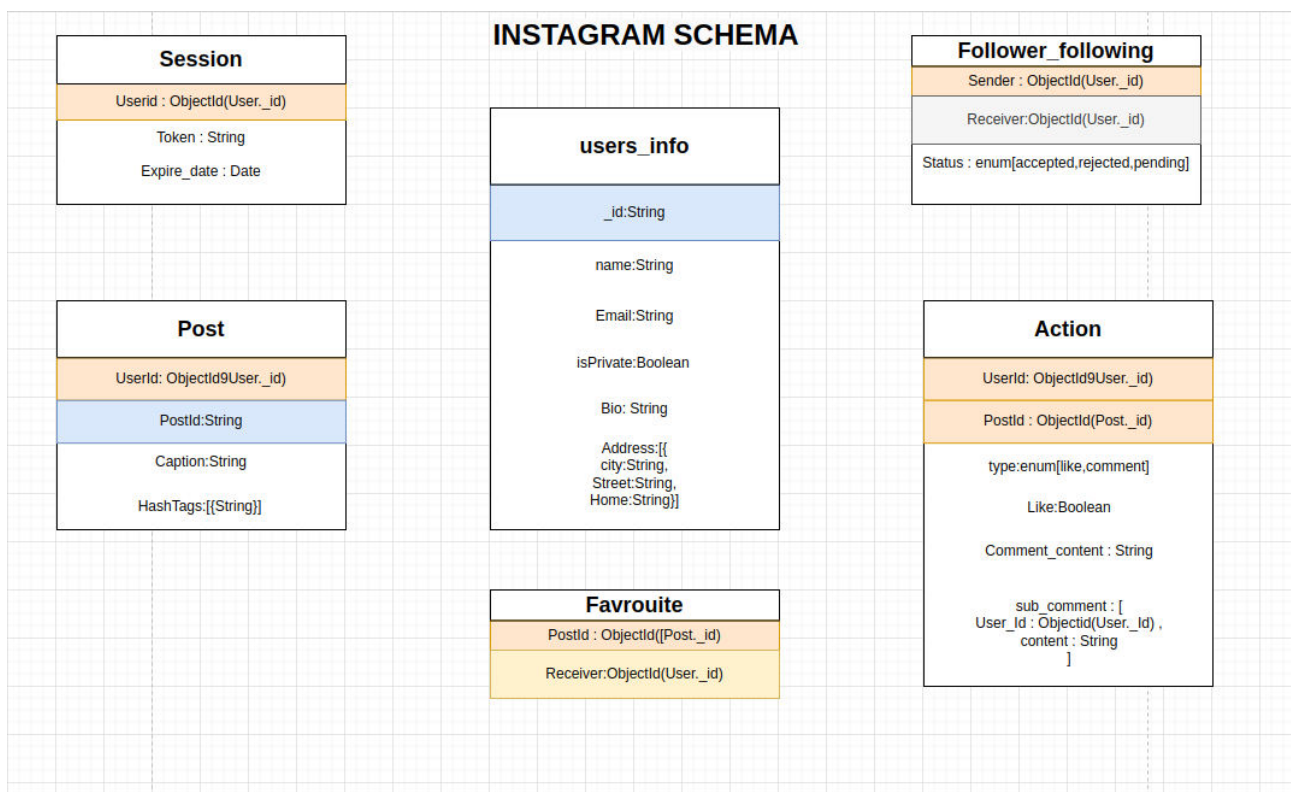
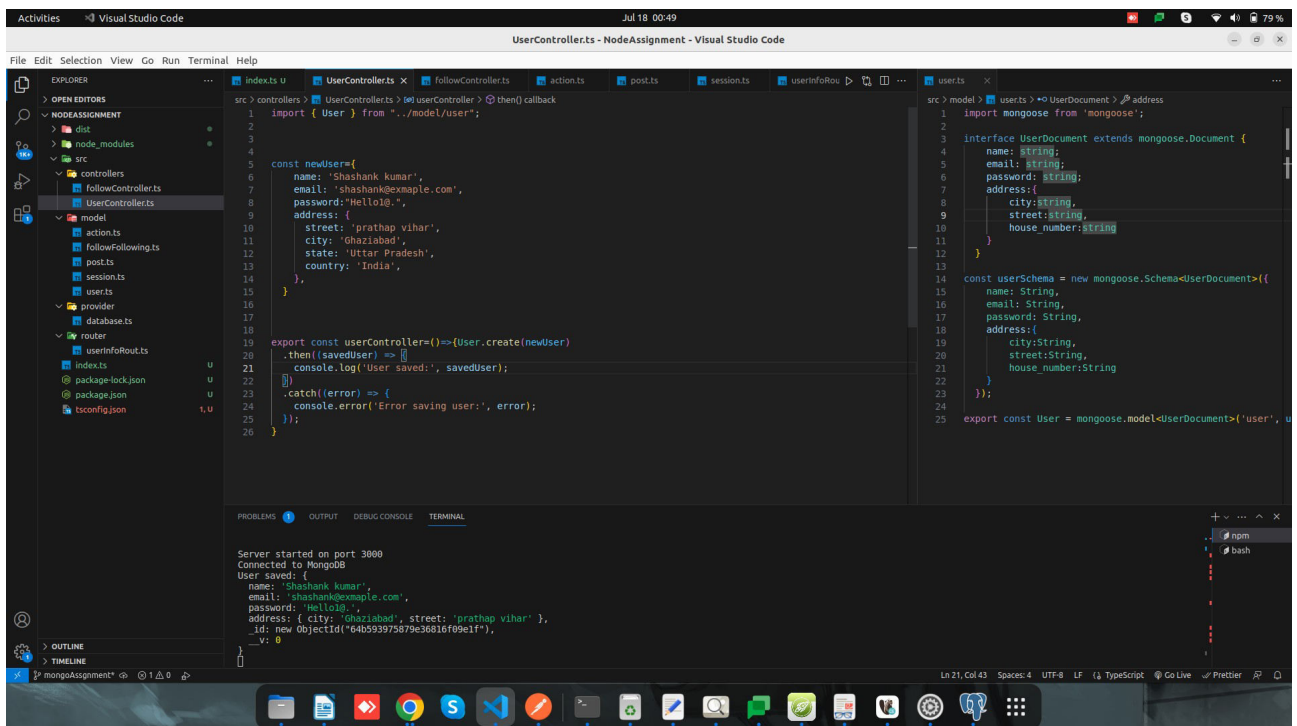
