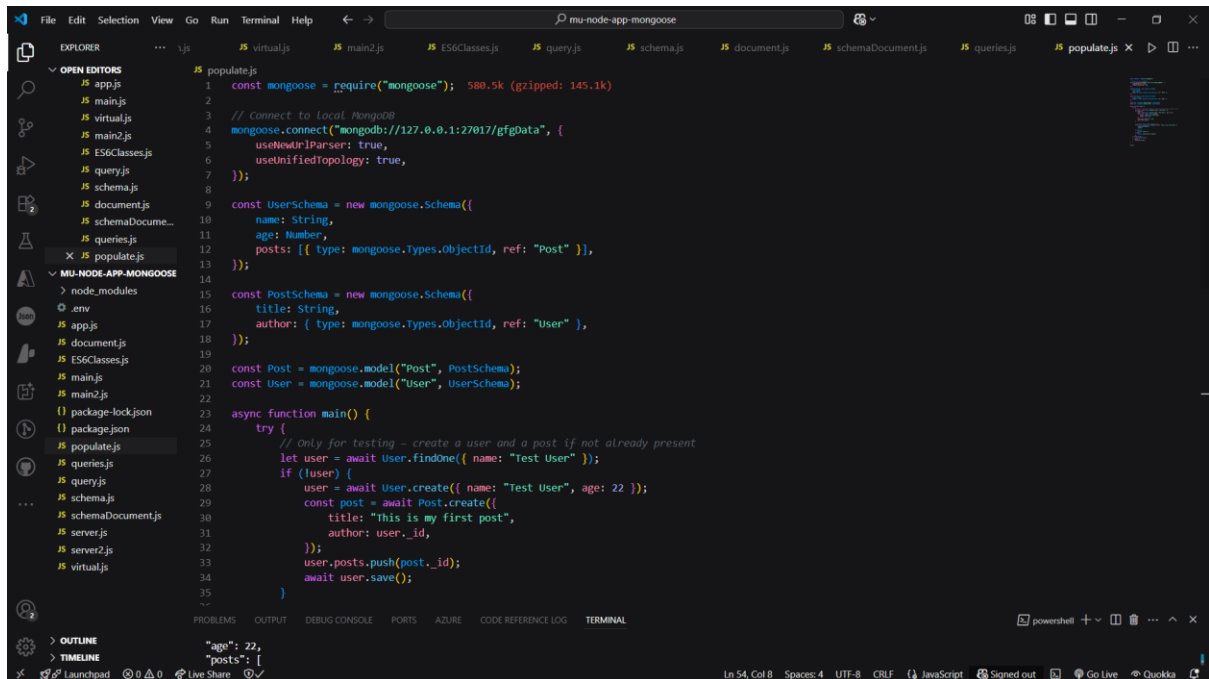
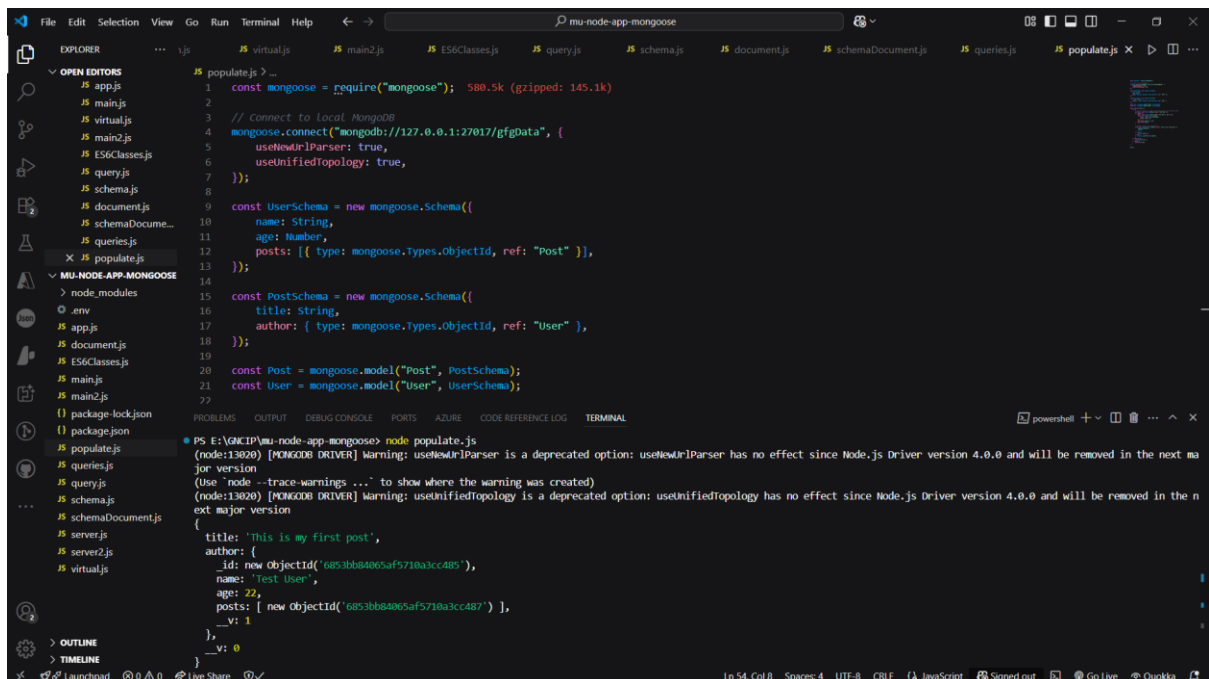


