

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

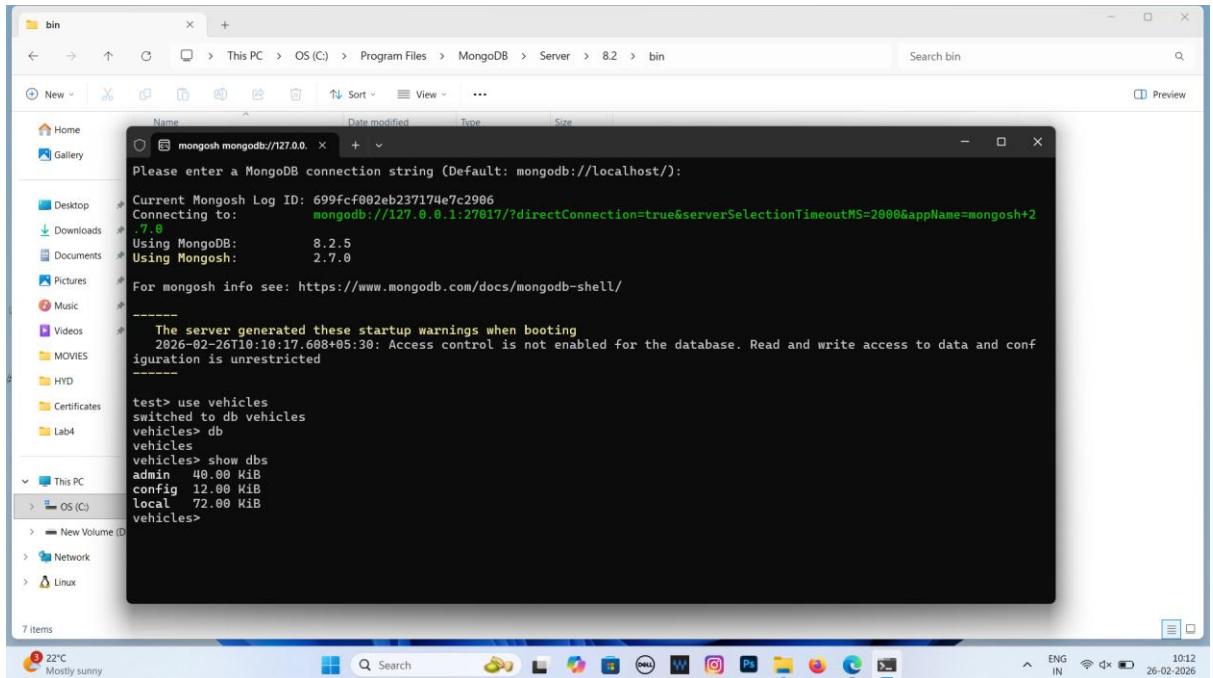
School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461

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”.

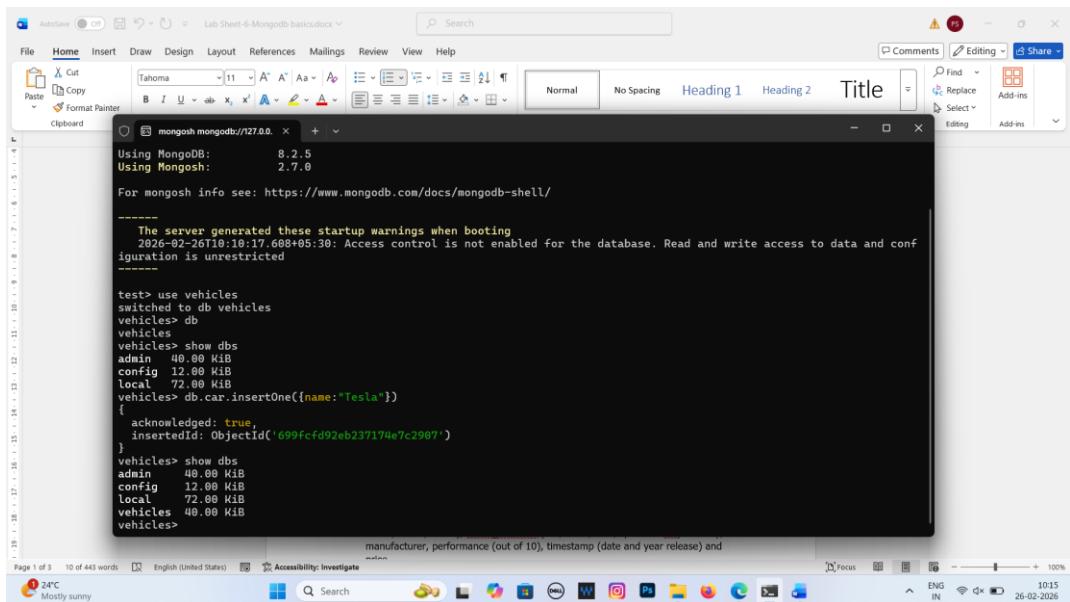


```

Please enter a MongoDB connection string (Default: mongodb://localhost/):
Current Mongosh Log ID: 699fcf002eb237174e7c2906
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
2026-02-26T10:18:17.608+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> db
vehicles
vehicles> show dbs
admin   40.00 KiB
config  12.00 KiB
local   72.00 KiB
vehicles>

```

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



```

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
2026-02-26T10:18:17.608+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> db
vehicles
vehicles> show dbs
admin   40.00 KiB
config  12.00 KiB
local   72.00 KiB
vehicles 40.00 KiB
vehicles>

```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

Date: 26/02/2026

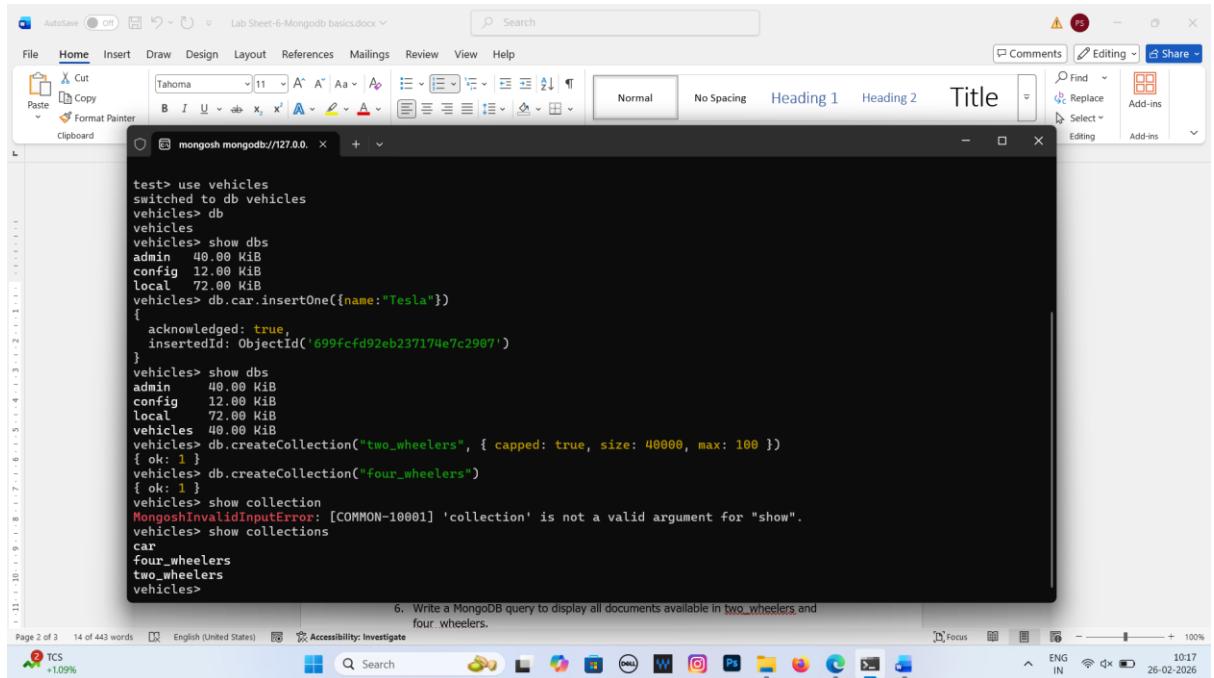
Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461

3. Create a collection called 'two_wHEELERS'. (use capping) and Create a collection called 'four_wHEELERS'.



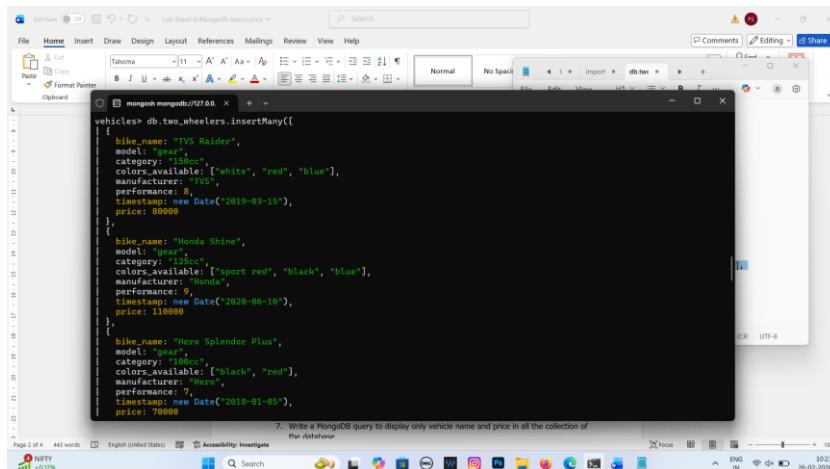
```

AutoSave Off
File Home Insert Draw Design Layout References Mailings Review View Help
Search
mongosh mongoDB://127.0.0.1:27017/test
test> use vehicles
switched to db vehicles
vehicles> db
vehicles
vehicles> show dbs
admin 40.00 KiB
config 12.00 KiB
local 72.00 KiB
vehicles> db.car.insertOne({name:"Tesla"})
{
  acknowledged: true,
  insertedId: ObjectId('699fcfd92eb237174e7c2907')
}
vehicles> show dbs
admin 40.00 KiB
config 12.00 KiB
local 72.00 KiB
vehicles 40.00 KiB
vehicles> db.createCollection("two_wHEELERS", { capped: true, size: 40000, max: 100 })
{
  ok: 1
}
vehicles> db.createCollection("four_wHEELERS")
{
  ok: 1
}
vehicles> show collection
MongoShellInvalidInputError: [COMMON-10001] 'collection' is not a valid argument for "show".
vehicles> show collections
car
four_wHEELERS
two_wHEELERS
vehicles>

