WEB ASSIGNMENT

Q1-

```javascript
const mongoose = require('mongoose');

async function main(){
    await
mongoose.connect('mongodb://127.0.0.1:27017/test3');
}

main()
.then(()=>{console.log("Connected Succesfully");})
.catch((err)=>{console.error(err);});

const userSchema = new mongoose.Schema({
    name:String,
    Roll_no:String,
    age:Number,
});

const User = mongoose.model("User",userSchema); //
Making the model or we can say making the collection
and giving the template to the collection.

const user1=new User({  // Inserting one by one or
making the document for the collection User.
    name: "Nitesh saini",
    Roll_no: "22cs2015",
    age: 20
});
```
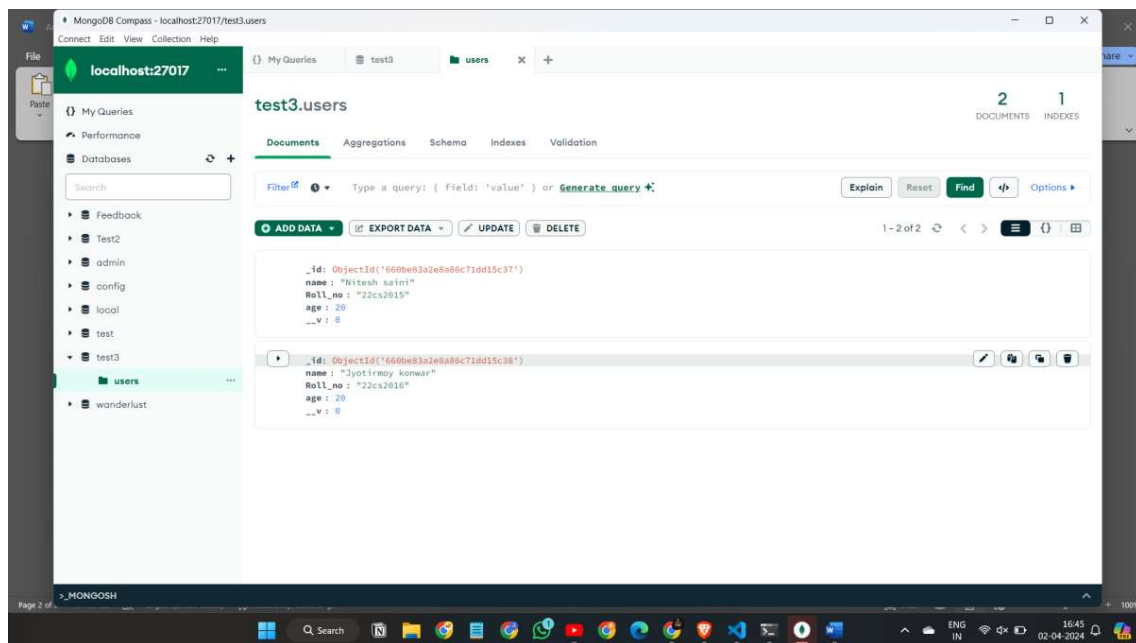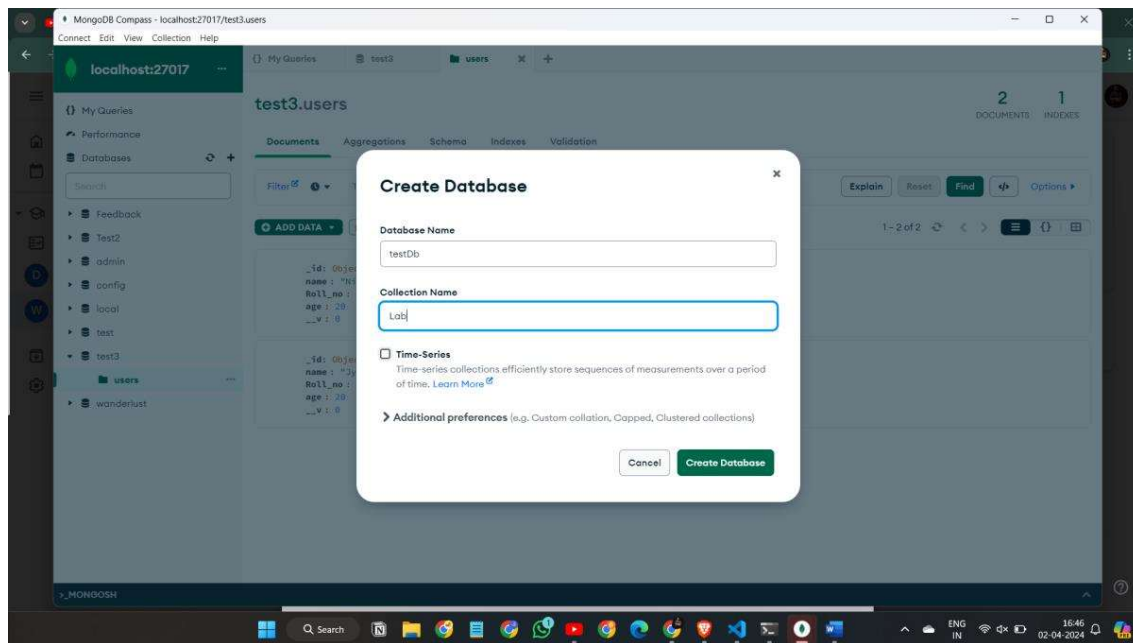
```
const user2=new User({  // Inserting or making the
document for the collection User.
    name: "Jyotirmoy konwar",