19 June 2025

Session 1



```
1 const mongoose = require("mongoose"); 588.5k (gzipped: 145.1k)
2
3 // Connect to Local MongoDB
4 mongoose.connect("mongodb://127.0.0.1:27017/gfgData", {
5   useNewUrlParser: true,
6   useUnifiedTopology: true,
7 });
8
9 const UserSchema = new mongoose.Schema({
10   name: String,
11   age: Number,
12   posts: [{ type: mongoose.Types.ObjectId, ref: "Post" }],
13 });
14
15 const PostSchema = new mongoose.Schema({
16   title: String,
17   author: { type: mongoose.Types.ObjectId, ref: "User" },
18 });
19
20 const Post = mongoose.model("Post", PostSchema);
21 const User = mongoose.model("User", UserSchema);
22
23 async function main() {
24   try {
25     // Only for testing - create a user and a post if not already present
26     let user = await User.findOne({ name: "Test User" });
27     if (!user) {
28       user = await User.create({ name: "Test User", age: 22 });
29       const post = await Post.create({
30         title: "This is my first post",
31         author: user._id,
32       });
33       user.posts.push(post._id);
34       await user.save();
35     }
36   }
37 }
```



```
PS E:\GNCIP\mu-node-app-mongoose> node populate.js
(node:13020) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(Use 'node --trace-warnings ...' to show where the warning was created)
(node:13080) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
{
  title: 'This is my first post',
  author: {
    _id: new ObjectId('6853bb84065af5710a3cc485'),
    name: 'Test User',
    age: 22,
    posts: [ new ObjectId('6853bb84065af5710a3cc487') ],
    _v: 1
  },
  _v: 0
}
```

```
1 const mongoose = require("mongoose"); 580.5k (gzipped: 145.1k)
2
3 // Connect to Local MongoDB
4 mongoose.connect("mongodb://127.0.0.1:27017/gfgData", {});
5
6 const UserSchema = new mongoose.Schema({
7   name: String,
8   age: Number,
9   posts: [{ type: mongoose.Types.ObjectId, ref: "Post" }],
10 });
11
12 const PostSchema = new mongoose.Schema({
13   title: String,
14   author: { type: mongoose.Types.ObjectId, ref: "User" },
15   comments: [{ type: mongoose.Types.ObjectId, ref: "Comment" }],
16 });
17
18 const commentSchema = new mongoose.Schema({
19   text: String,
20   author: { type: mongoose.Types.ObjectId, ref: "User" }
21 });
22
23 const Post = mongoose.model("Post", PostSchema);
24 const User = mongoose.model("User", UserSchema);
25 const Comment = mongoose.model("Comment", commentSchema);
26
27 async function main() {
28   try {
29     // Only for testing - create a user and a post if not already present
30
31     // Populate author when fetching post
32     const user = await User.findOne({ name: 'Test User' }).populate({
33       path: 'posts',
34       populate: {
35         path: 'comments',
36         populate: {
37           path: 'author',
38           select: '-password'
39         }
40       }
41     });
42     console.log('User and posts found:', user);
43   } catch (error) {
44     console.error('Error:', error);
45   }
46 }
47
48 main();
```

```
PS E:\GNCIP\mu-node-app-mongoose> node populate.js
{"_id": "6853bb84865af5718a3cc485",
  "name": "Test User",
  "age": 22,
  "posts": [
    {
      "comments": [],
      "_id": "6853bb84865af5718a3cc487",
      "title": "This is my first post",
      "author": "6853bb84865af5718a3cc485",
      "__v": 0
    }
  ],
  "__v": 1
}
```