```

6. Write a MongoDB query to display all documents available in two_wHEELERS and four_wHEELERS.

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.



```

mongosh mongoDB://127.0.0.1:27017/vehicles
vehicles> db.two_wHEELERS.insertMany([
  {
    bike_name: "TVS Raider",
    model: "gear",
    category: "150cc",
    colors_available: ["white", "red", "blue"],
    manufacturer: "TVS",
    performance: 8,
    timestamp: new Date("2019-03-15"),
    price: 80000
  },
  {
    bike_name: "Honda Shine",
    model: "gear",
    category: "125cc",
    colors_available: ["sport red", "black", "blue"],
    manufacturer: "Honda",
    performance: 9,
    timestamp: new Date("2020-06-10"),
    price: 110000
  },
  {
    bike_name: "Hero Splendor Plus",
    model: "gear",
    category: "100cc",
    colors_available: ["black", "red"],
    manufacturer: "Hero",
    performance: 7,
    timestamp: new Date("2019-01-05"),
    price: 70000
  }
])

```

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

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

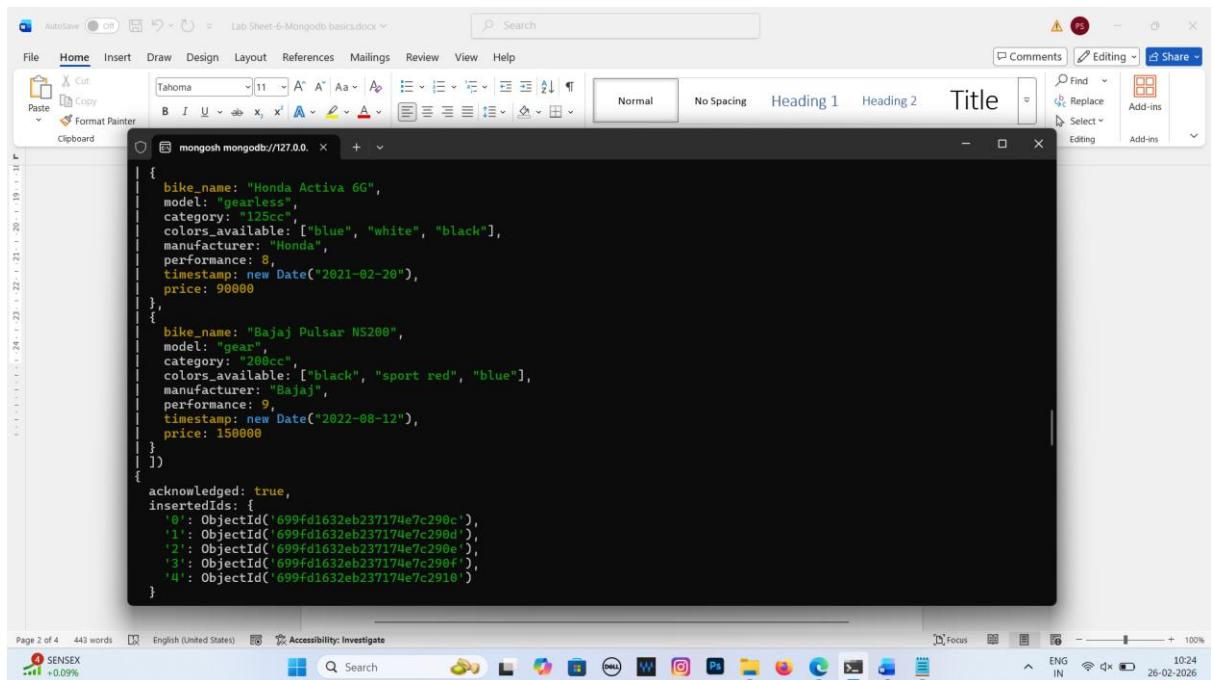
Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

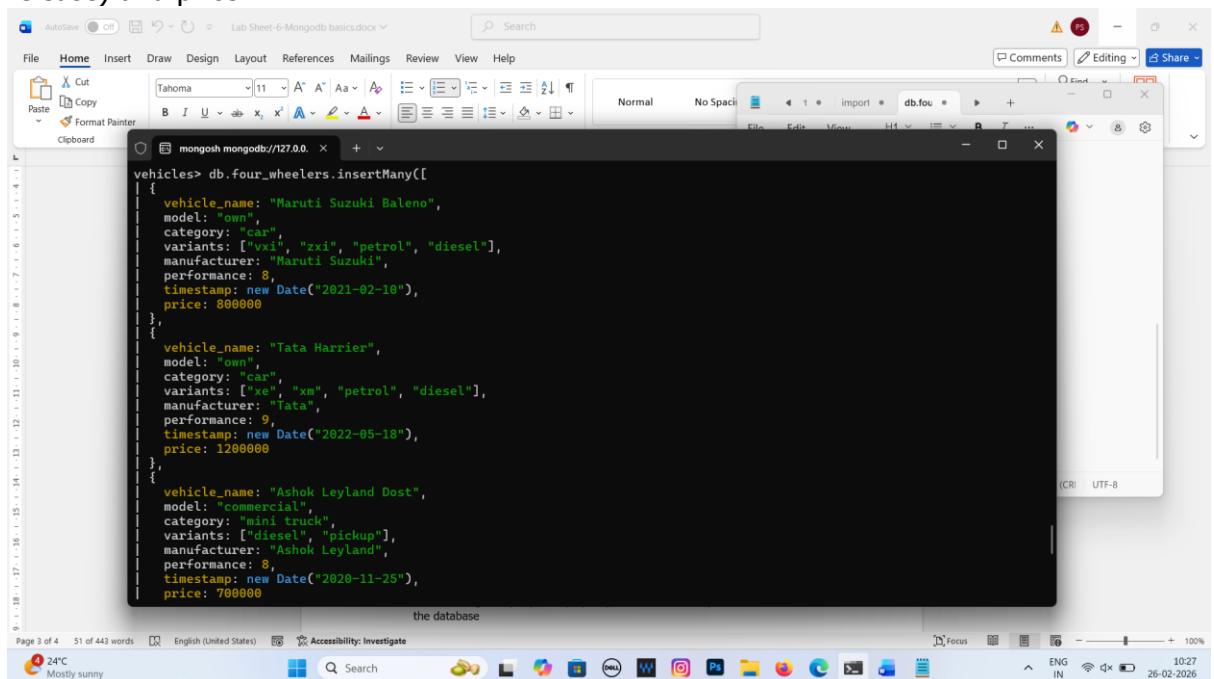
Student name: Ponhari S

Reg. no.: 23BCE8461



```
{
  "bike_name": "Honda Activa 6G",
  "model": "gearless",
  "category": "125cc",
  "colors_available": ["blue", "white", "black"],
  "manufacturer": "Honda",
  "performance": 8,
  "timestamp": new Date("2021-02-20"),
  "price": 90000
},
{
  "bike_name": "Bajaj Pulsar NS200",
  "model": "gear",
  "category": "200cc",
  "colors_available": ["black", "sport red", "blue"],
  "manufacturer": "Bajaj",
  "performance": 9,
  "timestamp": new Date("2022-08-12"),
  "price": 150000
}
]
{
  acknowledged: true,
  insertedIds: [
    '_0': ObjectId('699fd1632eb237174e7c290c'),
    '_1': ObjectId('699fd1632eb237174e7c290d'),
    '_2': ObjectId('699fd1632eb237174e7c290e'),
    '_3': ObjectId('699fd1632eb237174e7c290f'),
    '_4': ObjectId('699fd1632eb237174e7c2910')
  ]
}
```

- 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 Suzuki Baleno",
    model: "own",
    category: "car",
    variants: ["vxi", "zxi", "petrol", "diesel"],
    manufacturer: "Maruti Suzuki",
    performance: 8,
    timestamp: new Date("2021-02-10"),
    price: 800000
  },
  {
    vehicle_name: "Tata Harrier",
    model: "own",
    category: "car",
    variants: ["xe", "xm", "petrol", "diesel"],
    manufacturer: "Tata",
    performance: 9,
    timestamp: new Date("2022-05-18"),
    price: 1200000
  },
  {
    vehicle_name: "Ashok Leyland Dost",
    model: "commercial",
    category: "mini truck",
    variants: ["diesel", "pickup"],
    manufacturer: "Ashok Leyland",
    performance: 8,
    timestamp: new Date("2020-11-25"),
    price: 700000
  }
])
```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

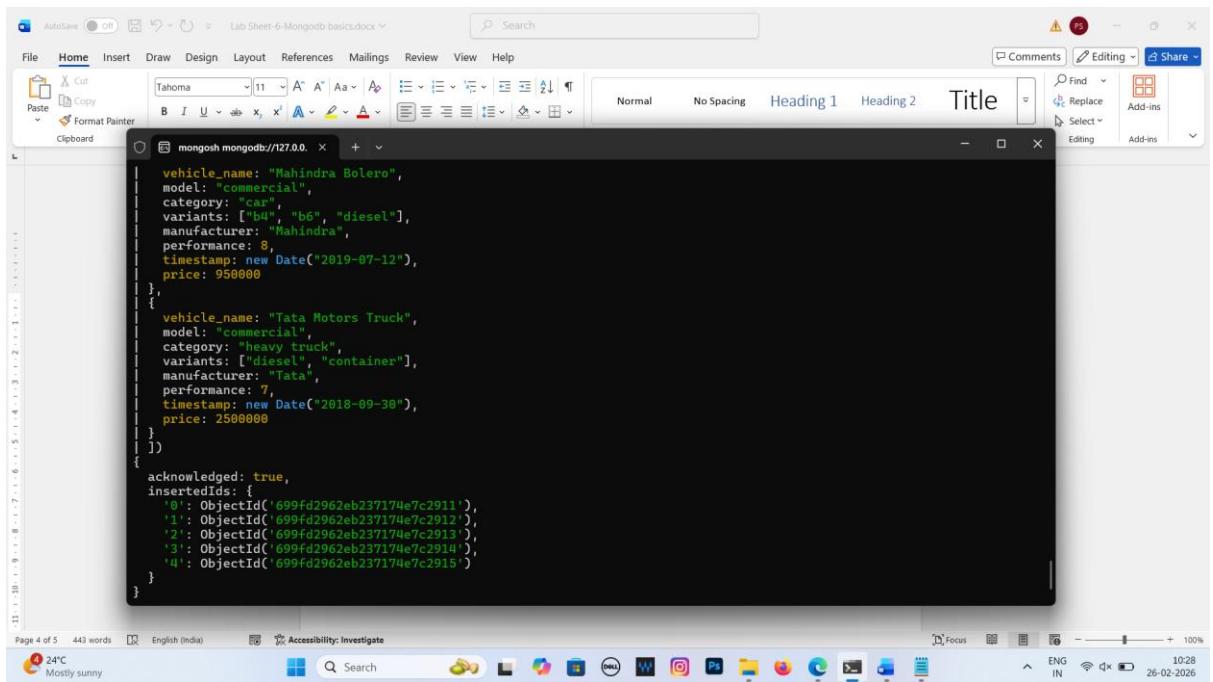
Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461



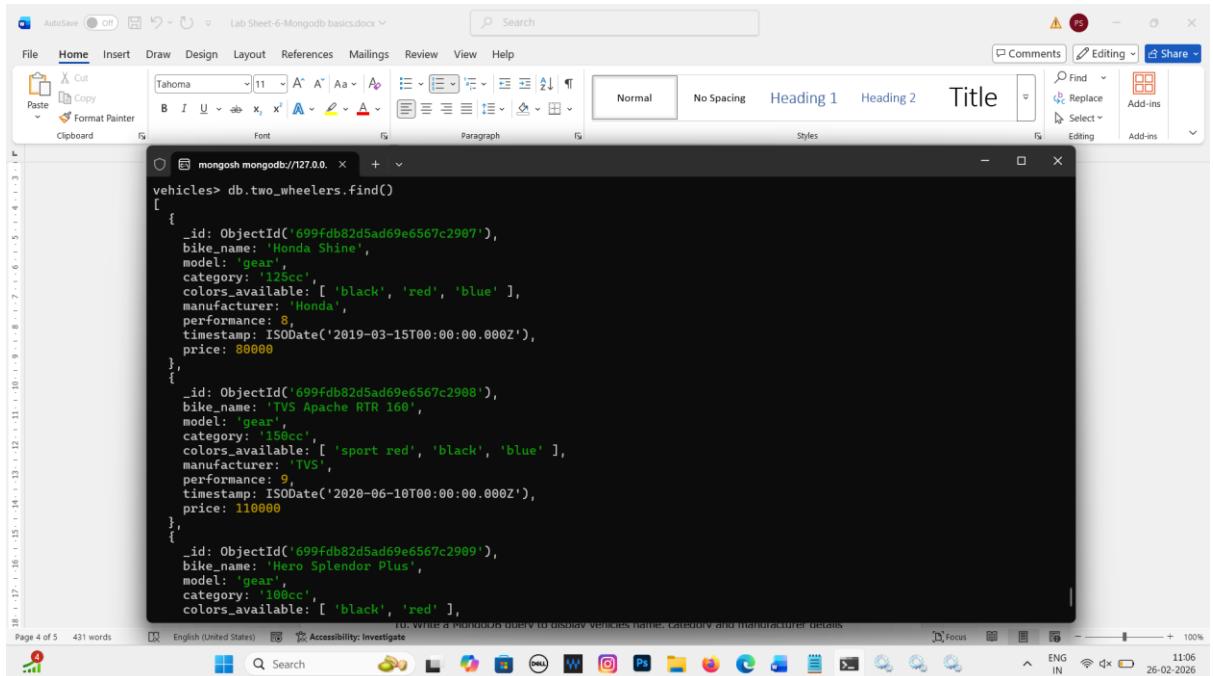
```

