

File Edit Selection View Go Run Terminal Help

node_modules

index.js package-lock.json package.json

JS index.js

```
7 let schemaClass = new mongoose.Schema({
8   name: {
12     unique: true,
13     index: true,
14     sparse: true,
15     validate: (value) => value.length > 3,
16     alias: 'full_name',
17     get: (value) => value.toUpperCase(),
18     set: (value) => value.toLowerCase()
19   }
20 });
21 // creating model from the schema
22 let Schema = mongoose.model('Schema', schemaClass);
23
24 let schema1 = new Schema({
25   name: "GeeksForGeeks"
26 });
27 // will have a default value of John Doe
28 let schema2 = new Schema({});
29
30 console.log(schema1);
31 console.log(schema2);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

node

```
The student data saved succesfully
PS C:\Users\ishit\Desktop\moongoose> node index.js
The student data saved succesfully
PS C:\Users\ishit\Desktop\moongoose> node index.js
{
  name: 'geeksforgeeks',
  _id: new ObjectId('68524fc4b92ad49b3e77d9dF')
}
{ name: 'john doe', _id: new ObjectId('68524fc4b92ad49b3e77d9e0') }
```

Ln 31, Col 22 Spaces: 4 UTF-8 CRLF JavaScript Go Live

Search

11:04 18-06-2025

VS Code interface showing a JavaScript file named `index.js` and its output in the terminal.

File Explorer (Left):

- node_modules
- index.js
- package-lock.json
- package.json

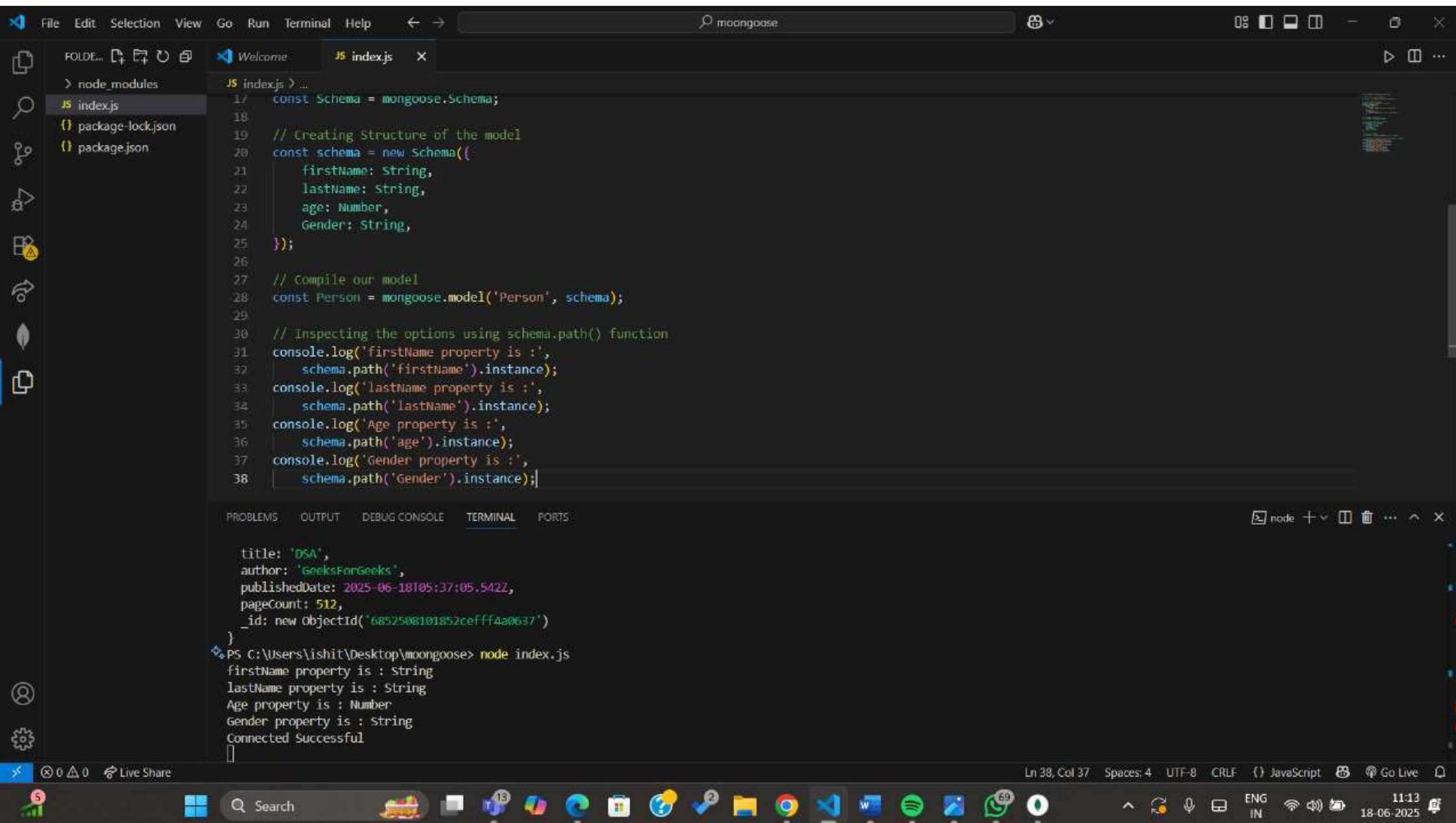
Editor (Center):