SESSION 2

```
1 const mongoose = require('mongoose'); 580.5k (gzipped: 145.1k)
2 const { User, Post } = require('./model'); // Import models
3
4 // Connect to MongoDB
5 mongoose.connect('mongodb://localhost:27017/GFG', {
6   useNewUrlParser: true,
7   useUnifiedTopology: true
8 })
9 .then(() => console.log("connected to MongoDB"))
10 .catch((err) => console.error("Connection error:", err));
11
12 // Function to create and populate
13 async function run() {
14   await mongoose.connection.dropDatabase(); // optional: clear old data
15
16   const user = new User({ username: 'vansh', email: 'vansh@example.com' });
17   await user.save();
18
19   const post = new Post({ title: 'My first post', postedBy: user._id });
20   await post.save();
21
22   // Populate and show results
23   const populatedPost = await Post.find().populate('postedBy');
24   console.log("Post with user:", populatedPost);
25
26   mongoose.connection.close(); // Close connection
27 }
28
29 run();
30
31 _id: new ObjectId('6853f29768960af1be36eaf'),
32 username: 'vansh',
33 email: 'vansh@example.com',
34 __v: 0
35 }
```

```
1 const mongoose = require('mongoose'); 580.5k (gzipped: 145.1k)
2
3 // User Schema
4 const userSchema = new mongoose.Schema({
5   username: String,
6   email: String
7 });
8
9 // Post Schema with reference to User
10 const postSchema = new mongoose.Schema({
11   title: String,
12   postedBy: {
13     type: mongoose.Schema.Types.ObjectId,
14     ref: "User"
15   }
16 });
17
18 // Create Models
19 const User = mongoose.model('User', userSchema);
20 const Post = mongoose.model('Post', postSchema);
21
22 // Export models only (do NOT export undefined variables)
23 module.exports = {
24   User,
25   Post
26 };
27
28 _id: new ObjectId('6853f29768960af1be36eaf'),
29 username: 'vansh',
30 email: 'vansh@example.com',
31 __v: 0
32 }
```

```
(Use 'node --trace-warnings ...' to show where the warning was created)
(node:9148) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the ne
xt major version
Connected to MongoDB
Post with user: [
  {
    id: new ObjectId('6853f29768960af1be36ea11'),
    title: 'My first post',
    postedBy: {
      id: new ObjectId('6853f29768960af1be36ea11'),
      title: 'My first post',
      username: 'vansh',
      email: 'vansh@example.com',
      _v: 0
    },
    _v: 0
  }
]
PS E:\GNCIP\mu-node-app-mongoose>
```

```
1 const mongoose = require('mongoose');
2
3 // Connect to MongoDB
4 mongoose.connect('mongodb://localhost:27017/GfG', {
5   useNewUrlParser: true,
6   useUnifiedTopology: true
7 });
8
9 .then(() => console.log("Connected to MongoDB"))
10 .catch((err) => console.error("Connection error:", err));
11
12 // Creating Schemas
13 const userSchema = new mongoose.Schema({
14   username: String,
15   email: String
16 });
17
18 const postSchema = new mongoose.Schema({
19   title: String,
20   postedBy: {
21     type: mongoose.Schema.Types.ObjectId,
22     ref: "Users"
23   }
24 });
25
26 for version