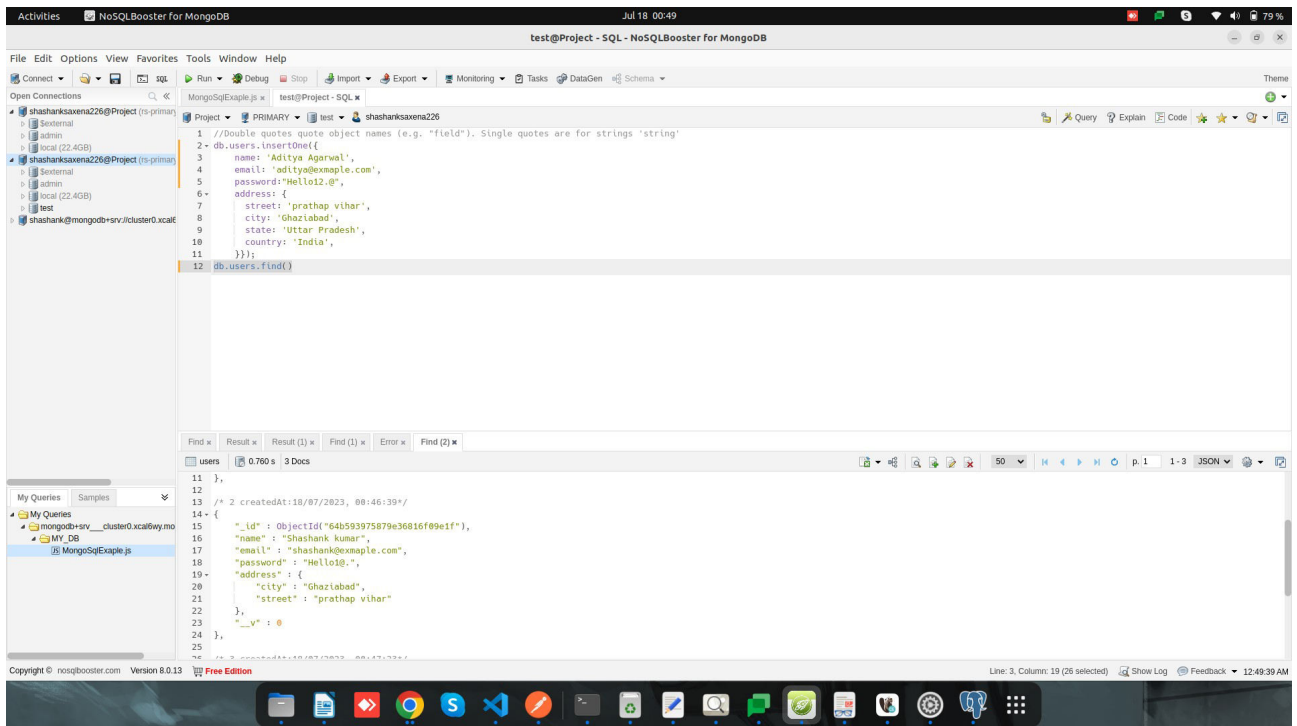


