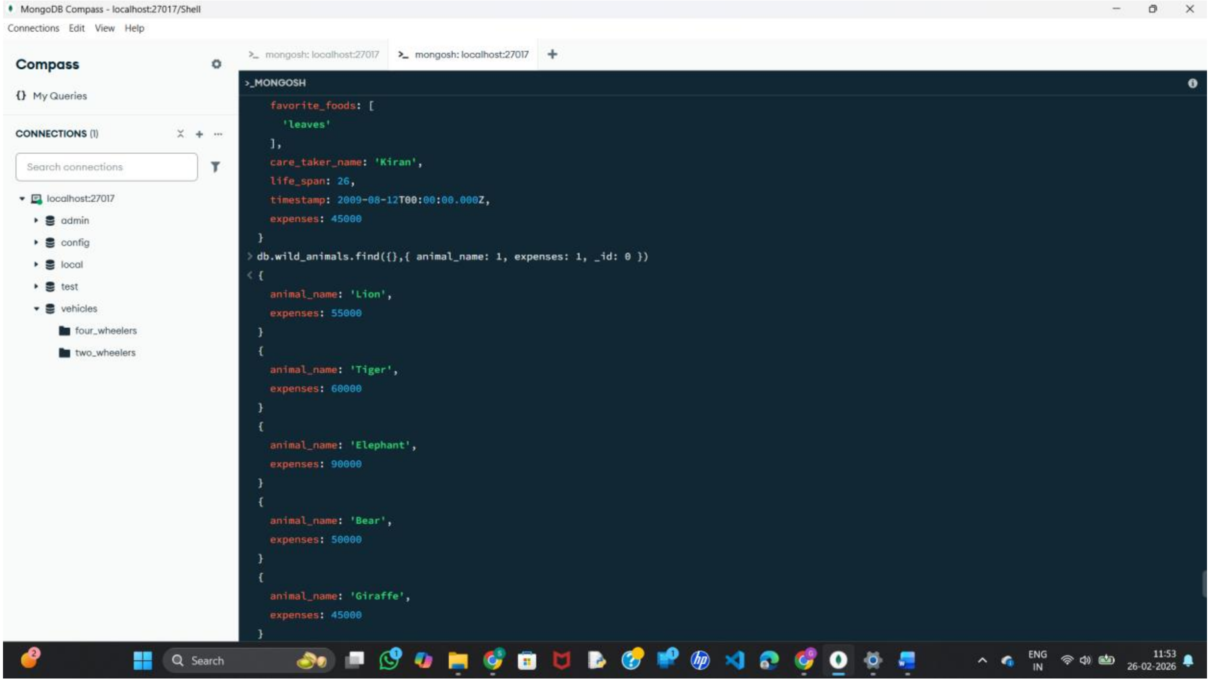


```
  {
    _id: ObjectId('699fe489615ac525cbbc8dc5'),
    animal_name: 'Cow',
    gender: 'female',
    favorite_foods: [
      'grass',
      'hay'
    ],
    animal_petname: 'lakshmi',
    life_span: 25,
    timestamp: 2019-03-25T00:00:00.000Z,
    expenses: 15000
  }
  {
    _id: ObjectId('699fe489615ac525cbbc8dc6'),
    animal_name: 'Goat',
    gender: 'male',
    favorite_foods: [
      'leaves',
      'grass'
    ],
    animal_petname: 'puri',
    life_span: 12,
    timestamp: 2022-01-05T00:00:00.000Z,
    expenses: 7000
  }
  {
    _id: ObjectId('699fe489615ac525cbbc8dc7'),
    animal_name: 'Horse',
```

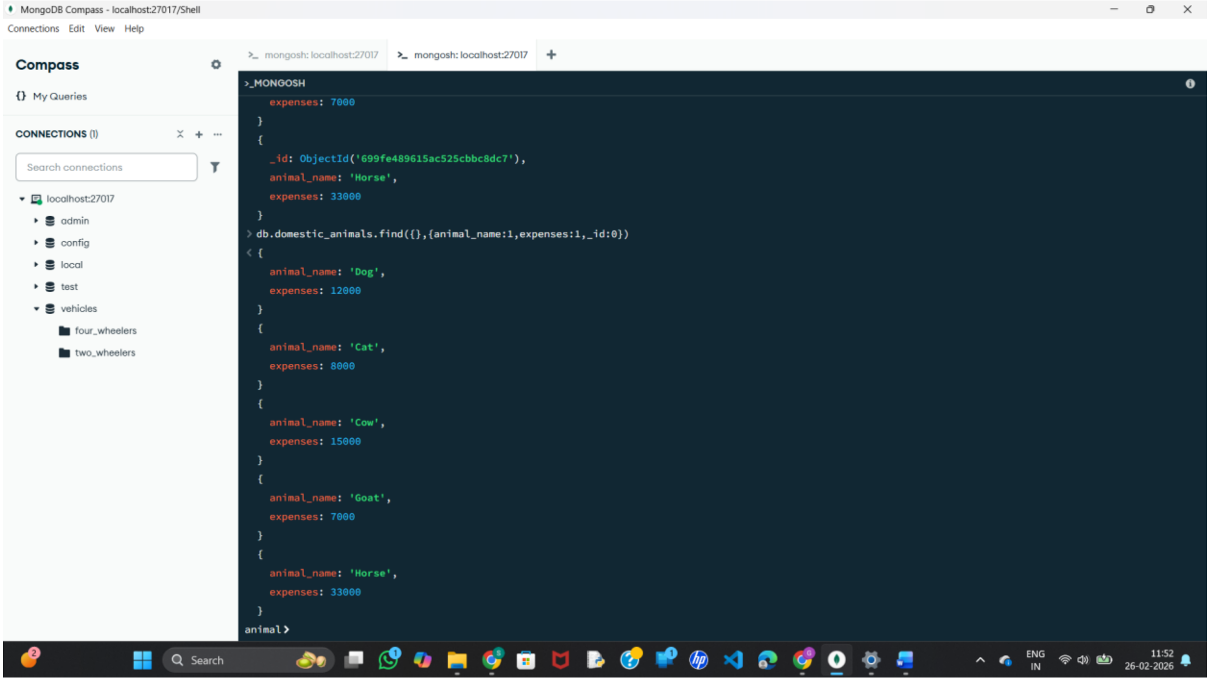


```
    expenses: 15000
  }
  {
    _id: ObjectId('699fe489615ac525cbbc8dc6'),
    animal_name: 'Goat',
    gender: 'male',
    favorite_foods: [
