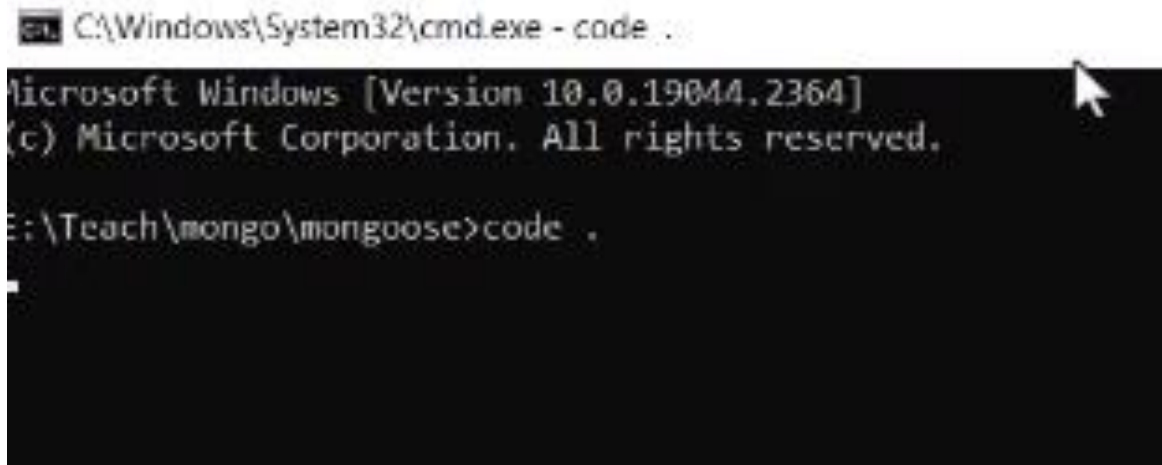


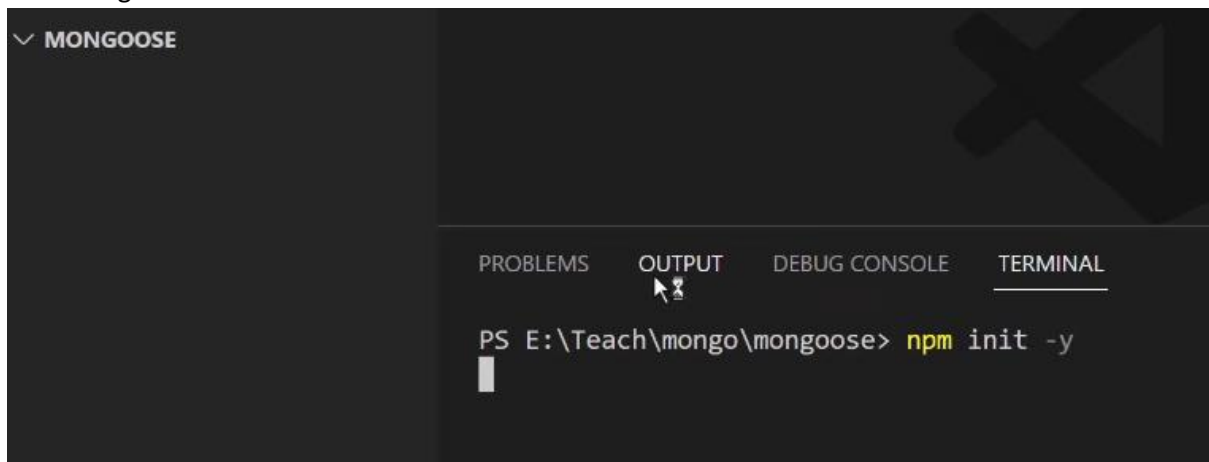
Create folder, type cmd in address bar, code .