mongosh mongodb://127.0.0.1:27017

db.vehicles.insert([
  {
    vehicle_name: "Mahindra Bolero",
    model: "commercial",
    category: "car",
    variants: ["b4", "b6", "diesel"],
    manufacturer: "Mahindra",
    performance: 8,
    timestamp: new Date("2019-07-12"),
    price: 950000
  },
  {
    vehicle_name: "Tata Motors Truck",
    model: "commercial",
    category: "heavy truck",
    variants: ["diesel", "container"],
    manufacturer: "Tata",
    performance: 7,
    timestamp: new Date("2018-09-30"),
    price: 2500000
  }
])
{
  acknowledged: true,
  insertedIds: [
    '_0': ObjectId('699fd2962eb237174e7c2911'),
    '_1': ObjectId('699fd2962eb237174e7c2912'),
    '_2': ObjectId('699fd2962eb237174e7c2913'),
    '_3': ObjectId('699fd2962eb237174e7c2914'),
    '_4': ObjectId('699fd2962eb237174e7c2915')
  ]
}

```

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



```

mongosh mongodb://127.0.0.1:27017

db.vehicles.find()
[{
  _id: ObjectId('699fdb82d5ad69e6567c2907'),
  bike_name: 'Honda Shine',
  model: 'gear',
  category: '125cc',
  colors_available: [ 'black', 'red', 'blue' ],
  manufacturer: 'Honda',
  performance: 8,
  timestamp: ISODate('2019-03-15T00:00:00.000Z'),
  price: 80000
},
{
  _id: ObjectId('699fdb82d5ad69e6567c2908'),
  bike_name: 'TVS Apache RTR 160',
  model: 'gear',
  category: '150cc',
  colors_available: [ 'sport red', 'black', 'blue' ],
  manufacturer: 'TVS',
  performance: 9,
  timestamp: ISODate('2020-06-10T00:00:00.000Z'),
  price: 110000
},
{
  _id: ObjectId('699fdb82d5ad69e6567c2909'),
  bike_name: 'Hero Splendor Plus',
  model: 'gear',
  category: '100cc',
  colors_available: [ 'black', 'red' ],
  manufacturer: 'Hero'
}]

```

VIT-AP UNIVERSITY, ANDHRA PRADESH

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461

A screenshot of a Microsoft Word document titled "Lab Sheet-6-Mongodb basics.docx". The Word ribbon is visible at the top, showing tabs for File, Home, Insert, Draw, Design, Layout, References, Mailings, Review, View, and Help. The Home tab is selected. The main content area displays a code editor window with the following MongoDB query results:

```
mongosh mongo://127.0.0.1:27017
vehicles> db.four_wheelers.find()
[ {
  _id: ObjectId('699fdb8fd5ad69e6567c298c'),
  vehicle_name: 'Maruti Suzuki Swift',
  model: 'own',
  category: 'car',
  variants: [ 'xxi', 'xsi', 'petrol', 'diesel' ],
  manufacturer: 'Maruti Suzuki',
  performance: 8,
  timestamp: ISODate('2021-02-10T00:00:00.000Z'),
  price: 800000
},
{
  _id: ObjectId('699fdb8fd5ad69e6567c298d'),
  vehicle_name: 'Tata Nexon',
  model: 'own',
  category: 'car',
  variants: [ 'xe', 'xmt', 'petrol', 'diesel' ],
  manufacturer: 'Tata',
  performance: 9,
  timestamp: ISODate('2022-05-18T00:00:00.000Z'),
  price: 1200000
},
{
  _id: ObjectId('699fdb8fd5ad69e6567c298e'),
  vehicle_name: 'Ashok Leyland Dost',
  model: 'commercial',
  category: 'mini truck',
  variants: [ 'diesel', 'pickup' ],
  manufacturer: 'Ashok Leyland',
  performance: 10,
  timestamp: ISODate('2022-05-18T00:00:00.000Z'),
  price: 1500000
} ]
```

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

A screenshot of a Microsoft Word document titled "Lab Sheet-6-Mongodb basics.docx". The Word ribbon is visible at the top. A code editor window is open, displaying MongoDB shell queries and their results. The queries are as follows:

```
[{ price: 80000 },
{ price: 110000 },
{ price: 70000 },
{ price: 90000 },
{ price: 150000 }]
vehicles> db.two_wHEELERS.find(
| {},
| { bike_name: 1, price: 1, _id: 0 }
| )
[ { bike_name: 'Honda Shine', price: 80000 },
{ bike_name: 'TVS Apache RTR 160', price: 110000 },
{ bike_name: 'Hero Splendor Plus', price: 70000 },
{ bike_name: 'Honda Activa 6G', price: 90000 },
{ bike_name: 'Bajaj Pulsar NS200', price: 150000 }]
vehicles> db.four_wHEELERS.find(
| {},
| { vehicle_name: 1, price: 1, _id: 0 }
| )
[ { vehicle_name: 'Maruti Suzuki Swift', price: 80000 },
{ vehicle_name: 'Tata Nexon', price: 120000 },
{ vehicle_name: 'Ashok Leyland Dost', price: 70000 },
{ vehicle_name: 'Mahindra Bolero', price: 95000 },
{ vehicle_name: 'Tata Motors Truck', price: 2500000 }]
vehicles>
```

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

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

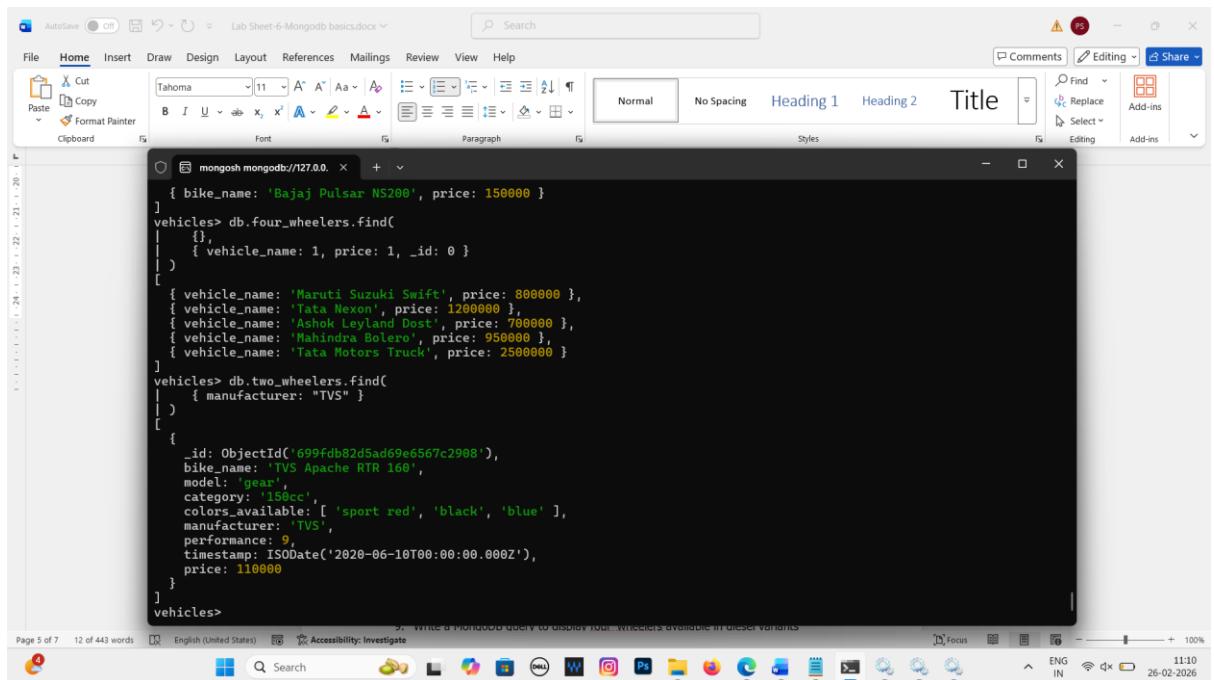
Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461

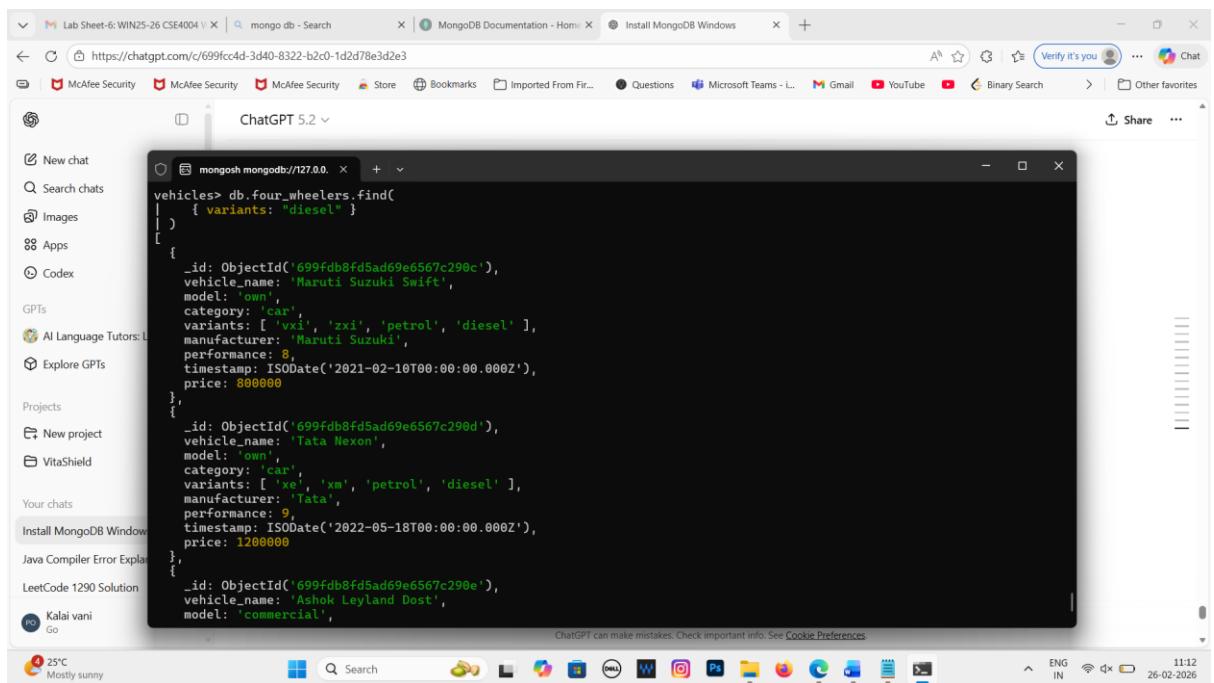


