

The screenshot shows a VS Code editor with a React.js project. The file explorer on the left shows the project structure, including a schema.js file. The main editor displays the schema.js file, which contains a schema definition for a user. The terminal window at the bottom shows the output of the server, including a warning about the deprecated use of the useFindById option and a message indicating that the server is listening on port 3000.

```

schema.js
10 app.get('/api/users', async (req, res) => {
11   //
12   })
13
14 // get a user by ID
15 app.get('/api/users/:id', async (req, res) => {
16   try {
17     const user = await User.findById(req.params.id);
18     if (!user) {
19       res.status(404).json({ message: 'User not found' });
20     }
21     res.json(user);
22   } catch (error) {
23     console.error(error);
24   }
25 });

```

Terminal Output:

```

(node:3000) [DEP0168] Warning: UseFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(node:3000) [DEP0168] Warning: UseFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server listening on port: 3000
MongoDB connected...
Documents inserted successfully...
(node:3000) [DEP0168] Warning: useFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(node:3000) [DEP0168] Warning: useFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server listening on port: 3000
MongoDB connected

```

The screenshot shows a VS Code editor with a React.js project. The file explorer on the left shows the project structure, including a schema.js file. The main editor displays the schema.js file, which contains a schema definition for a user. The terminal window at the bottom shows the output of the server, including a warning about the deprecated use of the useFindById option and a message indicating that the server is listening on port 3000.

```

schema.js
10 app.get('/api/users', async (req, res) => {
11   //
12   })
13
14 // get a user by ID
15 app.get('/api/users/:id', async (req, res) => {
16   try {
17     const user = await User.findById(req.params.id);
18     if (!user) {
19       res.status(404).json({ message: 'User not found' });
20     }
21     res.json(user);
22   } catch (error) {
23     console.error(error);
24   }
25 });

```

Terminal Output:

```

(node:3000) [DEP0168] Warning: UseFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(node:3000) [DEP0168] Warning: UseFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server listening on port: 3000
MongoDB connected...
Documents inserted successfully...
(node:3000) [DEP0168] Warning: useFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(node:3000) [DEP0168] Warning: useFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server listening on port: 3000
MongoDB connected

```

The screenshot shows a VS Code editor with a React.js project. The file explorer on the left shows the project structure, including a schema.js file. The main editor displays the schema.js file, which contains a schema definition for a user. The terminal window at the bottom shows the output of the server, including a warning about the deprecated use of the useFindById option and a message indicating that the server is listening on port 3000.

```

schema.js
10 app.get('/api/users', async (req, res) => {
11   //
12   })
13
14 // get a user by ID
15 app.get('/api/users/:id', async (req, res) => {
16   try {
17     const user = await User.findById(req.params.id);
18     if (!user) {
19       res.status(404).json({ message: 'User not found' });
20     }
21     res.json(user);
22   } catch (error) {
23     console.error(error);
24   }
25 });