    Roll_no: "22cs2016",
    age: 20
});

user1.save();
user2.save();
```
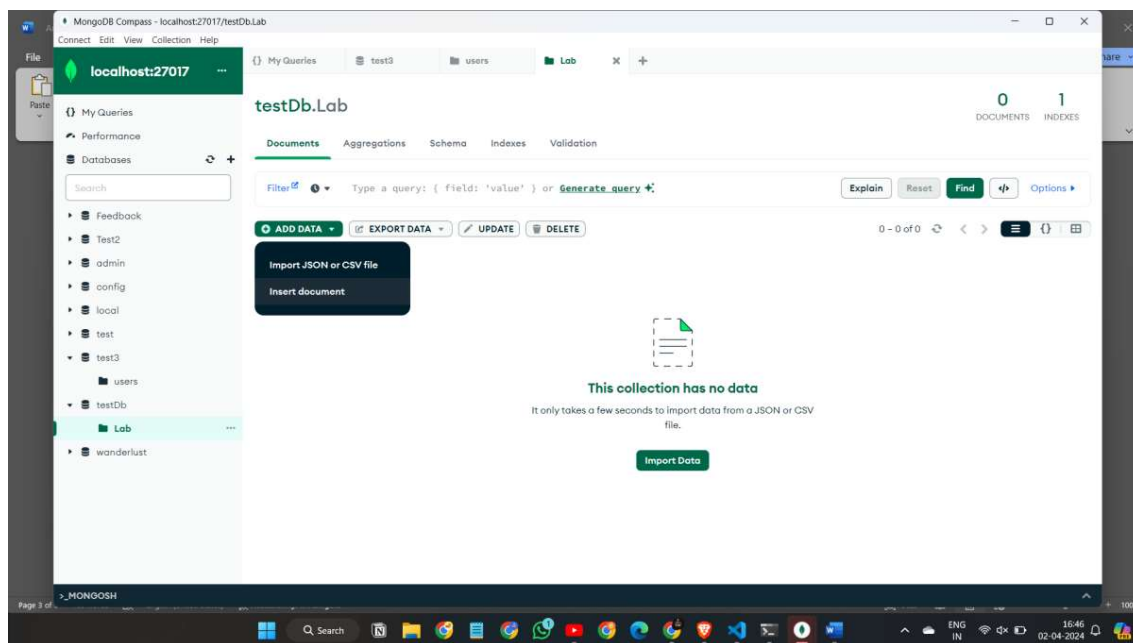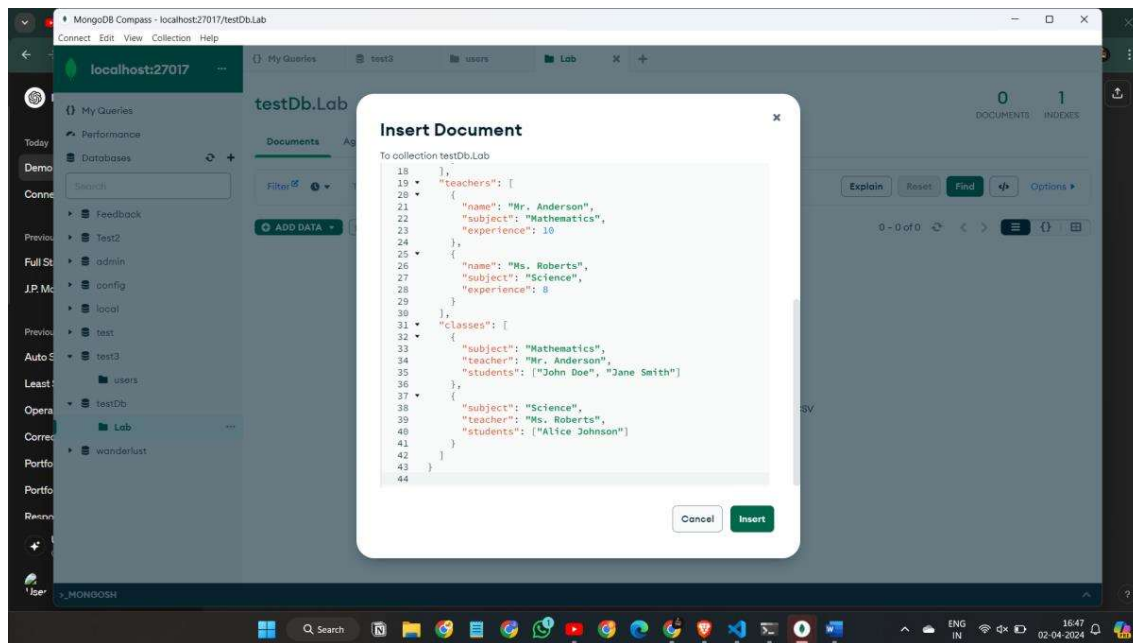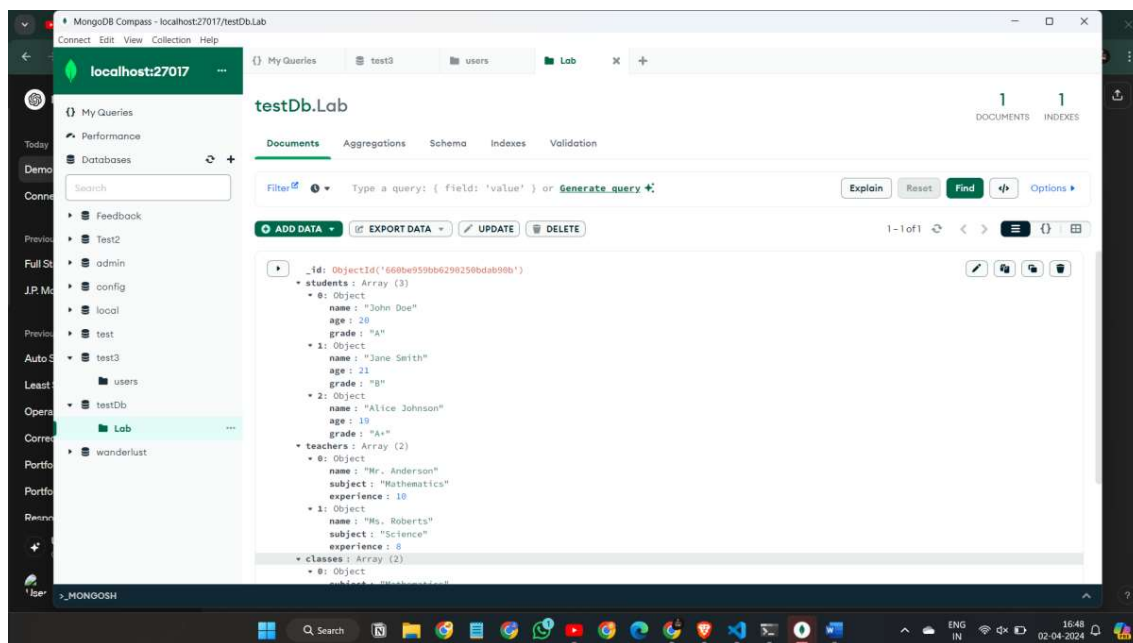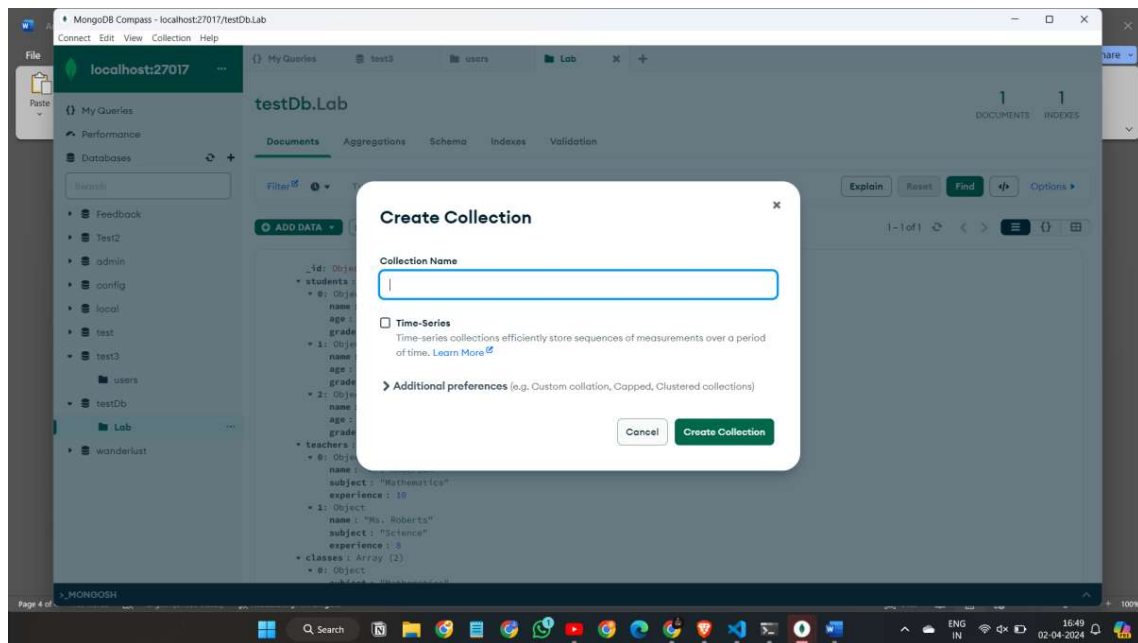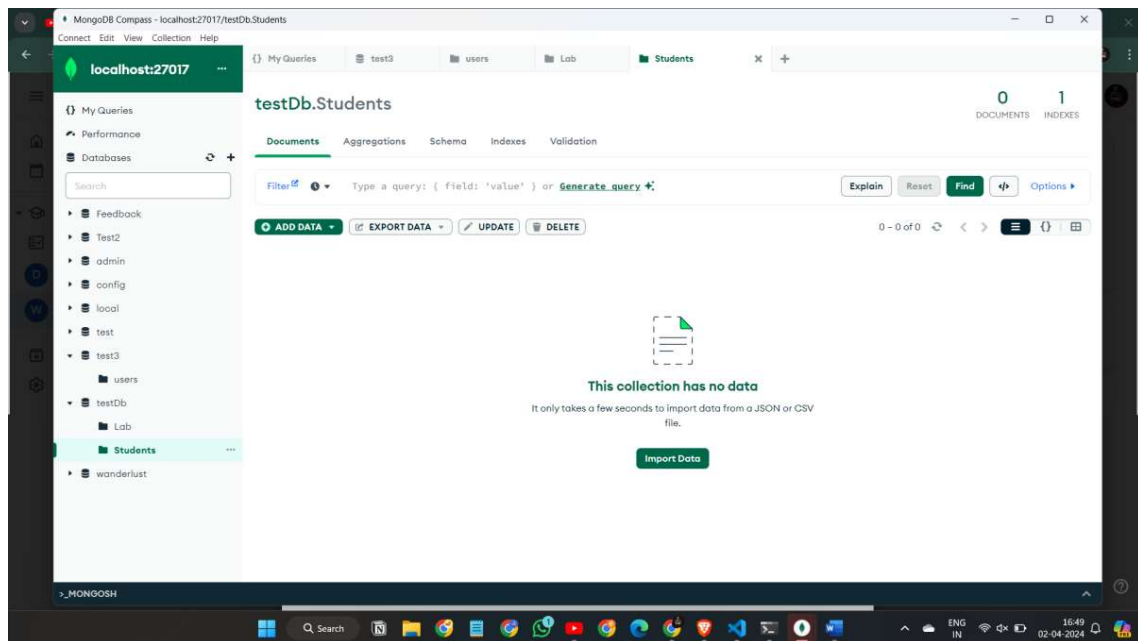