      'leaves',
      'grass'
    ],
    animal_petname: 'puri',
    life_span: 12,
    timestamp: 2022-01-05T00:00:00.000Z,
    expenses: 7000
  }
  {
    _id: ObjectId('699fe489615ac525cbbc8dc7'),
    animal_name: 'Horse',
    gender: 'male',
    favorite_foods: [
      'grass',
      'grains'
    ],
    animal_petname: 'badsha',
    life_span: 25,
    timestamp: 2016-09-14T00:00:00.000Z,
    expenses: 33000
  }
animal>
```

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

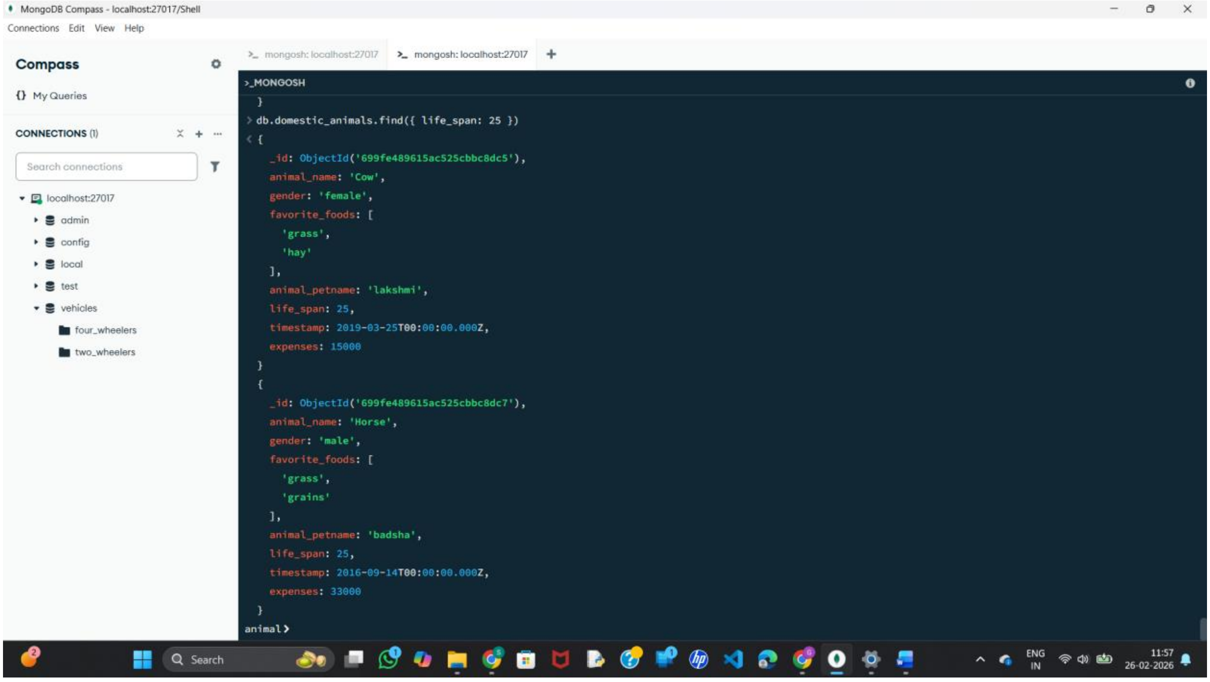


```
    favorite_foods: [
      'leaves'
    ],
    care_taker_name: 'Kiran',
    life_span: 26,
    timestamp: 2009-08-12T00:00:00.000Z,
    expenses: 45000
  }
> db.wild_animals.find({},{ animal_name: 1, expenses: 1, _id: 0 })
< {
    animal_name: 'Lion',
    expenses: 55000
  }
  {
    animal_name: 'Tiger',
    expenses: 60000
