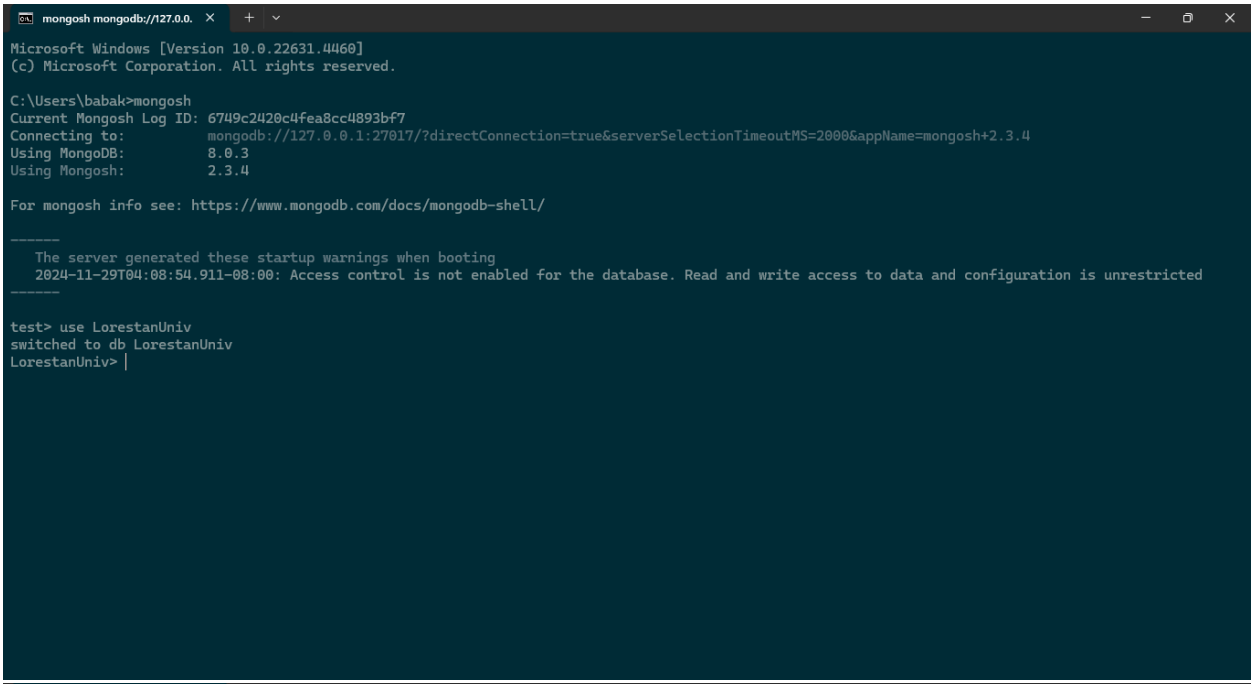
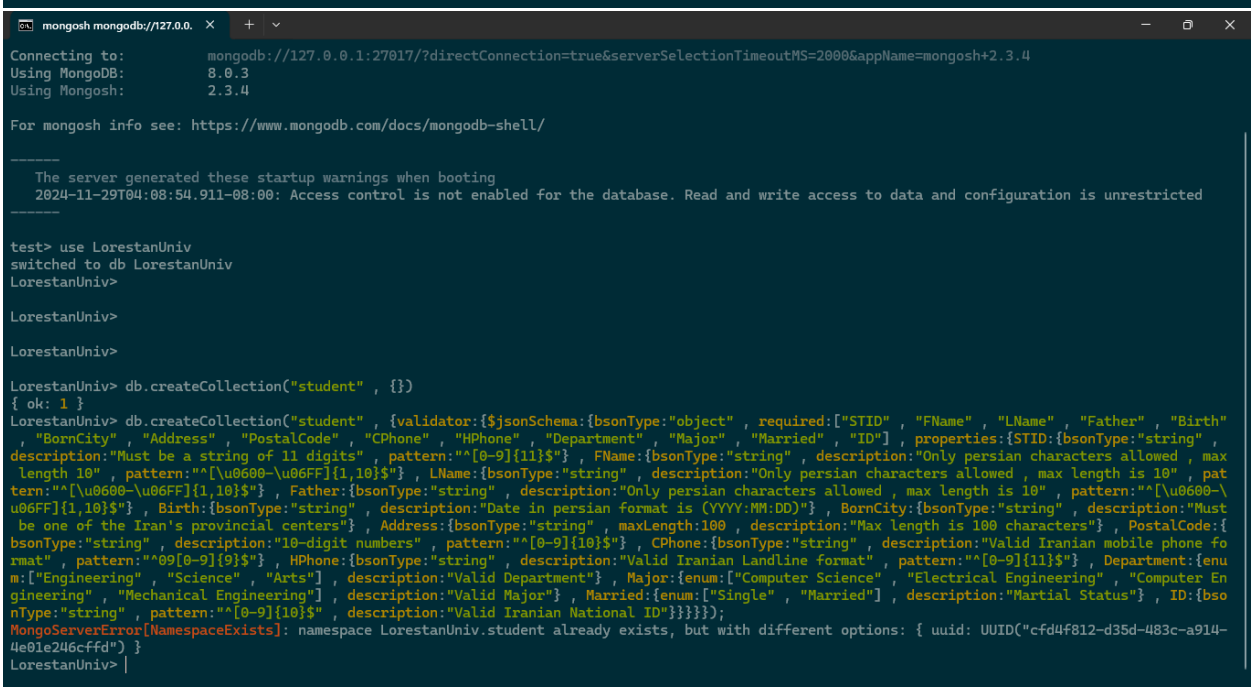
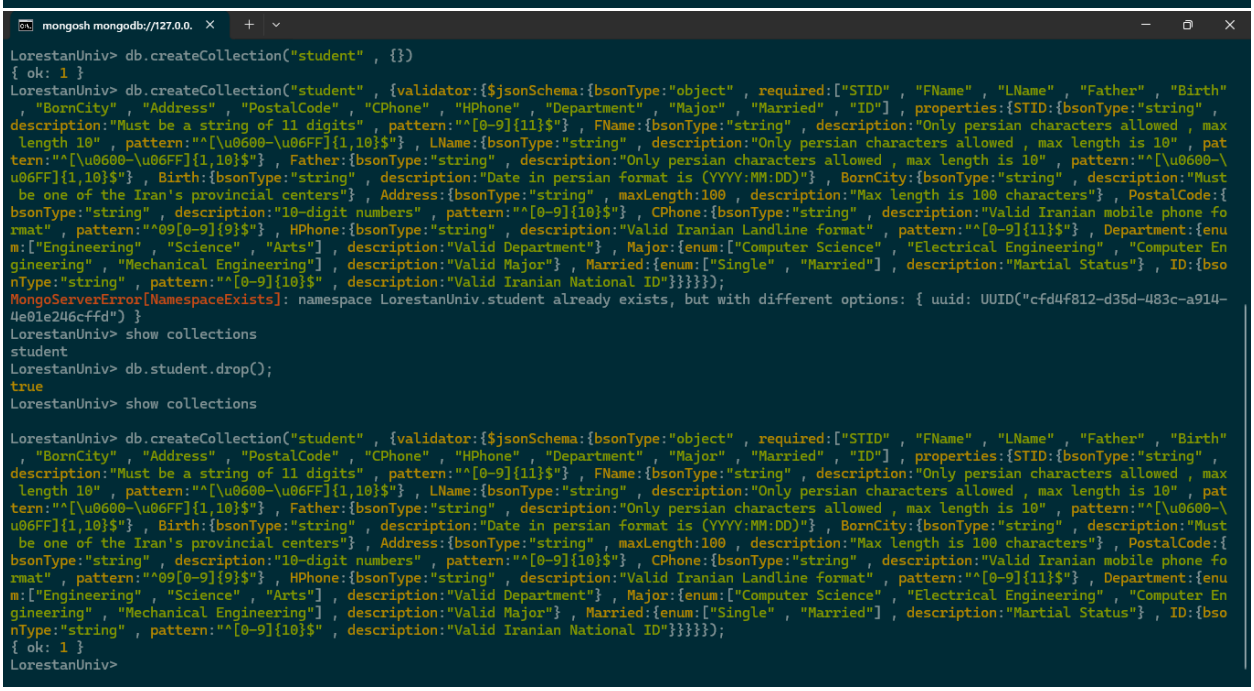
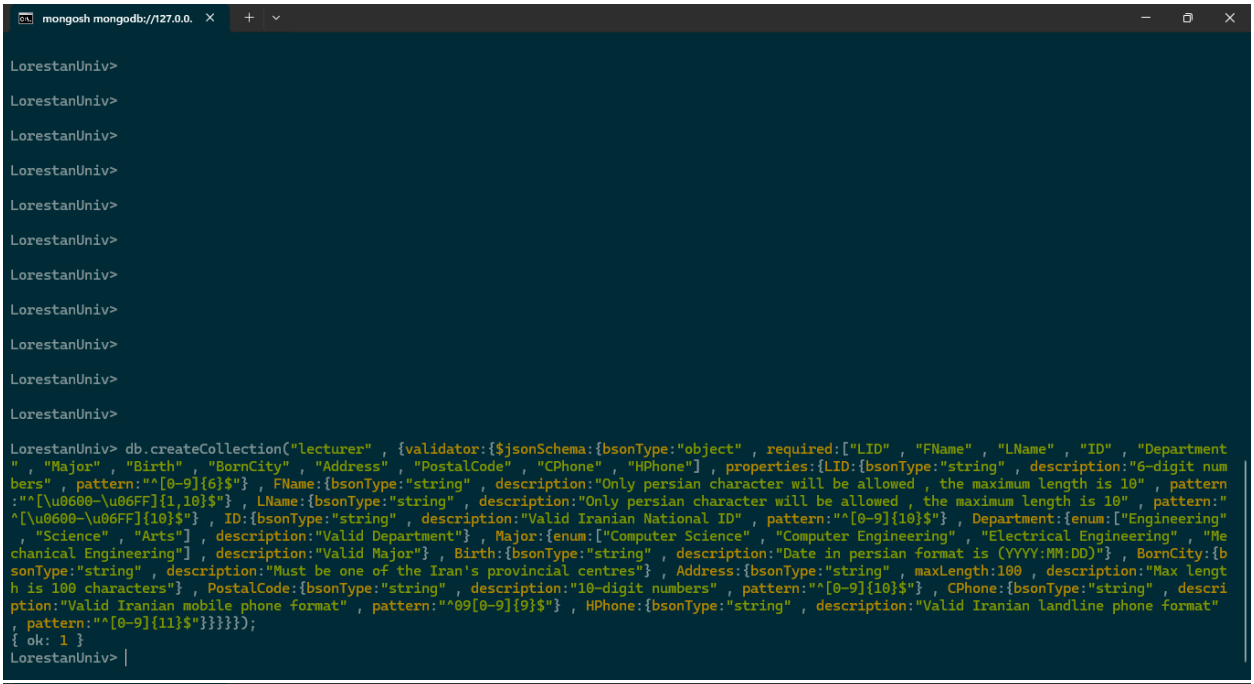

mongosh mongodb://127.0.0.
Microsoft Windows [Version 10.0.22631.4460]
(c) Microsoft Corporation. All rights reserved.
C:\Users\babak>mongosh
Current Mongosh Log ID: 6749c2420c4fea8cc4893bf7
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.4
Using MongoDB: 8.0.3
Using Mongosh: 2.3.4
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
The server generated these startup warnings when booting
2024-11-29T04:08:54.911-08:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
test> use LorestanUniv
switched to db LorestanUniv
LorestanUniv>


mongosh mongodb://127.0.0.
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.4
Using MongoDB: 8.0.3
Using Mongosh: 2.3.4
For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/
The server generated these startup warnings when booting
2024-11-29T04:08:54.911-08:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
test> use LorestanUniv
switched to db LorestanUniv
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv> db.createCollection("student" , {})
{ ok: 1 }
LorestanUniv> db.createCollection("student" , {validator:{$jsonSchema:{bsonType:"object" , required:["STID" , "FName" , "LName" , "Father" , "Birth" , "BornCity" , "Address" , "PostalCode" , "CPhone" , "HPhone" , "Department" , "Major" , "Married" , "ID"] , properties:{STID:{bsonType:"string" , description:"Must be a string of 11 digits" , pattern:"^[0-9]{11}$"} , FName:{bsonType:"string" , description:"Only persian characters allowed , max length 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , LName:{bsonType:"string" , description:"Only persian characters allowed , max length is 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , Father:{bsonType:"string" , description:"Only persian characters allowed , max length is 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , Birth:{bsonType:"string" , description:"Date in persian format is (YYYY:MM:DD)"} , BornCity:{bsonType:"string" , description:"Must be one of the Iran's provincial centers"} , Address:{bsonType:"string" , maxLength:100 , description:"Max length is 100 characters"} , PostalCode:{bsonType:"string" , description:"10-digit numbers" , pattern:"^[0-9]{10}$"} , CPhone:{bsonType:"string" , description:"Valid Iranian mobile phone format" , pattern:"^09[0-9]{9}$"} , HPhone:{bsonType:"string" , description:"Valid Iranian Landline format" , pattern:"^[0-9]{11}$"} , Department:{enum:["Engineering" , "Science" , "Arts"] , description:"Valid Department"} , Major:{enum:["Computer Science" , "Electrical Engineering" , "Computer Engineering" , "Mechanical Engineering"] , description:"Valid Major"} , Married:{enum:["Single" , "Married"] , description:"Martial Status"} , ID:{bsonType:"string" , pattern:"^[0-9]{10}$" , description:"Valid Iranian National ID"}}}});
MongoServerError[NamespaceExists]: namespace LorestanUniv.student already exists, but with different options: { uuid: UUID("cfd4f812-d35d-483c-a914-4e01e246cffd") }
LorestanUniv>