27 (Use 'node --trace-warnings ...' to show where the warning was created)
28 (node:10648) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in the n
ext major version
29 Connected to MongoDB
30 [
31   {
32     id: new ObjectId('6853f29768960af1be36ea11'),
33     title: 'My first post',
34     postedBy: new ObjectId('6853f29768960af1be36ea11'),
35     _v: 0
36   }
37 ]
```

```
File Edit Selection View Go Run Terminal Help mu-node-app-mongoose
EXPLORER
  JS virtual.js
  JS main2.js
  JS ES6Classes.js
  JS query.js
  JS schema.js
  JS document.js
  JS schemaDocument.js
  JS queries.js
  JS populate.js
  JS model.js
  MU-NODE-APP-MONGOOSE
    node_modules
    .env
    app.js
    document.js
    ES6Classes.js
    main.js
    main2.js
    model.js
    package-lock.json
    package.json
    populate.js
    queries.js
    query.js
    schema.js
    schemaDocument.js
    server.js
    server2.js
    virtual.js
    virtualpopulated.js
  OUTLINE
  TIMELINE
  Launchpad 0 0 0 Live Share
  JS populate.js
  8
  9
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20
  21
  22
  23
  24
  25
  26
  27
  28
  29
  30
  31
  PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE CODE REFERENCE LOG TERMINAL
  node
  xt major version
  Connected to MongoDB
  [
    {
      _id: new ObjectId('6853f29768960af1be36ea11'),
      title: 'My first post',
      postedBy: {
        _id: new ObjectId('6853f29768960af1be36ea0f'),
        username: 'vansh',
        email: 'vansh@example.com',
        __v: 0
      }
    }
  ]
  Ln 36, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Signed out Go Live Quokka
```

```
File Edit Selection View Go Run Terminal Help mu-node-app-mongoose
EXPLORER
  JS ES6Classes.js
  JS query.js
  JS schema.js
  JS document.js
  JS schemaDocument.js
  JS queries.js
  JS populate.js
  JS model.js
  JS virtualpopulated.js
  MU-NODE-APP-MONGOOSE
    node_modules
    .env
    app.js
    document.js
    ES6Classes.js
    main.js
    main2.js
    model.js
    package-lock.json
    package.json
    populate.js
    queries.js
    query.js
    schema.js
    schemaDocument.js
    server.js
    server2.js
    virtual.js
    virtualpopulated.js
  OUTLINE
  TIMELINE
  Launchpad 0 0 0 Live Share
  JS virtualpopulated.js
  1
  2
  3
  4
  5
  6
  7
  8
  9
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20
  21
  22
  23
  24
  25
  26
  27
  28
  29
  30
  31
  32
  PROBLEMS OUTPUT DEBUG CONSOLE PORTS AZURE CODE REFERENCE LOG TERMINAL
  powershell
  const mongoose = require("mongoose");
  mongoose.set("strictQuery", true);
  mongoose.connect("mongodb://localhost:27017/geeksforgoeks", {
    useNewUrlParser: true,
  });
  // Define User schema
  const userSchema = new mongoose.Schema({
    first: String,
    last: String,
  });
  // Add virtual field fullName
  userSchema.virtual("fullName").get(function () {
    return `${this.first} ${this.last}`;
  });
  // Enable virtuals in output
  userSchema.set("toObject", { virtuals: true });