```
JS index.js > ...
6 let bookSchema = new mongoose.Schema({
15   publishedDate: {
17     required: true
18   },
19   pageCount: {
20     type: Number,
21     required: true
22   },
23   publisher: String,
24   coverImageUrl: String
25 });
26 // creating model from the schema
27 let Book = mongoose.model('Book', bookSchema);
28
29 let book = new Book({
30   title: "DSA",
31   author: "GeeksForGeeks",
32   publishedDate: new Date(),
33   pageCount: 512,
34 });
35 console.log("Book:", book);
```

Terminal (Bottom):

```
name: 'geeksforgeeks',
  _id: new ObjectId('68524fc4b92ad49b3e77d9df')
}
{ name: 'john doe', _id: new ObjectId('68524fc4b92ad49b3e77d9e0') }
PS C:\Users\ishit\Desktop\mongoose> node index.js
Book: {
  title: 'DSA',
  author: 'GeeksForGeeks',
  publishedDate: 2025-06-18T05:37:05.542Z,
  pageCount: 512,
  _id: new ObjectId('6852508101852cefff4a0637')
}
```

Status Bar (Bottom): Ln 35, Col 28 | Spaces: 4 | UTF-8 | CRLF | JavaScript | Go Live



File Edit Selection View Go Run Terminal Help

moongoose

FOLDE... node_modules JS index.js

JS index.js > ...

```
9 mongoose.connect(uri)
10   .then((ans) => {
11     console.log("Connected Successful")
12   })
13   .catch((err) => {
14     console.log("Error in the Connection")
15   })
16
17 // Creating Structure of the model
18 const studentSchema = new mongoose.Schema({
19   name: { type: String },
20   rollNumber: { type: Number },
21   class: { type: Number },
22 });
23
24 console.log('Name property accepts: ',
25   studentSchema.path('name').instance);
26 console.log('Roll Number property accepts: ',
27   studentSchema.path('rollNumber').instance);
28 console.log('Class property accepts: ',
29   studentSchema.path('class').instance);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
}
PS C:\Users\ishit\Desktop\moongoose> node index.js
firstName property is : String
lastName property is : String
Age property is : Number
Gender property is : String
Connected Successful
PS C:\Users\ishit\Desktop\moongoose> node index.js
Name property accepts: String
Roll Number property accepts: Number
Class property accepts: Number
Connected Successful
```

Ln 29, Col 43 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live

Search

11:15 18-06-2025

File Edit Selection View Go Run Terminal Help

node_modules
index.js
package-lock.json
package.json

JS index.js > ...