```

File Home Insert Draw Design Layout References Mailings Review View Help
Cut Copy Format Painter
Clipboard Font Paragraph Styles
Normal No Spacing Heading 1 Heading 2 Title
Comments Editing Share Add-ins
Find Replace Select Add-ins
[mongo] > db.four_wheelers.find()
[1]
{
  vehicle_name: 'Bajaj Pulsar NS200', price: 150000
}
[2]
{
  vehicle_name: 'Maruti Suzuki Swift', price: 800000
}
[3]
{
  vehicle_name: 'Tata Nexon', price: 1200000
}
[4]
{
  vehicle_name: 'Ashok Leyland Dost', price: 700000
}
[5]
{
  vehicle_name: 'Mahindra Bolero', price: 950000
}
[6]
{
  vehicle_name: 'Tata Motors Truck', price: 2500000
}
[7]
{
  manufacturer: "TVS"
}
[8]
{
  _id: ObjectId('699fdb82d5ad69e6567c2908'),
  bike_name: 'TVS Apache RTR 160',
  model: 'gear',
  category: '150cc',
  colors_available: [ 'sport red', 'black', 'blue' ],
  manufacturer: 'TVS',
  performance: 9,
  timestamp: ISODate('2020-06-10T00:00:00.000Z'),
  price: 110000
}
[9]
vehicles>

```

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



```

[mongo] > db.four_wheelers.find()
[1]
{
  variants: "diesel"
}
[2]
{
  _id: ObjectId('699fdb8fd5ad69e6567c290c'),
  vehicle_name: 'Maruti Suzuki Swift',
  model: 'own',
  category: 'car',
  variants: [ 'x1i', 'xzi', 'patrol', 'diesel' ],
  manufacturer: 'Maruti Suzuki',
  performance: 8,
  timestamp: ISODate('2021-02-10T00:00:00.000Z'),
  price: 80000
},
[3]
{
  _id: ObjectId('699fdb8fd5ad69e6567c290d'),
  vehicle_name: 'Tata Nexon',
  model: 'own',
  category: 'car',
  variants: [ 'xe', 'xm', 'petrol', 'diesel' ],
  manufacturer: 'Tata',
  performance: 9,
  timestamp: ISODate('2022-05-18T00:00:00.000Z'),
  price: 120000
},
[4]
{
  _id: ObjectId('699fdb8fd5ad69e6567c290e'),
  vehicle_name: 'Ashok Leyland Dost',
  model: 'commercial',
}

```

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

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

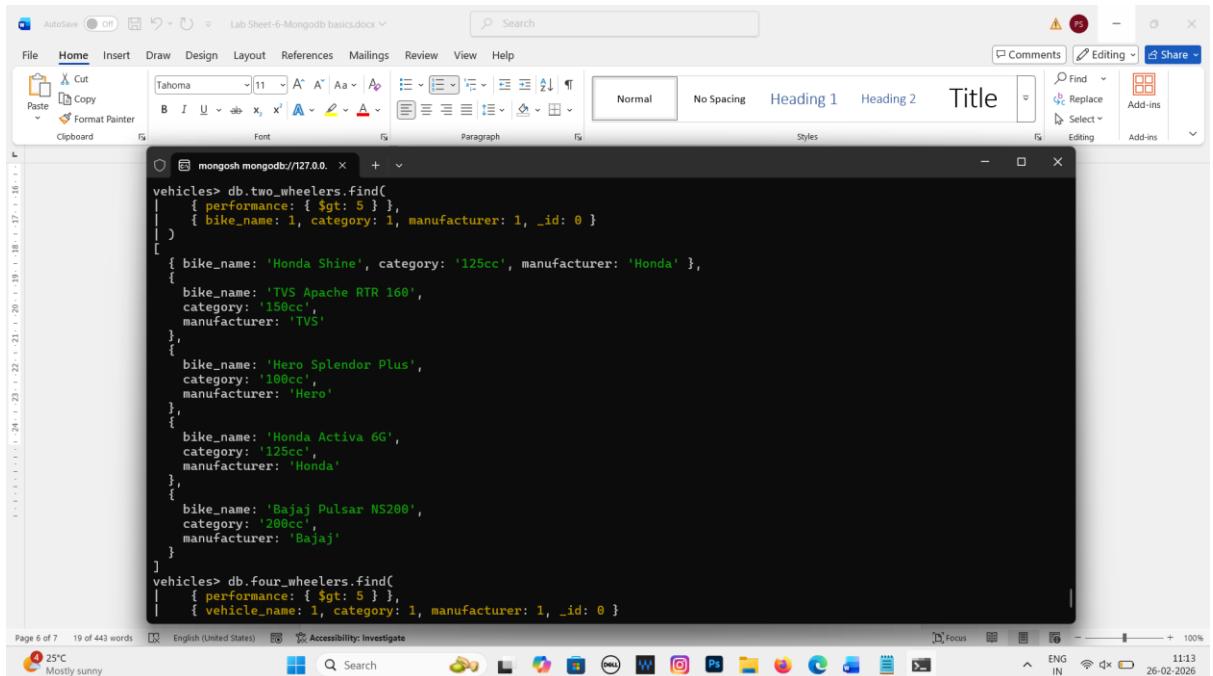
Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Ponhari S

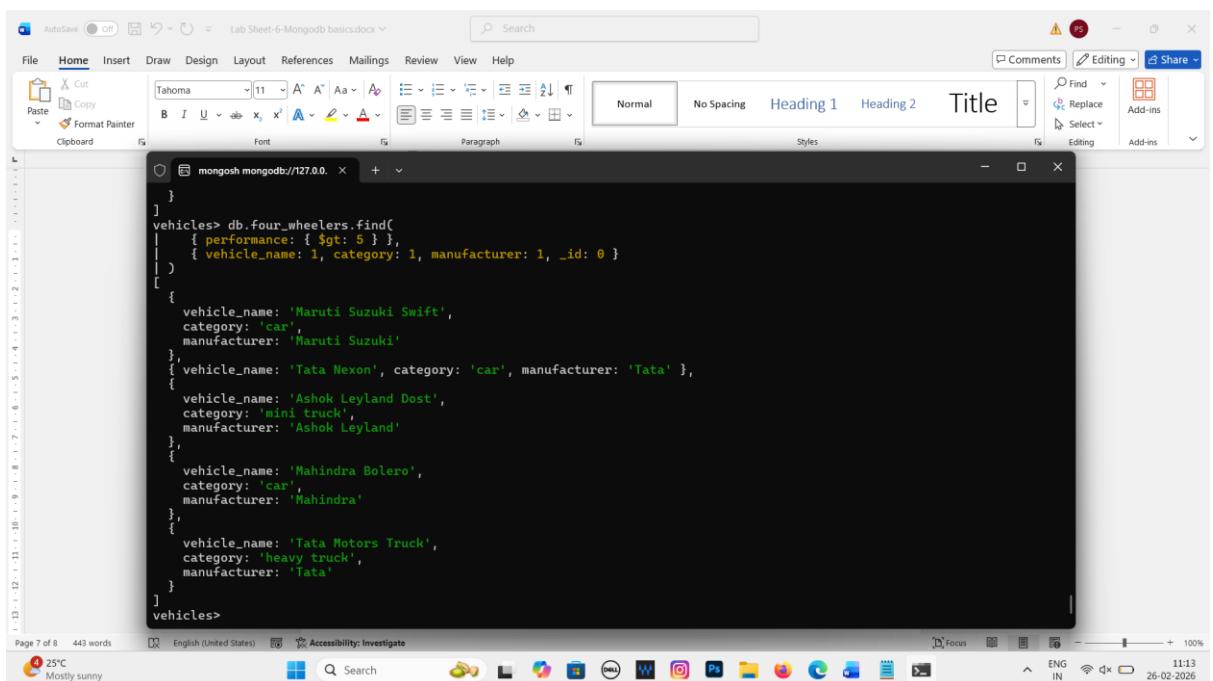
Reg. no.: 23BCE8461



The screenshot shows a Microsoft Word document titled "Lab Sheet-6-Mongodb basics.docx". An embedded terminal window displays MongoDB shell commands and their results. The commands are:

```
mongosh mongoDB://127.0.0.1:27017
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: 'Honda Activa 6G', category: '125cc', manufacturer: 'Honda' },
  { bike_name: 'Bajaj Pulsar NS200', category: '200cc', manufacturer: 'Bajaj' }
]
vehicles> db.four_wheelers.find(
  { performance: { $gt: 5 } },
  { vehicle_name: 1, category: 1, manufacturer: 1, _id: 0 }
)
```

The results show five motorcycle documents from the 'two_wheelers' collection and five car documents from the 'four_wheelers' collection.



The screenshot shows a Microsoft Word document titled "Lab Sheet-6-Mongodb basics.docx". An embedded terminal window displays MongoDB shell commands and their results. The commands are:

```
mongosh mongoDB://127.0.0.1:27017
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: 'Honda Activa 6G', category: '125cc', manufacturer: 'Honda' },
  { bike_name: 'Bajaj Pulsar NS200', category: '200cc', manufacturer: 'Bajaj' }
]
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 Nexus', category: 'car', manufacturer: 'Tata' },
  { vehicle_name: 'Ashok Leyland Dost', category: 'mini truck', manufacturer: 'Ashok Leyland' },
  { vehicle_name: 'Mahindra Bolero', category: 'car', manufacturer: 'Mahindra' },
  { vehicle_name: 'Tata Motors Truck', category: 'heavy truck', manufacturer: 'Tata' }
]
```

The results show five motorcycle documents from the 'two_wheelers' collection and five car documents from the 'four_wheelers' collection.

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461

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'.

The screenshot shows a Microsoft Word document window with a title bar 'Lab Sheet-6-Mongodb basics.docx'. The main content area contains a terminal-like interface for the MongoDB shell (mongosh). The output shows the connection string, MongoDB version, and a command to switch to the 'animal' database:

```
Please enter a MongoDB connection string (Default: mongodb://localhost/):
Current Mongosh Log ID: 699fde736f0748ff67c2986
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
2026-02-26T11:01:39.136+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----

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

The status bar at the bottom indicates 'Page 8 of 8' and '443 words'.

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

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

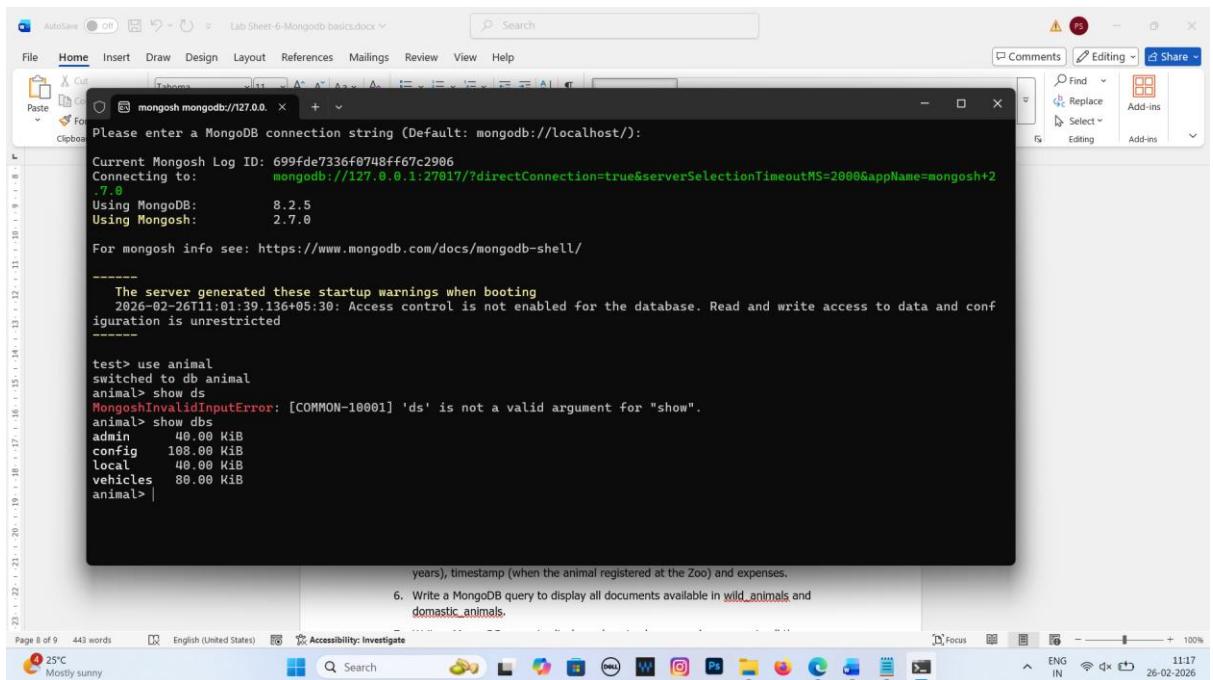
Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

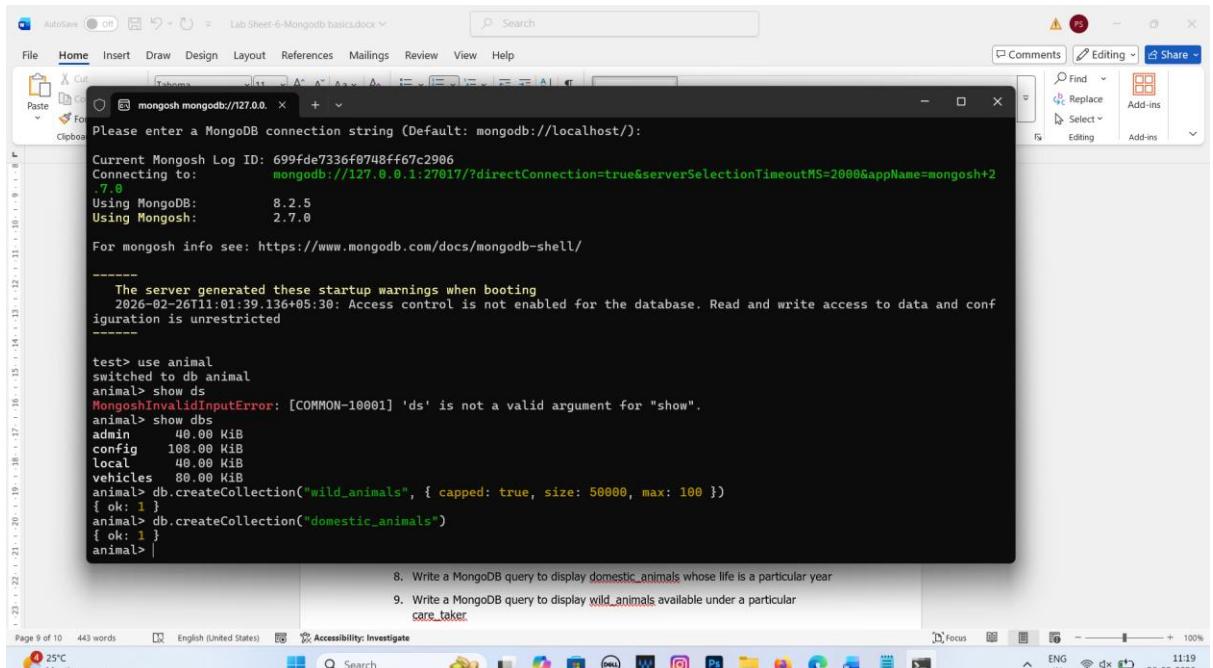
School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461



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



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).

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

Date: 26/02/2026

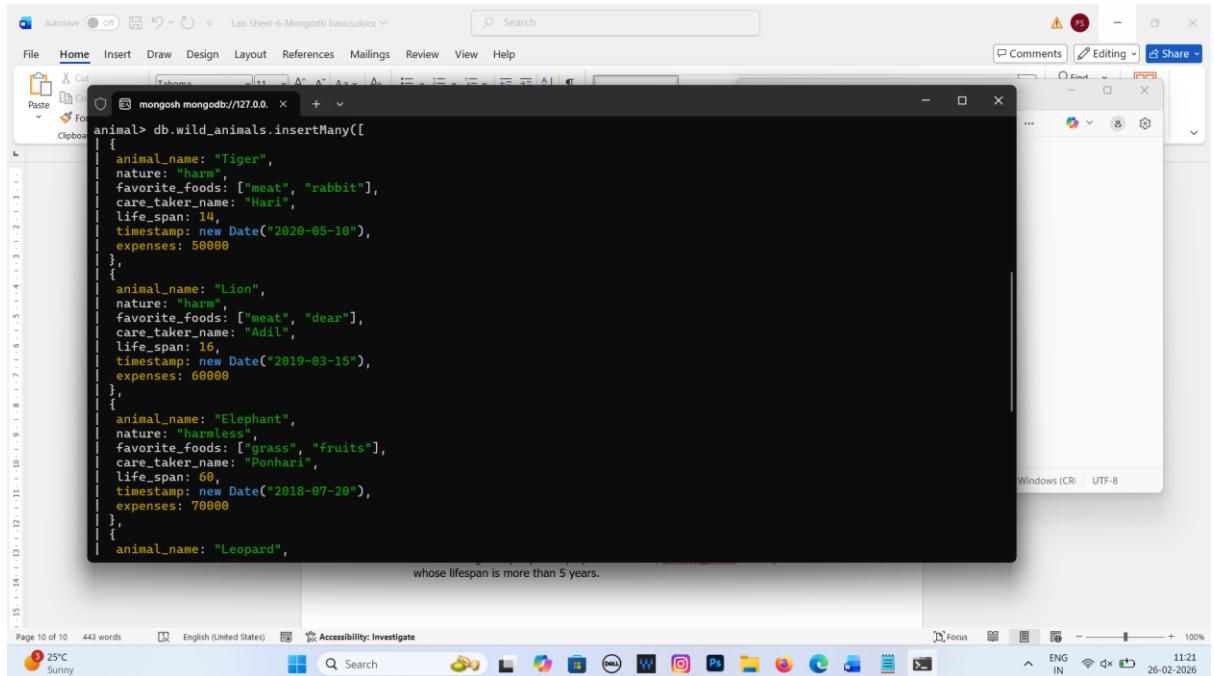
Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461

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.