Instagram Schema Design

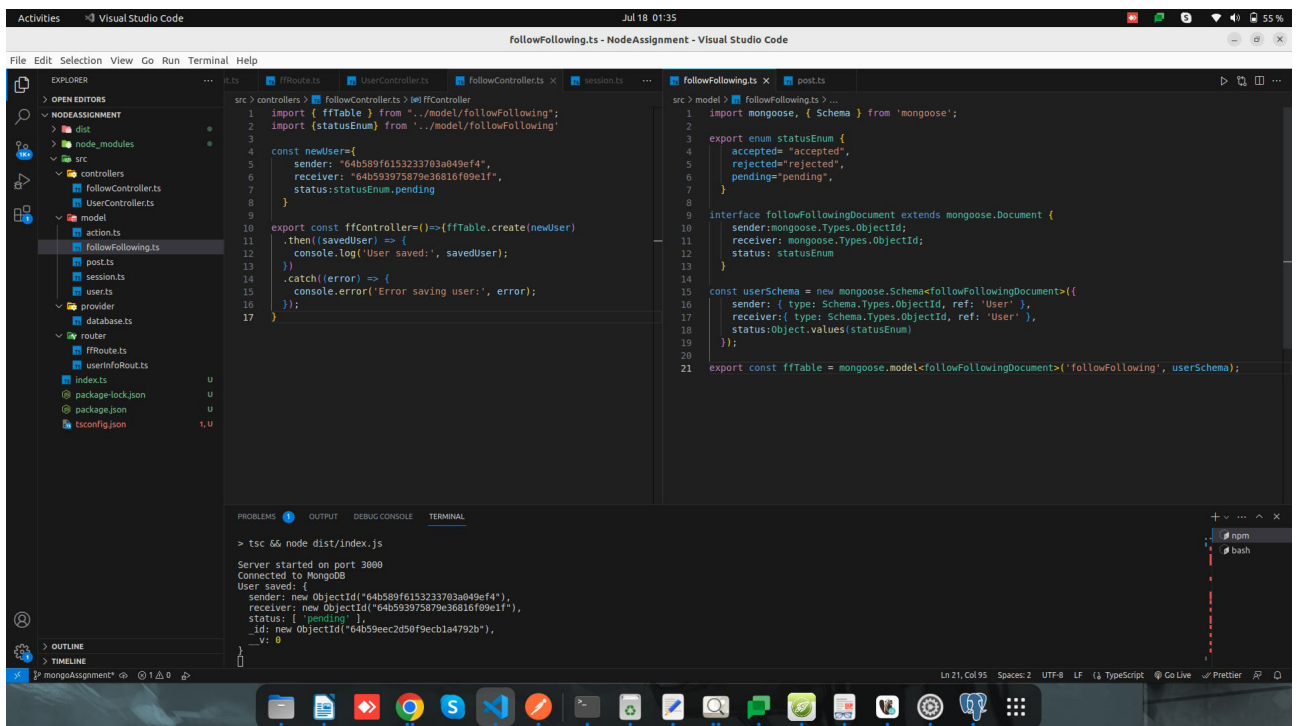
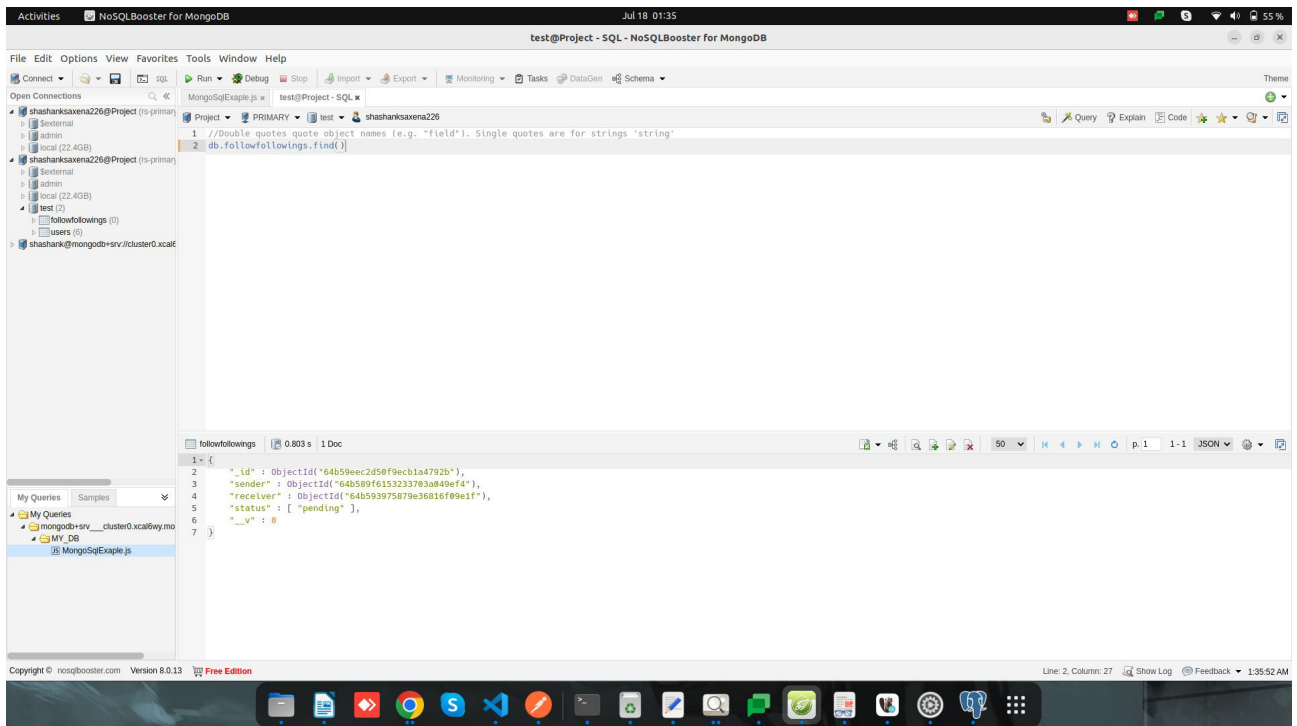
- Features
- user
- session management
- post
- action-comment like
- follower/following with private public account
- favrouite