  }
  {
    animal_name: 'Elephant',
    expenses: 90000
  }
  {
    animal_name: 'Bear',
    expenses: 50000
  }
  {
    animal_name: 'Giraffe',
    expenses: 45000
  }
```



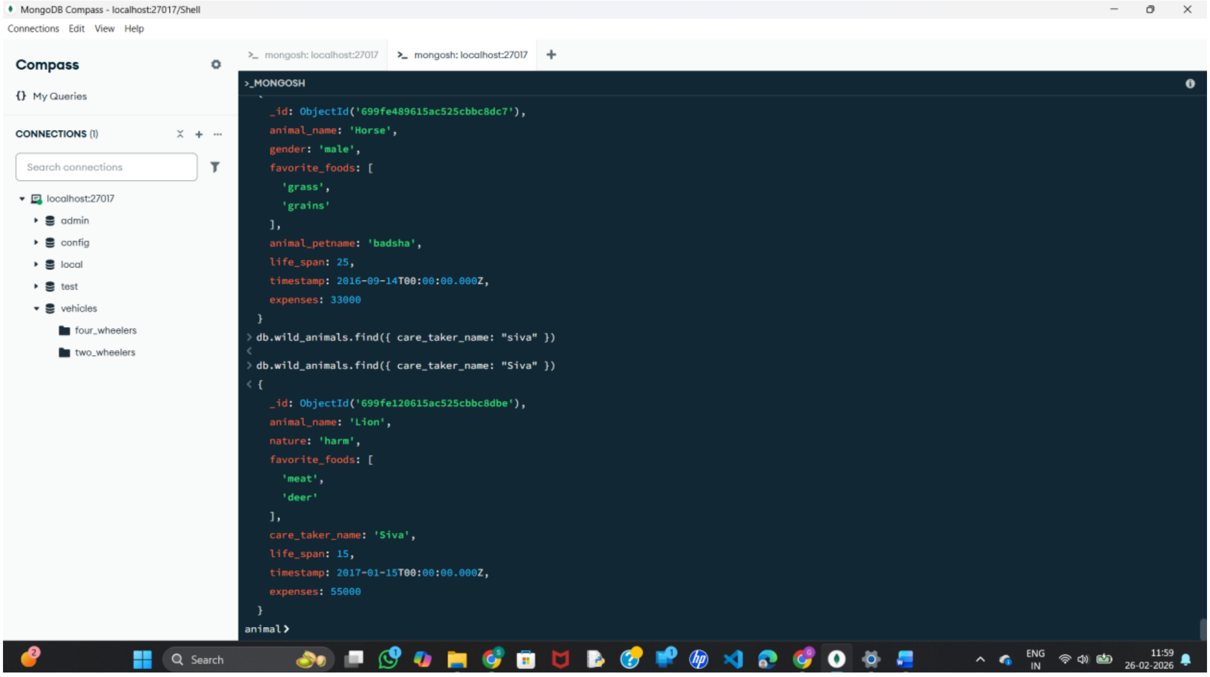
```
    expenses: 7000
  }
  {
    _id: ObjectId('699fe489615ac525cbbc8dc7'),
    animal_name: 'Horse',
    expenses: 33000
  }
> db.domestic_animals.find({},{animal_name:1,expenses:1,_id:0})
< {
    animal_name: 'Dog',
    expenses: 12000
  }
  {
    animal_name: 'Cat',
    expenses: 8000
  }
  {
    animal_name: 'Cow',
    expenses: 15000
  }
  {
    animal_name: 'Goat',
    expenses: 7000
  }
  {
    animal_name: 'Horse',
    expenses: 33000
  }
animal>
```

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



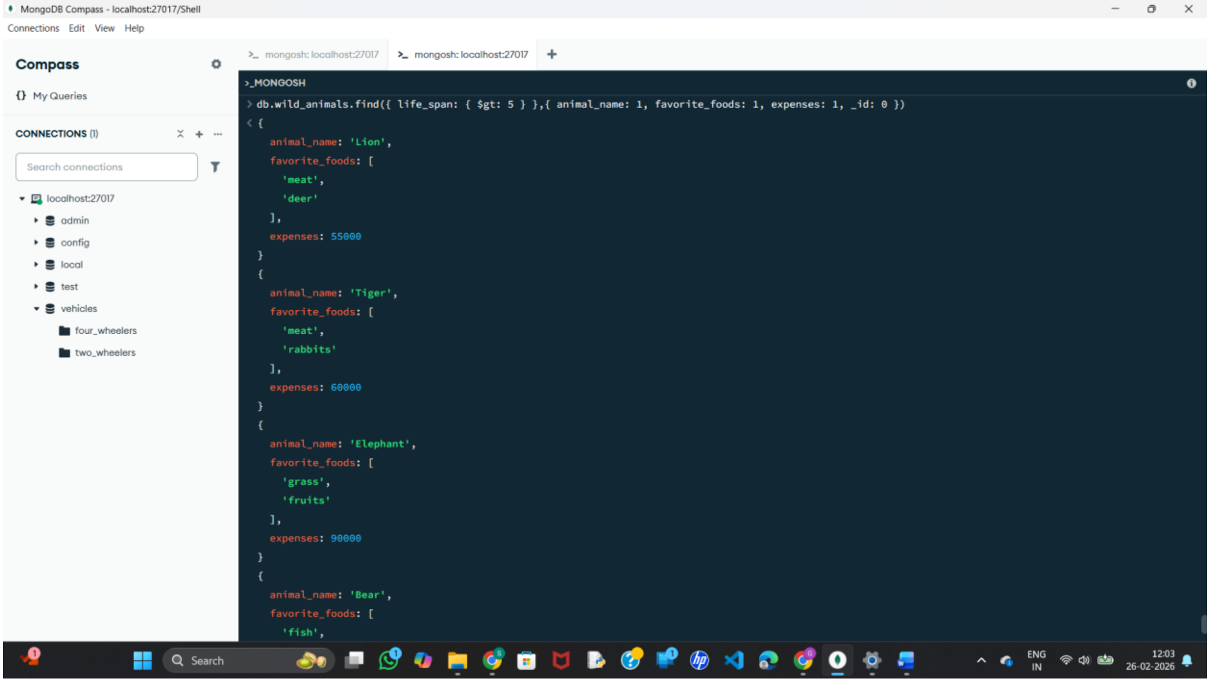
```
  }
> db.domestic_animals.find({ life_span: 25 })
< {