```

Terminal Output:

```

(node:3000) [DEP0168] Warning: UseFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(node:3000) [DEP0168] Warning: UseFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server listening on port: 3000
MongoDB connected...
Documents inserted successfully...
(node:3000) [DEP0168] Warning: useFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
(node:3000) [DEP0168] Warning: useFindById is a deprecated option; useFindById has no effect since Node.js Driver version 4.0.0 and will be removed in the next major version
Server listening on port: 3000
MongoDB connected

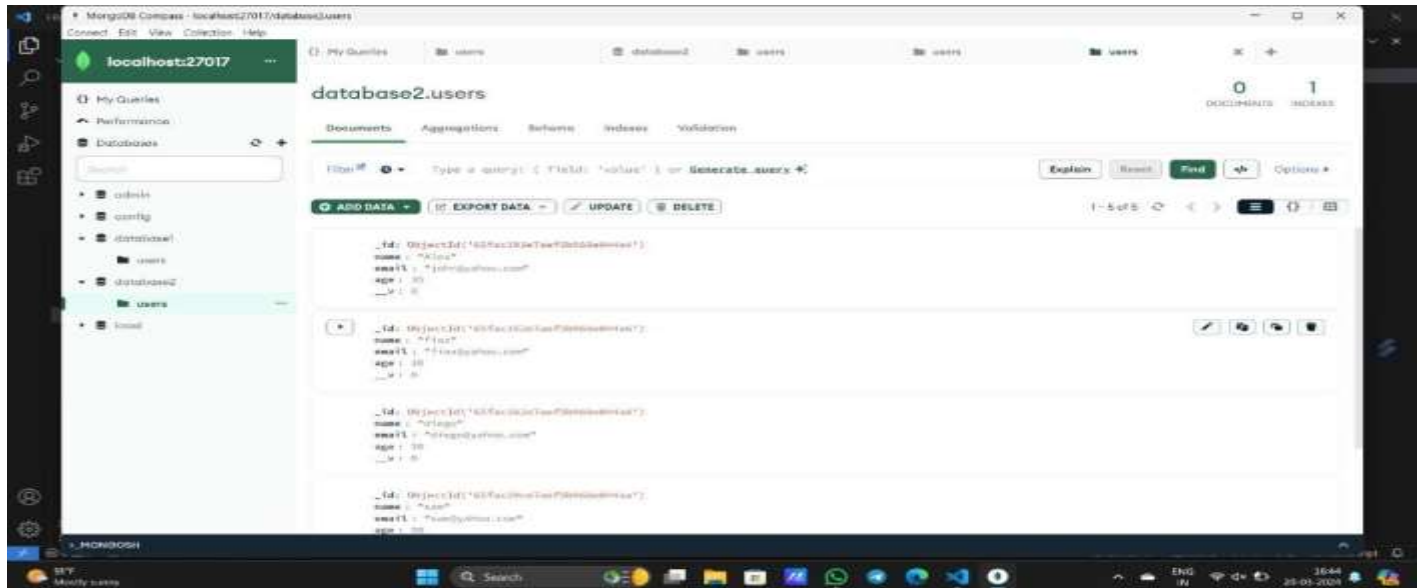
```

The screenshot shows a VS Code editor with a React.js project. The file explorer on the left shows the project structure. The main editor displays a REST client with two GET requests to `http://localhost:3000/api/users`. The first request is for a specific user ID, and the second is for all users. The terminal window shows the output of the REST client, displaying user details for 'John' and 'Fiaz'.

```

schemali@Reactjs:~$ curl -X GET http://localhost:3000/api/users/1
{"_id": "60fac33e7aef3b80e044a", "name": "John", "email": "john@yahoo.com", "age": 35, "sex": "M"}

schemali@Reactjs:~$ curl -X GET http://localhost:3000/api/users
[{"_id": "60fac33e7aef3b80e044a", "name": "John", "email": "john@yahoo.com", "age": 35, "sex": "M"}, {"_id": "60fac33e7aef3b80e044b", "name": "Fiaz", "email": "fiaz@yahoo.com", "age": 30, "sex": "M"}, {"_id": "60fac33e7aef3b80e044c", "name": "Raza", "email": "raza@yahoo.com", "age": 30, "sex": "M"}, {"_id": "60fac33e7aef3b80e044d", "name": "Quam", "email": "quam@yahoo.com", "age": 30, "sex": "M"}]
  
```



The screenshot shows a VS Code editor with a REST client. The main editor displays a POST request to `http://localhost:3000/api/users` with a JSON body containing user details. The terminal window shows the output of the REST client, displaying the user details for 'John'.

```

schemali@Reactjs:~$ curl -X POST http://localhost:3000/api/users -H "Content-Type: application/json" -d '{"name": "John", "email": "john@yahoo.com", "age": 35, "sex": "M"}'
{"_id": "60fac33e7aef3b80e044a", "name": "John", "email": "john@yahoo.com", "age": 35, "sex": "M"}
  
```