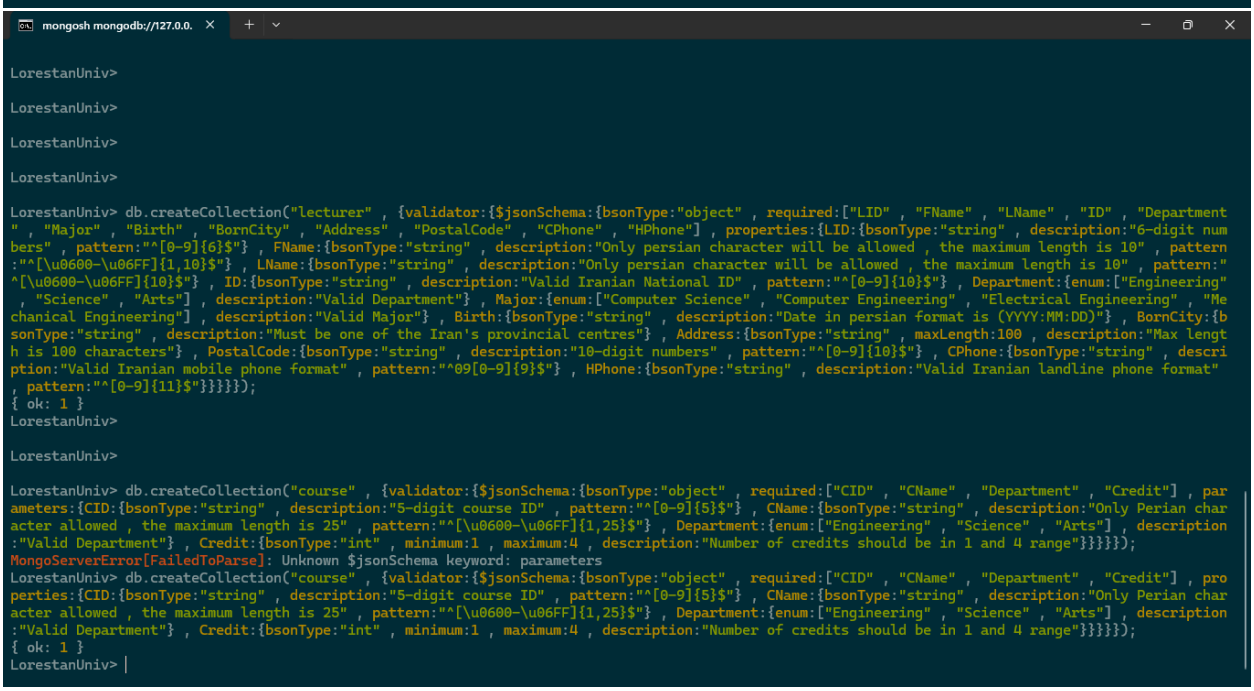

mongosh mongodb://127.0.0.

The server generated these startup warnings when booting
2024-11-29T04:08:54.911-08:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------
test> use LorestanUniv
switched to db LorestanUniv
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv> db.createCollection("student" , {})
{ ok: 1 }
LorestanUniv> db.createCollection("student" , {validator:{$jsonSchema:{bsonType:"object" , required:["STID" , "FName" , "LName" , "Father" , "Birth" , "BornCity" , "Address" , "PostalCode" , "CPhone" , "HPhone" , "Department" , "Major" , "Married" , "ID"] , properties:{STID:{bsonType:"string" , description:"Must be a string of 11 digits" , pattern:"^[0-9]{11}$"} , FName:{bsonType:"string" , description:"Only persian characters allowed , max length 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , LName:{bsonType:"string" , description:"Only persian characters allowed , max length is 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , Father:{bsonType:"string" , description:"Only persian characters allowed , max length is 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , Birth:{bsonType:"string" , description:"Date in persian format is (YYYY:MM:DD)"} , BornCity:{bsonType:"string" , description:"Must be one of the Iran's provincial centers"} , Address:{bsonType:"string" , maxLength:100 , description:"Max length is 100 characters"} , PostalCode:{bsonType:"string" , description:"10-digit numbers" , pattern:"^[0-9]{10}$"} , CPhone:{bsonType:"string" , description:"Valid Iranian mobile phone format" , pattern:"^09[0-9]{9}$"} , HPhone:{bsonType:"string" , description:"Valid Iranian Landline format" , pattern:"^[0-9]{11}$"} , Department:{enum:["Engineering" , "Science" , "Arts"] , description:"Valid Department"} , Major:{enum:["Computer Science" , "Electrical Engineering" , "Computer Engineering" , "Mechanical Engineering"] , description:"Valid Major"} , Married:{enum:["Single" , "Married"] , description:"Martial Status"} , ID:{bsonType:"string" , pattern:"^[0-9]{10}$" , description:"Valid Iranian National ID"}}}});
MongoServerError[NamespaceExists]: namespace LorestanUniv.student already exists, but with different options: { uuid: UUID("cfd4f812-d35d-483c-a914-4e01e246cffd") }
LorestanUniv> show collections
student
LorestanUniv> db.student.drop();
true
LorestanUniv> show collections
LorestanUniv>

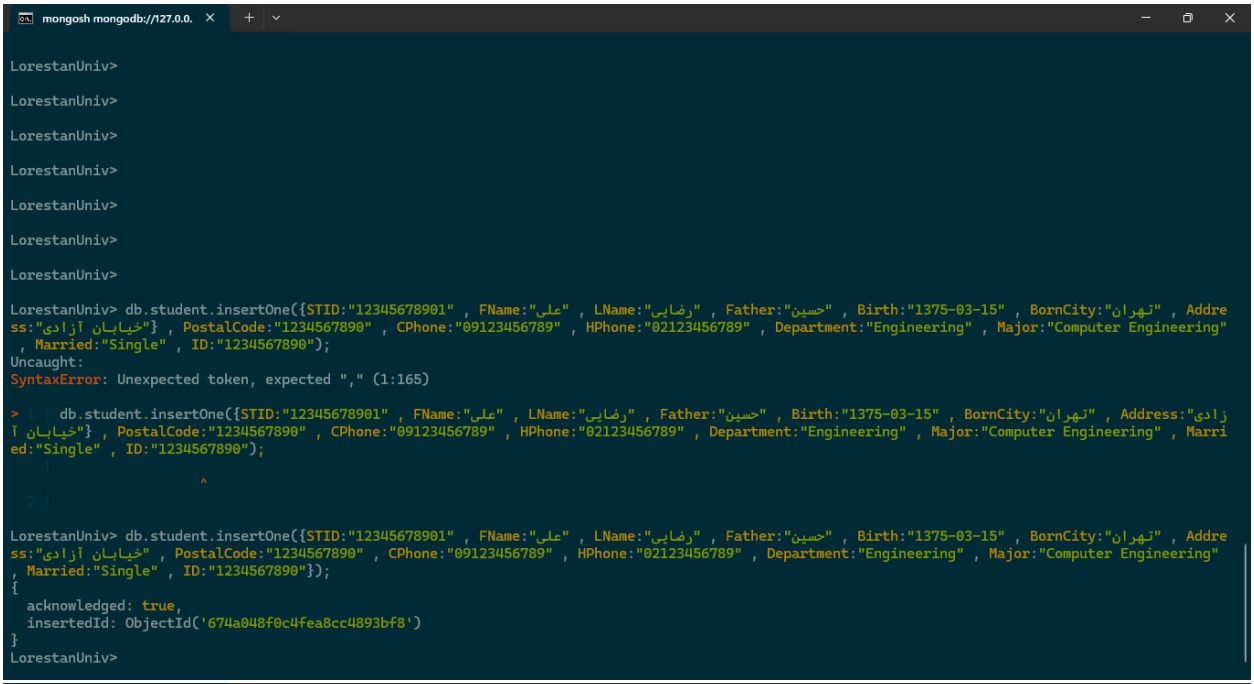

mongosh mongodb://127.0.0.
LorestanUniv> db.createCollection("student" , {})
{ ok: 1 }
LorestanUniv> db.createCollection("student" , {validator:{$jsonSchema:{bsonType:"object" , required:["STID" , "FName" , "LName" , "Father" , "Birth" , "BornCity" , "Address" , "PostalCode" , "CPhone" , "HPhone" , "Department" , "Major" , "Married" , "ID"] , properties:{STID:{bsonType:"string" , description:"Must be a string of 11 digits" , pattern:"^[0-9]{11}$"} , FName:{bsonType:"string" , description:"Only persian characters allowed , max length 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , LName:{bsonType:"string" , description:"Only persian characters allowed , max length is 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , Father:{bsonType:"string" , description:"Only persian characters allowed , max length is 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , Birth:{bsonType:"string" , description:"Date in persian format is (YYYY:MM:DD)"} , BornCity:{bsonType:"string" , description:"Must be one of the Iran's provincial centers"} , Address:{bsonType:"string" , maxLength:100 , description:"Max length is 100 characters"} , PostalCode:{bsonType:"string" , description:"10-digit numbers" , pattern:"^[0-9]{10}$"} , CPhone:{bsonType:"string" , description:"Valid Iranian mobile phone format" , pattern:"^09[0-9]{9}$"} , HPhone:{bsonType:"string" , description:"Valid Iranian Landline format" , pattern:"^[0-9]{11}$"} , Department:{enum:["Engineering" , "Science" , "Arts"] , description:"Valid Department"} , Major:{enum:["Computer Science" , "Electrical Engineering" , "Computer Engineering" , "Mechanical Engineering"] , description:"Valid Major"} , Married:{enum:["Single" , "Married"] , description:"Martial Status"} , ID:{bsonType:"string" , pattern:"^[0-9]{10}$" , description:"Valid Iranian National ID"}}}});
MongoServerError[NamespaceExists]: namespace LorestanUniv.student already exists, but with different options: { uuid: UUID("cfd4f812-d35d-483c-a914-4e01e246cffd") }
LorestanUniv> show collections
student
LorestanUniv> db.student.drop();
true
LorestanUniv> show collections
LorestanUniv> db.createCollection("student" , {validator:{$jsonSchema:{bsonType:"object" , required:["STID" , "FName" , "LName" , "Father" , "Birth" , "BornCity" , "Address" , "PostalCode" , "CPhone" , "HPhone" , "Department" , "Major" , "Married" , "ID"] , properties:{STID:{bsonType:"string" , description:"Must be a string of 11 digits" , pattern:"^[0-9]{11}$"} , FName:{bsonType:"string" , description:"Only persian characters allowed , max length 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , LName:{bsonType:"string" , description:"Only persian characters allowed , max length is 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , Father:{bsonType:"string" , description:"Only persian characters allowed , max length is 10" , pattern:"^[\u0600-\u06FF]{1,10}$"} , Birth:{bsonType:"string" , description:"Date in persian format is (YYYY:MM:DD)"} , BornCity:{bsonType:"string" , description:"Must be one of the Iran's provincial centers"} , Address:{bsonType:"string" , maxLength:100 , description:"Max length is 100 characters"} , PostalCode:{bsonType:"string" , description:"10-digit numbers" , pattern:"^[0-9]{10}$"} , CPhone:{bsonType:"string" , description:"Valid Iranian mobile phone format" , pattern:"^09[0-9]{9}$"} , HPhone:{bsonType:"string" , description:"Valid Iranian Landline format" , pattern:"^[0-9]{11}$"} , Department:{enum:["Engineering" , "Science" , "Arts"] , description:"Valid Department"} , Major:{enum:["Computer Science" , "Electrical Engineering" , "Computer Engineering" , "Mechanical Engineering"] , description:"Valid Major"} , Married:{enum:["Single" , "Married"] , description:"Martial Status"} , ID:{bsonType:"string" , pattern:"^[0-9]{10}$" , description:"Valid Iranian National ID"}}}});
{ ok: 1 }
LorestanUniv>



```
mongosh mongodb://127.0.0.

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv> db.createCollection("lecturer" , {validator:{$jsonSchema:{bsonType:"object" , required:["LID" , "FName" , "LName" , "ID" , "Department
" , "Major" , "Birth" , "BornCity" , "Address" , "PostalCode" , "CPhone" , "HPhone"] , properties:{LID:{bsonType:"string" , description:"6-digit num
bers" , pattern:"^[0-9]{6}$"} , FName:{bsonType:"string" , description:"Only persian character will be allowed , the maximum length is 10" , pattern
:"^[\u0600-\u06FF]{1,10}$"} , LName:{bsonType:"string" , description:"Only persian character will be allowed , the maximum length is 10" , pattern:"
^[\u0600-\u06FF]{10}$"} , ID:{bsonType:"string" , description:"Valid Iranian National ID" , pattern:"^[0-9]{10}$"} , Department:{enum:["Engineering"
 , "Science" , "Arts"] , description:"Valid Department"} , Major:{enum:["Computer Science" , "Computer Engineering" , "Electrical Engineering" , "Me
chanical Engineering"] , description:"Valid Major"} , Birth:{bsonType:"string" , description:"Date in persian format is (YYYY:MM:DD)"} , BornCity:{b
sonType:"string" , description:"Must be one of the Iran's provincial centres"} , Address:{bsonType:"string" , maxLength:100 , description:"Max lengt
h is 100 characters"} , PostalCode:{bsonType:"string" , description:"10-digit numbers" , pattern:"^[0-9]{10}$"} , CPhone:{bsonType:"string" , descri
ption:"Valid Iranian mobile phone format" , pattern:"^09[0-9]{9}$"} , HPhone:{bsonType:"string" , description:"Valid Iranian landline phone format"
, pattern:"^[0-9]{11}$"}}}}});
{ ok: 1 }
LorestanUniv>
```



```
mongosh mongodb://127.0.0.

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv> db.createCollection("lecturer" , {validator:{$jsonSchema:{bsonType:"object" , required:["LID" , "FName" , "LName" , "ID" , "Department
" , "Major" , "Birth" , "BornCity" , "Address" , "PostalCode" , "CPhone" , "HPhone"] , properties:{LID:{bsonType:"string" , description:"6-digit num
bers" , pattern:"^[0-9]{6}$"} , FName:{bsonType:"string" , description:"Only persian character will be allowed , the maximum length is 10" , pattern
:"^[\u0600-\u06FF]{1,10}$"} , LName:{bsonType:"string" , description:"Only persian character will be allowed , the maximum length is 10" , pattern:"
^[\u0600-\u06FF]{10}$"} , ID:{bsonType:"string" , description:"Valid Iranian National ID" , pattern:"^[0-9]{10}$"} , Department:{enum:["Engineering"
 , "Science" , "Arts"] , description:"Valid Department"} , Major:{enum:["Computer Science" , "Computer Engineering" , "Electrical Engineering" , "Me
chanical Engineering"] , description:"Valid Major"} , Birth:{bsonType:"string" , description:"Date in persian format is (YYYY:MM:DD)"} , BornCity:{b
sonType:"string" , description:"Must be one of the Iran's provincial centres"} , Address:{bsonType:"string" , maxLength:100 , description:"Max lengt
h is 100 characters"} , PostalCode:{bsonType:"string" , description:"10-digit numbers" , pattern:"^[0-9]{10}$"} , CPhone:{bsonType:"string" , descri
ption:"Valid Iranian mobile phone format" , pattern:"^09[0-9]{9}$"} , HPhone:{bsonType:"string" , description:"Valid Iranian landline phone format"
, pattern:"^[0-9]{11}$"}}}}});
{ ok: 1 }
LorestanUniv>

LorestanUniv>

LorestanUniv> db.createCollection("course" , {validator:{$jsonSchema:{bsonType:"object" , required:["CID" , "CName" , "Department" , "Credit"] , par
ameters:{CID:{bsonType:"string" , description:"5-digit course ID" , pattern:"^[0-9]{5}$"} , CName:{bsonType:"string" , description:"Only Perian char
acter allowed , the maximum length is 25" , pattern:"^[\u0600-\u06FF]{1,25}$"} , Department:{enum:["Engineering" , "Science" , "Arts"] , description
:"Valid Department"} , Credit:{bsonType:"int" , minimum:1 , maximum:4 , description:"Number of credits should be in 1 and 4 range"}}}});
MongoServerError[FailedToParse]: Unknown $jsonSchema keyword: parameters
LorestanUniv> db.createCollection("course" , {validator:{$jsonSchema:{bsonType:"object" , required:["CID" , "CName" , "Department" , "Credit"] , pro
perties:{CID:{bsonType:"string" , description:"5-digit course ID" , pattern:"^[0-9]{5}$"} , CName:{bsonType:"string" , description:"Only Perian char
acter allowed , the maximum length is 25" , pattern:"^[\u0600-\u06FF]{1,25}$"} , Department:{enum:["Engineering" , "Science" , "Arts"] , description
:"Valid Department"} , Credit:{bsonType:"int" , minimum:1 , maximum:4 , description:"Number of credits should be in 1 and 4 range"}}}});
{ ok: 1 }
LorestanUniv>
```