Q-2

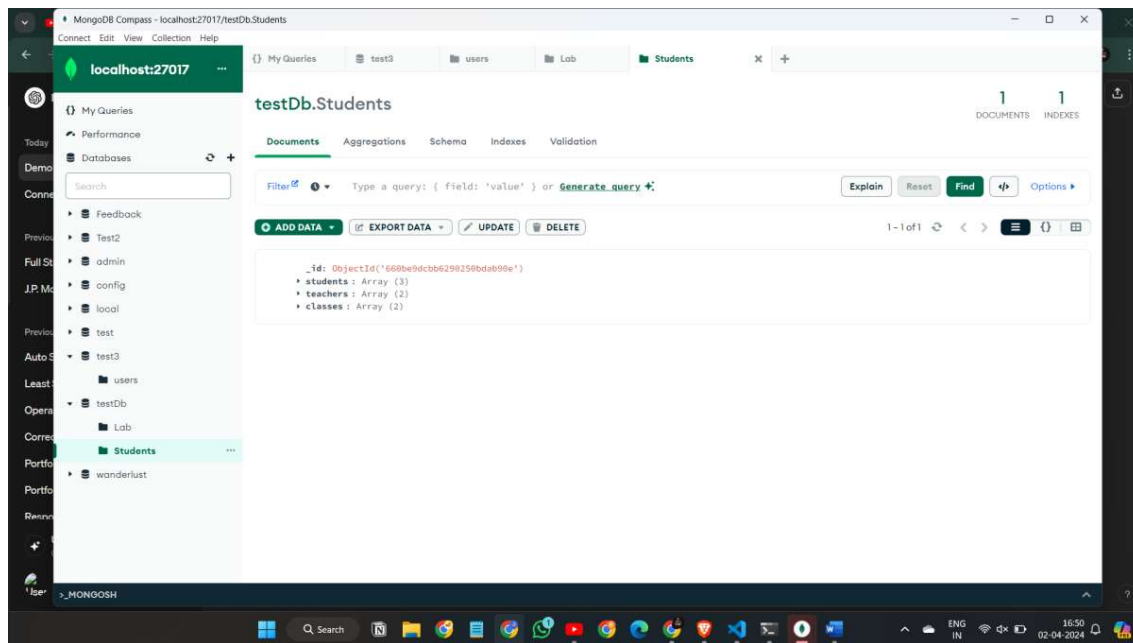Step -1

Step-2

Step-3

Step-4

Q-3

Step-1

Step-2

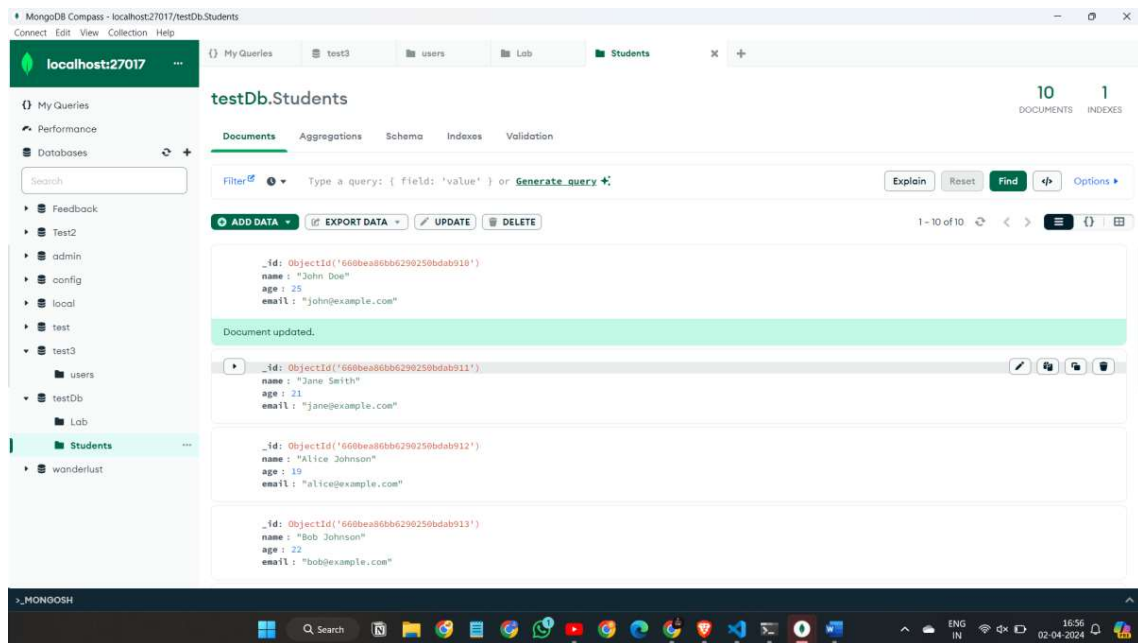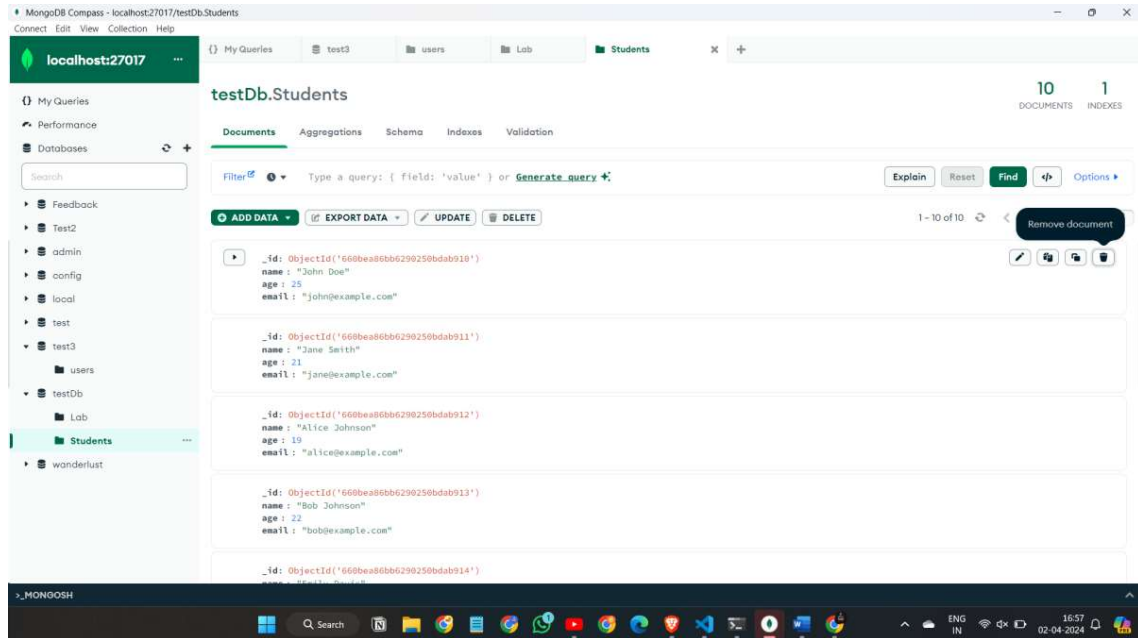Step-3