  userSchema.set("toJSON", { virtuals: true });
  // Book schema
  const bookSchema = new mongoose.Schema({
    title: String,
    author: {
      type: mongoose.Schema.Types.ObjectId,
      ref: "User"
    },
  });
  const User = mongoose.model("User", userSchema);
  const Book = mongoose.model("Book", bookSchema);
  first: 'John',
  last: 'Doe'
  },
  _id: new ObjectId('6853f8e8948465e28c699e41'),
  title: 'My Book Title',
  author: {
  Ln 29, Col 4 Spaces: 4 UTF-8 CRLF JavaScript Signed out Go Live Quokka
```

```
File Edit Selection View Go Run Terminal Help
mu-node-app-mongoose

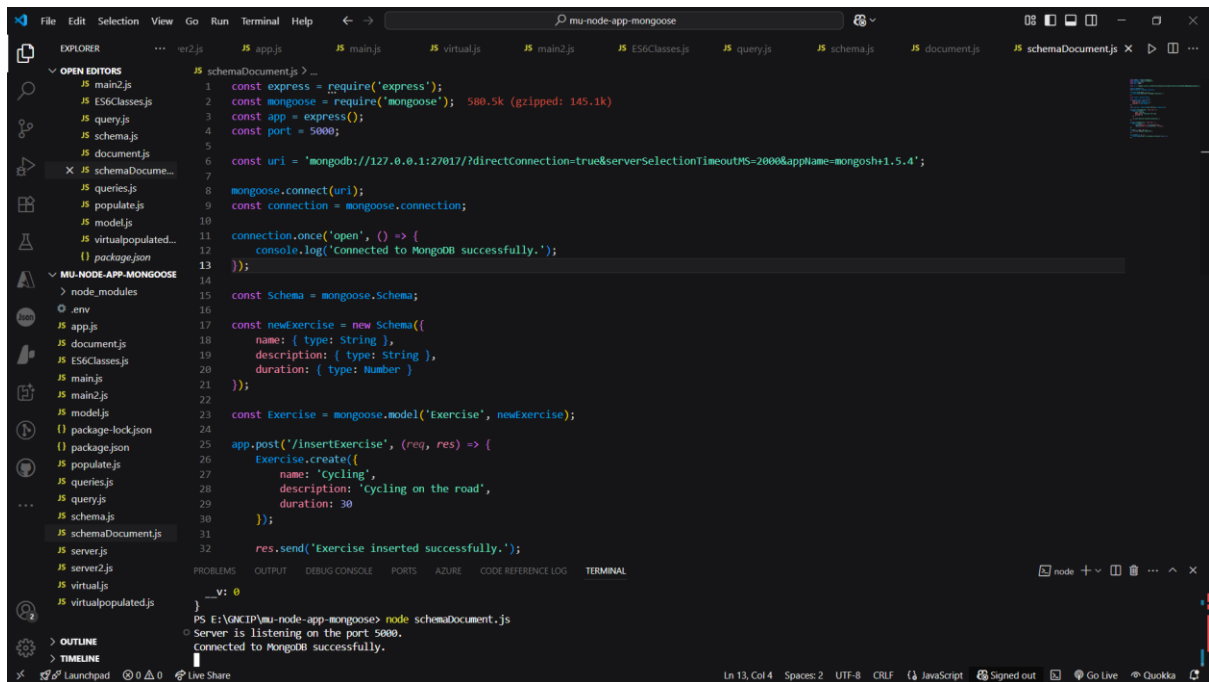
EXPLORER
virtual.js
main2.js
ES6Classes.js
query.js
schema.js
document.js
schemaDocument.js
queries.js
populate.js
model.js
virtualpopulated.js
mu-node-app-mongoose
node_modules
env
app.js
document.js
ES6Classes.js
main.js
main2.js
model.js
package-lock.json
package.json
populate.js
queries.js
query.js
schema.js
schemaDocument.js
server.js
server2.js
virtual.js
virtualpopulated.js

TERMINAL
PS E:\GKIP\mu-node-app-mongoose> node virtualpopulated.js
{
  _id: new ObjectId('6853f8e8948465e28c699e41'),
  title: 'My Book Title',
  author: {
    _id: new ObjectId('6853f8e8948465e28c699e3f'),
    first: 'John',
    last: 'Doe'
  },
  _id: new ObjectId('6853f8e8948465e28c699e41'),
  title: 'My Book Title',
  author: {
    _id: new ObjectId('6853f8e8948465e28c699e3f'),
    first: 'John',
    last: 'Doe'
  },
  _v: 0
}
Title: My Book Title
Author: John Doe
PS E:\GKIP\mu-node-app-mongoose>
```

```
File Edit Selection View Go Run Terminal Help
mu-node-app-mongoose

EXPLORER
query.js
schema.js
document.js
schemaDocument.js
queries.js
populate.js
model.js
virtualpopulated.js
package.json
env
app.js
document.js
ES6Classes.js
main.js
main2.js
model.js
package-lock.json
package.json
populate.js
queries.js
query.js
schema.js
schemaDocument.js
server.js
server2.js
virtual.js
virtualpopulated.js

TERMINAL
node virtualpopulated.js
{
  _id: new ObjectId('6853f8e8948465e28c699e41'),
  title: 'My Book Title',
  author: {
    _id: new ObjectId('6853f8e8948465e28c699e3f'),
    first: 'John',
    last: 'Doe'
  },
  _id: new ObjectId('6853f8e8948465e28c699e41'),
  title: 'My Book Title',
  author: {
    _id: new ObjectId('6853f8e8948465e28c699e3f'),
    first: 'John',
    last: 'Doe'
  },
  _v: 0
}
Title: My Book Title
Author: John Doe
PS E:\GKIP\mu-node-app-mongoose>
```



```
1 const express = require('express');
2 const mongoose = require('mongoose');
3 const app = express();
4 const port = 5000;
5
6 const uri = 'mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.5.4';
7
8 mongoose.connect(uri);
9 const connection = mongoose.connection;
10
11 connection.once('open', () => {
12   console.log('connected to MongoDB successfully.');

PS E:\GHC1P\mu-node-app-mongoose> node schemaDocument.js

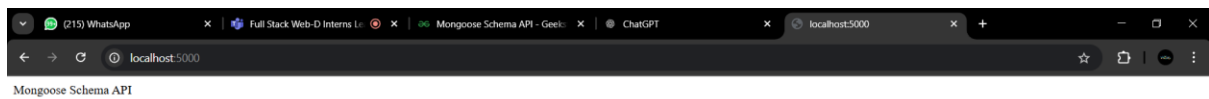


Server is listening on the port 5000.



connected to MongoDB successfully.