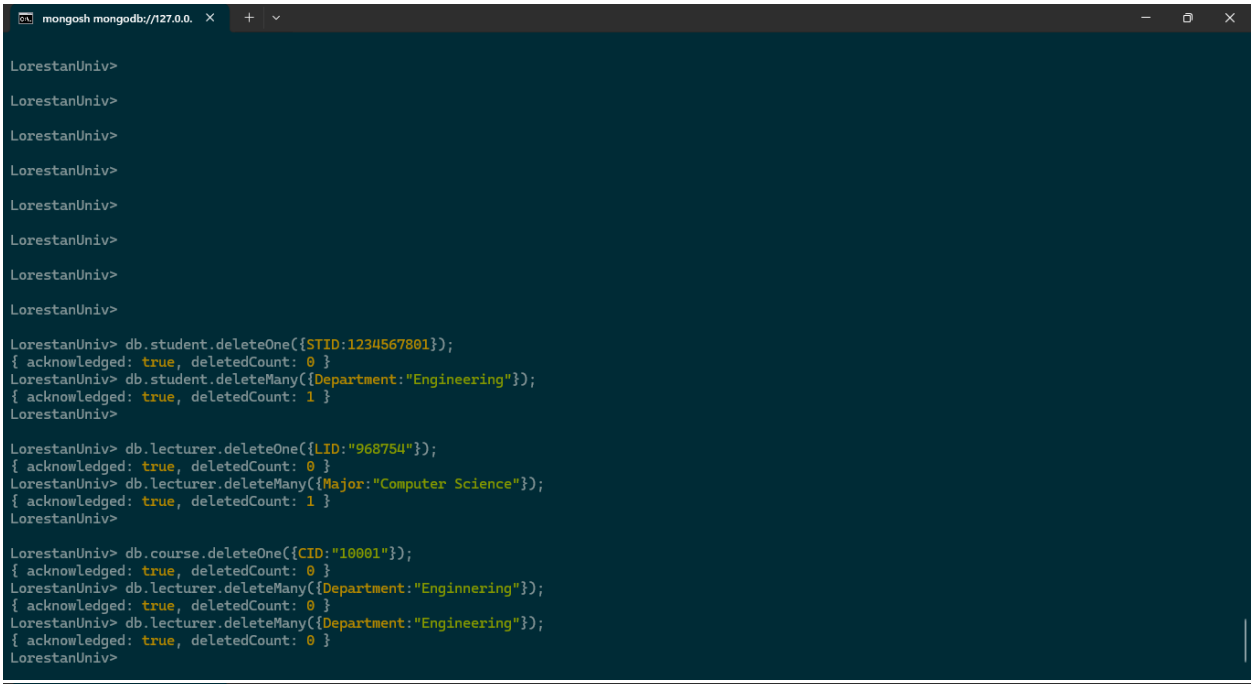


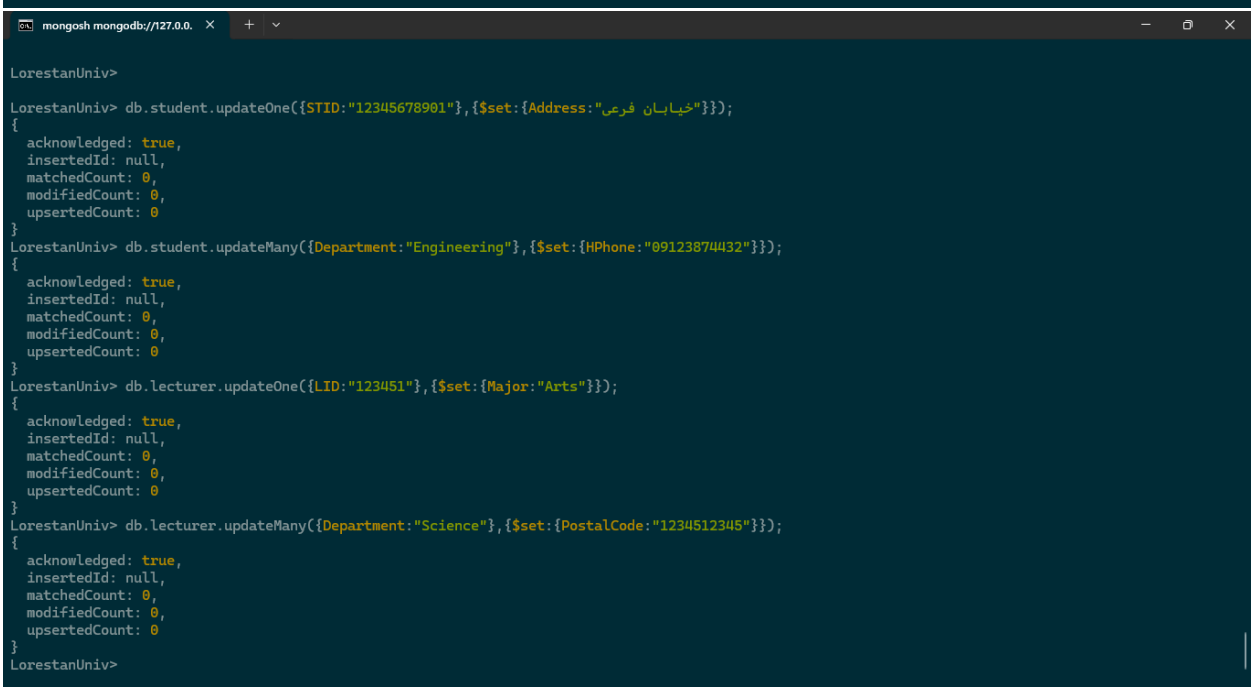
```
mongosh mongodb://127.0.0.

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv> db.student.insertOne({STID:"12345678901" , FName:"علی" , LName:"رضایی" , Father:"حسین" , Birth:"1375-03-15" , BornCity:"تهران" , Addre
ss:"خیابان آزادی"} , PostalCode:"1234567890" , CPhone:"09123456789" , HPhone:"02123456789" , Department:"Engineering" , Major:"Computer Engineering"
 , Married:"Single" , ID:"1234567890");
Uncaught:
SyntaxError: Unexpected token, expected "," (1:165)

> 1  db.student.insertOne({STID:"12345678901" , FName:"علی" , LName:"رضایی" , Father:"حسین" , Birth:"1375-03-15" , BornCity:"تهران" , Address:"زادی
  آ خیابان"} , PostalCode:"1234567890" , CPhone:"09123456789" , HPhone:"02123456789" , Department:"Engineering" , Major:"Computer Engineering" , Marri
ed:"Single" , ID:"1234567890");
    |                     ^
  2 |

LorestanUniv> db.student.insertOne({STID:"12345678901" , FName:"علی" , LName:"رضایی" , Father:"حسین" , Birth:"1375-03-15" , BornCity:"تهران" , Addre
ss:"خیابان آزادی" , PostalCode:"1234567890" , CPhone:"09123456789" , HPhone:"02123456789" , Department:"Engineering" , Major:"Computer Engineering"
, Married:"Single" , ID:"1234567890"});
{
  acknowledged: true,
  insertedId: ObjectId('674a048f0c4fea8cc4893bf8')
}
LorestanUniv>
```