Q-4

Step-1
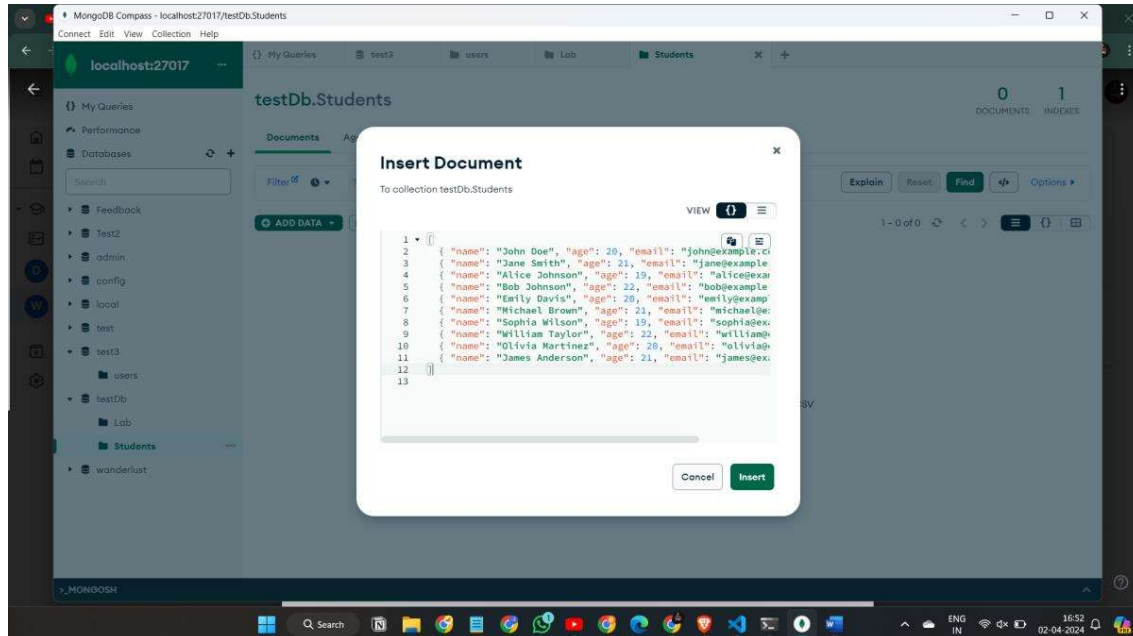
Q-5
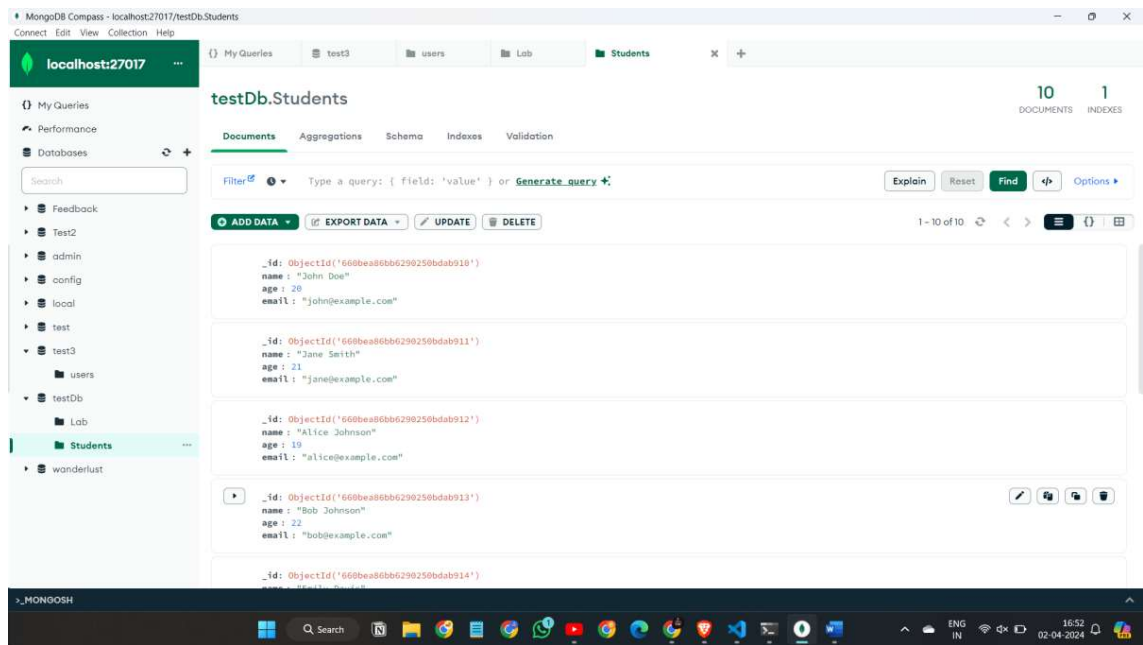
Q-6