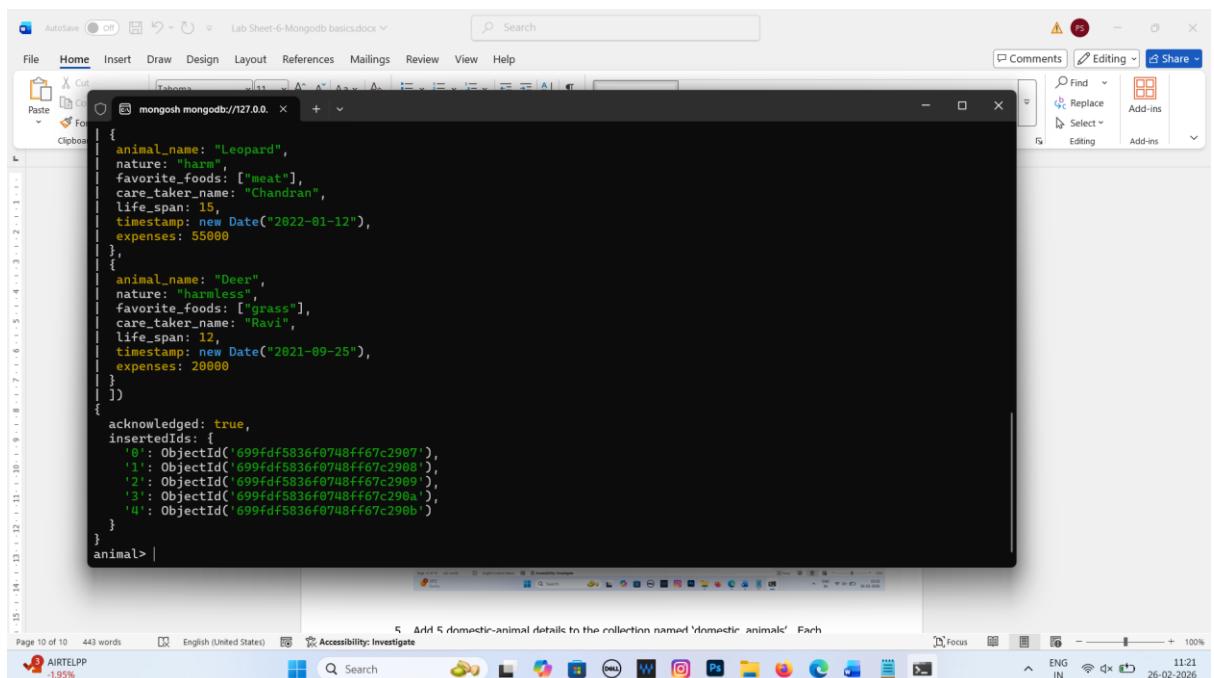


```

animal> db.wild_animals.insertMany([
  {
    animal_name: "Tiger",
    nature: "harm",
    favorite_foods: ["meat", "rabbit"],
    care_taker_name: "Hari",
    life_span: 14,
    timestamp: new Date("2020-05-10"),
    expenses: 50000
  },
  {
    animal_name: "Lion",
    nature: "harm",
    favorite_foods: ["meat", "dear"],
    care_taker_name: "Adil",
    life_span: 16,
    timestamp: new Date("2019-03-15"),
    expenses: 60000
  },
  {
    animal_name: "Elephant",
    nature: "harmless",
    favorite_foods: ["grass", "fruits"],
    care_taker_name: "Ponhari",
    life_span: 60,
    timestamp: new Date("2018-07-20"),
    expenses: 70000
  },
  {
    animal_name: "Leopard",
    ...
  }
])

```

whose lifespan is more than 5 years.



```

animal> db.domestic_animals.insertMany([
  {
    animal_name: "Leopard",
    nature: "harm",
    favorite_foods: ["meat"],
    care_taker_name: "Chandran",
    life_span: 15,
    timestamp: new Date("2022-01-12"),
    expenses: 55000
  },
  {
    animal_name: "Deer",
    nature: "harmless",
    favorite_foods: ["grass"],
    care_taker_name: "Ravi",
    life_span: 12,
    timestamp: new Date("2021-09-25"),
    expenses: 20000
  }
])

```

```

{
  acknowledged: true,
  insertedIds: [
    '_0': ObjectId('699fdff5836f0748ff67c2987'),
    '_1': ObjectId('699fdff5836f0748ff67c2988'),
    '_2': ObjectId('699fdff5836f0748ff67c2989'),
    '_3': ObjectId('699fdff5836f0748ff67c298a'),
    '_4': ObjectId('699fdff5836f0748ff67c298b')
  ]
}

```

5. Add 5 domestic-animal details to the collection named 'domestic_animals'. Each

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

Date: 26/02/2026

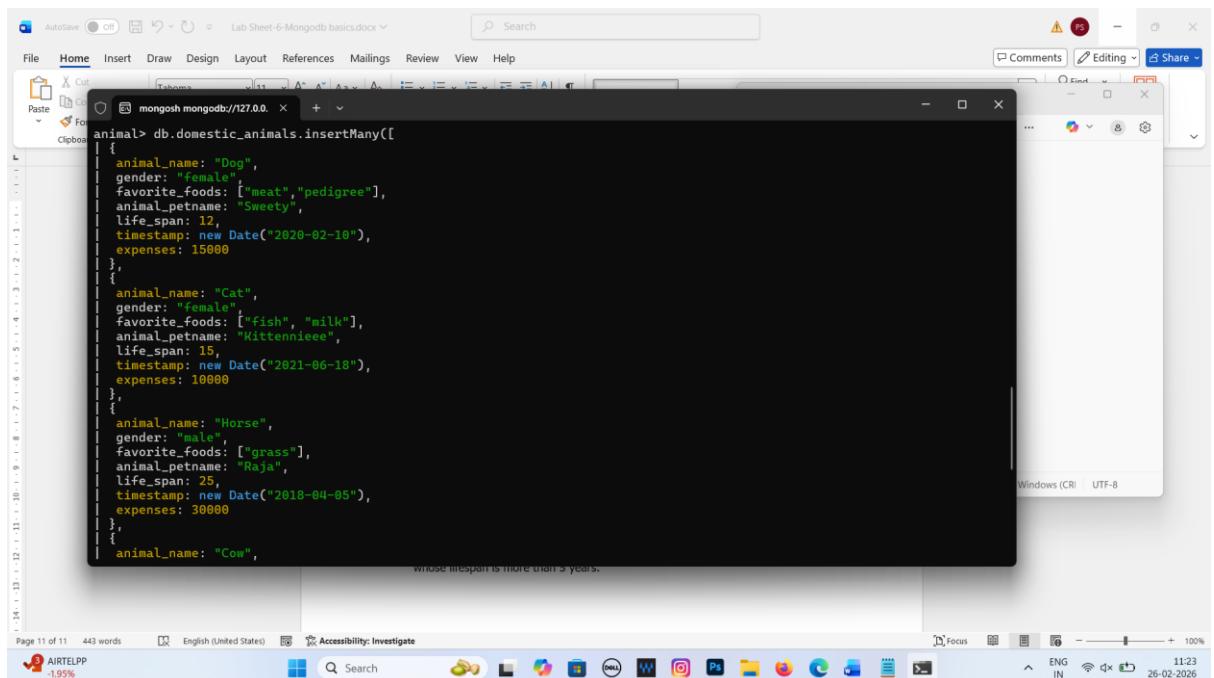
Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461

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.

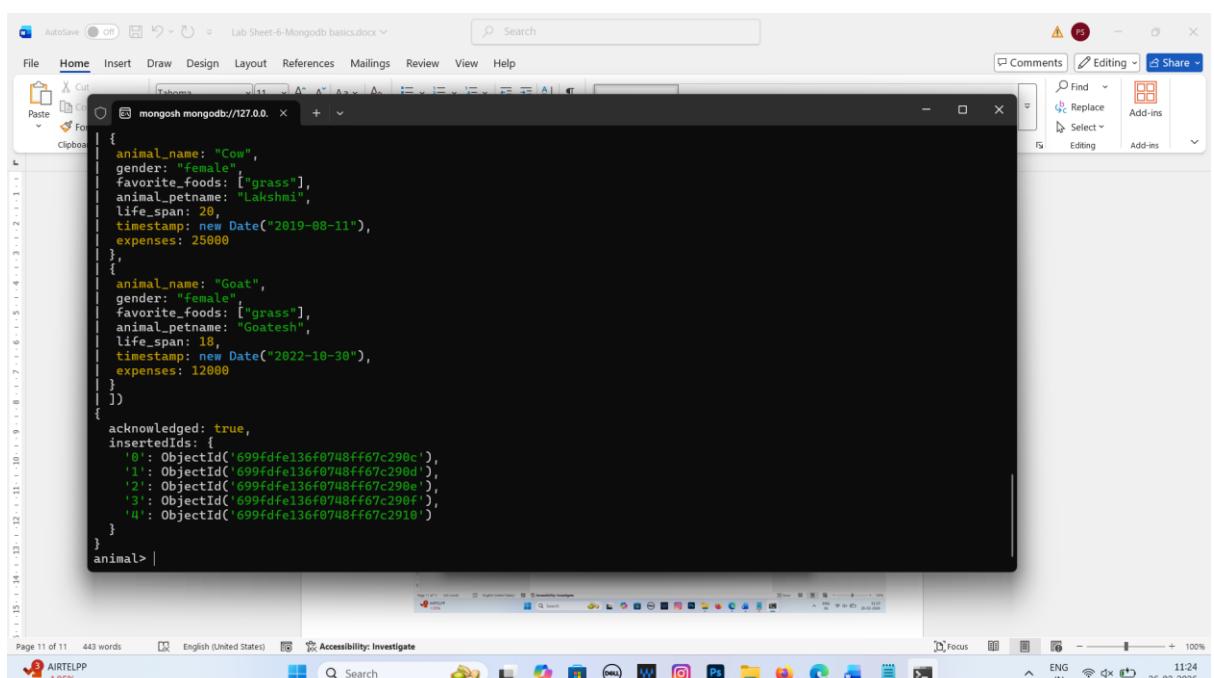


```

animal> db.domestic_animals.insertMany([
  {
    animal_name: "Dog",
    gender: "female",
    favorite_foods: ["meat", "pedigree"],
    animal_petname: "Sweetie",
    life_span: 12,
    timestamp: new Date("2020-02-10"),
    expenses: 15000
  },
  {
    animal_name: "Cat",
    gender: "female",
    favorite_foods: ["fish", "milk"],
    animal_petname: "Kittenniee",
    life_span: 15,
    timestamp: new Date("2021-06-18"),
    expenses: 10000
  },
  {
    animal_name: "Horse",
    gender: "male",
    favorite_foods: ["grass"],
    animal_petname: "Raja",
    life_span: 25,
    timestamp: new Date("2018-04-05"),
    expenses: 30000
  },
  {
    animal_name: "Cow",
    gender: "female",
    favorite_foods: ["grass"],
    animal_petname: "Lakshmi",
    life_span: 20,
    timestamp: new Date("2019-08-11"),
    expenses: 25000
  }
])