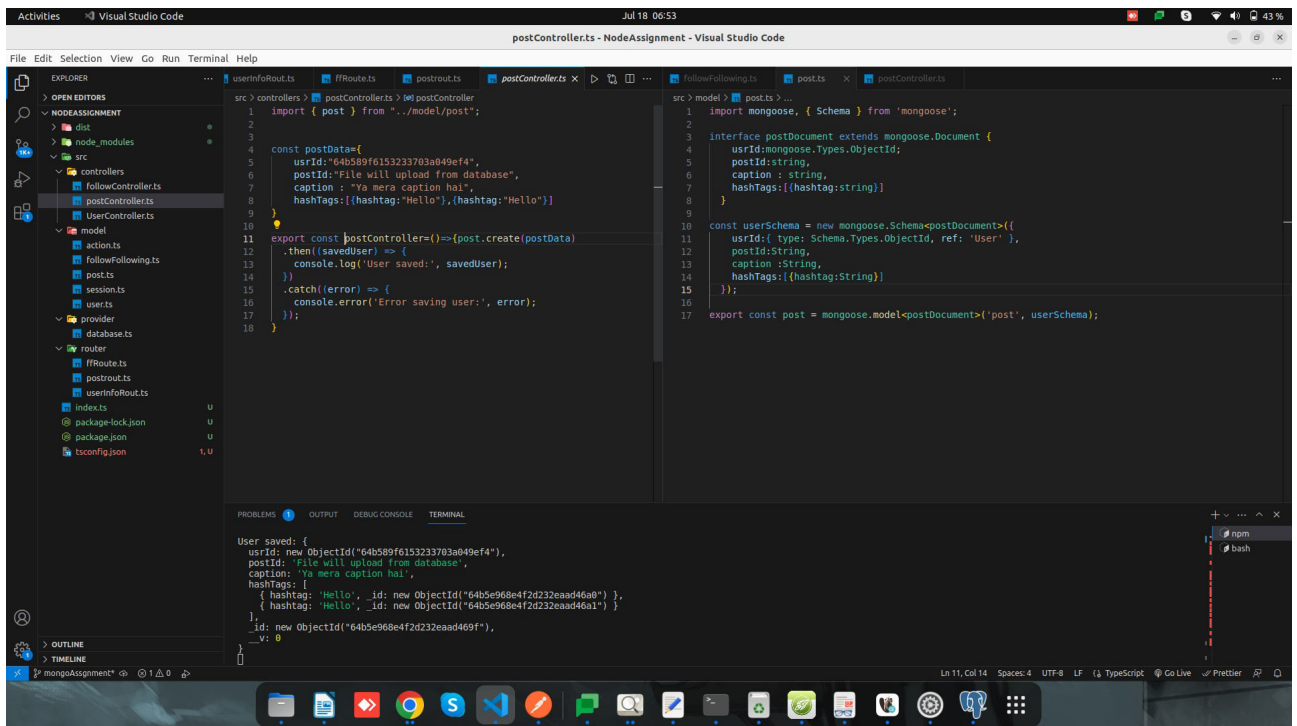
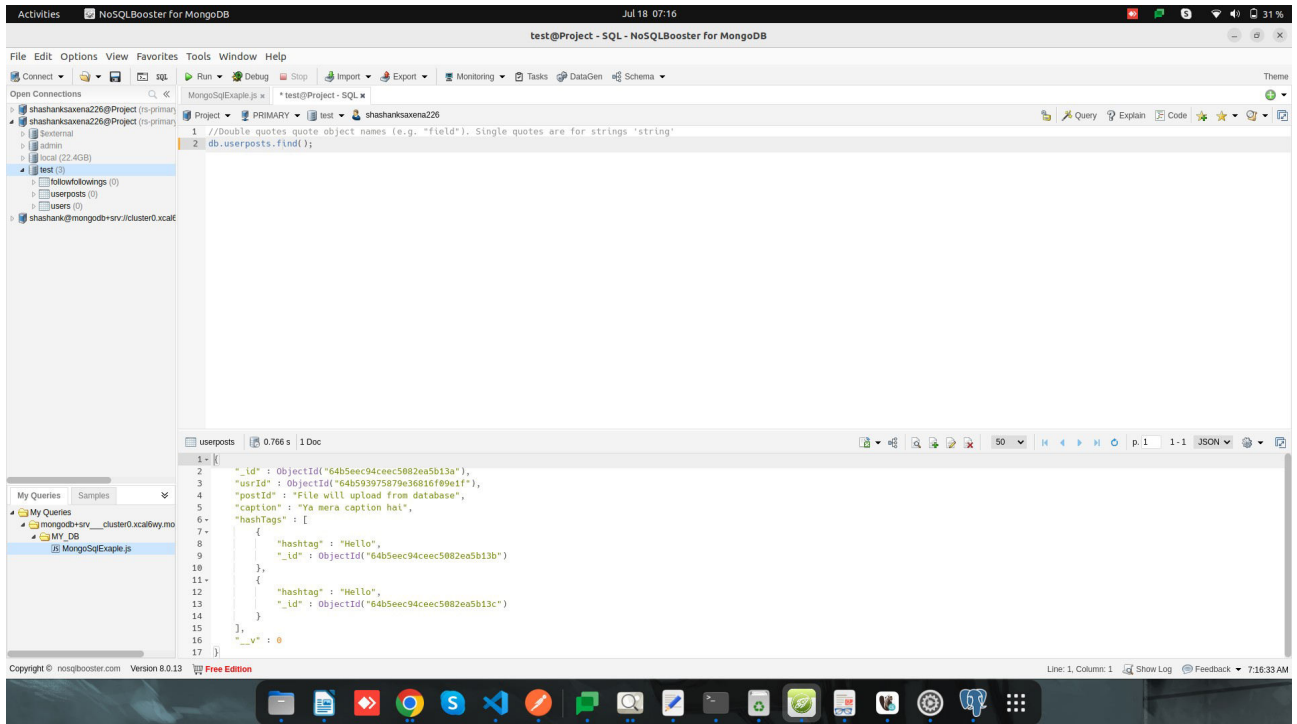
1.User Collection