Step-1
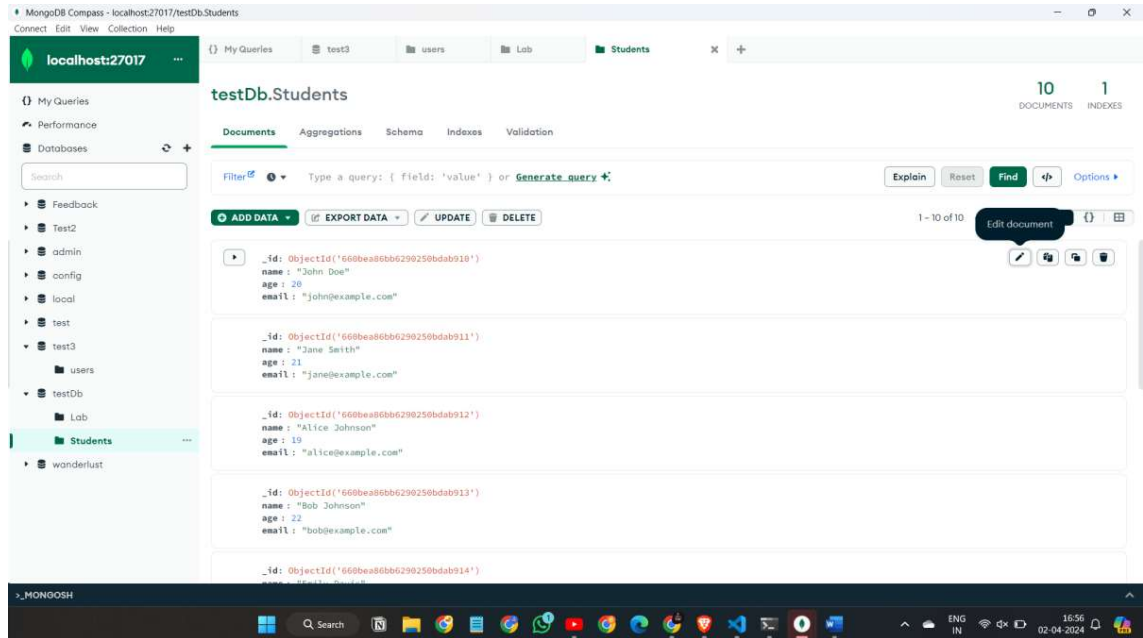
Step-2

Q-7

Step-1

Step-2
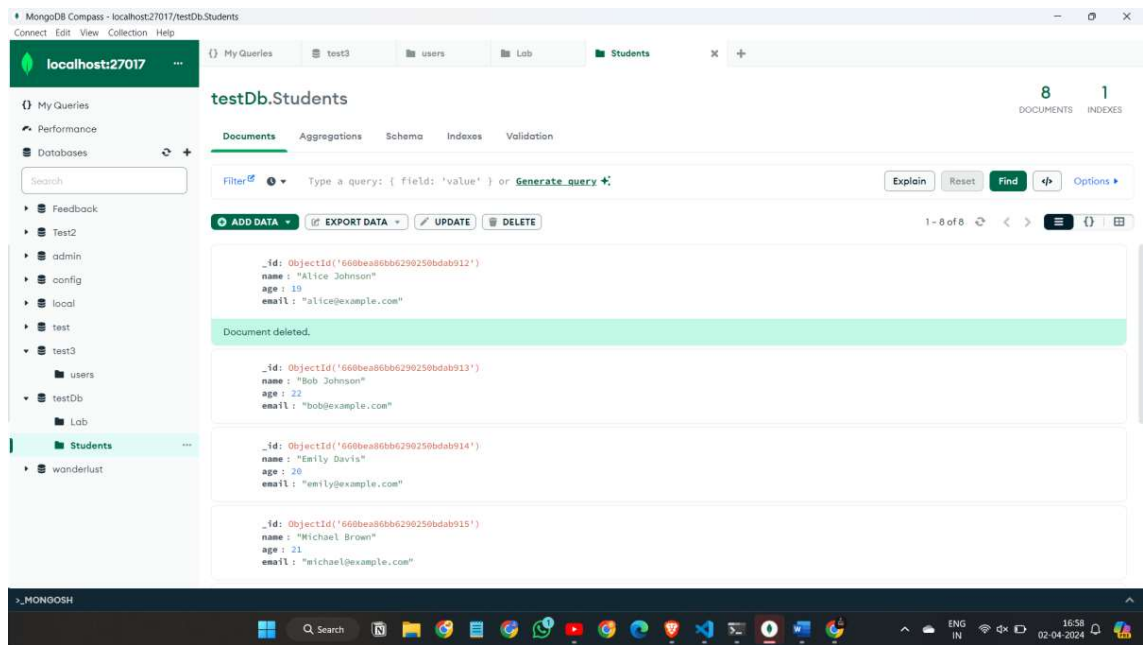
Q-8
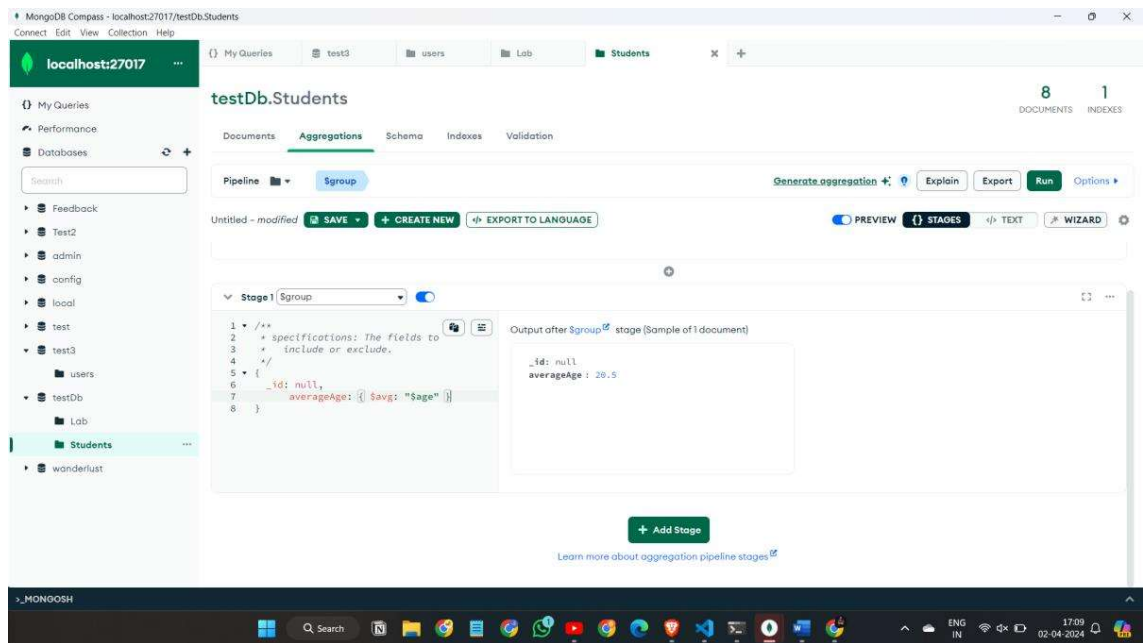
Q-9