```

whose lifespan is more than 5 years.



```

animal> [
  {
    animal_name: "Cow",
    gender: "female",
    favorite_foods: ["grass"],
    animal_petname: "Lakshmi",
    life_span: 20,
    timestamp: new Date("2019-08-11"),
    expenses: 25000
  },
  {
    animal_name: "Goat",
    gender: "female",
    favorite_foods: ["grass"],
    animal_petname: "Goatesh",
    life_span: 18,
    timestamp: new Date("2022-10-30"),
    expenses: 12000
  }
]

acknowledged: true,
insertedIds: [
  '0': ObjectId( 699fdf136f0748ff67c290c ),
  '1': ObjectId( 699fdf136f0748ff67c290d ),
  '2': ObjectId( 699fdf136f0748ff67c290e ),
  '3': ObjectId( 699fdf136f0748ff67c290f ),
  '4': ObjectId( 699fdf136f0748ff67c2910' )
]
animal>

```

Lab Sheet 6: MongoDB Basic commands

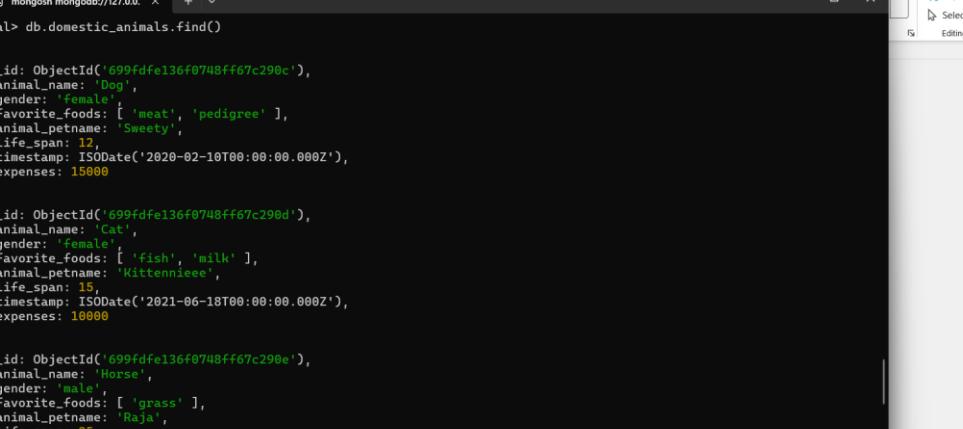
Branch/ Class: B.Tech
Faculty Name: Prof. S.Gopikrishnam
Student name: Ponhari S

Date: 26/02/2026
School: SCOPE
Req. no.: 23BCE8461

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

A screenshot of a Microsoft Word document titled "Lab Sheet-6-Mongodb basics.docx". The document contains a single code block representing MongoDB query results. The code shows three documents from a collection named "wild_animals". Each document has fields: _id, animal_name, nature, favorite_foods, care_taker_name, life_span, timestamp, and expenses.

```
animal> db.wild_animals.find()
[ {
    _id: ObjectId('699fdf5836f0748ff67c2907'),
    animal_name: 'Tiger',
    nature: 'harm',
    favorite_foods: [ 'meat', 'rabbit' ],
    care_taker_name: 'Hari',
    life_span: 14,
    timestamp: ISODate('2020-05-10T00:00:00.000Z'),
    expenses: 50000
},
{
    _id: ObjectId('699fdf5836f0748ff67c2908'),
    animal_name: 'Lion',
    nature: 'harm',
    favorite_foods: [ 'meat', 'dear' ],
    care_taker_name: 'Adil',
    life_span: 16,
    timestamp: ISODate('2019-03-15T00:00:00.000Z'),
    expenses: 60000
},
{
    _id: ObjectId('699fdf5836f0748ff67c2909'),
    animal_name: 'Elephant',
    nature: 'harmless',
    favorite_foods: [ 'grass', 'fruits' ],
    care_taker_name: 'Ponhari',
    life_span: 60,
    timestamp: ISODate('2018-07-20T00:00:00.000Z'),
    expenses: 70000
}]
```



The screenshot shows a Microsoft Word document titled "Lab Sheet-6-Mongodb basics.docx". The document contains a single line of code:

```
animal> db.domestic_animals.find()
```

This line represents a MongoDB query executed in the mongo shell. The query uses the find() method on the "domestic_animals" collection to retrieve all documents.

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

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

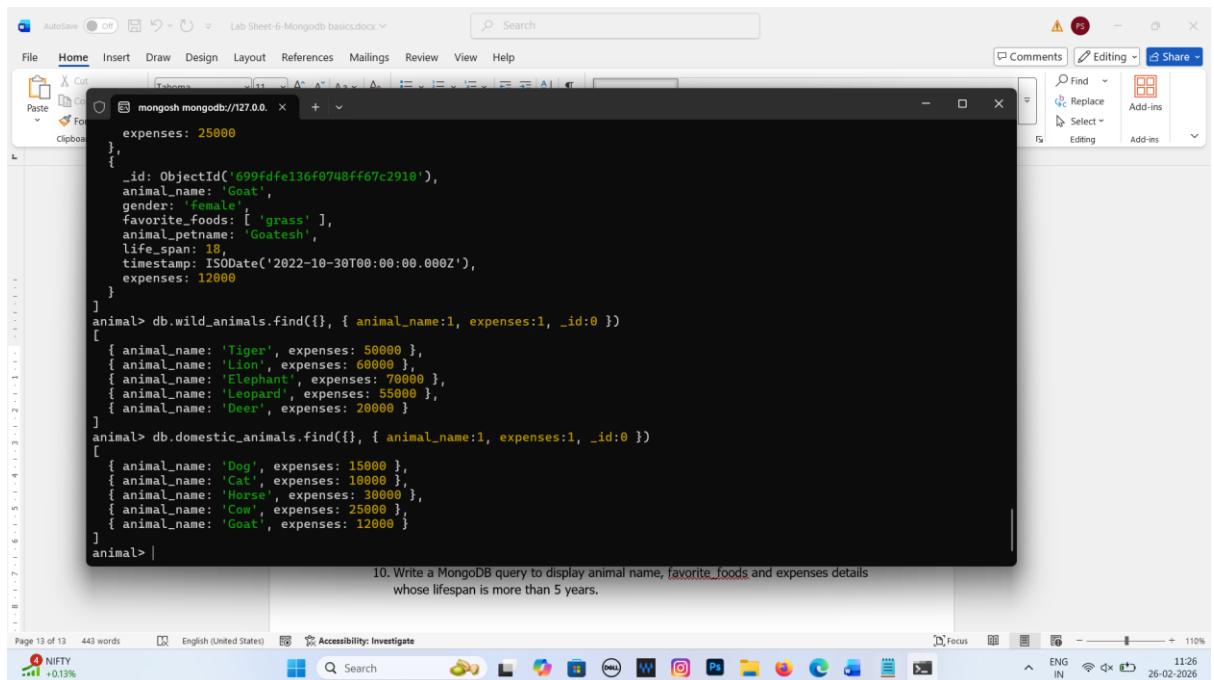
Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

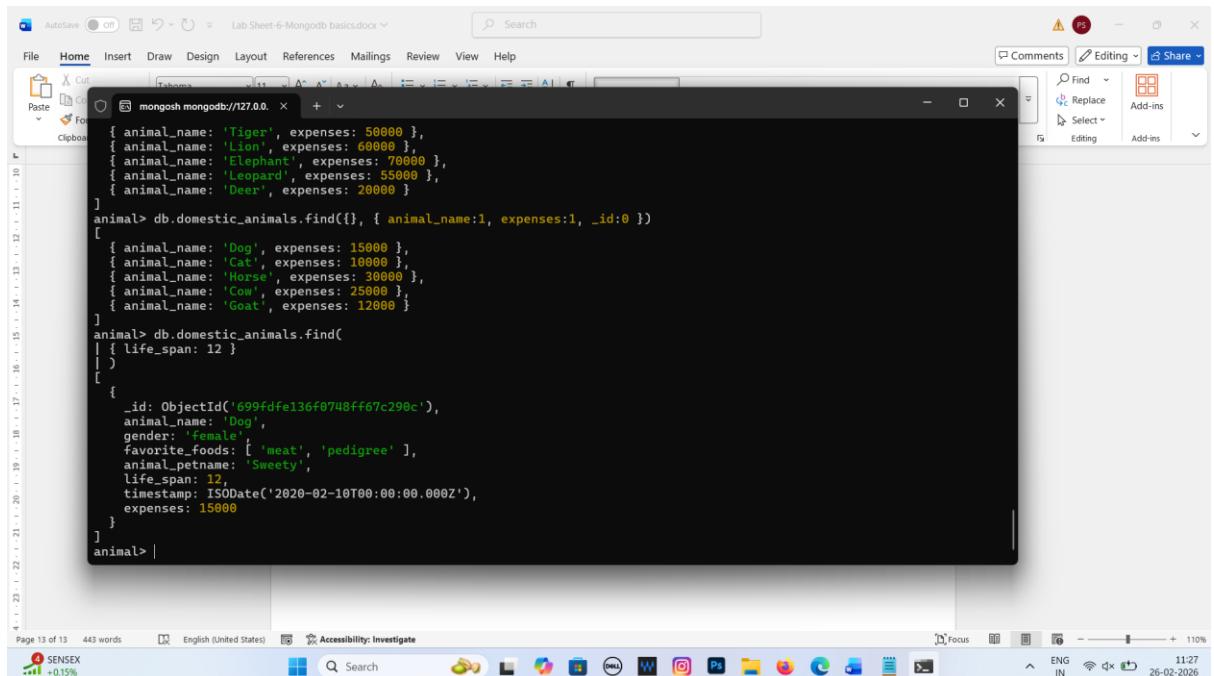
School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461