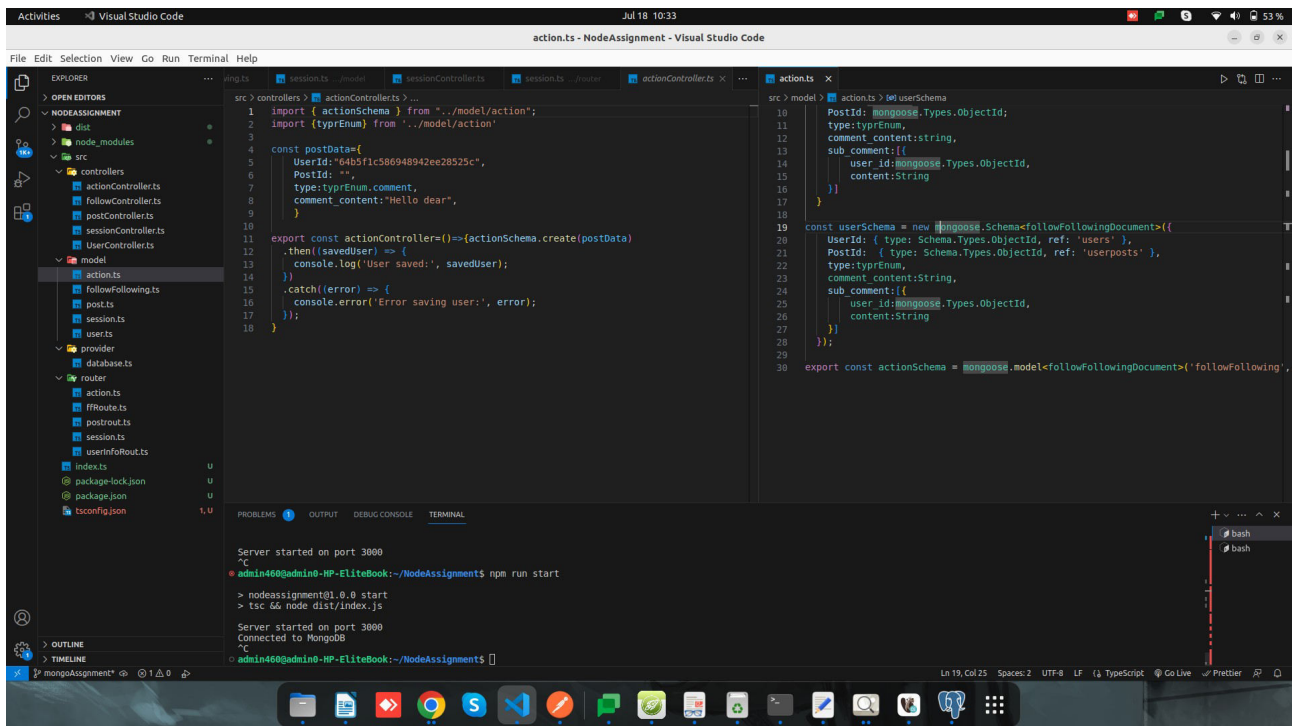
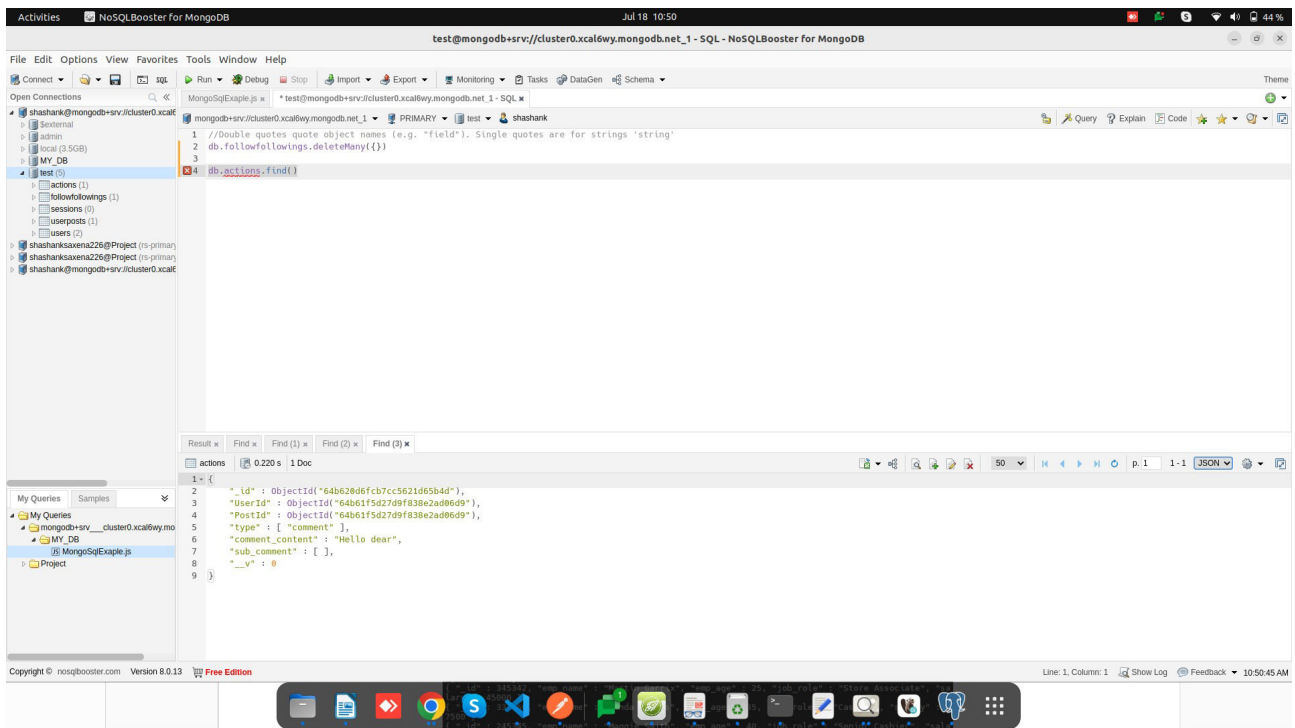
2. Follower/Following Table