```
9   const userSchema = new mongoose.Schema({  
54     password: {  
55       // Set the data type to string  
56       type: String,  
57       // Define a getter function for password  
58       // field to encrypt the value using MD5  
59       get: (type) => md5(type)  
60     }  
61   });  
62  
63   // Create a user model from the schema  
64   const User = mongoose.model('User', userSchema);  
65  
66   const user = new User({  
67     email: 'abc1452@gmail.com',  
68     password: '1234'  
69   });  
70  
71   // Log the masked email and encrypted  
72   // password of the user  
73   // Output: a*****@gmail.com  
74   console.log(user.email);  
75  
76   // Output: 81dc9bdb52d04dc20036dbd8313ed055  
77   console.log(user.password);  
78  
79  
80
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\ishit\Desktop\moongoose> node index.js
a*****2@gmail.com
81dc9bdb52d04dc20036dbd8313ed055
PS C:\Users\ishit\Desktop\moongoose>

Ln 67, Col 2 Spaces: 4 UTF-8 CRLF JavaScript Go Live

6 13 2 70

11:45 18-06-2025

File Edit Selection View Go Run Terminal Help

miongoose

FOLDE...
node_modules
index.js
package-lock.json
package.json

JS index.js

JS index.js > ...

```
38 }
39
40 // Fetch a post and populate its author
41 async function fetchPost() {
42   try {
43     const post = await Post.findOne({ title: 'Mongoose Guide' }).populate('author');
44     if (post && post.author) {
45       console.log('\n🔥 Populated Post:');
46       console.log('Title:', post.title);
47       console.log('Author Name:', post.author.name);
48       console.log('Author Email:', post.author.email);
49     } else {
50       console.log('Post or author not found.');
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

powerShell +

MongoDB connected
✔ Sample data seeded.

🔥 Populated Post:
Title: Mongoose Guide
Author Name: Vaishnavi
Author Email: vaishnavi@example.com

PS C:\Users\ishit\Desktop\mongoose>

Ln 1, Col 38 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live

Search

13 2 66

12:06 18-06-2025

File Edit Selection View Go Run Terminal Help

← →

mongoose

80% 100% 150% 200%

⌵

FOLDE...
node_modules
JS index.js
package-lock.json
package.json
JS script2.js

JS script2.js > ...

```
7 // Connecting to database
8 mongoose.set('strictQuery', false);
9
10 mongoose.connect(url)
11   .then((ans) => {
12     console.log("Connected Successful")
13   })
14   .catch((err) => {
15     console.log("Error in the Connection")
16   })
17
18 // Creating structure of the model
19 const studentSchema = new mongoose.Schema({
20   name: { type: String },
21   rollNumber: { type: Number },
22   class: { type: Number },
23 });
24
25
26 console.log('Name property accepts: ',
27   studentSchema.path('name').instance);
28 console.log('Roll Number property accepts: ',
29   studentSchema.path('rollNumber').instance);
30 console.log('Class property accepts: ',
31   studentSchema.path('class').instance);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

node +

⌵

⌵

⌵

⌵

Author Name: Vaishnavi
Author Email: vaishnavi@example.com
PS C:\Users\ishit\Desktop\mongoose> node script2.js
Name property accepts: String
Roll Number property accepts: Number
Class property accepts: Number
Connected Successful

Ln 31, Col 43 Spaces: 4 UTF-8 CRLF {} JavaScript

Go Live

12:16
18-06-2025

SENSEX
-0.30%

Search

13

2

66

ENG
IN

12:16
18-06-2025

File Edit Selection View Go Run Terminal Help

moongoose

FOLDE... JS index.js JS script2.js

> node_modules

JS index.js

package-lock.json

package.json

JS script2.js

JS script2.js > ...

```
14 .catch((err) => {
15   })
16
17 // Calling Schema class
18 const Schema = mongoose.Schema;
19
20 // Creating Structure of the model
21 const schema = new Schema({
22   name: {
23     type: String,
24     firstName: {
25       type: String,
26       require: true,
27     },
28     lastName: {
29       type: String,
30       require: true,
31     }
32   }
33 });
34
35 // compile our model
36 const Person = mongoose.model('Person', schema);
37
38 // Inspecting the options using schema.path() function
39 console.log(schema.path('name').options);
40
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

node script2.js

SchemaStringOptions {
 type: [Function: String],
 firstName: { type: [Function: String], require: true },
 lastName: { type: [Function: String], require: true }
}
Connected Successful

Ln 40, Col 42 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live

High UV Now

Search

14

2

66

12:21 18-06-2025

File Edit Selection View Go Run Terminal Help

moongoose

node_modules
index.js
package-lock.json
package.json
script2.js

Welcome JS index.js JS script2.js