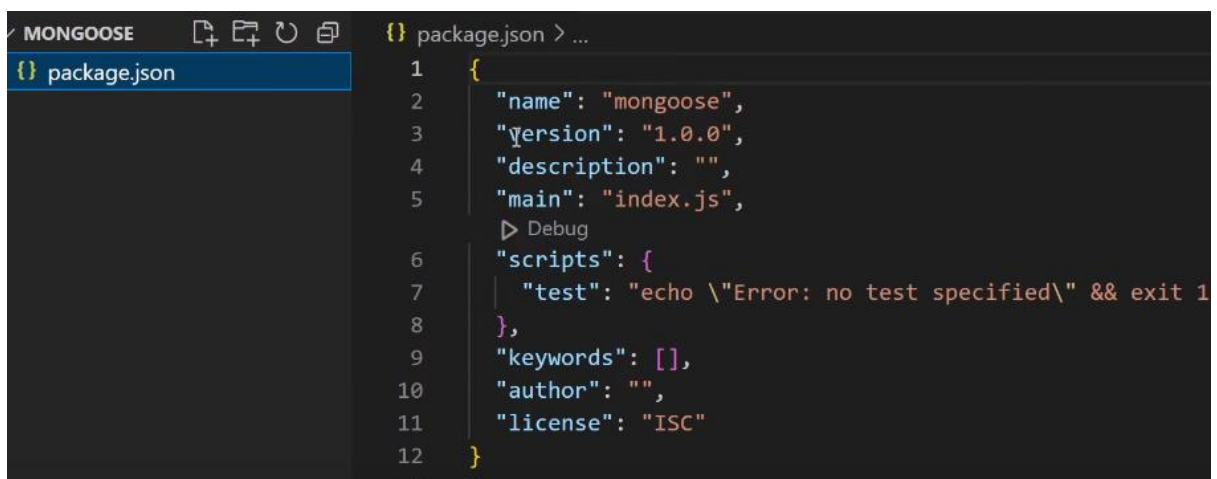
```
C:\Windows\System32\cmd.exe - code .  
Microsoft Windows [Version 10.0.19044.2364]  
(c) Microsoft Corporation. All rights reserved.  
E:\Teach\mongo\mongoose>code .
```

npm init : Initialize a new project and create a package. json file. npm install <package-name> : Install a package and add it to the dependencies object in your package. json file.

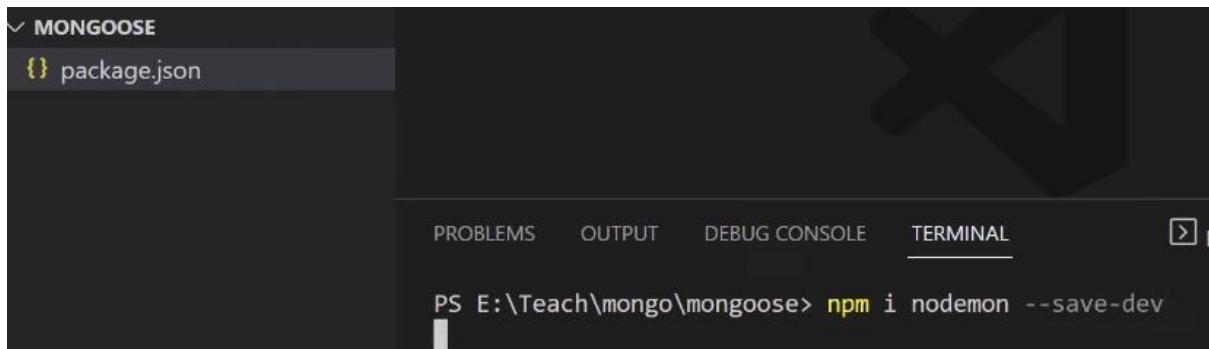
Y meaning : Yes



```
PS E:\Teach\mongo\mongoose> npm init -y
```



```
{  
  "name": "mongoose",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC"  
}
```