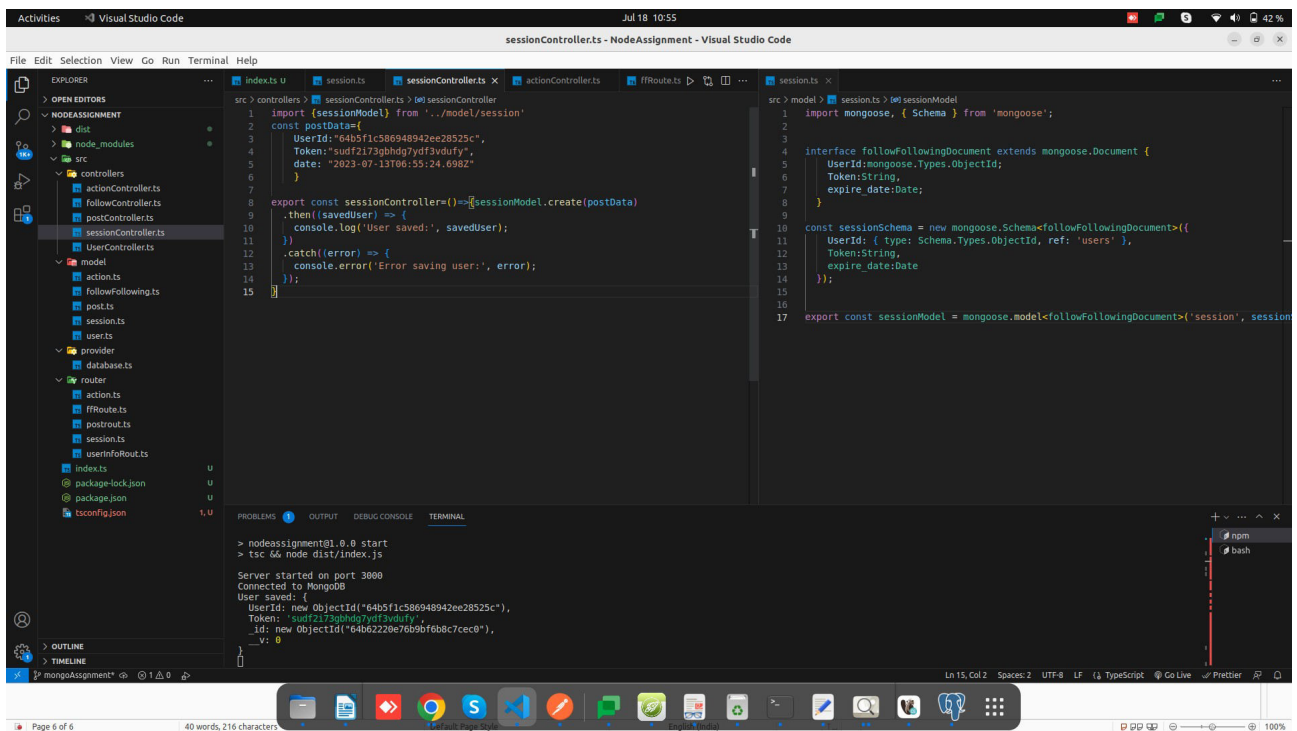
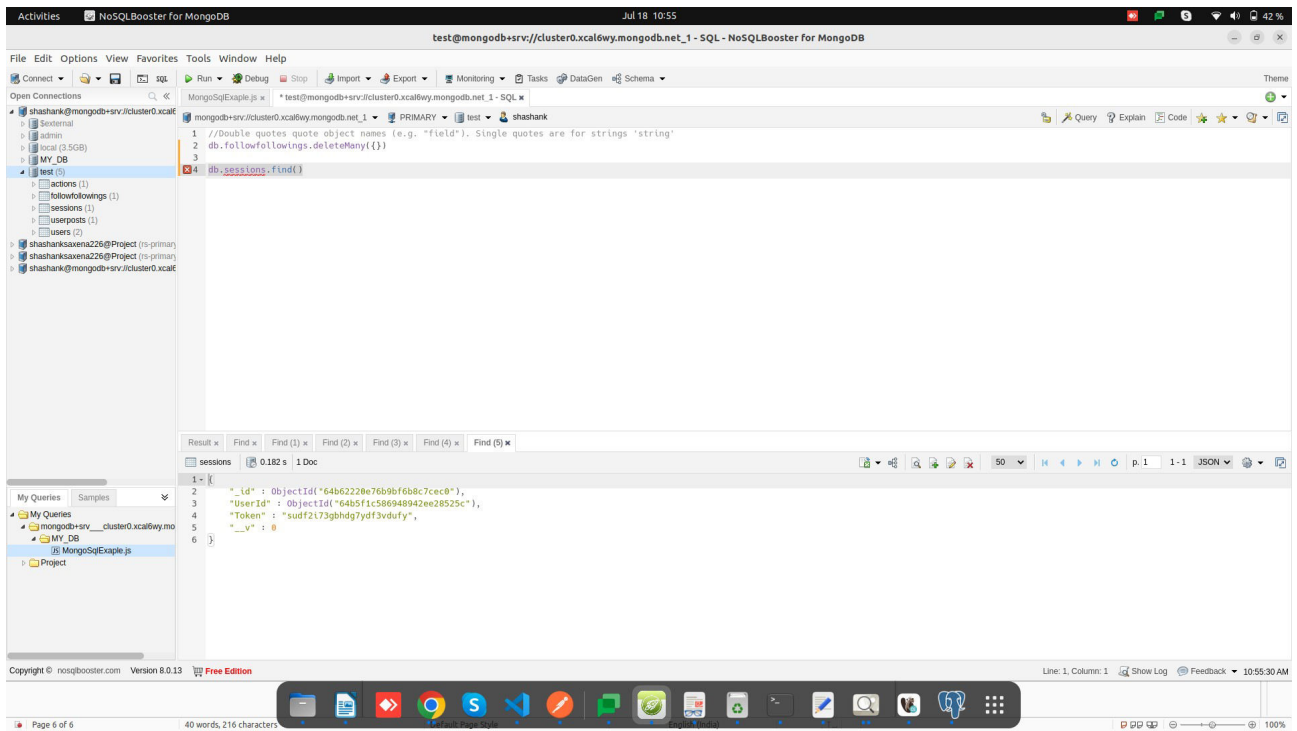
3. Post



4. Actions



5. Session



Queries

Look-up

Here we look-up from post to users details

The screenshot displays the NoSQLBooster for MongoDB interface. The main window shows a MongoDB query editor with the following aggregate query:

```
1 db.posts.aggregate([
2   $lookup: {
3     from: "users",
4     localField: "userId",
5     foreignField: "_id",
6     as: "common_id"
7   }
8 ])
```

The results pane shows the output of the query, which is a JSON array of documents. The first document is:

```
{
  "_id": "64b67e7311925a3774282b40",
  "userId": "64b61f5d27d9f838e2ad86d9",
  "postId": "Post1",
  "caption": "Caption",
  "hashtags": [
    {
      "hashtag": "post",
      "_id": "64b67e7311925a3774282b41"
    },
    {
      "hashtag": "newpost",
      "_id": "64b67e7311925a3774282b42"
    }
  ],
  "_v": 0,
  "common_id": [
    {
      "_id": "64b61f5d27d9f838e2ad86d9",
      "name": "Aditya",
      "email": "Aditya@example.com",
      "password": "Aditya1@.",
      "isPrivateAccount": true,
      "address": {
        "city": "Ghaziabad",
        "street": "prathap vihar"
      },
      "_v": 0
    }
  ]
}
```

The interface also shows a sidebar with a tree view of the database structure, including collections like 'posts', 'users', and 'common_id'. The bottom status bar indicates the version of NoSQLBooster (8.0.13) and the current document (Line 19, Column 4).