```
JS script2.js > ...
12 // Creating a new document
13 async function start() {
14   let user1 = new User({
15     name: "Geeks",
16     age: 20,
17   });
18   user1.save().then(async (doc) => {
19     if (doc) {
20       console.log("The document is saved successfully");
21       console.log(doc._id);
22     }
23   });
24   // ...existing code...
25   const user2 = await User.findOne({ email: "someemail@example.com" });
26   if (user2) {
27     user2.name = "GeeksforGeeks ";
28     await user2.save();
29     console.log("User updated:", user2);
30   } else {
31     console.log("User not found with the given email.");
32   }
33   // ...existing code...
34 }
35 start();
36
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

river version 4.0.0 and will be removed in the next major version✔ Document saved successfully
ID: new ObjectId('685269b366600b1ae17ef2a8')
✔ Document updated successfully
Updated ID: new ObjectId('685269b366600b1ae17ef2a8')
PS C:\Users\ishit\Desktop\moongoose> node script2.js
✔ The document is saved successfully
new ObjectId('685269d886f54c9e7c317a6c')
User not found with the given email.

0 0 0 Live Share

Ln 36, Col 2 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live

Hot days ahead 33°C

Search

13 2 68

ENG IN 12:55 18-06-2025

File Edit Selection View Go Run Terminal Help

node_modules
index.js
package-lock.json
package.json
script2.js

JS script2.js > ...

```
3  async function main() {  
21  });  
22  
23  const Person = mongoose.model('Person', personSchema);  
24  
25  // Document creation  
26  const person1 = new Person({ name: 'john', age: 19 });  
27  
28  
29  try {  
30    await person1.validate(); // Only validates, doesn't save  
31    console.log('✅ Validation passed');  
32  } catch (err) {  
33    console.error('❌ Validation failed');  
34    console.error(err.message);  
35  }  
36  
37  } catch (err) {  
38    console.error('❌ Connection failed:', err);  
39  } finally {  
40    await mongoose.disconnect();  
41  }  
42  }  
43  
44  main();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
✅ Document updated successfully  
Updated ID: new ObjectId('685269b366600b1ae17ef2a8')  
PS C:\Users\ishit\Desktop\moongoose> node script2.js  
❌ The document is saved successfully  
new ObjectId('685269d886f54c9e7c317a6c')  
User not found with the given email.  
• PS C:\Users\ishit\Desktop\moongoose> node script2.js  
✅ Connected to database  
✅ Validation passed  
❖ PS C:\Users\ishit\Desktop\moongoose>
```

Ln 45, Col 2 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live

33°C Cloudy Search 13 18-06-2025

```

> node_modules
JS app.js
JS index.js
() package-lock.json
() package.json
JS script2.js

JS app.js > ...
16 const runQueries = async () => {
17   console.log('First function call:', usersByName);
18
19   // 2. Find users by name, exclude 'age' field
20   const usersNoAge = await User.find({ name: 'Punit' }, { age: 0 });
21   console.log('Second function call (no age):', usersNoAge);
22
23   // 3. Find users age ≥ 10, limit 2 results
24   const limitedUsers = await User.find({ age: { $gte: 10 } }, null, { limit: 2 });
25   console.log('Third function call (age ≥ 10, limit 2):', limitedUsers);
26
27   } catch (err) {
28     console.error('Query error:', err);
29   }
30 };
31
32 runQueries();
33
34
35

```

```

PS C:\Users\ishit\Desktop\mongoose> node app.js
HellFire was Updated
PS C:\Users\ishit\Desktop\mongoose> node app.js
[x] Connected to database
[log] First function call: []
[log] Second function call (no age): []
[log] Third function call (age ≥ 10, limit 2): [
  {
    _id: new ObjectId('685269b366600b1ae17ef2a8'),
    name: 'GeeksforGeeks',
    age: 20,
    __v: 0
  },
  {
    _id: new ObjectId('685269d886f54c9e7c317a6c'),
    name: 'Geeks',
    age: 20,
    __v: 0
  }
]

```