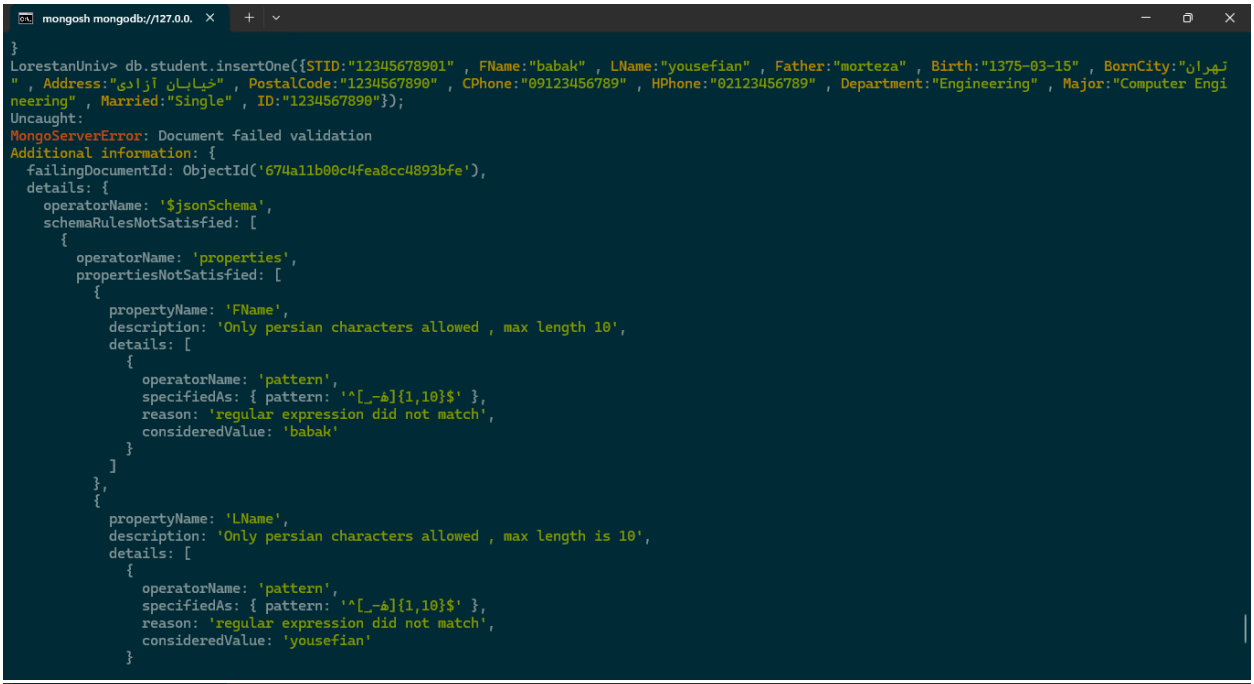
```
mongosh mongodb://127.0.0.

LorestanUniv> db.lecturer.insertOne({LID:"123456" , FName:"رضا" , LName:"احمدی" , ID:"1234567890" , Department:"Engineering" , Major:"Computer Scien
ce" , Birth:"1381-10-10" , BornCity:"اهواز" , Address:"خیابان اصلی" , PostalCode:"2934586712" , CPhone:"09129845761" , HPhone:"06612349876"});
Uncaught:
MongoServerError: Document failed validation
Additional information: {
  failingDocumentId: ObjectId('674a061b0c4fea8cc4893bf9'),
  details: {
    operatorName: '$jsonSchema',
    schemaRulesNotSatisfied: [
      {
        operatorName: 'properties',
        propertiesNotSatisfied: [
          {
            propertyName: 'LName',
            description: 'Only persian character will be allowed , the maximum length is 10',
            details: [
              {
                operatorName: 'pattern',
                specifiedAs: { pattern: '^[_-ه]{10}$' },
                reason: 'regular expression did not match',
                consideredValue: 'احمدی'
              }
            ]
          }
        ]
      }
    ]
  }
}
LorestanUniv> db.lecturer.insertOne({LID:"123456" , FName:"رضا" , LName:"احمدوندیان" , ID:"1234567890" , Department:"Engineering" , Major:"Computer
Science" , Birth:"1381-10-10" , BornCity:"اهواز" , Address:"خیابان اصلی" , PostalCode:"2934586712" , CPhone:"09129845761" , HPhone:"06612349876"});
{
  acknowledged: true,
  insertedId: ObjectId('674a080b0c4fea8cc4893bfa')
}
LorestanUniv>
```

```
mongosh mongodb://127.0.0.
LorestanUniv>

LorestanUniv> db.course.insertOne({CID:"19384" , CName:"هوش مصنوعی" , Department:"Engineering" , Credit:3});
Uncaught:
MongoServerError: Document failed validation
Additional information: {
  failingDocumentId: ObjectId('674a09410c4fea8cc4893bfb'),
  details: {
    operatorName: '$jsonSchema',
    schemaRulesNotSatisfied: [
      {
        operatorName: 'properties',
        propertiesNotSatisfied: [
          {
            propertyName: 'CName',
            description: 'Only Perian character allowed , the maximum length is 25',
            details: [
              {
                operatorName: 'pattern',
                specifiedAs: { pattern: '^[ـ-ۿ]{1,25}$' },
                reason: 'regular expression did not match',
                consideredValue: 'هوش مصنوعی'
              }
            ]
          }
        ]
      }
    ]
  }
}
LorestanUniv> db.course.insertOne({CID:"19384" , CName:"هوشمصنوعی" , Department:"Engineering" , Credit:3});
{
  acknowledged: true,
  insertedId: ObjectId('674a09d70c4fea8cc4893bfc')
}
LorestanUniv>
```

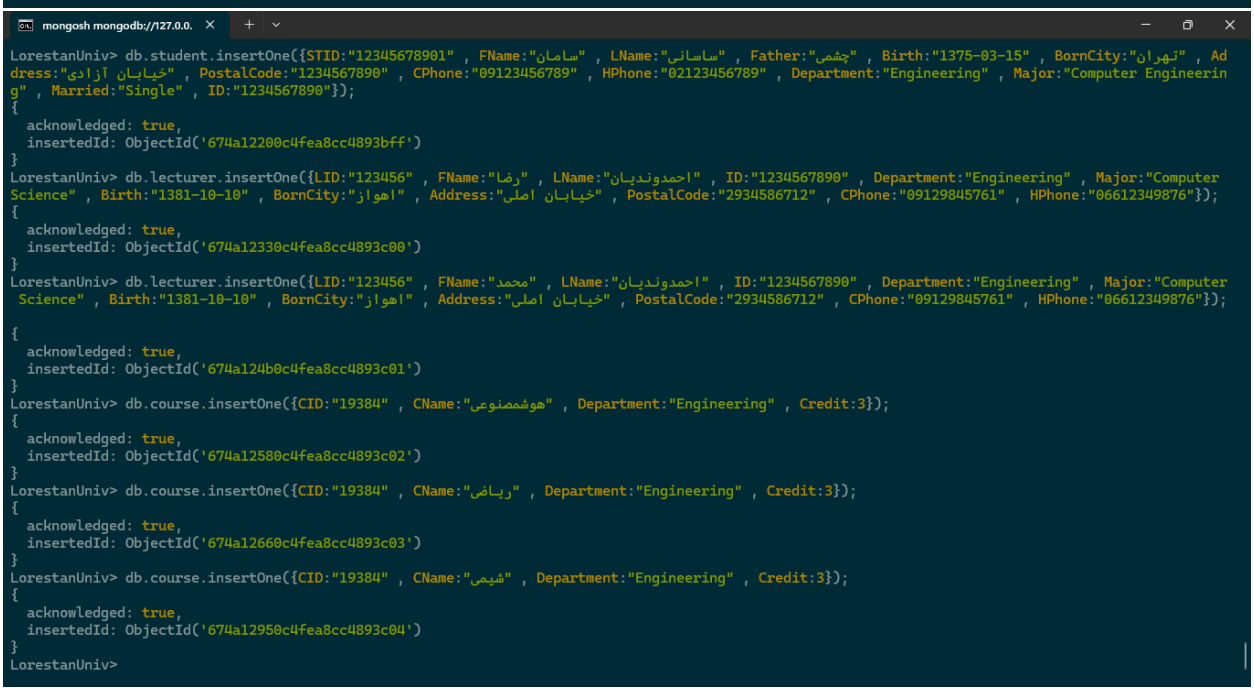
```
mongosh mongodb://127.0.0.

LorestanUniv> db.createView("CourseRegister" , "student" , [{$lookup:{from:"course" , localField:"STID" , foreignField:"CID" , as:"RegisteredCourses
"}},{$project:{SID:"$STID" , FName:1 , LName:1 , RegisteredCourses:{CID:1 , CName:1 , Departnemt:1 , Credit:1}}}]);
{ ok: 1 }
LorestanUniv>

LorestanUniv>

LorestanUniv> db.createView("PresentedCourses" , "lecturer" , [{$lookup:{from:"course" , localField:"LID" , foreignField:"CID" , as:"PresentedCourse
s"}},{$project:{LID:"$LID" , FName:1 , LName:1 , PresentedCourses:{CID:1 , CName:1 , Department:1 , Credit:1}}}]);
{ ok: 1 }
LorestanUniv>

LorestanUniv>

LorestanUniv> db.CourseRegister.find();
[
  {
    _id: ObjectId('674a048f0c4fea8cc4893bf8'),
    FName: 'علی',
    LName: 'رضایی',
    RegisteredCourses: [],
    SID: '12345678901'
  }
]
LorestanUniv> db.PresentedCourses.find();
[
  {
    _id: ObjectId('674a080b0c4fea8cc4893bfa'),
    FName: 'رضا',
    LName: 'احمدوندیان',
    PresentedCourses: [],
    LID: '123456'
  }
]
LorestanUniv>
```


mongosh mongodb://127.0.0.
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv> db.student.deleteOne({STID:1234567801});
{ acknowledged: true, deletedCount: 0 }
LorestanUniv> db.student.deleteMany({Department:"Engineering"});
{ acknowledged: true, deletedCount: 1 }
LorestanUniv>
LorestanUniv> db.lecturer.deleteOne({LID:"968754"});
{ acknowledged: true, deletedCount: 0 }
LorestanUniv> db.lecturer.deleteMany({Major:"Computer Science"});
{ acknowledged: true, deletedCount: 1 }
LorestanUniv>
LorestanUniv> db.course.deleteOne({CID:"10001"});
{ acknowledged: true, deletedCount: 0 }
LorestanUniv> db.lecturer.deleteMany({Department:"Enginnering"});
{ acknowledged: true, deletedCount: 0 }
LorestanUniv> db.lecturer.deleteMany({Department:"Engineering"});
{ acknowledged: true, deletedCount: 0 }
LorestanUniv>

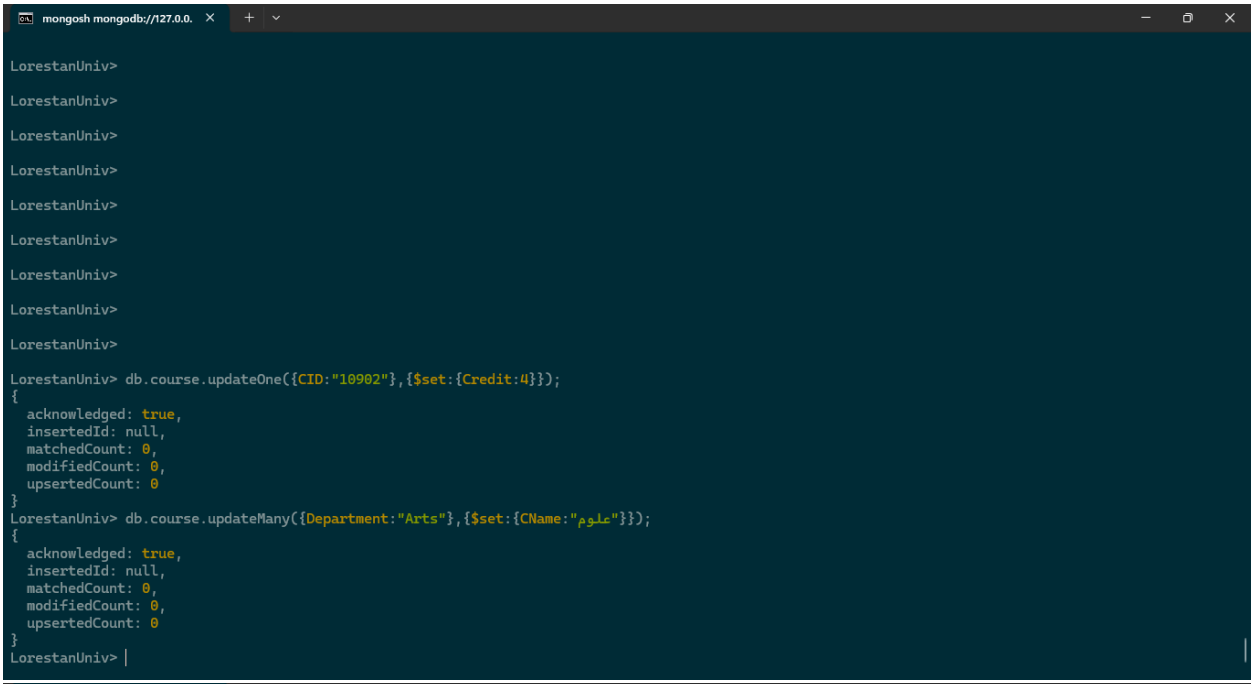

mongosh mongodb://127.0.0.
LorestanUniv>
LorestanUniv> db.student.updateOne({STID:"12345678901"},{$set:{Address:"خیابان فرعی"}});
{
acknowledged: true,
insertedId: null,
matchedCount: 0,
modifiedCount: 0,
upsertedCount: 0
}
LorestanUniv> db.student.updateMany({Department:"Engineering"},{$set:{HPhone:"09123874432"}});
{
acknowledged: true,
insertedId: null,
matchedCount: 0,
modifiedCount: 0,
upsertedCount: 0
}
LorestanUniv> db.lecturer.updateOne({LID:"123451"},{$set:{Major:"Arts"}});
{
acknowledged: true,
insertedId: null,
matchedCount: 0,
modifiedCount: 0,
upsertedCount: 0
}
LorestanUniv> db.lecturer.updateMany({Department:"Science"},{$set:{PostalCode:"1234512345"}});
{
acknowledged: true,
insertedId: null,
matchedCount: 0,
modifiedCount: 0,
upsertedCount: 0
}
LorestanUniv>