Step-1
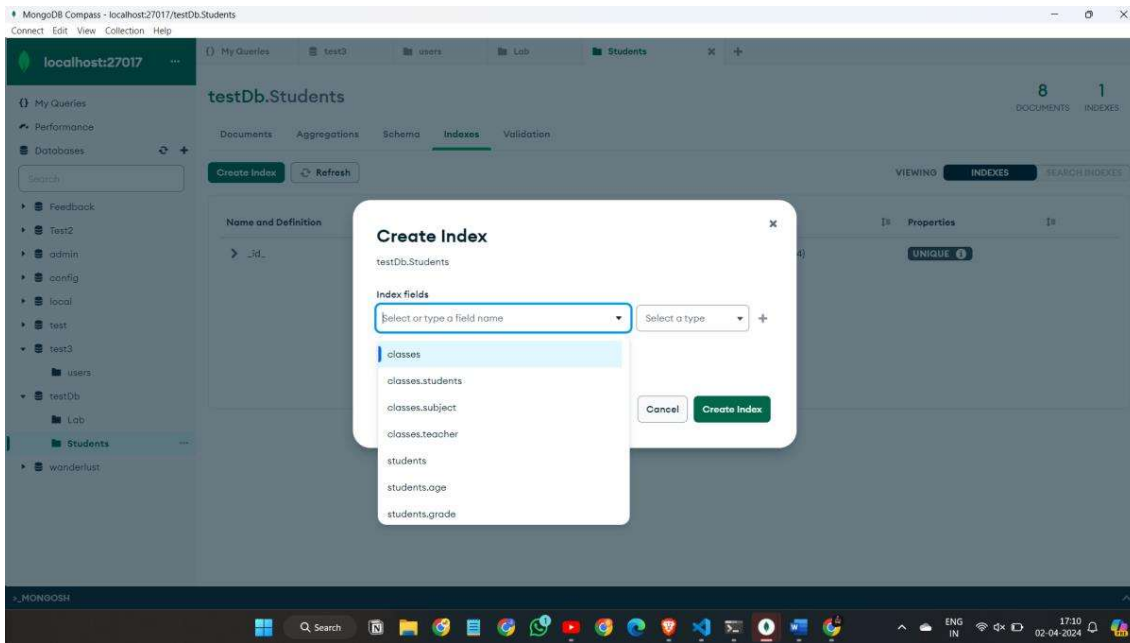
Step-2

Q-10

Step-1

Step-2

Step-3

Q-13

Q-14

Connect   Edit   View   Collection   Help

localhost:27017 ...

{} My Queries
Performance
Databases

Feedback
Test2
admin
config
local
test
test3
  users
testDb
  Lab
  Students
wanderlust

{} My Queries   test3   users   Lab   **Students**   ×   +

testDb.Students

8 DOCUMENTS
1 INDEXES

Documents   Aggregations   Schema   Indexes   **Validation**

Validation Action ⓘ   Error ▾   Validation Level ⓘ   Strict ▾

```
 1 ▾ {
 2 ▾   $jsonSchema: {
 3       bsonType: 'object',
 4 ▾     required: [
 5         'name'
 6       ],
 7 ▾     properties: {
 8 ▾       name: {
 9           bsonType: 'string',
10           minLength: 1,
11           description: 'must be a non-empty string'
12         }
13       }
14     }
15   }
```

Sample documents

✓ Passed validation   ⊘ Failed validation

_id: ObjectId('660bea86bb6290250bdab915')
name : "Michael Brown"
age : 21
email : "michael@example.com"

                                    No Preview Documents

>_MONGOSH