    _id: ObjectId('699fe489615ac525cbbc8dc5'),
    animal_name: 'Cow',
    gender: 'female',
    favorite_foods: [
      'grass',
      'hay'
    ],
    animal_petname: 'lakshmi',
    life_span: 25,
    timestamp: 2019-03-25T00:00:00.000Z,
    expenses: 15000
  }
  {
    _id: ObjectId('699fe489615ac525cbbc8dc7'),
    animal_name: 'Horse',
    gender: 'male',
    favorite_foods: [
      'grass',
      'grains'
    ],
    animal_petname: 'badsha',
    life_span: 25,
    timestamp: 2016-09-14T00:00:00.000Z,
    expenses: 33000
  }
animal>
```

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

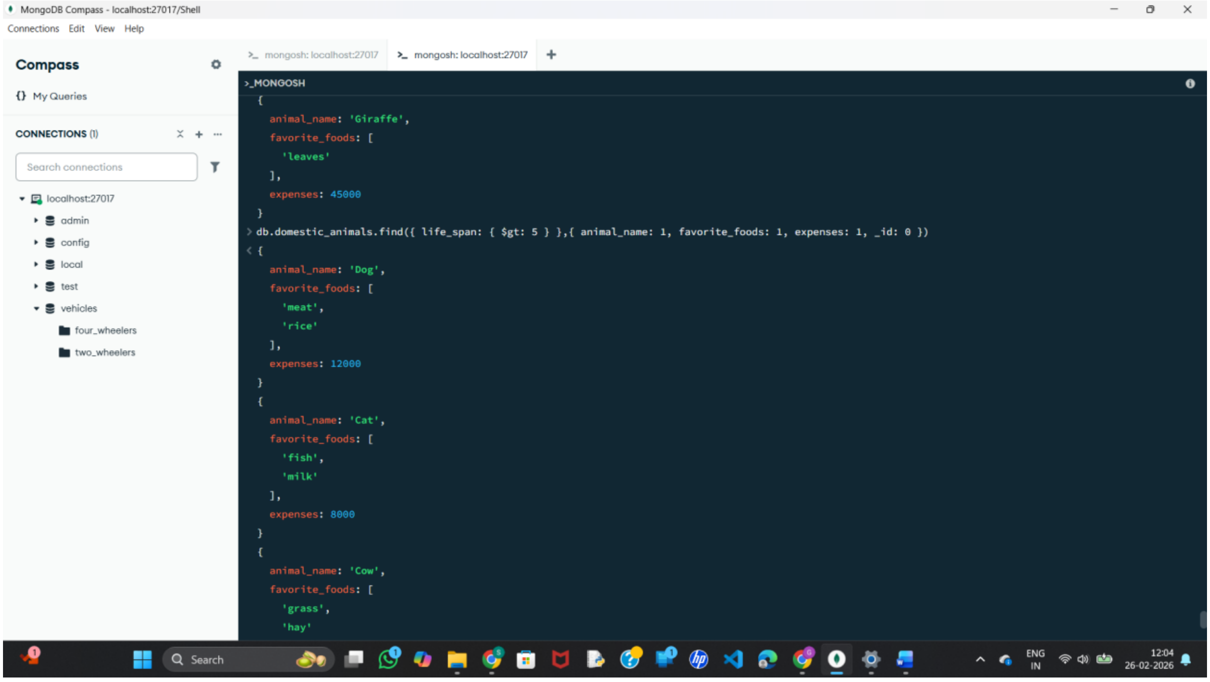


```
    _id: ObjectId('699fe489615ac525cbbc8dc7'),
    animal_name: 'Horse',
    gender: 'male',
    favorite_foods: [
      'grass',
      'grains'
    ],
    animal_petname: 'badsha',
    life_span: 25,
    timestamp: 2016-09-14T00:00:00.000Z,