Operators

Seq operator

The screenshot shows the NoSQLBooster for MongoDB interface. The query editor contains the following JavaScript code:

```
1 db.users.find({"name" : {$seq:"Shashank"}})
```

The results pane shows two documents from the 'users' collection:

```
1 /* 1 createdAt:18/07/2023, 10:44:19+*/
2 {
3   "_id": ObjectId("64b61fab10bec633eb7daa88"),
4   "name": "Shashank",
5   "email": "Shashank@example.com",
6   "password": "Shashank@",
7   "isPrivateAccount": true,
8   "address": {
9     "city": "Ghaziabad",
10    "street": "prathap vihar"
11  },
12   "__v": 0
13 },
14
15 /* 2 createdAt:18/07/2023, 17:19:25+*/
16 {
17   "_id": ObjectId("64b6745307bda4538ead84a"),
18   "name": "Shashank",
19   "email": "Shashank@example.com",
20   "password": "Shashank@",
21   "isPrivateAccount": true,
22   "address": {
23     "city": "Ghaziabad",
24     "street": "prathap vihar"
25   },
26   "__v": 0
27 }
```

\$ne

The screenshot shows the NoSQLBooster for MongoDB interface. The query editor contains the following JavaScript code:

```
1 db.users.find({"name":{"$ne":"Shashank"}})
```

The results pane shows three documents from the 'users' collection:

```
1 /* 1 createdAt:18/07/2023, 10:43:01+*/
2 {
3   "_id": ObjectId("64b61f5d2709f838e2ad9ed9"),
4   "name": "Aditya",
5   "email": "Aditya@example.com",
6   "password": "Aditya1@",
7   "isPrivateAccount": true,
8   "address": {
9     "city": "Ghaziabad",
10    "street": "prathap vihar"
11  },
12   "__v": 0
13 },
14
15 /* 2 createdAt:18/07/2023, 16:48:03+*/
16 {
17   "_id": ObjectId("64b674eba177ac64572b4a33"),
18   "name": "Ashish",
19   "email": "Ashish@example.com",
20   "password": "Ashish@",
21   "isPrivateAccount": true,
22   "address": {
23     "street": "prathap vihar",
24     "city": "Ghaziabad",
25     "state": "Uttar Pradesh",
26     "country": "India"
27   },
28 },
29
30 /* 3 createdAt:18/07/2023, 16:48:30+*/
31 {
32   "_id": ObjectId("64b67586a177ac64572b4a34"),
33   "name": "Ashu",
34   "email": "Ashu@example.com",
35   "password": "Ashu@",
36   "isPrivateAccount": true,
37   "address": {
38     "street": "nathan vihar"
39   },
40 }
```


\$in

test@mongodb+srv://cluster0.xcalfwy.mongodb.net_1 - SQL - NoSQLBooster for MongoDB

```
1 db.users.find({'name':{'$in':['Aditya','Ashish']}})
```

Find (1) x Find (1) x

users 0.134 s 4 Docs

```
1 /* 1 createdAt:18/07/2023, 10:43:01*/
2 {
3   "_id" : ObjectId("64b61f5d27d9f838e2ad06d9"),
4   "name" : "Aditya",
5   "email" : "Aditya@example.com",
6   "password" : "Aditya@123",
7   "isPrivateAccount" : true,
8   "address" : {
9     "city" : "Ghaziabad",
10    "street" : "prathap vihar"
11  },
12   "_v" : 0
13 },
14 },
15 /* 2 createdAt:18/07/2023, 16:48:03*/
16 {
17   "_id" : ObjectId("64b674eba177ac64572b4a33"),
18   "name" : "Ashish",
19   "email" : "Ashish@example.com",
20   "password" : "Ashish@123",
21   "isPrivateAccount" : true,
22   "address" : {
23     "street" : "prathap vihar",
24     "city" : "Ghaziabad",
25     "state" : "Uttar Pradesh",
26     "country" : "India"
27   },
28 },
29 },
30 /* 3 createdAt:18/07/2023, 17:28:18*/
31 {
32   "_id" : ObjectId("64b67c7a673ad69ad895732e"),
33   "name" : "Aditya",
34   "email" : "Aditya@example.com",
35   "password" : "Aditya@123",
36   "isPrivateAccount" : true,
37   "address" : {
38     "city" : "Ghaziabad",
```

\$nin

test@mongodb+srv://cluster0.xcalfwy.mongodb.net_1 - SQL - NoSQLBooster for MongoDB

```
1 db.users.find({'name':{'$nin':['Aditya','Shashank']}})
```

Find (1) x Find (2) x

users 0.213 s 4 Docs

```
1 /* 1 createdAt:18/07/2023, 16:48:03*/
2 {
3   "_id" : ObjectId("64b674eba177ac64572b4a33"),
4   "name" : "Ashish",
5   "email" : "Ashish@example.com",
6   "password" : "Ashish@123",
7   "isPrivateAccount" : true,
8   "address" : {
9     "street" : "prathap vihar",
10    "city" : "Ghaziabad",
11    "state" : "Uttar Pradesh",
12    "country" : "India"
13   },
14 },
15 },
16 /* 2 createdAt:18/07/2023, 16:48:38*/
17 {
18   "_id" : ObjectId("64b67596a177ac64572b4a34"),
19   "name" : "Ashu",
20   "email" : "Ashu@example.com",
21   "password" : "Ashu@123",
22   "isPrivateAccount" : true,
23   "address" : {
24     "street" : "prathap vihar",
25     "city" : "Ghaziabad",
26     "state" : "Uttar Pradesh",
27     "country" : "India"
28   },
29 },
30 },
31 /* 3 createdAt:18/07/2023, 16:48:53*/
32 {
33   "_id" : ObjectId("64b6751da177ac64572b4a35"),
34   "name" : "Dhairya",
35   "email" : "Dhairya@example.com",
36   "password" : "Dhairya@123",
37   "isPrivateAccount" : true,
38   "address" : {
```

