LABORATORY PROGRAM -1
MONGO DB CRUD OPERATIONS

I. CREATE DATABASE IN MONGODB.

```
mongosh
use myDB;
db;
show dbs;
```

II. CRUD (CREATE, READ, UPDATE, DELETE) OPERATIONS

```
db.createCollection(“Student”);
db.Student.insert({id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfi
ng"});
db.Student.update({id:3,StudName:"AryanDavid",Grade:"VII"},{$set:{Hobbies:"Skating"}}
,{upsert:true});
db.Student.find({StudName:"Aryan David"});
db.Student.find({},{StudName:1,Grade:1,_id:0});
db.Student.find({Grade:{$eq:'VII'}}).pretty();
db.Student.find({Hobbies :{ $in: ['Chess','Skating']}}).pretty ();
db.Student.find({StudName:/^M/}).pretty();
db.Student.find({StudName:/e/}).pretty();
```

III.

```
db.Student.count();
```

IV.

```
db.Student.find().sort({StudName:-1}).pretty();
db.Student.drop();
```

V.

```
db.Students.insert({StudName:”Vamsi”, Grade:”VI”})
```

VI.

```
db.Students.update({_id:4},{$set:{Location:”Network”}})
```

VII.

```
db.Students.update({_id:4},{$unset:{Location:”Network”}})
```

VIII.

```
db.Student.find({_id:1},{StudName:1,Grade:1,_id:0});
db.Student.find({Grade:{$ne:&amp;#39;VII&amp;#39;}}).pretty();
db.Student.find({StudName:/s$/}).pretty();
```

IX.

```
db.Students.update({_id:3},{$set:{Location:null}})
```

X.

```
db.Students.count()
```

XI.

```
db.Students.count({Grade:”VII”})
db.Students.find({Grade:”VII”}).limit(3).pretty();
```

db.Students.find().sort({StudName:1}).pretty();
db.Students.find().skip(2).pretty()

XII.

db.food.insert({ id:1, fruits:['grapes','mango','apple'] })
db.food.insert({ _id:2, fruits:['grapes','mango','cherry'] })
db.food.insert({ _id:3, fruits:['banana','mango'] })
To find in “fruits” array having “mango” in the first index position.

db.food.find ({'fruits.1':'grapes'})
To find those documents from the “food” collection where the size of the array is two.
db.food.find ({“fruits”: {$size:2}})
To find the document with a particular id and display the first two elements from the array “fruits”
db.food.find({_id:1},{“fruits”:{$slice:2}})
To find all the documets from the food collection which have elements mango and grapes in the array “fruits”
db.food.find({fruits:{$all:[“mango”,”grapes”]}})
update on Array:
using particular id replace the element present in the 1 st index position of the fruits array with apple