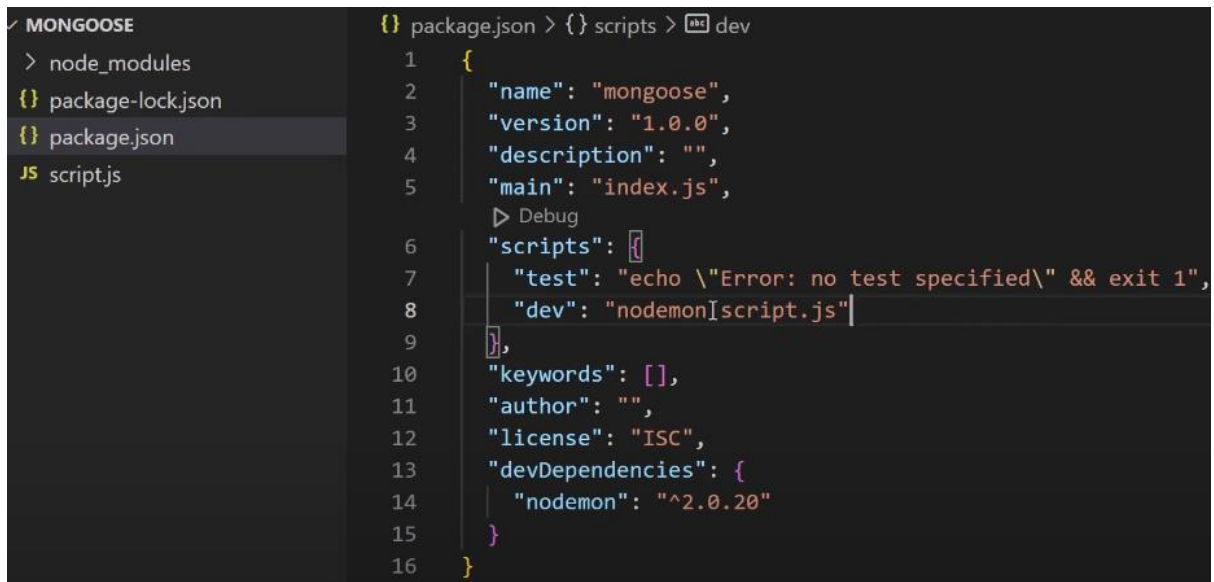
The screenshot shows the VS Code interface for a project named 'MONGOOSE'. The Explorer sidebar on the left shows a file tree with 'package.json' selected. The bottom panel shows the 'TERMINAL' tab with a PowerShell prompt at 'E:\Teach\mongo\mongoose' where the command 'npm i nodemon --save-dev' has been entered.

--save is default use (npm install, it will work after clone in github)

--save-dev is for development

When you use the --save-dev flag, the package is added to your devDependencies object.

if you download a repo from github and type npm install, the devDependencies are ignored



The screenshot shows the VS Code editor with the 'package.json' file open. The Explorer sidebar on the left shows the file tree with 'package.json' selected. The editor displays the following JSON content:

```
1 {
2   "name": "mongoose",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1",
8     "dev": "nodemon script.js"
9   },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "devDependencies": {
14    "nodemon": "^2.0.20"
15  }
16 }
```

Don't run every time: node script.js

Run only one time : npm run dev (it will take automatically refresh when saved)

```

{} package.json > {} scripts > dev
1 {
2   "name": "mongoose",
3   "version": "1.0.0",
4   "description": "",
}

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS E:\Teach\mongo\mongoose> npm run dev

> mongoose@1.0.0 dev
> nodemon script.js

[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node script.js`
[nodemon] clean exit - waiting for changes before restart

```

```

JS script.js
1

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS E:\Teach\mongo\mongoose> npm i mongoose

added 101 packages, and audited 134 packages in 18s

