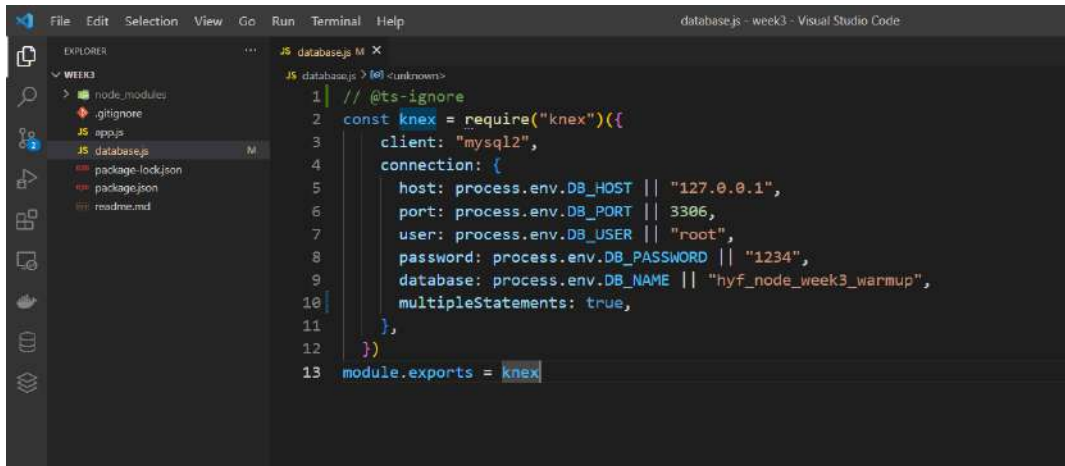


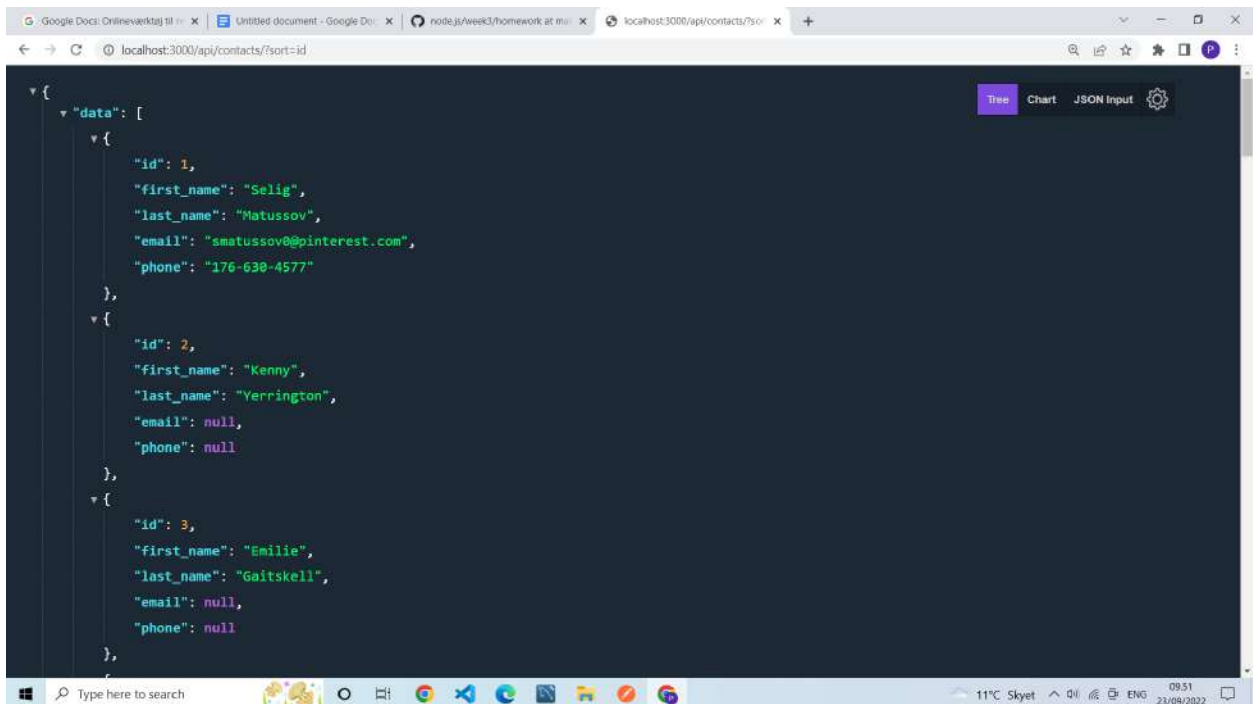
As mentioned above, the `sort` query parameter has been introduced with a SQL injection vulnerability.



```
1 // @ts-ignore
2 const knex = require("knex")({
3   client: "mysql2",
4   connection: {
5     host: process.env.DB_HOST || "127.0.0.1",
6     port: process.env.DB_PORT || 3306,
7     user: process.env.DB_USER || "root",
8     password: process.env.DB_PASSWORD || "1234",
9     database: process.env.DB_NAME || "hyf_node_week3_warmup",
10    multipleStatements: true,
11  },
12 });
13 module.exports = knex
```