Logical Operator

\$and

The screenshot shows the NoSQLBooster for MongoDB interface. The query editor contains the following SQL query:

```
1 db.users.find({$and:({name:'Aditya'},({isPrivateAccount:true})})}
```

The results pane shows two documents:

```
1 /* 1 createdAt:18/07/2023, 10:43:01+*/
2 {
3   "_id": ObjectId("64b61f5d27d9f838e2ad86d9"),
4   "name": "Aditya",
5   "email": "Aditya@exampl.com",
6   "password": "Aditya1@.",
7   "isPrivateAccount": true,
8   "address": {
9     "city": "Ghaziabad",
10    "street": "prathap vihar"
11  },
12  "_v": 0
13 },
14
15 /* 2 createdAt:18/07/2023, 17:20:18+*/
16 {
17   "_id": ObjectId("64b67c7a673ad69ad895732e"),
18   "name": "Aditya",
19   "email": "Aditya@exampl.com",
20   "password": "Aditya@.",
21   "isPrivateAccount": true,
22   "address": {
23     "city": "Ghaziabad",
24     "street": "prathap vihar"
25   },
26   "_v": 0
27 }
```

\$or

The screenshot shows the NoSQLBooster for MongoDB interface. The query editor contains the following SQL query:

```
1 db.users.find({$or:({name:'Aditya'},({email:'Ashish@gmail.com'})})}
```

The results pane shows two documents:

```
1 /* 1 createdAt:18/07/2023, 10:43:01+*/
2 {
3   "_id": ObjectId("64b61f5d27d9f838e2ad86d9"),
4   "name": "Aditya",
5   "email": "Aditya@exampl.com",
6   "password": "Aditya1@.",
7   "isPrivateAccount": true,
8   "address": {
9     "city": "Ghaziabad",
10    "street": "prathap vihar"
11  },
12  "_v": 0
13 },
14
15 /* 2 createdAt:18/07/2023, 17:20:18+*/
16 {
17   "_id": ObjectId("64b67c7a673ad69ad895732e"),
18   "name": "Aditya",
19   "email": "Aditya@exampl.com",
20   "password": "Aditya@.",
21   "isPrivateAccount": true,
22   "address": {
23     "city": "Ghaziabad",
24     "street": "prathap vihar"
25   },
26   "_v": 0
27 }
```

\$exist

The screenshot shows the NoSQLBooster for MongoDB interface. The query editor contains the following code:

```
1 db.users.find({'name':{'$exists':true,$eq:"Shashank"}})
```

The results pane shows two documents:

Key	Value	Type
(1) 64b61fab1f6ec533eb7daa88	{ name: "Shashank", email: "Shashank@example.com" }	Document
(2) 64b67c45307bda4538ead04a	{ name: "Shashank", email: "Shashank@example.com" }	Document

\$type

The screenshot shows the NoSQLBooster for MongoDB interface. The query editor contains the following code:

```
1 db.users.find({'name':{'$type':"string"}})
```

The results pane shows eight documents:

Key	Value	Type
(1) 64b61f5d27d9838e2ad06d9	{ name: "Aditya", email: "Aditya@example.com" }	Document
(2) 64b61fab1f6ec533eb7daa88	{ name: "Shashank", email: "Shashank@example.com" }	Document
(3) 64b674eb177ac54572b4a33	{ name: "Ashish", email: "Ashish@example.com" }	Document
(4) 64b67506a177ac54572b4a34	{ name: "Ashu", email: "Ashu@example.com" }	Document
(5) 64b6751a177ac54572b4a35	{ name: "Shairya", email: "Shairya@example.com" }	Document
(6) 64b67c45307bda4538ead04a	{ name: "Shashank", email: "Shashank@example.com" }	Document
(7) 64b67c7a673ad69a895732e	{ name: "Aditya", email: "Aditya@example.com" }	Document
(8) 64b67c9f77b4ddc1246c393	{ name: "Ashish", email: "Ashish@example.com" }	Document

\$Pipeline and sort function

The screenshot displays the NoSQLBooster for MongoDB application interface. The main window shows an aggregation pipeline being executed on a MongoDB database. The pipeline consists of a single stage: `db.actions.aggregate([{$match: {'type': 'comment'}}, {$project: {'comment_content': 1, 'type': 1}}, {$sort: {'_id': -1}}])`. The results are displayed in a table format, showing four documents. Each document has an `_id` (ObjectId), a `type` (comment), and a `comment_content` (Hello friends). The documents are sorted by `_id` in descending order.

The interface includes a sidebar with a tree view of the database structure, showing collections like `actions`, `comments`, `sessions`, `users`, and `usersposts`. The main window has a menu bar (File, Edit, Options, View, Favorites, Tools, Window, Help) and a toolbar with various icons for connecting, running, debugging, and exporting data. The status bar at the bottom indicates the version (8.0.13) and the edition (Free Edition).

```
1
2
3 db.actions.aggregate([
4   $match: {'type': 'comment'},
5   $project: {'comment_content': 1, 'type': 1},
6   $sort: {'_id': -1}
7 ])
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
```

Results (4 documents):

```
1 {
2   "_id": ObjectId("64b6814e5dceea811768b83b"),
3   "type": "comment",
4   "comment_content": "Hello friends"
5 },
6
7
8 /* 2 createdAt: 18/07/2023, 17:39:08 */
9 {
10  "_id": ObjectId("64b680e4a31236083bad71ec"),
11  "type": "comment",
12  "comment_content": "Hello friends"
13 },
14
15 /* 3 createdAt: 18/07/2023, 17:38:48 */
16 {
17  "_id": ObjectId("64b680c8dd569a4f2baaa28a"),
18  "type": "comment",
19  "comment_content": "Hello friends"
20 },
21
22 /* 4 createdAt: 18/07/2023, 17:38:17 */
23 {
24  "_id": ObjectId("64b680b1e60cc695f3b1ac"),
25  "type": "comment",
26  "comment_content": "Hello friends"
27 }
```