db.food.update({_id:3},{$set:{'fruits.1':'apple'}})
insert new key value pairs in the fruits array
db.food.update({_id:2},{$push:{price:{grapes:80,mango:200,cherry:100}}})

```
test> use myDB;
switched to db myDB
myDB> db;
myDB
myDB> show dbs;
admin      40.00 KiB
config     60.00 KiB
local     128.00 KiB
products   72.00 KiB
myDB> db.createCollection("Student");
{ ok: 1 }
myDB> db.Student.inset({id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfing"});
TypeError: db.Student.inset is not a function
myDB>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
myDB> db.Student.insert({id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfing"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67cffa11805b3fcd68e6a3b7') }
}
myDB> db.Student.update({id:3,StudName:"AryanDavid",Grade:"VII"},{$set:{Hobbies:"Skating"}},{upsert:true});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: ObjectId('67cffbaf7e2cb31b9f820cab'),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
myDB> db.Student.find({StudName:"AryanDavid"});
[
  {
    _id: ObjectId('67cffbaf7e2cb31b9f820cab'),
    Grade: 'VII',
    StudName: 'AryanDavid',
    id: 3,
    Hobbies: 'Skating'
  }
]
myDB> db.Student.find({},{StudName:1,Grade:1,id:0});
MongoServerError[Location31254]: Cannot do exclusion on field id in inclusion projection
myDB> db.Student.find({Grade:{$eq:'VII'}}).pretty();
[
  {
    _id: ObjectId('67cffa11805b3fcd68e6a3b7'),
    id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  {
    _id: ObjectId('67cffbaf7e2cb31b9f820cab'),
    Grade: 'VII',
    StudName: 'AryanDavid',
    id: 3,
    Hobbies: 'Skating'
  }
]
```

```
myDB> db.Student.find({Hobbies:{$in:['Chess','Skating']}}).pretty();
[
  {
    _id: ObjectId('67cffbaf7e2cb31b9f820cab'),
    Grade: 'VII',
    StudName: 'AryanDavid',
    id: 3,
    Hobbies: 'Skating'
  }
]
myDB> db.Student.find({StudName:/^M/}).pretty()
[
  {
    _id: ObjectId('67cffa11805b3fcd68e6a3b7'),
    id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  }
]
myDB> db.Student.find({StudName:/e/}).pretty()
[
  {
    _id: ObjectId('67cffa11805b3fcd68e6a3b7'),
    id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  }
]
myDB> db.Student.count()
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
2
myDB> db.Student.find().sort({StudName:-1}).pretty()
[
  {
    _id: ObjectId('67cffa11805b3fcd68e6a3b7'),
    id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  {
    _id: ObjectId('67cffbaf7e2cb31b9f820cab'),
    Grade: 'VII',
    StudName: 'AryanDavid',
    id: 3,
    Hobbies: 'Skating'
  }
]
myDB> db.Students.save({StudName:"Vamsi",Grade:"VI"})
TypeError: db.Students.save is not a function
myDB> db.Student.save({StudName:"Vamsi",Grade:"VI"})
TypeError: db.Student.save is not a function
myDB>
(To exit, press Ctrl+C again or Ctrl+D or type .exit)
myDB> db.Students.insert({StudName:"Vamsi",Grade:"VI"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67d000c1805b3fcd68e6a3b8') }
}
```

```
myDB> db.Student.insert({StudName:"Vamsi",Grade:"VI"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67d00115805b3fcd68e6a3b9') }
}
myDB> db.Student.update({id:4},{$set:{Location:"Network"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
myDB> db.Student.update({id:4},{$unset:{Location:"Network"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
myDB> db.Student.find({id:1},{StudName:1,Grade:1,id:0})
MongoServerError[Location31254]: Cannot do exclusion on field id in inclusion projection
myDB> db.Student.find({id:1},{StudName:1,Grade:0,id:1})
MongoServerError[Location31254]: Cannot do exclusion on field Grade in inclusion projection
myDB> db.Student.find({Grade:{$ne:'VI'}}).pretty();
[
  {
    _id: ObjectId('67cffa11805b3fcd68e6a3b7'),
    id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  {
    _id: ObjectId('67cffbaf7e2cb31b9f820cab'),
    Grade: 'VII',
    StudName: 'AryanDavid',
    id: 3,
    Hobbies: 'Skating'
  }
]
myDB> db.Student.find({StudName:/s$/}).pretty()

myDB> db.Student.update({id:3},{$set:{Location:null}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
myDB> db.Student.count()
3
myDB> db.Student.count({Grade:"VII"})
2
```

```
myDB> db.Student.find({Grade:"VII"}).limit(3).pretty()
[
  {
    _id: ObjectId('67cffa11805b3fcd68e6a3b7'),
    id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  {
    _id: ObjectId('67cffbaf7e2cb31b9f820cab'),
    Grade: 'VII',
    StudName: 'AryanDavid',
    id: 3,
    Hobbies: 'Skating',
    Location: null
  }
]
myDB> db.Student.find().sort({StudName:1}).pretty()
[
  {
    _id: ObjectId('67cffbaf7e2cb31b9f820cab'),
    Grade: 'VII',
    StudName: 'AryanDavid',
    id: 3,
    Hobbies: 'Skating',
    Location: null
  },
  {
    _id: ObjectId('67cffa11805b3fcd68e6a3b7'),
    id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  {
    _id: ObjectId('67d00115805b3fcd68e6a3b9'),
    StudName: 'Vamsi',
    Grade: 'VI'
  }
]
myDB> db.Student.find().skip(2).pretty()
[
  {
    _id: ObjectId('67d00115805b3fcd68e6a3b9'),
    StudName: 'Vamsi',
    Grade: 'VI'
  }
]
myDB> db.createCollection("Food");
{ ok: 1 }
myDB> db.food.insert({id:1,fruits:['grapes','mango','apple']})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67d004af805b3fcd68e6a3ba') }
}
myDB> db.food.insert({id:2,fruits:['grapes','mango','cherry']})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67d004d6805b3fcd68e6a3bb') }
}
```

```
myDB> db.food.insert({id:3,fruits:['banana','mango']})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('67d004f3805b3fcd68e6a3bc') }
}
myDB> db.food.find({fruits:['grapes','mango','apple']}).pretty()
[
  {
    _id: ObjectId('67d004af805b3fcd68e6a3ba'),
    id: 1,
    fruits: [ 'grapes', 'mango', 'apple' ]
  }
]
myDB> db.food.find({'fruits.1':'grapes'})

myDB> db.food.find({"fruits":{$size:2}})
[
  {
    _id: ObjectId('67d004f3805b3fcd68e6a3bc'),
    id: 3,
    fruits: [ 'banana', 'mango' ]
  }
]
myDB> db.food.find({id:1},{"fruits":{$slice:2}})
[
  {
    _id: ObjectId('67d004af805b3fcd68e6a3ba'),
    id: 1,
    fruits: [ 'grapes', 'mango' ]
  }
]
myDB> db.food.find({fruits:{$all:["mango","grapes"]}})
[
  {
    _id: ObjectId('67d004af805b3fcd68e6a3ba'),
    id: 1,
    fruits: [ 'grapes', 'mango', 'apple' ]
  },
  {
    _id: ObjectId('67d004d6805b3fcd68e6a3bb'),
    id: 2,
    fruits: [ 'grapes', 'mango', 'cherry' ]
  }
]
myDB> db.food.update({id:2},{$push:{price:{grapes:80,mango:200,cherry:100}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```