I have run only one query at browser ;

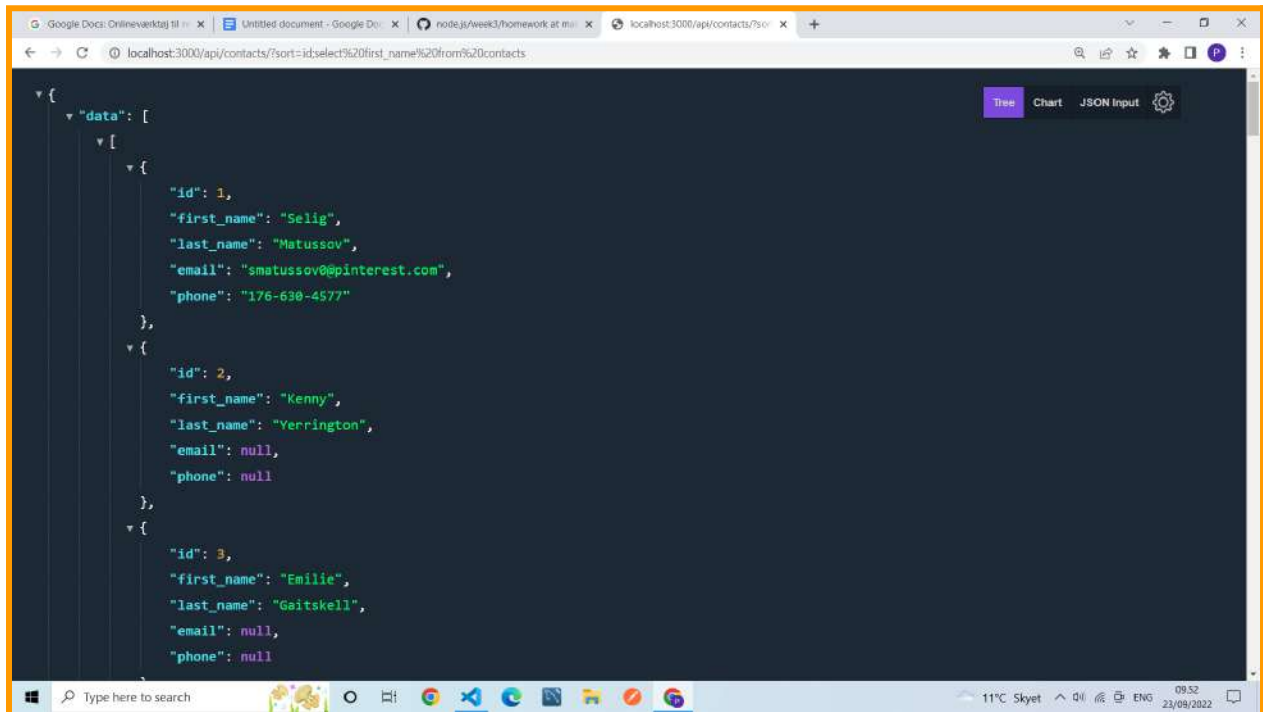
<http://localhost:3000/api/contacts/?sort=id>