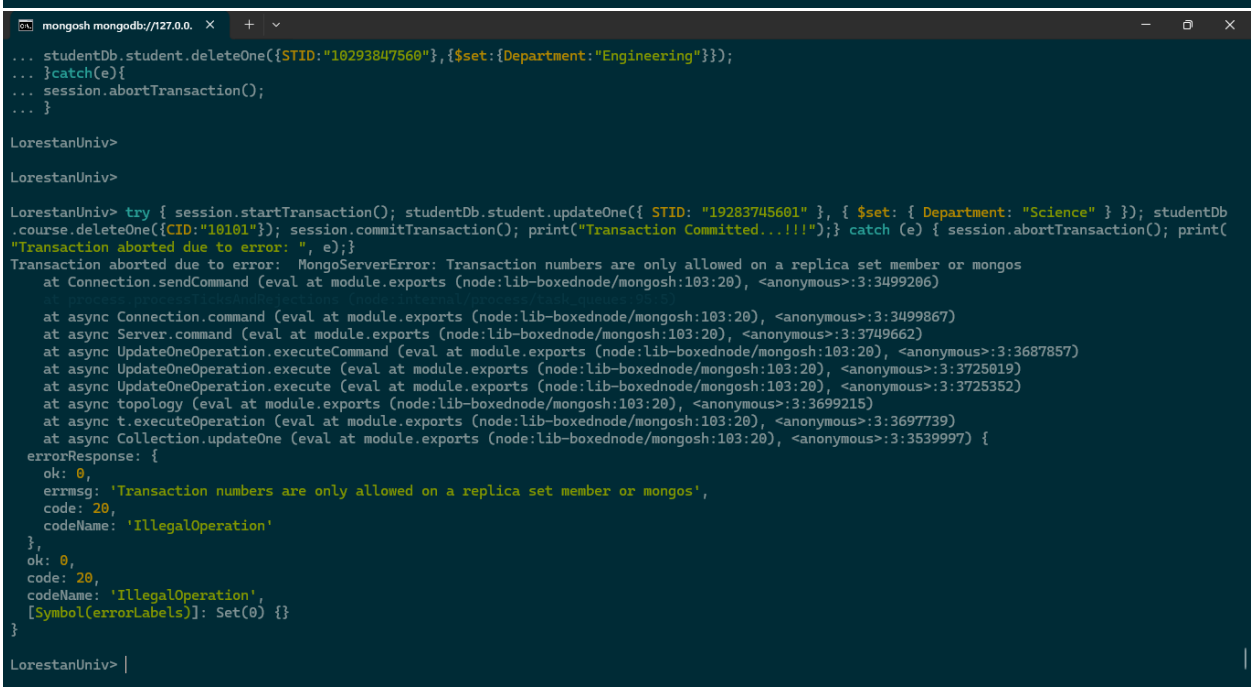


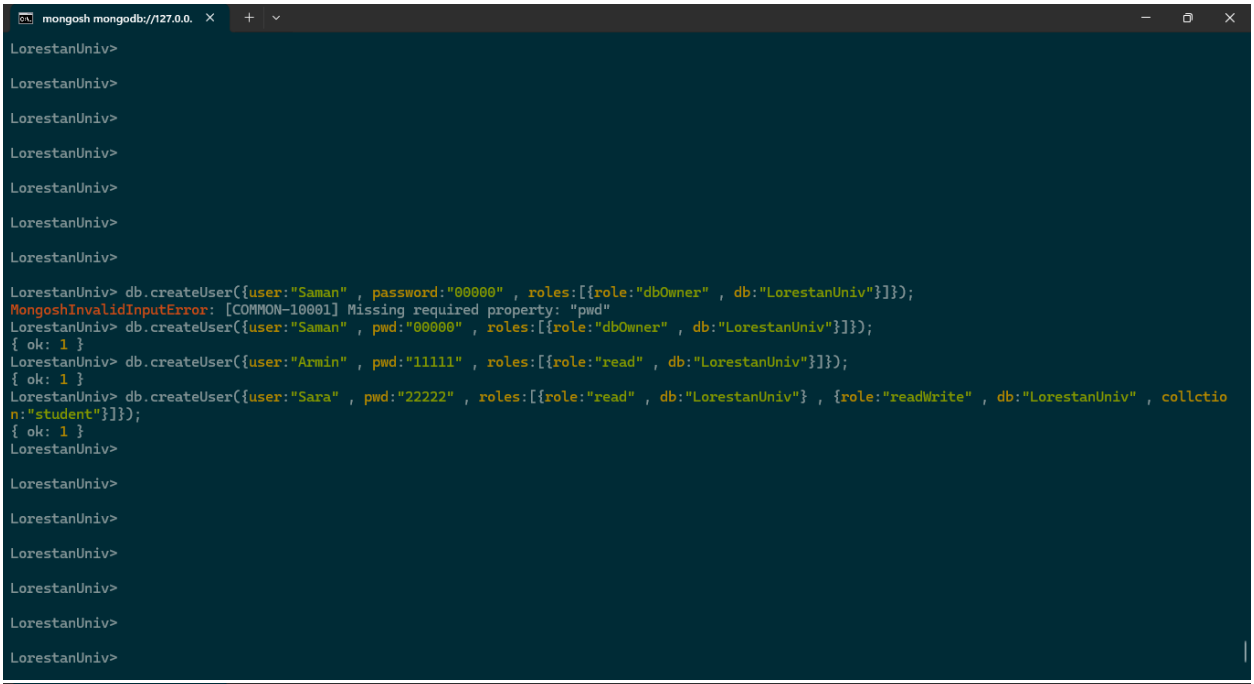
```
mongosh mongodb://127.0.0.
}
LorestanUniv> db.student.insertOne({STID:"12345678901" , FName:"babak" , LName:"yousefian" , Father:"morteza" , Birth:"1375-03-15" , BornCity:"تهران
" , Address:"خیابان آزادی" , PostalCode:"1234567890" , CPhone:"09123456789" , HPhone:"02123456789" , Department:"Engineering" , Major:"Computer Engi
neering" , Married:"Single" , ID:"1234567890"});
Uncaught:
MongoServerError: Document failed validation
Additional information: {
  failingDocumentId: ObjectId('674a11b00c4fea8cc4893bfe'),
  details: {
    operatorName: '$jsonSchema',
    schemaRulesNotSatisfied: [
      {
        operatorName: 'properties',
        propertiesNotSatisfied: [
          {
            propertyName: 'FName',
            description: 'Only persian characters allowed , max length 10',
            details: [
              {
                operatorName: 'pattern',
                specifiedAs: { pattern: '^[ـ-ه]{1,10}$' },
                reason: 'regular expression did not match',
                consideredValue: 'babak'
              }
            ]
          },
          {
            propertyName: 'LName',
            description: 'Only persian characters allowed , max length is 10',
            details: [
              {
                operatorName: 'pattern',
                specifiedAs: { pattern: '^[ـ-ه]{1,10}$' },
                reason: 'regular expression did not match',
                consideredValue: 'yousefian'
              }
```



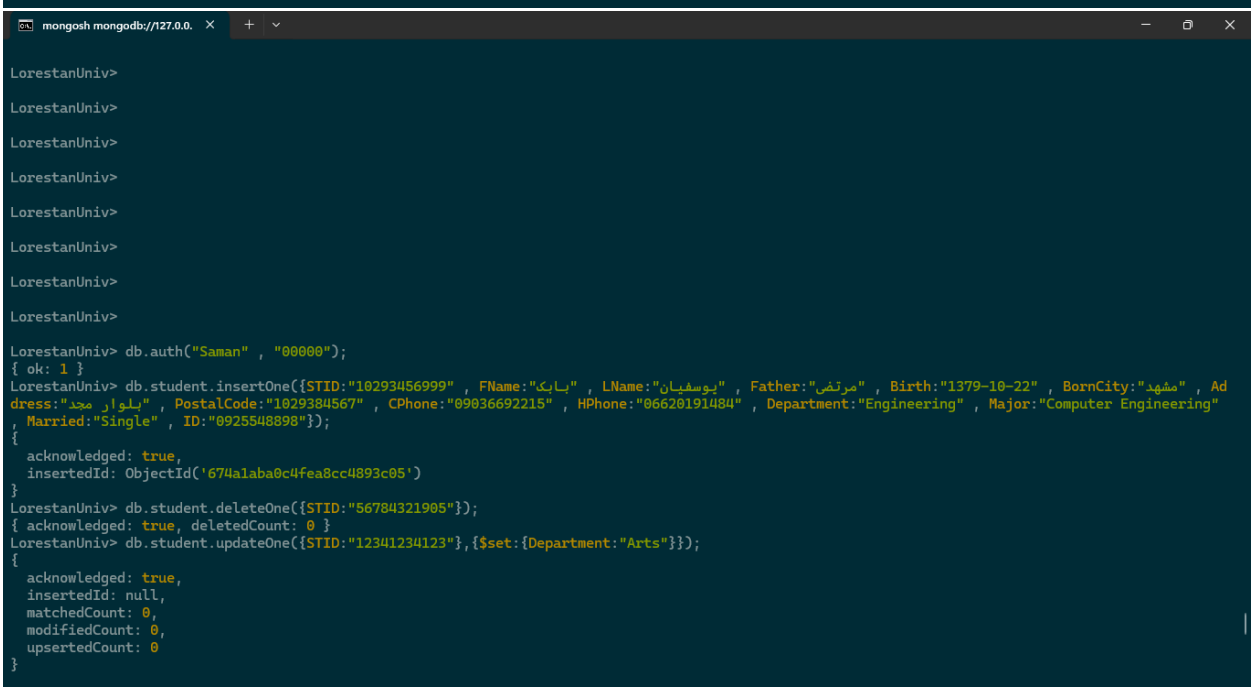
```
mongosh mongodb://127.0.0.
LorestanUniv> db.student.insertOne({STID:"12345678901" , FName:"سامان" , LName:"ساسانی" , Father:"چشمی" , Birth:"1375-03-15" , BornCity:"تهران" , Ad
dress:"خیابان آزادی" , PostalCode:"1234567890" , CPhone:"09123456789" , HPhone:"02123456789" , Department:"Engineering" , Major:"Computer Engineerin
g" , Married:"Single" , ID:"1234567890"});
{
  acknowledged: true,
  insertedId: ObjectId('674a12200c4fea8cc4893bff')
}
LorestanUniv> db.lecturer.insertOne({LID:"123456" , FName:"رضا" , LName:"احمدوندیان" , ID:"1234567890" , Department:"Engineering" , Major:"Computer
Science" , Birth:"1381-10-10" , BornCity:"اهواز" , Address:"خیابان اصلی" , PostalCode:"2934586712" , CPhone:"09129845761" , HPhone:"06612349876"});
{
  acknowledged: true,
  insertedId: ObjectId('674a12330c4fea8cc4893c00')
}
LorestanUniv> db.lecturer.insertOne({LID:"123456" , FName:"محمد" , LName:"احمدوندیان" , ID:"1234567890" , Department:"Engineering" , Major:"Computer
 Science" , Birth:"1381-10-10" , BornCity:"اهواز" , Address:"خیابان اصلی" , PostalCode:"2934586712" , CPhone:"09129845761" , HPhone:"06612349876"});

{
  acknowledged: true,
  insertedId: ObjectId('674a124b0c4fea8cc4893c01')
}
LorestanUniv> db.course.insertOne({CID:"19384" , CName:"هوشمصنوعی" , Department:"Engineering" , Credit:3});
{
  acknowledged: true,
  insertedId: ObjectId('674a12580c4fea8cc4893c02')
}
LorestanUniv> db.course.insertOne({CID:"19384" , CName:"ریاض" , Department:"Engineering" , Credit:3});
{
  acknowledged: true,
  insertedId: ObjectId('674a12660c4fea8cc4893c03')
}
LorestanUniv> db.course.insertOne({CID:"19384" , CName:"شیمی" , Department:"Engineering" , Credit:3});
{
  acknowledged: true,
  insertedId: ObjectId('674a12950c4fea8cc4893c04')
}
LorestanUniv>
```


mongosh mongodb://127.0.0.
LorestanUniv>
LorestanUniv> db.course.updateOne({CID:"10902"},{$set:{Credit:4}});
{
acknowledged: true,
insertedId: null,
matchedCount: 0,
modifiedCount: 0,
upsertedCount: 0
}
LorestanUniv> db.course.updateMany({Department:"Arts"},{$set:{CName:"علوم"}});
{
acknowledged: true,
insertedId: null,
matchedCount: 0,
modifiedCount: 0,
upsertedCount: 0
}
LorestanUniv>

