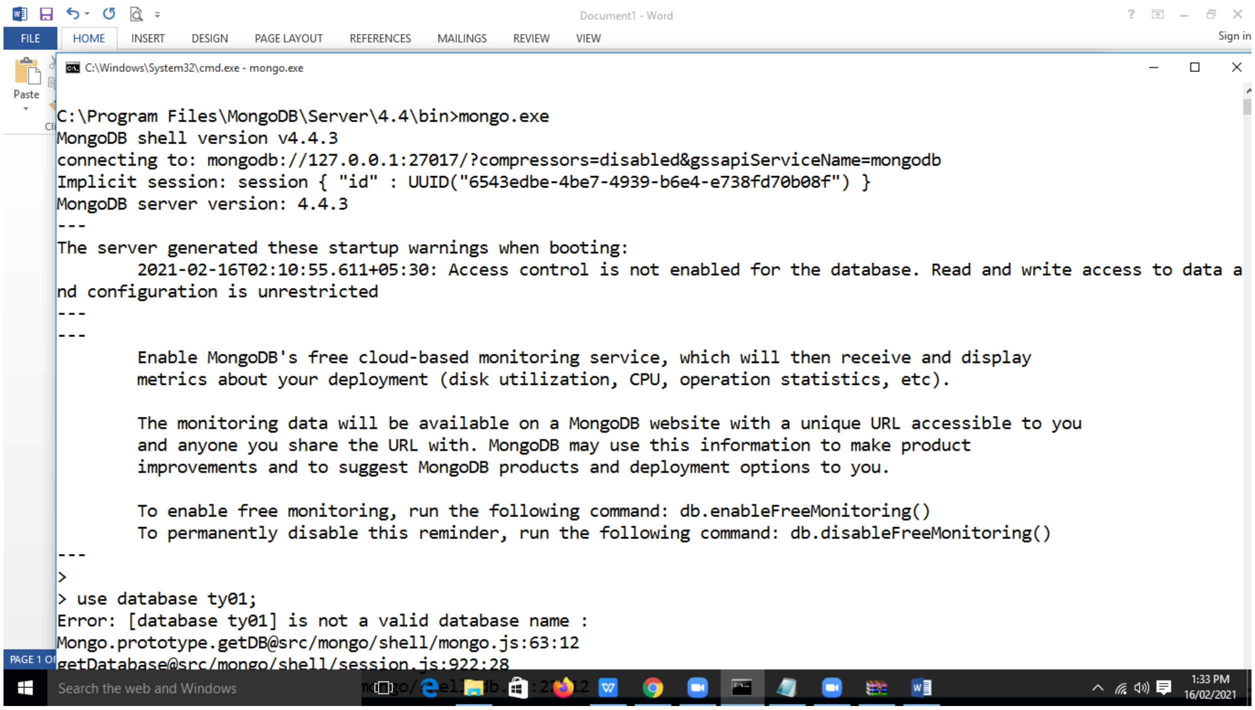
in cmd try:

C:\Program Files\MongoDB\Server\4.4\bin>mongo.exe



```
C:\Program Files\MongoDB\Server\4.4\bin>mongo.exe
MongoDB shell version v4.4.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("6543edbe-4be7-4939-b6e4-e738fd70b08f") }
MongoDB server version: 4.4.3
---
The server generated these startup warnings when booting:
        2021-02-16T02:10:55.611+05:30: Access control is not enabled for the database. Read and write access to data a
nd configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
>
> use database ty01;
Error: [database ty01] is not a valid database name :
Mongo.prototype.getDB@src/mongo/shell/mongo.js:63:12
getDatabase@src/mongo/shell/session.js:922:28
```

Commands :

Create database “inventory”, & create collection “products” & add 5 product documents in it.

For collection “products”, details like (productno, description, company, price) need to be

maintained.

2) List products of “Samsung”.

3) List products in ascending/descending order of their price.

4) List products whose price is more than 15000.

5) Update price of product “p1” to 10000.

```
use inventory;

db.product.insert({"productno":"p1","description":"earphone","company":"onepl
us","price":

8000});

db.product.insert({"productno":"p2","description":"desktop","company":"apple",
"price":120000});


db.product.insert({"productno":"p3","description":"smartwatch","company":"sa
msung","price":17000});


db.product.insert({"productno":"p4","description":"speaker","company":"jbl","pri
ce":13000});


db.product.insert({"productno":"p5","description":"smartphone","company":"sa
msung","price":14000});


 db.product.find();

 db.product.find({productno:"p3"});

db.product.find({company:"samsung"});

db.product.find().sort({"price":1}).pretty();

db.product.find().sort({"price":-1}).pretty();

db.product.find({"price":{$gt:15000}}).pretty();

  db.result.find({"marks":{$gt:500}}).pretty();
 db.product.update(
{"seatno": "202"},
{$set:{"price":700}}
);
db.product.find({"productno":"p1"}).pretty();

db.dropDatabase()
```

Create database “exam” & create collection “result” to store student marks. For collection

“result”, details like (seatno, name, class, marks, grade) need to be maintained. Grades are (F,

D, C, B, A, O)

2) List all student details.