```



HOPPSCOTCH

Try "Import collections" Ctrl K

Save My Workspace Login

POST Untitled +

Select environment

POST http://localhost:5000/insertExercise Send Save

Parameters Body Headers Authorization Pre-request Script Tests Variables

Content Type application/x-www-form-urlencoded Override

Request Body

name	Vaishnavi	✓	✗
_csrf	Y3ZAdH7l--mlc_8m06MfhMVmb9xJVqJRIU	✓	✗

Status: 200 OK Time: 83 ms Size: 259 B

HTML Raw Headers Test Results

Response Body

```
1 Exercise inserted successfully.
```

Personal Workspace > Collections

Search

+ New

Collections are empty

Import or create a collection

Import

+ Add new

Help & feedback

HOPPSCOTCH

Try "Create environment" Ctrl K

Save My Workspace Login

GET Untitled +

Select environment

GET http://localhost:5000/Findexercis Send Save

Parameters Body Headers Authorization Pre-request Script Tests Variables

Content Type application/x-www-form-urlencoded Override

Request Body

name	Vaishnavi	✓	✗
_csrf	Y3ZAdH7l--mlc_8m06MfhMVmb9xJVqJRIU	✓	✗

Status: 200 OK Time: 15 ms Size: 345 B

JSON Raw Headers Test Results

Response Body

```
1 {
2   {
3     "_id": "6854826772f94ab7b3ccab73",
4     "name": "Cycling",
5     "description": "Cycling on the road",
6     "duration": 30,
7     "_v": 0
8   }
9 }
```

Personal Workspace > Collections

Search

+ New

Collections are empty

Import or create a collection

Import

+ Add new

Help & feedback


```
localhost:5000/Findexercises

Pretty-print

[
  {
    "_id": "6854826772f94ab7b3ccab73",
    "name": "Cycling",
    "description": "Cycling on the road",
    "duration": 30,
    "__v": 0
  }
]
```

HOOPSCOTCH

Try "Create environment" Ctrl K

Save My Workspace Login

POST Untitled +

Select environment

POST http://localhost:5000/insertMobile Send Save

Parameters Body Headers Authorization Pre-request Script Tests Variables

Query Parameters

Key	Value	Description
-----	-------	-------------

Status: 200 OK Time: 93 ms Size: 256 B

HTML Raw Headers Test Results

Response Body

```
1 Mobile inserted successfully.
```

Personal Workspace Collections

Search

+ New

Collections are empty

Import or create a collection

Import

+ Add new

Type here to search

34°C T-storms 18:13 19-06-2025

Browser tabs: (215) WhatsApp, Full Stack Web-D Inter, Mongoose Schema API, ChatGPT, Error, Hoppscotch • Open sou...

Address bar: hoppscotch.io

HOPPSCOTCH

Try "Share request" Ctrl K

Save My Workspace Login

GET **Untitled** +

Select environment

Parameters Body Headers Authorization Pre-request Script Tests Variables

Query Parameters

Key	Value	Description
-----	-------	-------------

Status: 200 • OK Time: 65 ms Size: 339 B

JSON Raw Headers Test Results

Response Body

```
1 {
2   "_id": "685405d4dcccff9be4f8d9e3",
3   "name": "Samsung 38s",
4   "price": 22000,
5   "rating": 4.5,
6   "isBig": true,
7   "__v": 0
8 }
```

Personal Workspace > Collections

Search

+ New

Collections are empty

Import or create a collection

Import

+ Add new

Help & feedback

Browser tabs: (215) WhatsApp, Full Stack Web-D Inter, Mongoose Schema API, ChatGPT, localhost:5000/findBefore..., Hoppscotch • Open sou...

Address bar: localhost:5000/findBeforeUpdatedMobile

Pretty-print

```
{
  "_id": "685405d4dcccff9be4f8d9e3",
  "name": "Samsung 38s",
  "price": 22000,
  "rating": 4.5,
  "isBig": true,
  "__v": 0
}
```