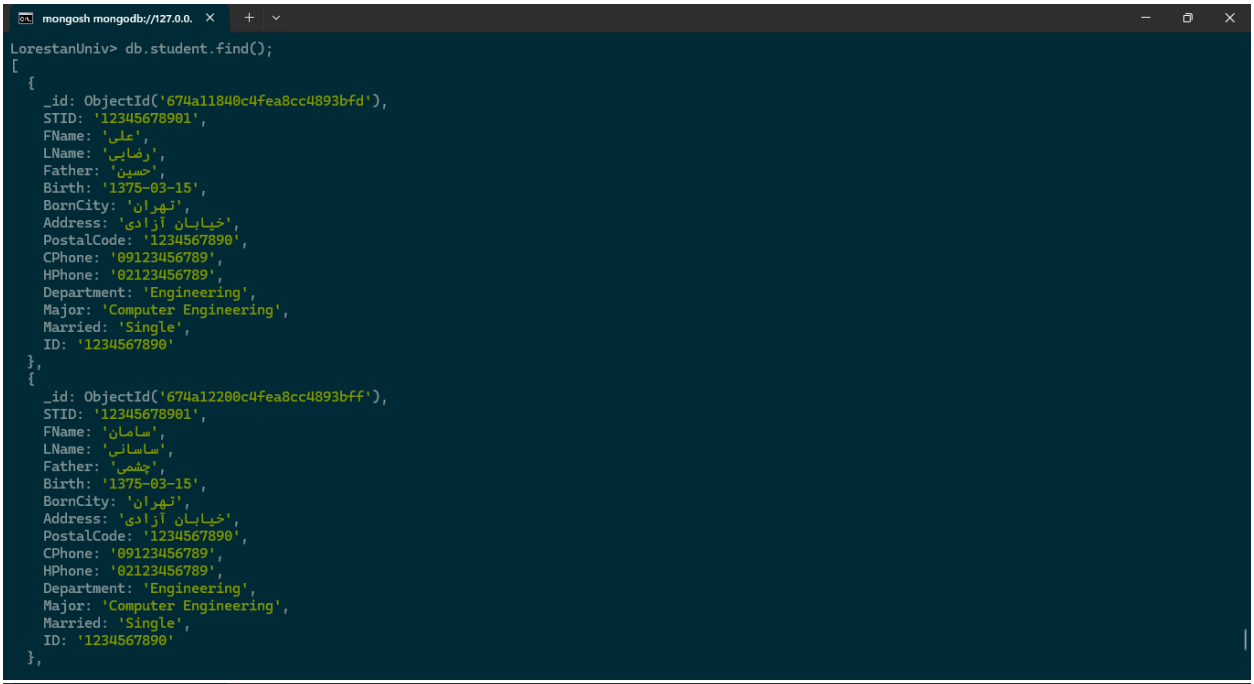

mongosh mongodb://127.0.0.
... studentDb.student.deleteOne({STID:"10293847560"},{$set:{Department:"Engineering"}});
... }catch(e){
... session.abortTransaction();
... }
LorestanUniv>
LorestanUniv> try { session.startTransaction(); studentDb.student.updateOne({ STID: "19283745601" }, { $set: { Department: "Science" } }); studentDb
.course.deleteOne({CID:"10101"}); session.commitTransaction(); print("Transaction Committed...!!!");} catch (e) { session.abortTransaction(); print(
"Transaction aborted due to error: ", e);}
Transaction aborted due to error: MongoServerError: Transaction numbers are only allowed on a replica set member or mongos
at Connection.sendCommand (eval at module.exports (node:lib-boxednode/mongosh:103:20), <anonymous>:3:3499206)
at async Connection.command (eval at module.exports (node:lib-boxednode/mongosh:103:20), <anonymous>:3:3499867)
at async Server.command (eval at module.exports (node:lib-boxednode/mongosh:103:20), <anonymous>:3:3749662)
at async UpdateOneOperation.executeCommand (eval at module.exports (node:lib-boxednode/mongosh:103:20), <anonymous>:3:3687857)
at async UpdateOneOperation.execute (eval at module.exports (node:lib-boxednode/mongosh:103:20), <anonymous>:3:3725019)
at async UpdateOneOperation.execute (eval at module.exports (node:lib-boxednode/mongosh:103:20), <anonymous>:3:3725352)
at async topology (eval at module.exports (node:lib-boxednode/mongosh:103:20), <anonymous>:3:3699215)
at async t.executeOperation (eval at module.exports (node:lib-boxednode/mongosh:103:20), <anonymous>:3:3697739)
at async Collection.updateOne (eval at module.exports (node:lib-boxednode/mongosh:103:20), <anonymous>:3:3539997) {
errorResponse: {
ok: 0,
errmsg: 'Transaction numbers are only allowed on a replica set member or mongos',
code: 20,
codeName: 'IllegalOperation'
},
ok: 0,
code: 20,
codeName: 'IllegalOperation',
[Symbol(errorLabels)]: Set(0) {}
}
LorestanUniv>

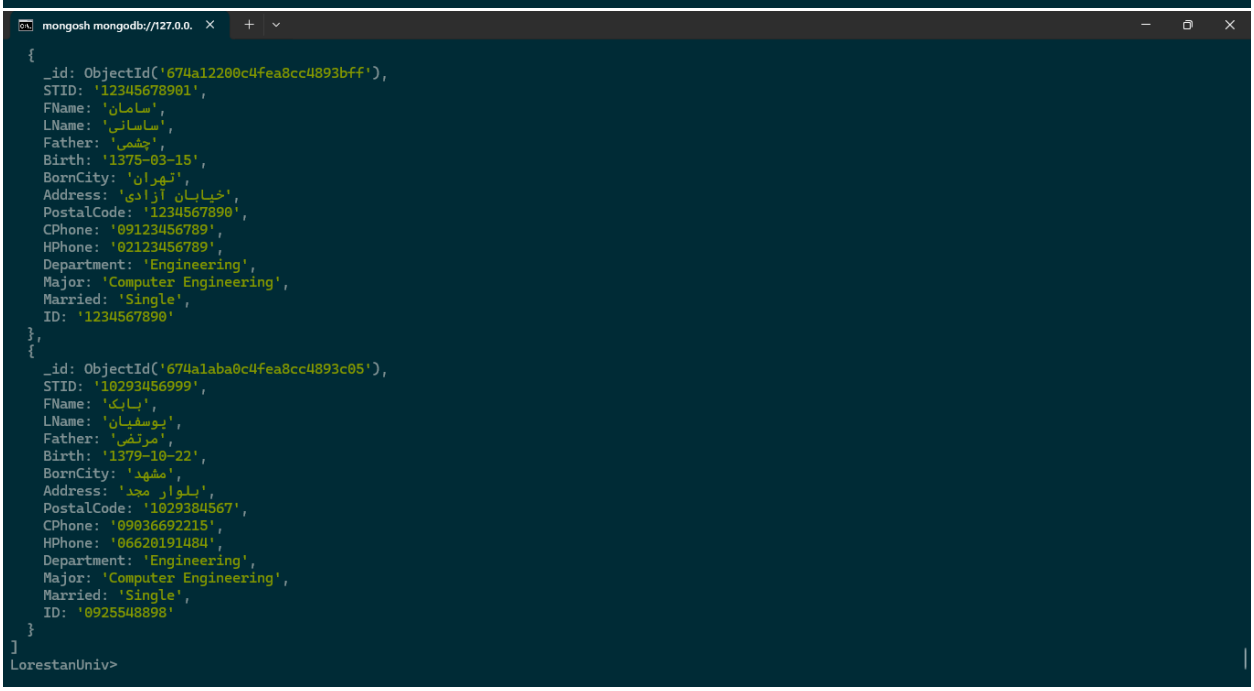


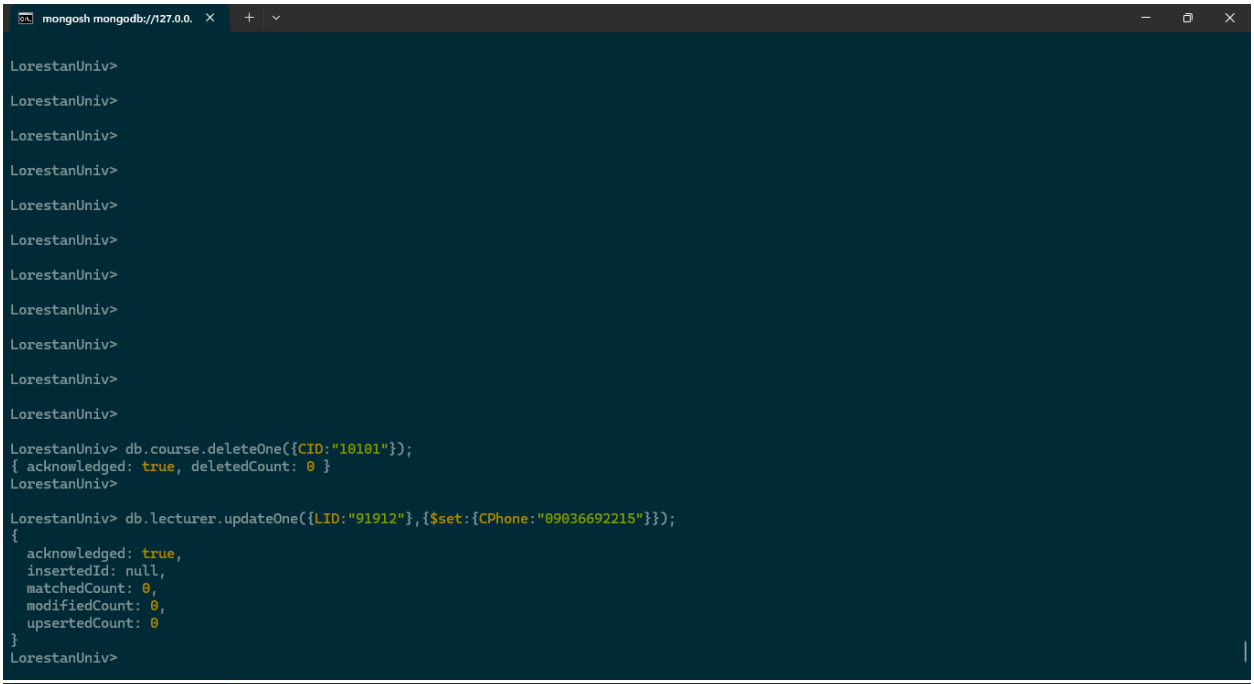
```
mongosh mongodb://127.0.0.
LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv> db.createUser({user:"Saman" , password:"00000" , roles:[{role:"dbOwner" , db:"LorestanUniv"}]});
MongoshInvalidInputError: [COMMON-10001] Missing required property: "pwd"
LorestanUniv> db.createUser({user:"Saman" , pwd:"00000" , roles:[{role:"dbOwner" , db:"LorestanUniv"}]});
{ ok: 1 }
LorestanUniv> db.createUser({user:"Armin" , pwd:"11111" , roles:[{role:"read" , db:"LorestanUniv"}]});
{ ok: 1 }
LorestanUniv> db.createUser({user:"Sara" , pwd:"22222" , roles:[{role:"read" , db:"LorestanUniv"} , {role:"readWrite" , db:"LorestanUniv" , collctio
n:"student"}]});
{ ok: 1 }
LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>
```