```

Terminal hide/show shortcut :ctrl +~

```

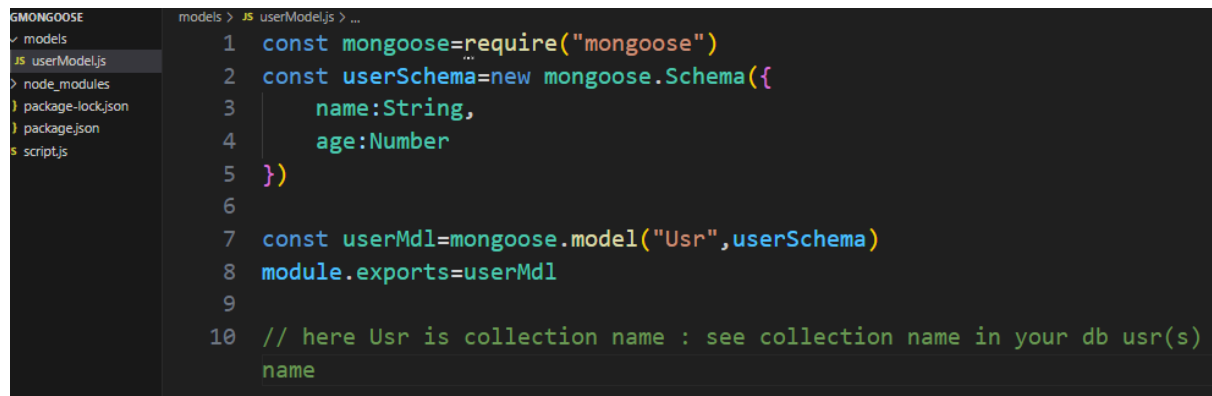
JS script.js > ...
1 const mongoose=require('mongoose')
2
3 mongoose.connect('mongodb://127.0.0.1:27017/gtest').then
4 (()=>{
5   console.log('connected')
6 }).catch(()=>{
7   console.log('connection error')
8 })

```

Schema : inbuilt class of mongoose

Use: only allow to store schema defined keys and datatype.

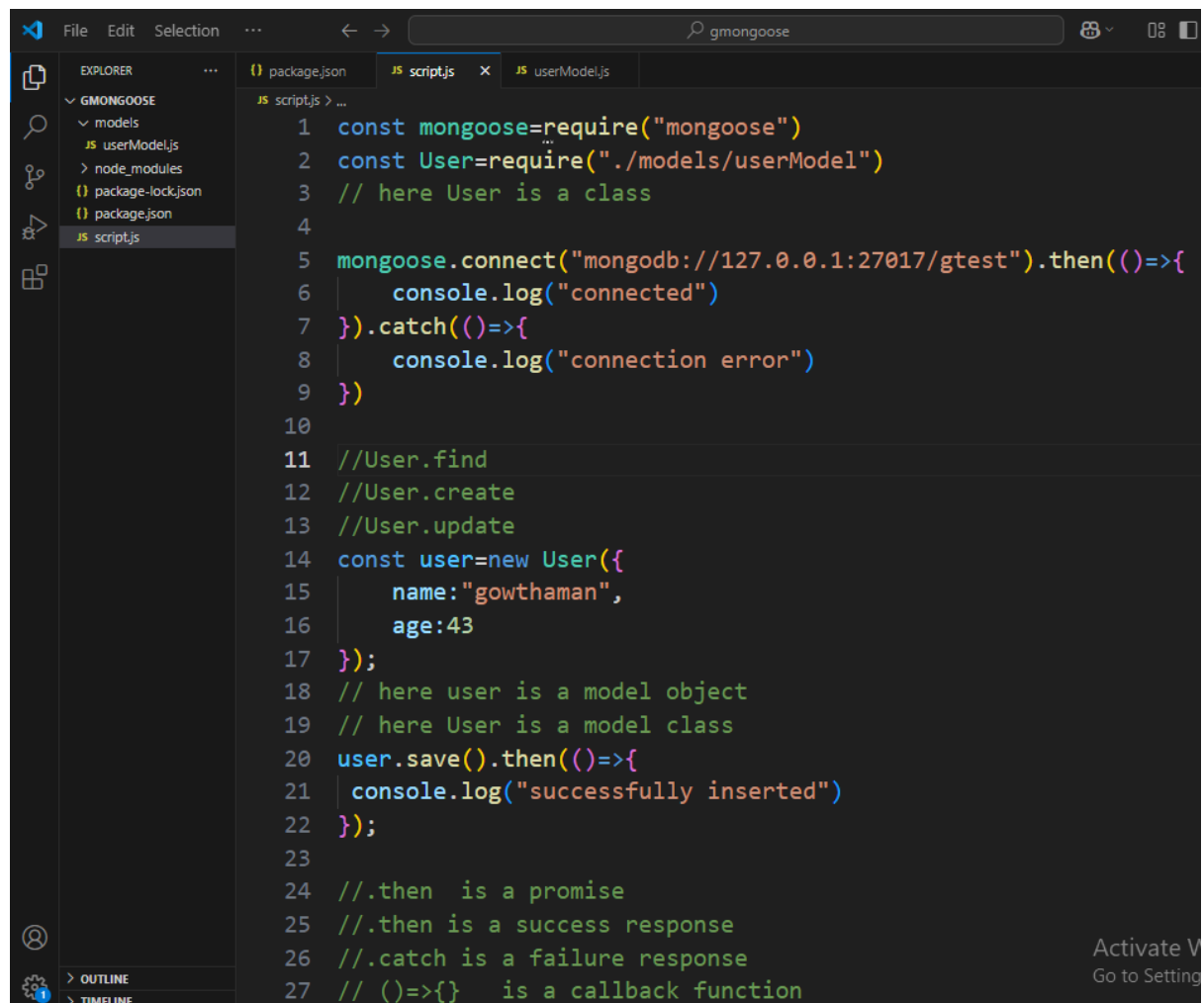
userSchema: userDefined object.



```

1  const mongoose=require("mongoose")
2  const userSchema=new mongoose.Schema({
3    name:String,
4    age:Number
5  })
6
7  const userMdl=mongoose.model("usr",userSchema)
8  module.exports=userMdl
9
10 // here usr is collection name : see collection name in your db usr(s)
    name
  