    expenses: 33000
  }
> db.wild_animals.find({ care_taker_name: "siva" })
<
> db.wild_animals.find({ care_taker_name: "Siva" })
< {
    _id: ObjectId('699fe120615ac525cbbc8dbe'),
    animal_name: 'Lion',
    nature: 'harm',
    favorite_foods: [
      'meat',
      'deer'
    ],
    care_taker_name: 'Siva',
    life_span: 15,
    timestamp: 2017-01-15T00:00:00.000Z,
    expenses: 55000
  }
animal>
```

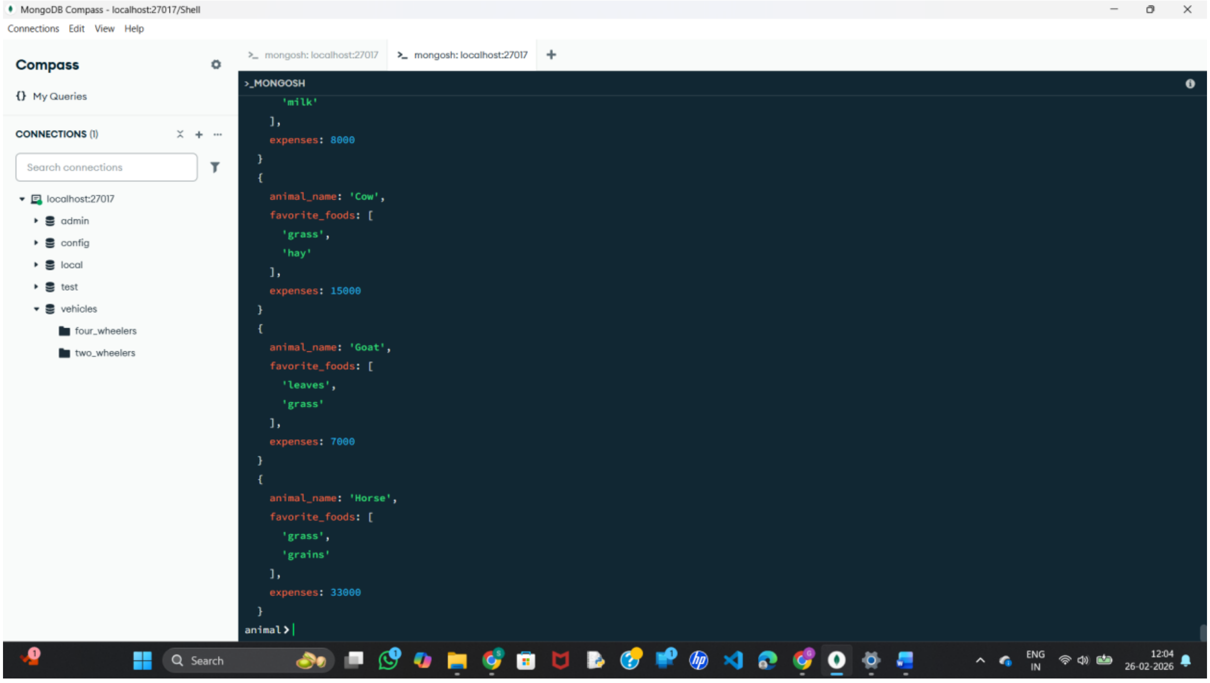
10. Write a MongoDB query to display animal name, favorite_foods and expenses details whose lifespan is more than 5 years.



```
> 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: 55000
  }
  {
    animal_name: 'Tiger',
    favorite_foods: [
      'meat',
      'rabbits'
    ],
    expenses: 60000
  }
  {
    animal_name: 'Elephant',
    favorite_foods: [
      'grass',
      'fruits'
    ],
    expenses: 90000
  }
  {
    animal_name: 'Bear',
    favorite_foods: [
      'fish',
```



```
  {
    animal_name: 'Giraffe',
    favorite_foods: [
      'leaves'
    ],
    expenses: 45000
  }
> 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: 12000
  }
  {
    animal_name: 'Cat',
    favorite_foods: [
      'fish',
      'milk'
    ],
    expenses: 8000
  }
  {
    animal_name: 'Cow',
    favorite_foods: [
      'grass',
      'hay'
```



```
      'milk'
    ],
    expenses: 8000
  }
  {
    animal_name: 'Cow',
    favorite_foods: [
      'grass',
      'hay'
    ],
    expenses: 15000
  }
  {
    animal_name: 'Goat',
    favorite_foods: [
      'leaves',
      'grass'
    ],
    expenses: 7000
  }
  {
    animal_name: 'Horse',
    favorite_foods: [
      'grass',
      'grains'
    ],
    expenses: 33000
  }
animal>
```