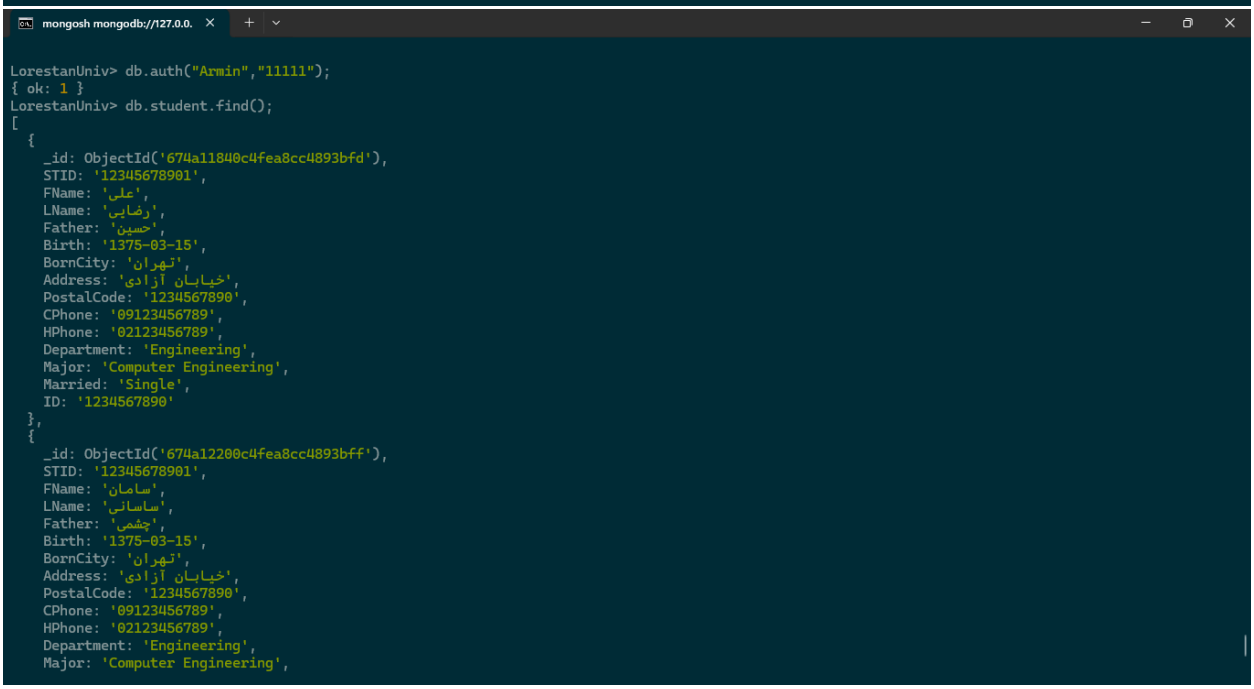
```
mongosh mongodb://127.0.0.

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv> db.auth("Saman" , "00000");
{ ok: 1 }
LorestanUniv> db.student.insertOne({STID:"10293456999" , FName:"بابک" , LName:"یوسفیان" , Father:"مرتضی" , Birth:"1379-10-22" , BornCity:"مشهد" , Ad
dress:"بلوار مجد" , PostalCode:"1029384567" , CPhone:"09036692215" , HPhone:"06620191484" , Department:"Engineering" , Major:"Computer Engineering"
, Married:"Single" , ID:"0925548898"});
{
  acknowledged: true,
  insertedId: ObjectId('674a1aba0c4fea8cc4893c05')
}
LorestanUniv> db.student.deleteOne({STID:"56784321905"});
{ acknowledged: true, deletedCount: 0 }
LorestanUniv> db.student.updateOne({STID:"12341234123"},{$set:{Department:"Arts"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```


mongosh mongodb://127.0.0.
LorestanUniv> db.student.find();
[
{
_id: ObjectId('674a11840c4fea8cc4893bfd'),
STID: '12345678901',
FName: 'علی',
LName: 'رضایی',
Father: 'حسین',
Birth: '1375-03-15',
BornCity: 'تهران',
Address: 'خیابان آزادی',
PostalCode: '1234567890',
CPhone: '09123456789',
HPhone: '02123456789',
Department: 'Engineering',
Major: 'Computer Engineering',
Married: 'Single',
ID: '1234567890'
},
{
_id: ObjectId('674a12200c4fea8cc4893bff'),
STID: '12345678901',
FName: 'سامان',
LName: 'ساسانی',
Father: 'چشمی',
Birth: '1375-03-15',
BornCity: 'تهران',
Address: 'خیابان آزادی',
PostalCode: '1234567890',
CPhone: '09123456789',
HPhone: '02123456789',
Department: 'Engineering',
Major: 'Computer Engineering',
Married: 'Single',
ID: '1234567890'
},

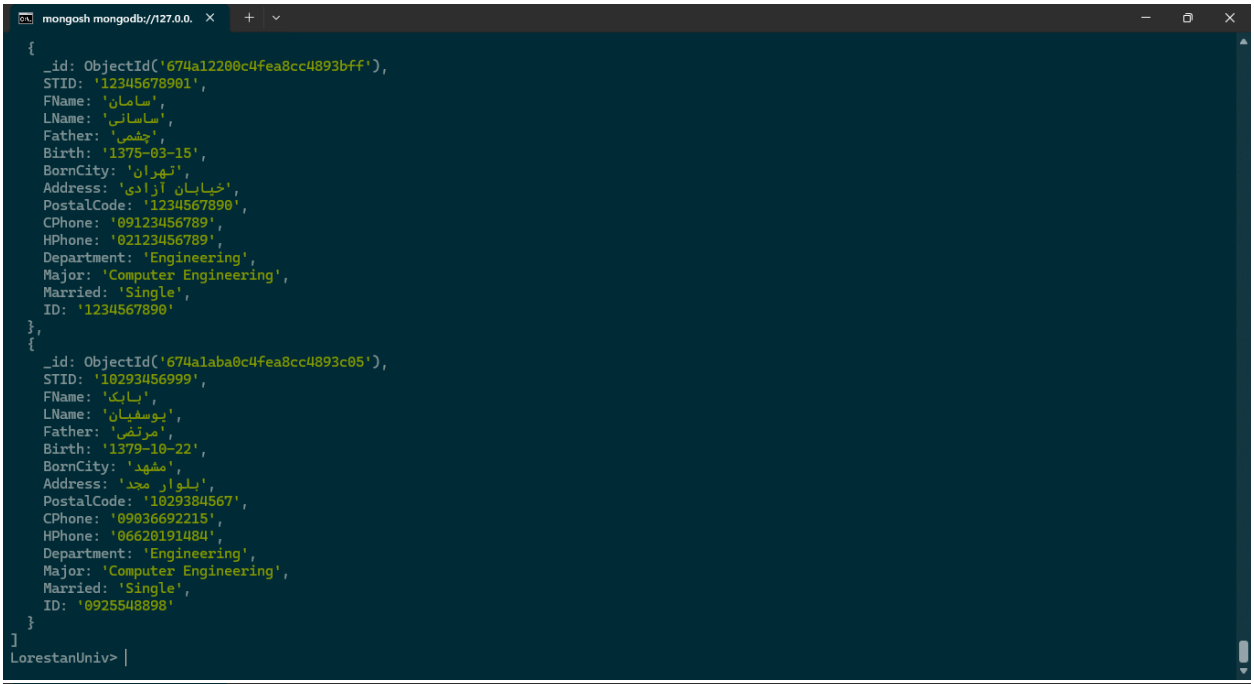

mongosh mongodb://127.0.0.
{
_id: ObjectId('674a12200c4fea8cc4893bff'),
STID: '12345678901',
FName: 'سامان',
LName: 'ساسانی',
Father: 'چشمی',
Birth: '1375-03-15',
BornCity: 'تهران',
Address: 'خیابان آزادی',
PostalCode: '1234567890',
CPhone: '09123456789',
HPhone: '02123456789',
Department: 'Engineering',
Major: 'Computer Engineering',
Married: 'Single',
ID: '1234567890'
},
{
_id: ObjectId('674a1aba0c4fea8cc4893c05'),
STID: '10293456999',
FName: 'بابک',
LName: 'یوسفیان',
Father: 'مرتضی',
Birth: '1379-10-22',
BornCity: 'مشهد',
Address: 'بلوار مجد',
PostalCode: '1029384567',
CPhone: '09036692215',
HPhone: '06620191484',
Department: 'Engineering',
Major: 'Computer Engineering',
Married: 'Single',
ID: '0925548898'
}
]
LorestanUniv>

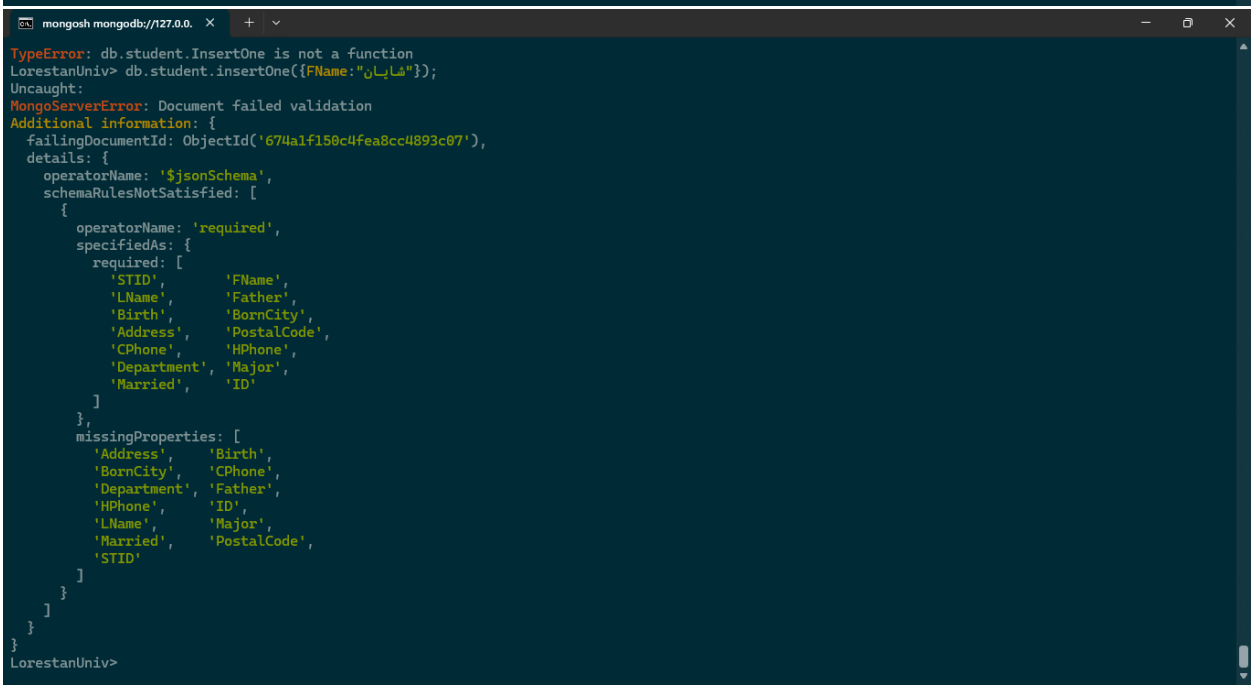


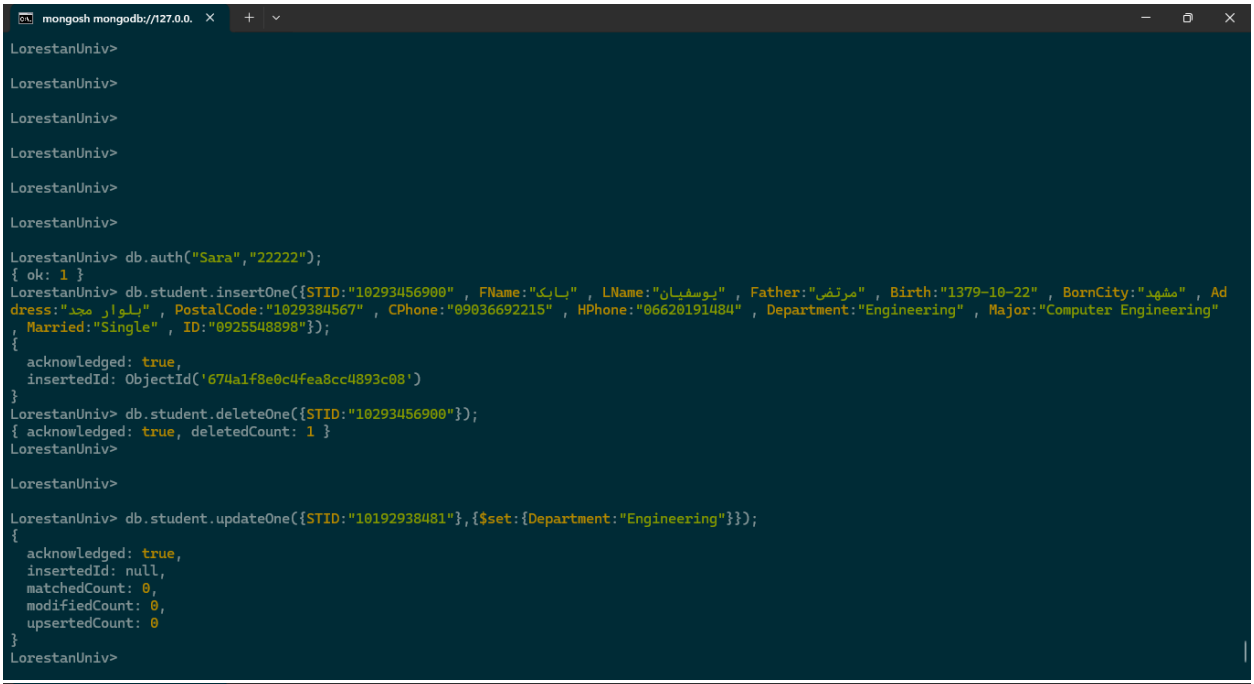
```
mongosh mongodb://127.0.0.

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv> db.course.deleteOne({CID:"10101"});
{ acknowledged: true, deletedCount: 0 }
LorestanUniv>

LorestanUniv> db.lecturer.updateOne({LID:"91912"},{$set:{CPhone:"09036692215"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
LorestanUniv>
```



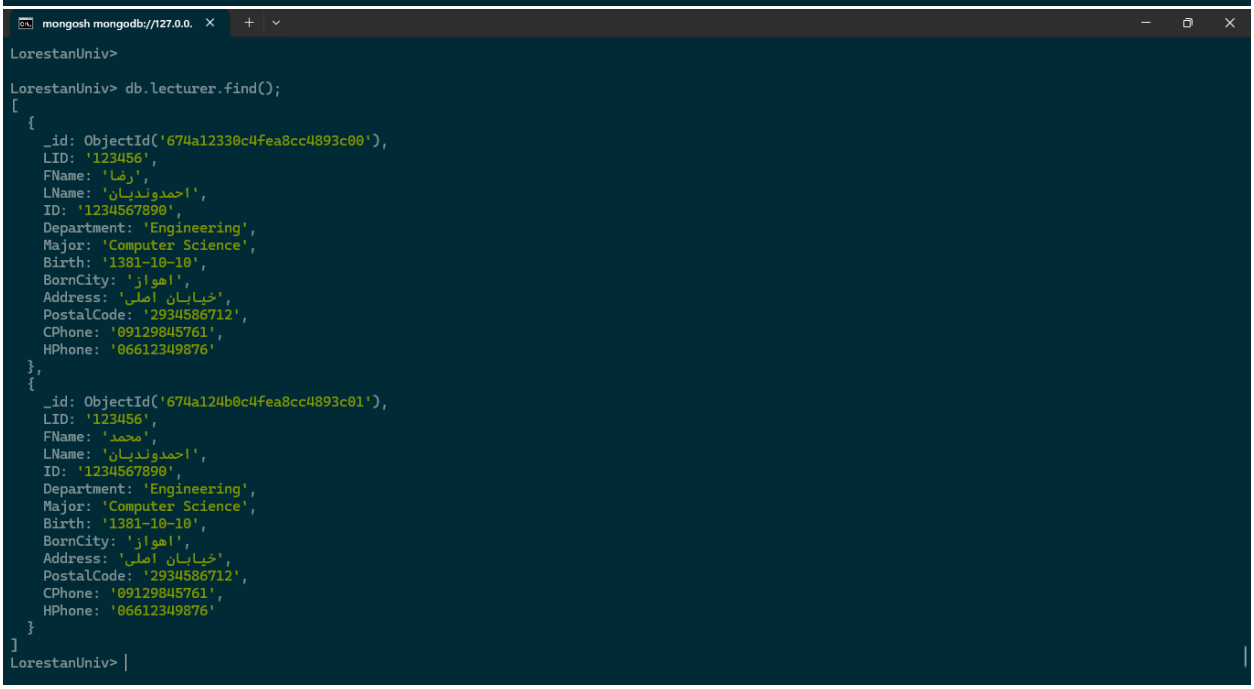
```
mongosh mongodb://127.0.0.