```

Insert method 1: (with out using async await)



```

1  const mongoose=require("mongoose")
2  const User=require("./models/userModel")
3  // here User is a class
4
5  mongoose.connect("mongodb://127.0.0.1:27017/gtest").then(()=>{
6    console.log("connected")
7  }).catch(()=>{
8    console.log("connection error")
9  })
10
11 //User.find
12 //User.create
13 //User.update
14 const user=new User({
15   name:"gowthaman",
16   age:43
17 });
18 // here user is a model object
19 // here User is a model class
20 user.save().then(()=>{
21   console.log("successfully inserted")
22 });
23
24 // .then is a promise
25 // .then is a success response
26 // .catch is a failure response
27 // ()=>{} is a callback function
  
```

Do not change name & age key. because schema defined.

Do not add extra field. Only name and age accept

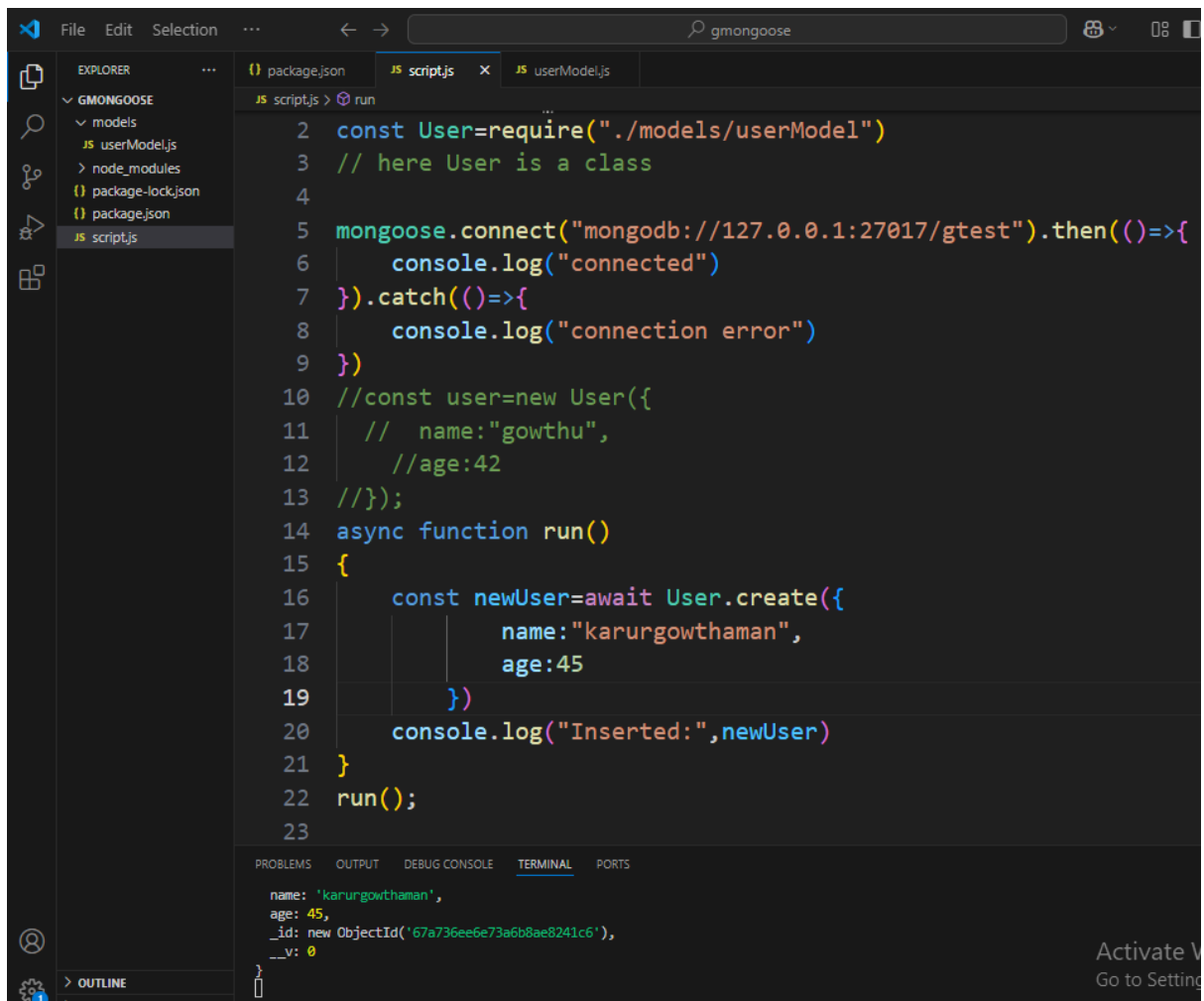
Same insert method using async function because promise return (.then)

Method 2: insertion using save with async function

```
1  const mongoose=require("mongoose")
2  const User=require("../models/userModel")
3  // here User is a class
4
5  mongoose.connect("mongodb://127.0.0.1:27017/gtest").then(()=>{
6    console.log("connected")
7  }).catch(()=>{
8    console.log("connection error")
9  })
10 //User.find
11 //User.create
12 //User.update
13 const user=new User({
14   name:"gowthu",
15   age:42
16 });
17 // here user is a model object