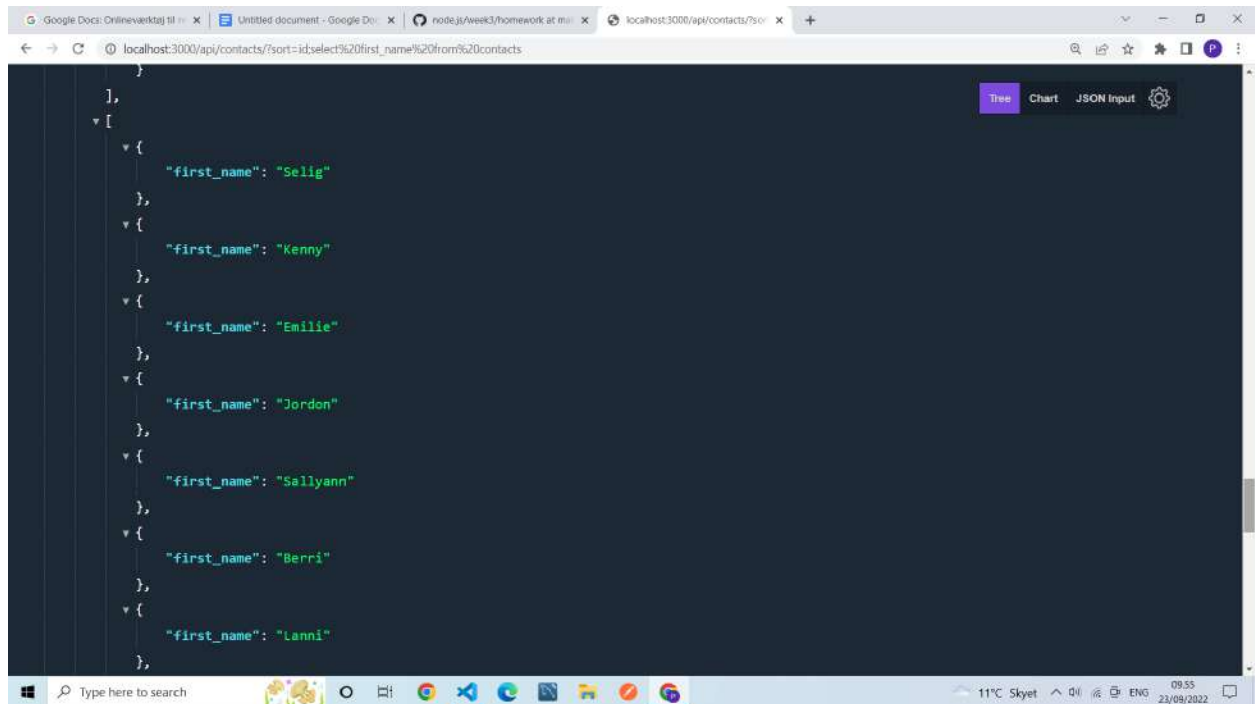
```
{
  "data": [
    {
      "id": 1,
      "first_name": "Selig",
      "last_name": "Matussov",
      "email": "smatussov0@pinterest.com",
      "phone": "176-630-4577"
    },
    {
      "id": 2,
      "first_name": "Kenny",
      "last_name": "Yerrington",
      "email": null,
      "phone": null
    },
    {
      "id": 3,
      "first_name": "Emilie",
      "last_name": "Gaitskell",
      "email": null,
      "phone": null
    }
  ]
}
```

I have run two queries at browser;

http://localhost:3000/api/contacts/?sort=id;select%20first_name%20from%20contacts



```
{
  "data": [
    {
      "id": 1,
      "first_name": "Selig",
      "last_name": "Matussov",
      "email": "smatussov@pinterest.com",
      "phone": "176-630-4577"
    },
    {
      "id": 2,
      "first_name": "Kenny",
      "last_name": "Yerrington",
      "email": null,
      "phone": null
    },
    {
      "id": 3,
      "first_name": "Emilie",
      "last_name": "Gaitskell",
      "email": null,
      "phone": null
    }
  ]
}
```



First, you should demonstrate the SQL injection and that it for instance is possible to drop/delete the `contacts` table with the `sort` query parameter. You can for instance demonstrate this with a screen recording and include it in the PR description.

Answer: Yes, It is possible to run drop table sql injection with Sort query.

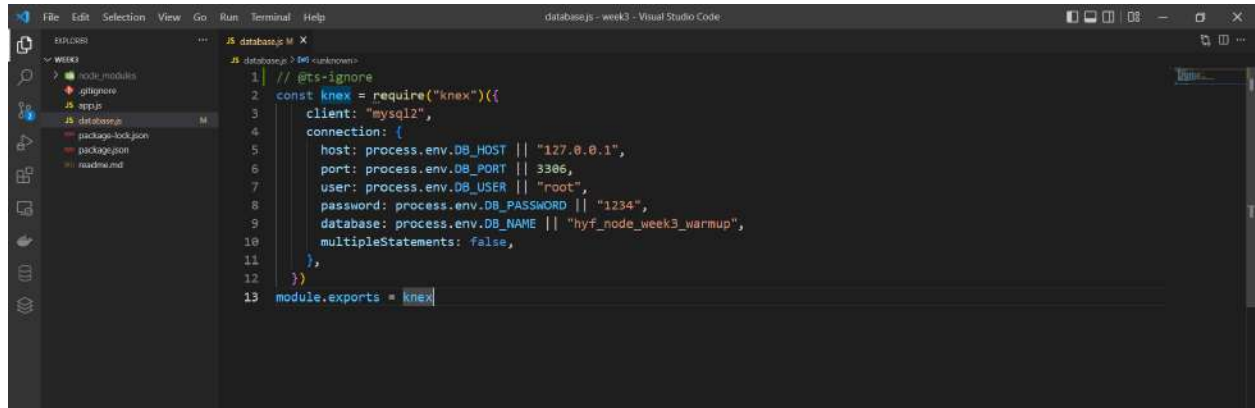
```
{
  "data": [
    {
      "id": 1,
      "first_name": "Selig",
      "last_name": "Matussov",
      "email": "smatussov@pinterest.com",
      "phone": "176-630-4577"
    },
    {
      "id": 2,
      "first_name": "Kenny",
      "last_name": "Yerrington",
      "email": null,
      "phone": null
    },
    {
      "id": 3,
      "first_name": "Emilie",
      "last_name": "Gaitskell",
      "email": null,
      "phone": null
    }
  ]
}
```

Now Table can't find in database.

```
{
  "error": "Internal server error"
}
```

After having demonstrated the SQL injection vulnerability, the goal is then to fix the issue by updating `app.js`.

