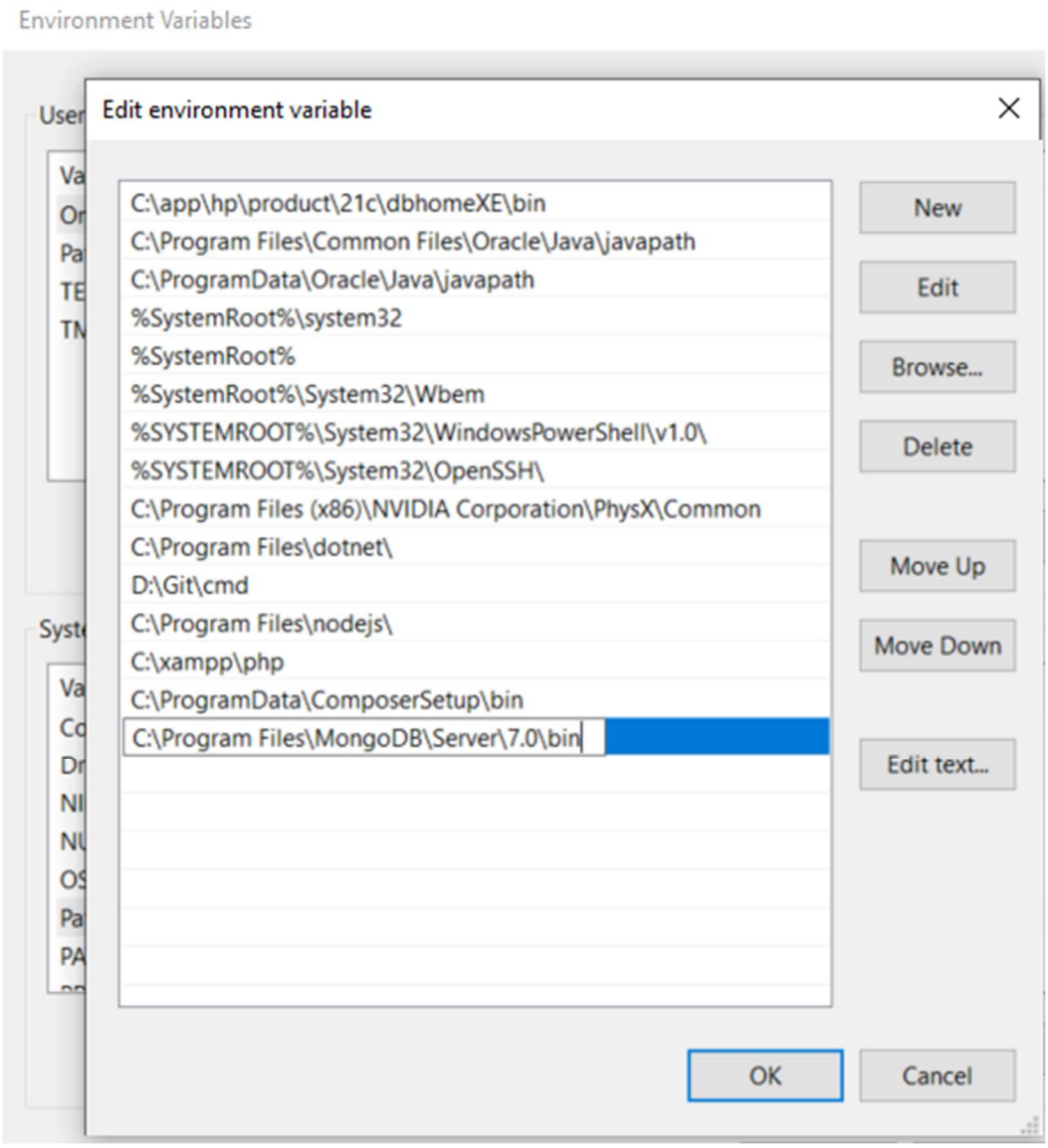
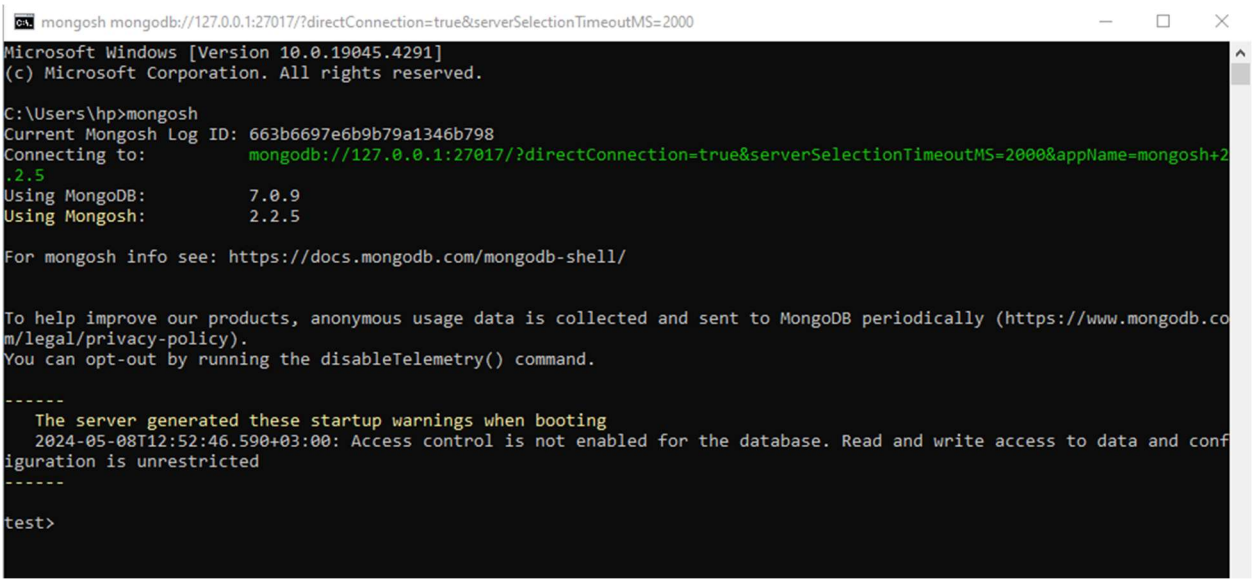
Student name:

Student number:

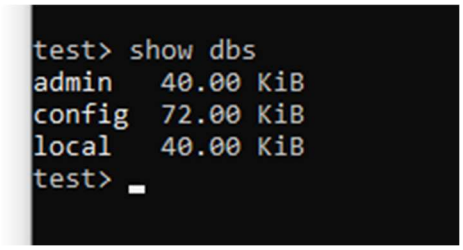
First we need to started the mongo DB and set the environment variable for mongo DB in the System:



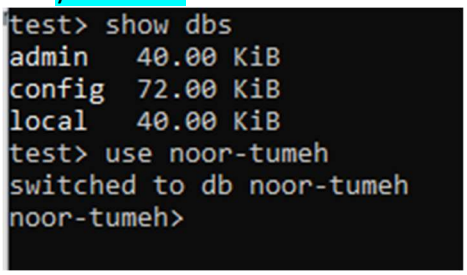
Open mongo shell:



To show Database:



1- Create database with Your-Name and switched to it:
Use your-name



2- After create the database I switched to it

```
test> use noor-tumeh
switched to db noor-tumeh
noor-tumeh>
```

3- Create collection with name namecollection:
Db.createCollection("nameCollection);
Show dbs

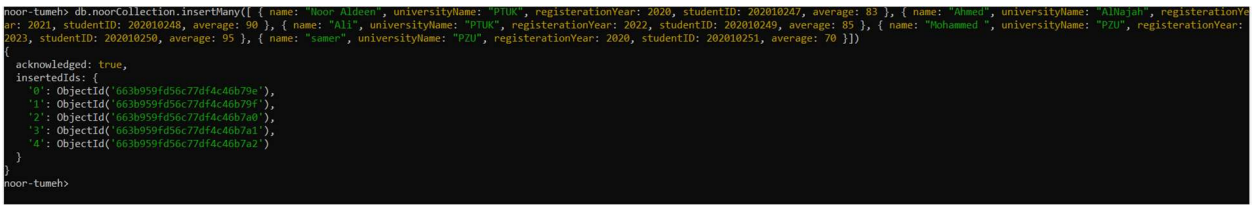
```
noor-tumeh> db.createCollection("noorCollection");
{ ok: 1 }
noor-tumeh> show dbs
admin       40.00 KiB
config      72.00 KiB
local       72.00 KiB
noor-tumeh   8.00 KiB
```