LorestanUniv> db.auth("Armin","11111");
{ ok: 1 }
LorestanUniv> db.student.find();
[
  {
    _id: ObjectId('674a11840c4fea8cc4893bfd'),
    STID: '12345678901',
    FName: 'علی',
    LName: 'رضایی',
    Father: 'حسین',
    Birth: '1375-03-15',
    BornCity: 'تهران',
    Address: 'خیابان آزادی',
    PostalCode: '1234567890',
    CPhone: '09123456789',
    HPhone: '02123456789',
    Department: 'Engineering',
    Major: 'Computer Engineering',
    Married: 'Single',
    ID: '1234567890'
  },
  {
    _id: ObjectId('674a12200c4fea8cc4893bff'),
    STID: '12345678901',
    FName: 'سامان',
    LName: 'ساسانی',
    Father: 'چشمی',
    Birth: '1375-03-15',
    BornCity: 'تهران',
    Address: 'خیابان آزادی',
    PostalCode: '1234567890',
    CPhone: '09123456789',
    HPhone: '02123456789',
    Department: 'Engineering',
    Major: 'Computer Engineering',
```


mongosh mongodb://127.0.0.
{
_id: ObjectId('674a12200c4fea8cc4893bff'),
STID: '12345678901',
FName: 'سامان',
LName: 'ساسانی',
Father: 'چشمی',
Birth: '1375-03-15',
BornCity: 'تهران',
Address: 'خیابان آزادی',
PostalCode: '1234567890',
CPhone: '09123456789',
HPhone: '02123456789',
Department: 'Engineering',
Major: 'Computer Engineering',
Married: 'Single',
ID: '1234567890'
},
{
_id: ObjectId('674a1aba0c4fea8cc4893c05'),
STID: '10293456999',
FName: 'بابک',
LName: 'یوسفیان',
Father: 'مرتضی',
Birth: '1379-10-22',
BornCity: 'مشهد',
Address: 'بلوار مجد',
PostalCode: '1029384567',
CPhone: '09036692215',
HPhone: '06620191484',
Department: 'Engineering',
Major: 'Computer Engineering',
Married: 'Single',
ID: '0925548898'
}
]
LorestanUniv>

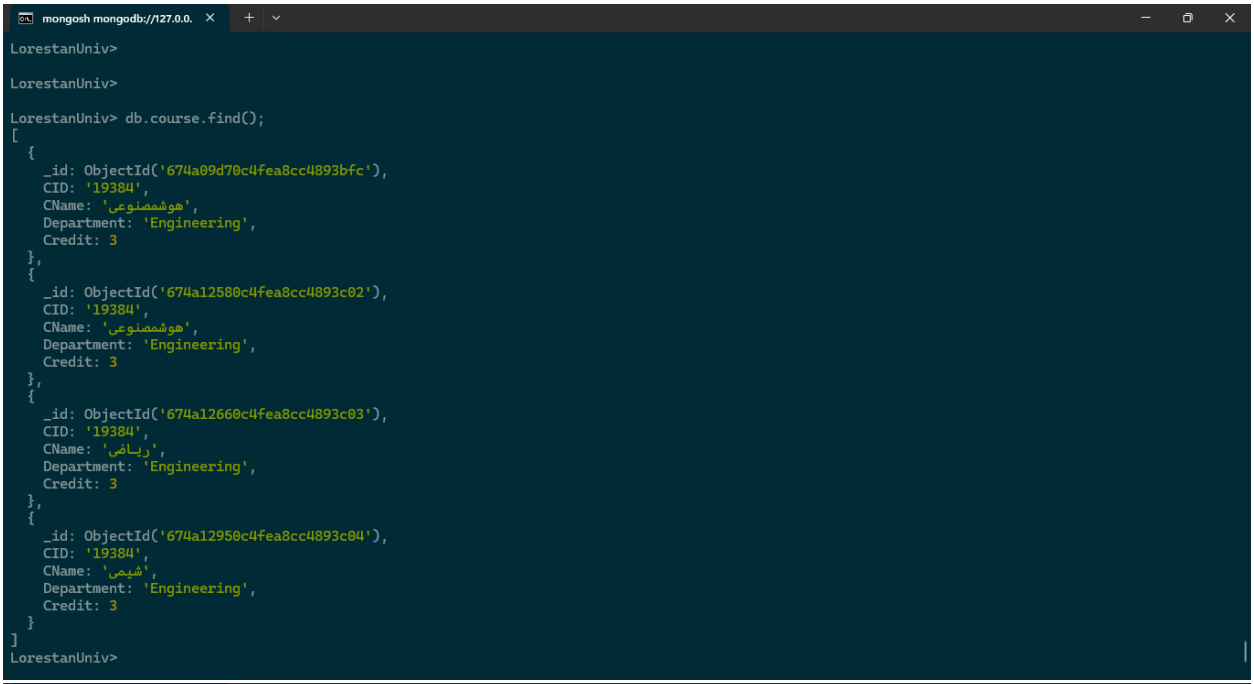

mongosh mongodb://127.0.0.
TypeError: db.student.InsertOne is not a function
LorestanUniv> db.student.insertOne({FName:"شایان"});
Uncaught:
MongoServerError: Document failed validation
Additional information: {
failingDocumentId: ObjectId('674a1f150c4fea8cc4893c07'),
details: {
operatorName: '$jsonSchema',
schemaRulesNotSatisfied: [
{
operatorName: 'required',
specifiedAs: {
required: [
'STID', 'FName',
'LName', 'Father',
'Birth', 'BornCity',
'Address', 'PostalCode',
'CPhone', 'HPhone',
'Department', 'Major',
'Married', 'ID'
]
},
missingProperties: [
'Address', 'Birth',
'BornCity', 'CPhone',
'Department', 'Father',
'HPhone', 'ID',
'LName', 'Major',
'Married', 'PostalCode',
'STID'
]
}
]
}
}
LorestanUniv>

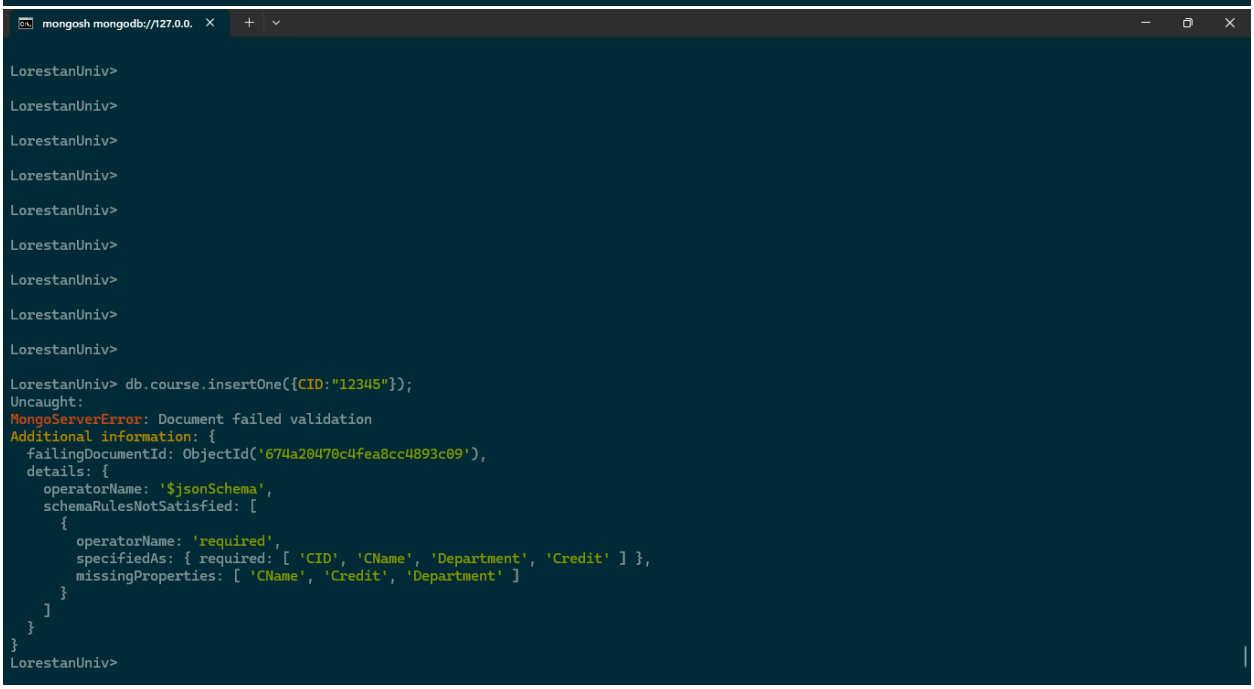


```
mongosh mongodb://127.0.0.
LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv>

LorestanUniv> db.auth("Sara","22222");
{ ok: 1 }
LorestanUniv> db.student.insertOne({STID:"10293456900" , FName:"بابک" , LName:"یوسفیان" , Father:"مرتضی" , Birth:"1379-10-22" , BornCity:"مشهد" , Ad
dress:"بلوار مجد" , PostalCode:"1029384567" , CPhone:"09036692215" , HPhone:"06620191484" , Department:"Engineering" , Major:"Computer Engineering"
, Married:"Single" , ID:"0925548898"});
{
  acknowledged: true,
  insertedId: ObjectId('674a1f8e0c4fea8cc4893c08')
}
LorestanUniv> db.student.deleteOne({STID:"10293456900"});
{ acknowledged: true, deletedCount: 1 }
LorestanUniv>

LorestanUniv>

LorestanUniv> db.student.updateOne({STID:"10192938481"},{$set:{Department:"Engineering"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
LorestanUniv>
```



```
mongosh mongodb://127.0.0.
LorestanUniv>

LorestanUniv> db.lecturer.find();
[
  {
    _id: ObjectId('674a12330c4fea8cc4893c00'),
    LID: '123456',
    FName: 'رضا',
    LName: 'احمدوندیان',
    ID: '1234567890',
    Department: 'Engineering',
    Major: 'Computer Science',
    Birth: '1381-10-10',
    BornCity: 'اهواز',
    Address: 'خیابان اصلی',
    PostalCode: '2934586712',
    CPhone: '09129845761',
    HPhone: '06612349876'
  },
  {
    _id: ObjectId('674a124b0c4fea8cc4893c01'),
    LID: '123456',
    FName: 'محمد',
    LName: 'احمدوندیان',
    ID: '1234567890',
    Department: 'Engineering',
    Major: 'Computer Science',
    Birth: '1381-10-10',
    BornCity: 'اهواز',
    Address: 'خیابان اصلی',
    PostalCode: '2934586712',
    CPhone: '09129845761',
    HPhone: '06612349876'
  }
]
LorestanUniv>
```


mongosh mongodb://127.0.0.
LorestanUniv>
LorestanUniv>
LorestanUniv> db.course.find();
[
{
_id: ObjectId('674a09d70c4fea8cc4893bfc'),
CID: '19384',
CName: 'هوشمصنوعی',
Department: 'Engineering',
Credit: 3
},
{
_id: ObjectId('674a12580c4fea8cc4893c02'),
CID: '19384',
CName: 'هوشمصنوعی',
Department: 'Engineering',
Credit: 3
},
{
_id: ObjectId('674a12660c4fea8cc4893c03'),
CID: '19384',
CName: 'ریاضی',
Department: 'Engineering',
Credit: 3
},
{
_id: ObjectId('674a12950c4fea8cc4893c04'),
CID: '19384',
CName: 'شیمی',
Department: 'Engineering',
Credit: 3
}
]
LorestanUniv>


mongosh mongodb://127.0.0.
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv>
LorestanUniv> db.course.insertOne({CID:"12345"});
Uncaught:
MongoServerError: Document failed validation
Additional information: {
failingDocumentId: ObjectId('674a20470c4fea8cc4893c09'),
details: {
operatorName: '$jsonSchema',
schemaRulesNotSatisfied: [
{
operatorName: 'required',
specifiedAs: { required: ['CID', 'CName', 'Department', 'Credit'] },
missingProperties: ['CName', 'Credit', 'Department']
}
]
}
}
LorestanUniv>

```
mongosh mongodb://127.0.0.

LorestanUniv> try{
... //Begin
... session.startTransaction();
... print("Transaction started. state is ACTIVE");
... //delete one record from course table
... const courseId = "12345";
... const deleteCourseResult = db.course.deleteOne({CID:courseId},{session});
...
... if(deleteCourseResult.deleteCount === 0){
... throw new ERROR(`No record found in course table and state is FAILED`);
... }
...
... print(`Deleted one Record from course and the state is PARTIALLY COMMITTED`);
...
... //SAVEPOINT
... session.commitTransaction();
... session.startTransaction();
... print(`state is SAVEPOINT and the transaction is PARTIALLY COMMITTED`);
...
... const deleteCourseRegisterResult = db.CourseRegister.deleteMany({CID:courseId},{session});
... print(`Delete this record and the state is PARTIALLY COMMITTED`);
...
... const deletePresentedCoursesResult = db.PresentedCourses.deleteMany({CID:courseId},{session});
... print(`Delete this record and the state is PARTIALLY COMMITTED`);
...
... session.commitTransaction();
...
... print(`Transaction committed successfully and the state is COMMITTED`);
...
... }catch(e){
...
... print(`an ERROR occurred and the state is FAILED`);
...
... session.abortTransaction();
... print(`Transaction ABORTED and the state is ABORTED`);
```

```
mongosh mongodb://127.0.0.

... print(`Deleted one Record from course and the state is PARTIALLY COMMITTED`);
...
... //SAVEPOINT
... session.commitTransaction();
... session.startTransaction();
... print(`state is SAVEPOINT and the transaction is PARTIALLY COMMITTED`);
...
... const deleteCourseRegisterResult = db.CourseRegister.deleteMany({CID:courseId},{session});
... print(`Delete this record and the state is PARTIALLY COMMITTED`);
...
... const deletePresentedCoursesResult = db.PresentedCourses.deleteMany({CID:courseId},{session});
... print(`Delete this record and the state is PARTIALLY COMMITTED`);
...
... session.commitTransaction();
...
... print(`Transaction committed successfully and the state is COMMITTED`);
...
... }catch(e){
...
... print(`an ERROR occurred and the state is FAILED`);
...
... session.abortTransaction();
... print(`Transaction ABORTED and the state is ABORTED`);
...
... }finally{
...
... //END THE SESSION
... session.endSession();
... print(`Session Ended...!!!`);
... }
Transaction started. state is ACTIVE
an ERROR occurred and the state is FAILED
Transaction ABORTED and the state is ABORTED
Session Ended...!!!

LorestanUniv>
```