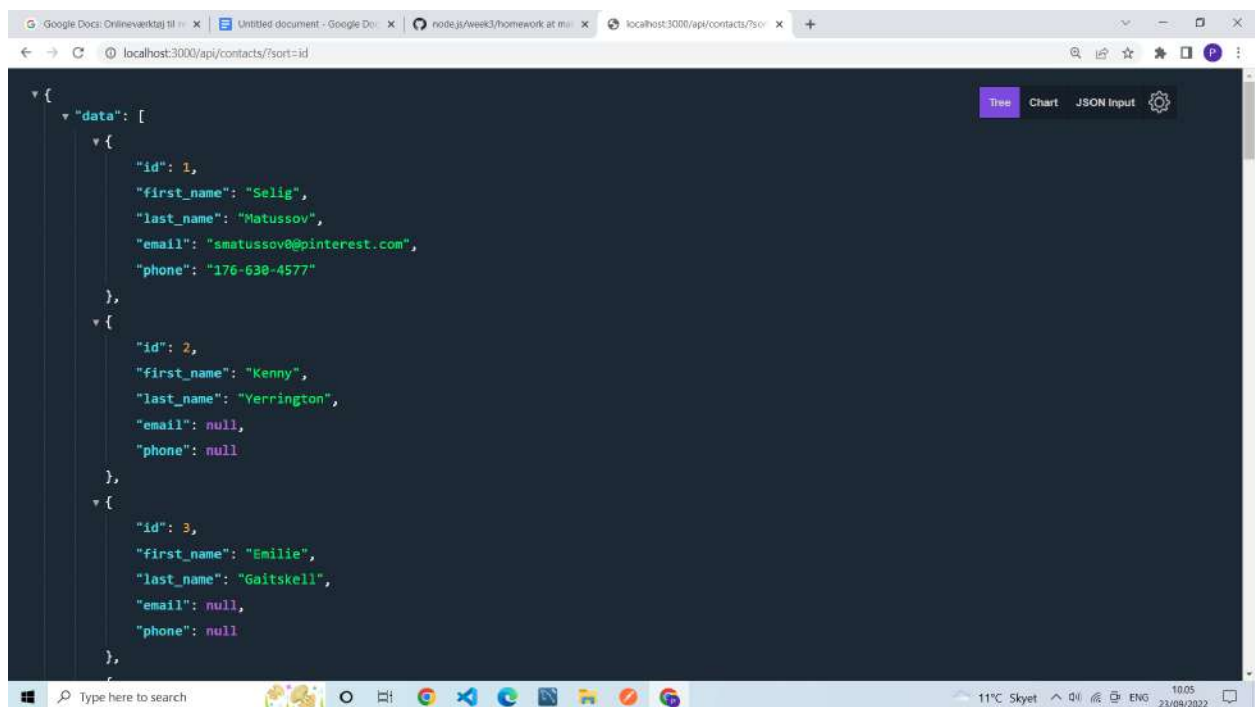
Answer: Issue can solve by `multipleStatements: false`;



```
1 // @ts-ignore
2 const knex = require("knex")({
3   client: "mysql2",
4   connection: {
5     host: process.env.DB_HOST || "127.0.0.1",
6     port: process.env.DB_PORT || 3306,
7     user: process.env.DB_USER || "root",
8     password: process.env.DB_PASSWORD || "1234",
9     database: process.env.DB_NAME || "hyf_node_week3_warmup",
10    multipleStatements: false,
11  },
12 });
13 module.exports = knex
```

Created a table again in Database.

Run sort query parameters only.



```
{
  "data": [
    {
      "id": 1,
      "first_name": "Selig",
      "last_name": "Matussov",
      "email": "smatussov0@pinterest.com",
      "phone": "176-638-4577"
    },
    {
      "id": 2,
      "first_name": "Kenny",
      "last_name": "Yerrington",
      "email": null,
      "phone": null
    },
    {
      "id": 3,
      "first_name": "Emilie",
      "last_name": "Gaitskell",
      "email": null,
      "phone": null
    }
  ]
}
```

Run sort query parameters with drop table sql query. It shows an error.