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



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

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

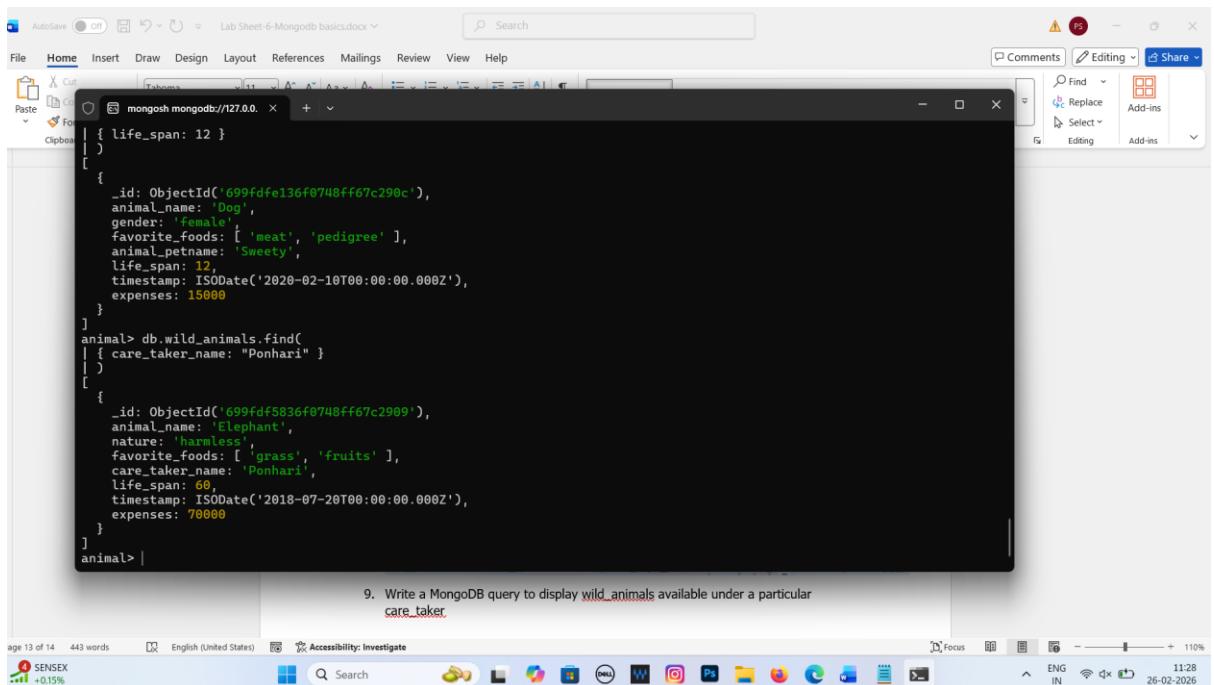
Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461



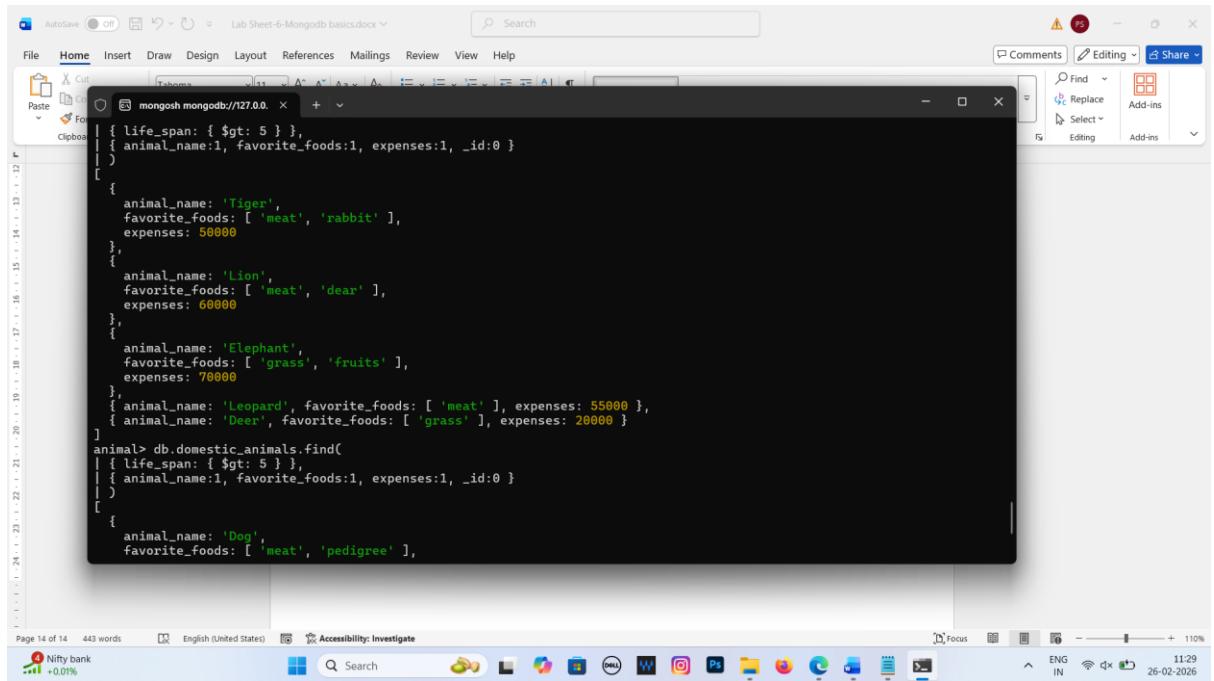
```

File Home Insert Draw Design Layout References Mailings Review View Help
|mongosh mongodb://127.0.0.1:27017/wild_animals>
| { life_span: 12 }
| [
| {
|   _id: ObjectId('699fdf136f0748ff67c290c'),
|   animal_name: 'Dog',
|   gender: 'female',
|   favorite_foods: [ 'meat', 'pedigree' ],
|   animal_petsname: 'Sweety',
|   life_span: 12,
|   timestamp: ISODate('2020-02-10T00:00:00.000Z'),
|   expenses: 15000
| }
| ]
animal> db.wild_animals.find(
| { care_taker_name: "Ponhari" }
| )
[
| {
|   _id: ObjectId('699fdf5836f0748ff67c2909'),
|   animal_name: 'Elephant',
|   nature: 'harmless',
|   favorite_foods: [ 'grass', 'fruits' ],
|   care_taker_name: 'Ponhari',
|   life_span: 60,
|   timestamp: ISODate('2018-07-20T00:00:00.000Z'),
|   expenses: 70000
| }
| ]
animal> |

```

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

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



```

File Home Insert Draw Design Layout References Mailings Review View Help
|mongosh mongodb://127.0.0.1:27017/domestic_animals>
| { life_span: { $gt: 5 } },
| { animal_name:1, favorite_foods:1, expenses:1, _id:0 }
| [
| {
|   animal_name: 'Tiger',
|   favorite_foods: [ 'meat', 'rabbit' ],
|   expenses: 50000
| },
| {
|   animal_name: 'Lion',
|   favorite_foods: [ 'meat', 'dear' ],
|   expenses: 60000
| },
| {
|   animal_name: 'Elephant',
|   favorite_foods: [ 'grass', 'fruits' ],
|   expenses: 70000
| },
| {
|   animal_name: 'Leopard',
|   favorite_foods: [ 'meat' ],
|   expenses: 55000
| },
| {
|   animal_name: 'Deer',
|   favorite_foods: [ 'grass' ],
|   expenses: 20000
| }
| ]
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', 'pedigree' ],
|   expenses: 15000
| }
| ]
animal> |

```

Lab Sheet 6: MongoDB Basic commands

Branch/ Class: B.Tech

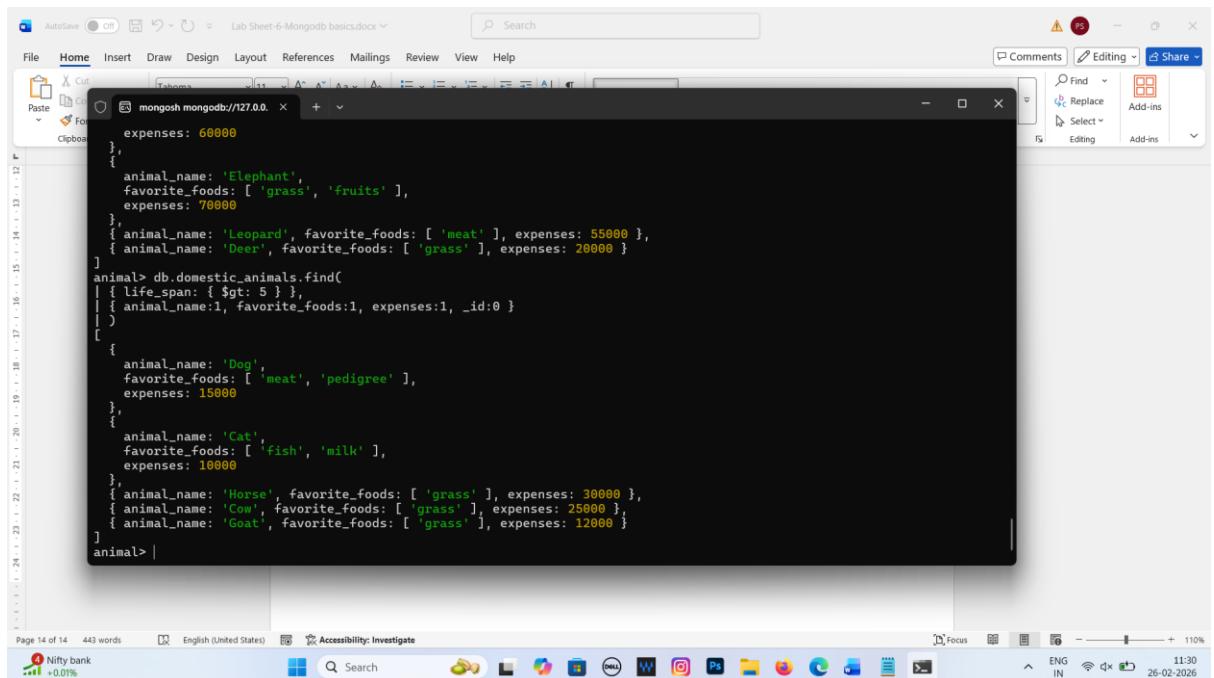
Date: 26/02/2026

Faculty Name: Prof. S.Gopikrishnan

School: SCOPE

Student name: Ponhari S

Reg. no.: 23BCE8461



The screenshot shows a Microsoft Word document titled "Lab Sheet-6-Mongodb basics.docx". A code editor window is open within the Word interface, displaying MongoDB shell commands and their results. The commands involve querying a collection named "domestic_animals" and filtering documents based on life_span and favorite_foods. The results are presented as an array of objects, each containing animal_name, favorite_foods, and expenses.

```
expenses: 60000
},
{
  animal_name: 'Elephant',
  favorite_foods: [ 'grass', 'fruits' ],
  expenses: 70000
},
{ animal_name: 'Leopard', favorite_foods: [ 'meat' ], expenses: 55000 },
{ animal_name: 'Deer', favorite_foods: [ 'grass' ], expenses: 20000 }
]
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', 'pedigree' ],
    expenses: 15000
  },
  {
    animal_name: 'Cat',
    favorite_foods: [ 'fish', 'milk' ],
    expenses: 10000
  },
  {
    animal_name: 'Horse',
    favorite_foods: [ 'grass' ],
    expenses: 30000
  },
  { animal_name: 'Cow', favorite_foods: [ 'grass' ],
    expenses: 25000 },
  { animal_name: 'Goat', favorite_foods: [ 'grass' ],
    expenses: 12000 }
]
animal> |
```