3) List students who have scored more than 500 marks.

4) List students of tycs in ascending order of marks.

5) Update marks of seatno: 202 to 750.

Insert in bluck :

```
db.inventory.insertMany(
[

  { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" },status: "A" },
  { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },

   { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },

  { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },

  { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
]);
```

---

Todays practice:

```
C:\Program Files\MongoDB\Server\4.4\bin>dir
 Volume in drive C has no label.
 Volume Serial Number is 14FA-3C5C


 Directory of C:\Program Files\MongoDB\Server\4.4\bin


16/02/2021  02:06 AM    <DIR>          .
16/02/2021  02:06 AM    <DIR>          ..
22/12/2020  12:29 AM             1,558 InstallCompass.ps1
21/12/2020  11:59 PM        21,613,056 mongo.exe
16/02/2021  02:10 AM               616 mongod.cfg
22/12/2020  12:24 AM        38,308,864 mongod.exe
22/12/2020  12:24 AM       387,690,496 mongod.pdb
```

```
21/12/2020  11:58 PM        27,293,184 mongos.exe
21/12/2020  11:58 PM       261,132,288 mongos.pdb
               7 File(s)    736,040,062 bytes
               2 Dir(s)  96,549,498,880 bytes free

C:\Program Files\MongoDB\Server\4.4\bin>mongo.exe
MongoDB shell version v4.4.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("6543edbe-4be7-4939-b6e4-e738fd70b08f") }
MongoDB server version: 4.4.3
---
The server generated these startup warnings when booting:
        2021-02-16T02:10:55.611+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to
make product
        improvements and to suggest MongoDB products and deployment options to
you.

        To enable free monitoring, run the following command:
db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command:
db.disableFreeMonitoring()
---
>
> use database ty01;
Error: [database ty01] is not a valid database name :
Mongo.prototype.getDB@src/mongo/shell/mongo.js:63:12
getDatabase@src/mongo/shell/session.js:922:28
DB.prototype.getSiblingDB@src/mongo/shell/db.js:22:12
shellHelper.use@src/mongo/shell/utils.js:832:10
shellHelper@src/mongo/shell/utils.js:819:15
@(shellhelp2):1:1
> use ty01;
switched to db ty01
> db
ty01
> db.student.insert({'stdid':'01','stname':'Roshal'});
WriteResult({ "nInserted" : 1 })
> db.student.find();
```

```
{ "_id" : ObjectId("602b712e7455d8e8ef6117f9"), "stdid" : "01", "stname" :
"Roshal" }
> db.student.find().pretty();
{
        "_id" : ObjectId("602b712e7455d8e8ef6117f9"),
        "stdid" : "01",
        "stname" : "Roshal"
}
> db.student.insert({'stdid':'02','stname':'Zeeshan'});
WriteResult({ "nInserted" : 1 })
> db.student.find().pretty();
{
        "_id" : ObjectId("602b712e7455d8e8ef6117f9"),
        "stdid" : "01",
        "stname" : "Roshal"
}
{
        "_id" : ObjectId("602b71a67455d8e8ef6117fa"),
        "stdid" : "02",
        "stname" : "Zeeshan"
}
> db.student.insert({'stdid':'03','stname':'Santosh','telno':98200200});
WriteResult({ "nInserted" : 1 })
> db.student.insert({'stdid':'04','stname':'Maya','telno':98200201,'age':20});
WriteResult({ "nInserted" : 1 })
```

```
> db.student.find().pretty();
{
        "_id" : ObjectId("602b712e7455d8e8ef6117f9"),
        "stdid" : "01",
        "stname" : "Roshal"
}
{
        "_id" : ObjectId("602b71a67455d8e8ef6117fa"),
        "stdid" : "02",
        "stname" : "Zeeshan"
}
{
        "_id" : ObjectId("602b71ea7455d8e8ef6117fb"),
        "stdid" : "03",
        "stname" : "Santosh",
        "telno" : 98200200
}
{
        "_id" : ObjectId("602b720d7455d8e8ef6117fc"),
        "stdid" : "04",
        "stname" : "Maya",
        "telno" : 98200201,
        "age" : 20
}
```

```
> db.student.find({'stdid':'03'}).pretty();
{
        "_id" : ObjectId("602b71ea7455d8e8ef6117fb"),
        "stdid" : "03",
        "stname" : "Santosh",
        "telno" : 98200200
}
> db.student.find({'stname':'Santosh'}).pretty();
{
        "_id" : ObjectId("602b71ea7455d8e8ef6117fb"),
        "stdid" : "03",
        "stname" : "Santosh",