4- Write a query to make sure that the collection was created:
Show collections

```
noor-tumeh> show collections
noorCollection
```

5- Add 5 persons to the collection; each one of them has the following properties: name, university name, registration year, student ID, and average (value from 0 to 100), one of these entries must has your info:

db.nameCollection.insertMany([
{name:"name ", universityName:"PTUK", registerationYear: 2020, studentID: universityID, average:00},
{name:"aaaaaa", universityName:"AlNajah", registerationYear: 2021, studentID: 000000, average:90},
{name:"aaaa", universityName:"PTUK", registerationYear: 2022, studentID: 000000, average:85},
{name:"aaaaaa ", universityName:"PZU", registerationYear: 2023, studentID: 0000, average:95},
{name:"aaaaa", universityName:"PZU", registerationYear: 2020, studentID: 000000, average:70}
]);



6- Write a query to return all data in the collection:
db.nameCollection.find();

```
noor-tumeh> db.noorCollection.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
5
noor-tumeh> db.noorCollection.find()
[
  {
    _id: ObjectId('663b959fd56c77df4c46b79e'),
    name: 'Noor Aldeen',
    universityName: 'PTUK',
    registerationYear: 2020,
    studentID: 202010247,
    average: 83
  },
  {
    _id: ObjectId('663b959fd56c77df4c46b79f'),
    name: 'Ahmed',
    universityName: 'AlNajah',
    registerationYear: 2021,
    studentID: 202010248,
    average: 90
  },
  {
    _id: ObjectId('663b959fd56c77df4c46b7a0'),
    name: 'Ali',
    universityName: 'PTUK',
    registerationYear: 2022,
    studentID: 202010249,
    average: 85
  },
  {
    _id: ObjectId('663b959fd56c77df4c46b7a1'),
    name: 'Mohammed ',
    universityName: 'PZU',
    registerationYear: 2023,
    studentID: 202010250,
    average: 95
  },
  {
    _id: ObjectId('663b959fd56c77df4c46b7a2'),
    name: 'samer',
    universityName: 'PZU',
    registerationYear: 2020,
    studentID: 202010251,
    average: 70
  }
]
```

7- Write a query to return only 3 person:
db.nameCollection.find().limit(3);

8- write a query to return the top 3 persons based on their average :
db.nameCollection.find().sort({average: -1}).limit(3);

```
noor-tumeh> db.noorCollection.find().sort({average: -1}).limit(3);
[
  {
    _id: ObjectId('663b959fd56c77df4c46b7a1'),
    name: 'Mohammed ',
    universityName: 'PZU',
    registerationYear: 2023,
    studentID: 202010250,
    average: 95
  },
  {
    _id: ObjectId('663b959fd56c77df4c46b79f'),
    name: 'Ahmed',
    universityName: 'AlNajah',
    registerationYear: 2021,
    studentID: 202010248,
    average: 90
  },
  {
    _id: ObjectId('663b959fd56c77df4c46b7a0'),
    name: 'Ali',
    universityName: 'PTUK',
    registerationYear: 2022,
    studentID: 202010249,
    average: 85
  }
]
noor-tumeh>
```

9- write a query that return persons whose average is above 89 and who registered in university after 2021:

db.nameCollection.find({$and: [{average: {$gt:89}}, {registerationYear:{$gt:2021}}]});

```
noor-tumeh> db.noorCollection.find({$and: [{average: {$gt:89}}, {registerationYear:{$gt:2021}}]});
[
  {
    _id: ObjectId('663b959fd56c77df4c46b7a1'),
    name: 'Mohammed ',
    universityName: 'PZU',
    registerationYear: 2023,
    studentID: 202010250,
    average: 95
  }
]
noor-tumeh>
```

10- update the person that has your info (based on your student Id) , change average instead from 100 scale to 4 by dividing it by 25

db.nameCollection.updateMany({ studentID: yourID }, [{ $set: { average: { $divide: ["$average", 25] } } }])

```
]
...   { studentID: 202010247 },
...   [
...     {
...       $set: {
...         average: { $divide: ["$average", 25] }
...       }
...     }
...   ]
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

To show the answer :

```
}
noor-tumeh> db.noorCollection.find()
[
  {
    _id: ObjectId('663b959fd56c77df4c46b79e'),
    name: 'Noor Aldeen',
    universityName: 'PTUK',
    registerationYear: 2020,
    studentID: 202010247,
    average: 3.32
  },
  {
```