18 // here User is a model class
19
20 async function run()
21 {
22   const newUser=await user.save()
23   //meaning: wait till to store data in db and assign result to newUser.
24   console.log("Inserted:",newUser)
25 }
26 run(); // calling function
27
```

Activate Windows
Go to Settings to activate

Method:3 insertion



```

2  const User=require("./models/userModel")
3  // here User is a class
4
5  mongoose.connect("mongodb://127.0.0.1:27017/gtest").then(()=>{
6      console.log("connected")
7  }).catch(()=>{
8      console.log("connection error")
9  })
10 //const user=new User({
11 //    name:"gowthu",
12 //    age:42
13 //});
14 async function run()
15 {
16     const newUser=await User.create({
17         name:"karurgowthaman",
18         age:45
19     })
20     console.log("Inserted:",newUser)
21 }
22 run();
23

```

Terminal Output:

```

name: 'karurgowthaman',
age: 45,
_id: new ObjectId('67a736ee6e73a6b8ae8241c6'),
__v: 0

```

```

14  async function run()
15  {
16      const newUser=await User.create({
17          name:"karurgowthaman",
18          age:45
19      })
20      //-----change-----
21      newUser.name="gow";
22      newUser.age=18;
23      await newUser.save()
24      //-----
25      console.log("Inserted:",newUser)
26  }
27  run();
28

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

Inserted: {
  name: 'gow',
  age: 18,
  _id: new ObjectId('67a73853c94d7c8bcd5f356e'),
  __v: 0
}

```