        "telno" : 98200200
}
> db.student.find({'stdid':-1}).pretty();
> db.student.find({'stdid':-1}).pretty();
> db.student.find().sort({'stdid':-1}).pretty();
{
        "_id" : ObjectId("602b720d7455d8e8ef6117fc"),
        "stdid" : "04",
        "stname" : "Maya",
        "telno" : 98200201,
        "age" : 20
}
```

```
{
        "_id" : ObjectId("602b71ea7455d8e8ef6117fb"),
        "stdid" : "03",
        "stname" : "Santosh",
        "telno" : 98200200
}
{
        "_id" : ObjectId("602b71a67455d8e8ef6117fa"),
        "stdid" : "02",
        "stname" : "Zeeshan"
}
{
        "_id" : ObjectId("602b712e7455d8e8ef6117f9"),
        "stdid" : "01",
        "stname" : "Roshal"
}
> db.student.find().sort({'stdid':1}).pretty();
{
        "_id" : ObjectId("602b712e7455d8e8ef6117f9"),
        "stdid" : "01",
        "stname" : "Roshal"
}
{
        "_id" : ObjectId("602b71a67455d8e8ef6117fa"),
        "stdid" : "02",
        "stname" : "Zeeshan"
}
{
        "_id" : ObjectId("602b71ea7455d8e8ef6117fb"),
        "stdid" : "03",
        "stname" : "Santosh",
        "telno" : 98200200
}
{
        "_id" : ObjectId("602b720d7455d8e8ef6117fc"),
        "stdid" : "04",
        "stname" : "Maya",
        "telno" : 98200201,
        "age" : 20
}
> db.student.find().sort({'stname':1}).pretty();
{
        "_id" : ObjectId("602b720d7455d8e8ef6117fc"),
        "stdid" : "04",
        "stname" : "Maya",
        "telno" : 98200201,
        "age" : 20
}
```

```
{
        "_id" : ObjectId("602b712e7455d8e8ef6117f9"),
        "stdid" : "01",
        "stname" : "Roshal"
}
{
        "_id" : ObjectId("602b71ea7455d8e8ef6117fb"),
        "stdid" : "03",
        "stname" : "Santosh",
        "telno" : 98200200
}
{
        "_id" : ObjectId("602b71a67455d8e8ef6117fa"),
        "stdid" : "02",
        "stname" : "Zeeshan"
}
> db.student.find().sort({'stname':-1}).pretty();
{
        "_id" : ObjectId("602b71a67455d8e8ef6117fa"),
        "stdid" : "02",
        "stname" : "Zeeshan"
}
{
        "_id" : ObjectId("602b71ea7455d8e8ef6117fb"),
        "stdid" : "03",
        "stname" : "Santosh",
        "telno" : 98200200
}
{
        "_id" : ObjectId("602b712e7455d8e8ef6117f9"),
        "stdid" : "01",
        "stname" : "Roshal"
}
{
        "_id" : ObjectId("602b720d7455d8e8ef6117fc"),
        "stdid" : "04",
        "stname" : "Maya",
        "telno" : 98200201,
        "age" : 20
}
> db.student.find({"age":{$gt:18}}).pretty();
{
        "_id" : ObjectId("602b720d7455d8e8ef6117fc"),
        "stdid" : "04",
        "stname" : "Maya",
        "telno" : 98200201,
        "age" : 20
}
```

```
> db.student.find({"age":{$gt:20}}).pretty();
> db.student.find({"age":{$ge:20}}).pretty();
Error: error: {
        "ok" : 0,
        "errmsg" : "unknown operator: $ge",
        "code" : 2,
        "codeName" : "BadValue"
}
> db.student.find({"age":{$gt:19}}).pretty();
{
        "_id" : ObjectId("602b720d7455d8e8ef6117fc"),
        "stdid" : "04",
        "stname" : "Maya",
        "telno" : 98200201,
        "age" : 20
}
> db.student.find({"age":{$ge:19}}).pretty();
Error: error: {
        "ok" : 0,
        "errmsg" : "unknown operator: $ge",
        "code" : 2,
        "codeName" : "BadValue"
}
> db.student.update({"stdid": "01"},{$set:{"stname":"Roshal Singh"}});
```

```
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.student.find().pretty();
{
        "_id" : ObjectId("602b712e7455d8e8ef6117f9"),
        "stdid" : "01",
        "stname" : "Roshal Singh"
}
{
        "_id" : ObjectId("602b71a67455d8e8ef6117fa"),
        "stdid" : "02",
        "stname" : "Zeeshan"
}
{
        "_id" : ObjectId("602b71ea7455d8e8ef6117fb"),
        "stdid" : "03",
        "stname" : "Santosh",
        "telno" : 98200200
}
{
        "_id" : ObjectId("602b720d7455d8e8ef6117fc"),
        "stdid" : "04",
        "stname" : "Maya",
        "telno" : 98200201,
        "age" : 20
}

> use inventory;

switched to db inventory

>
db.product.insert({"productno":"p1","description":"earphone","company":"onepl
us","price":

... 8000});

WriteResult({ "nInserted" : 1 })
```