

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The top bar includes the standard menu items: File, Edit, Selection, View, Go, Run, Terminal, Help, and a set of icons for back, forward, and search. The title bar displays "Day19_assignment". The left sidebar has sections for EXPLORER, OPEN EDITORS, and DAY19_ASSIGNMENT, with files like package.json, server.js, middleware, node_modules, routes, validators, package-lock.json, and package.json listed. The main editor area shows the content of the server.js file:

```
4  const logger = require('./middleware/logger');
5  const { notFoundHandler, errorHandler } = require('./middleware/errorHandler');
6
7  const app = express();
8  const PORT = 4000;
9
10 app.use(cors());
11 app.use(express.json());
12 app.use(logger);
13
14 app.get('/', (req, res) => {
15   res.send('Welcome to Express Server');
16 });
17
18 app.get('/status', (req, res) => {
19   res.json({ server: 'running', uptime: 'OK' });
20 });
21
22 app.get('/products', (req, res) => {
23   const name = req.query.name;
24   if (name) {
```

Below the editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing a PowerShell session (PS) in the directory C:\Users\kushw\OneDrive\Desktop\Wipro NGA - MERN\Day19\Day19_assignment. The user runs the command `node server.js`, which outputs:

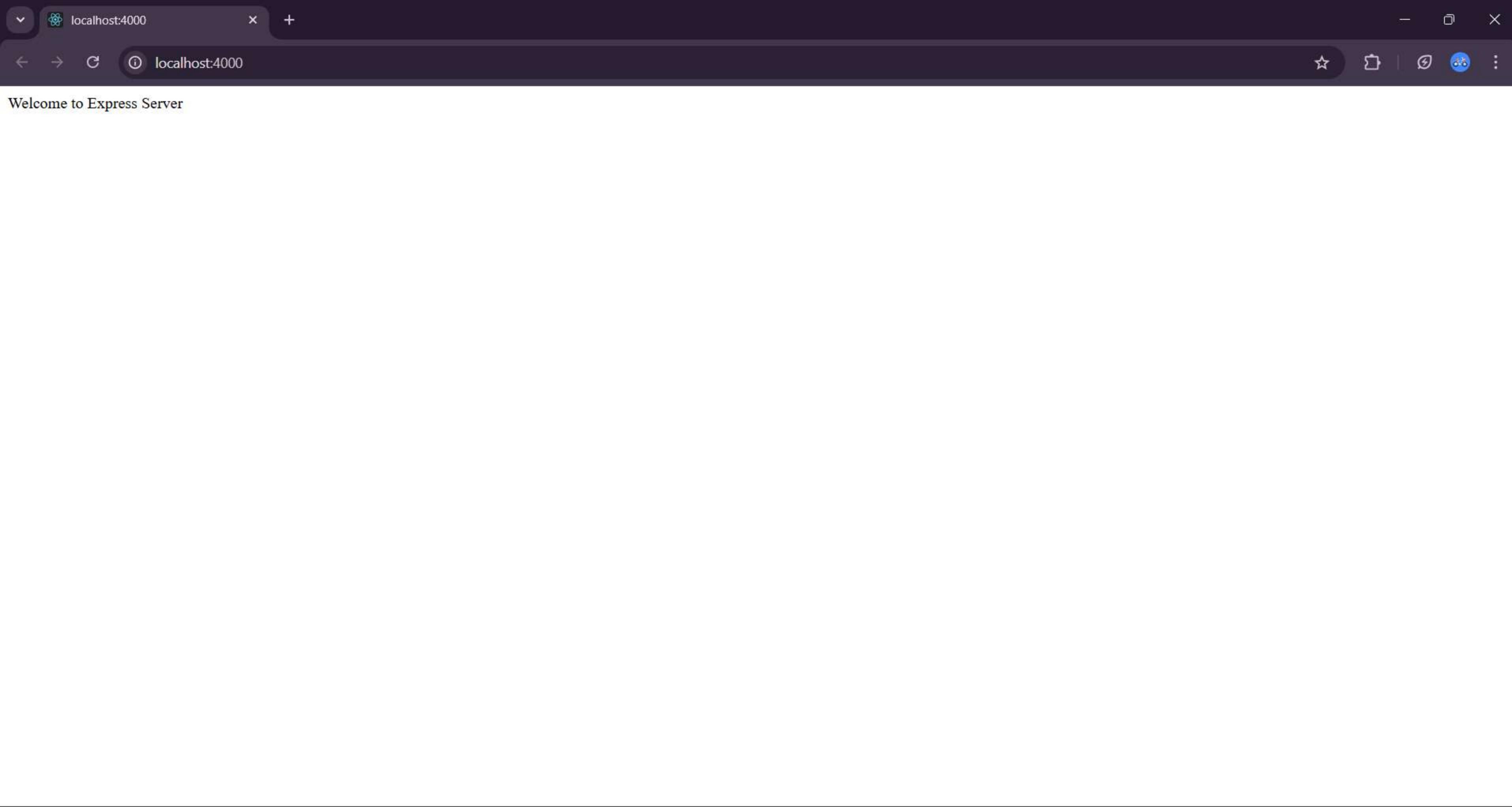
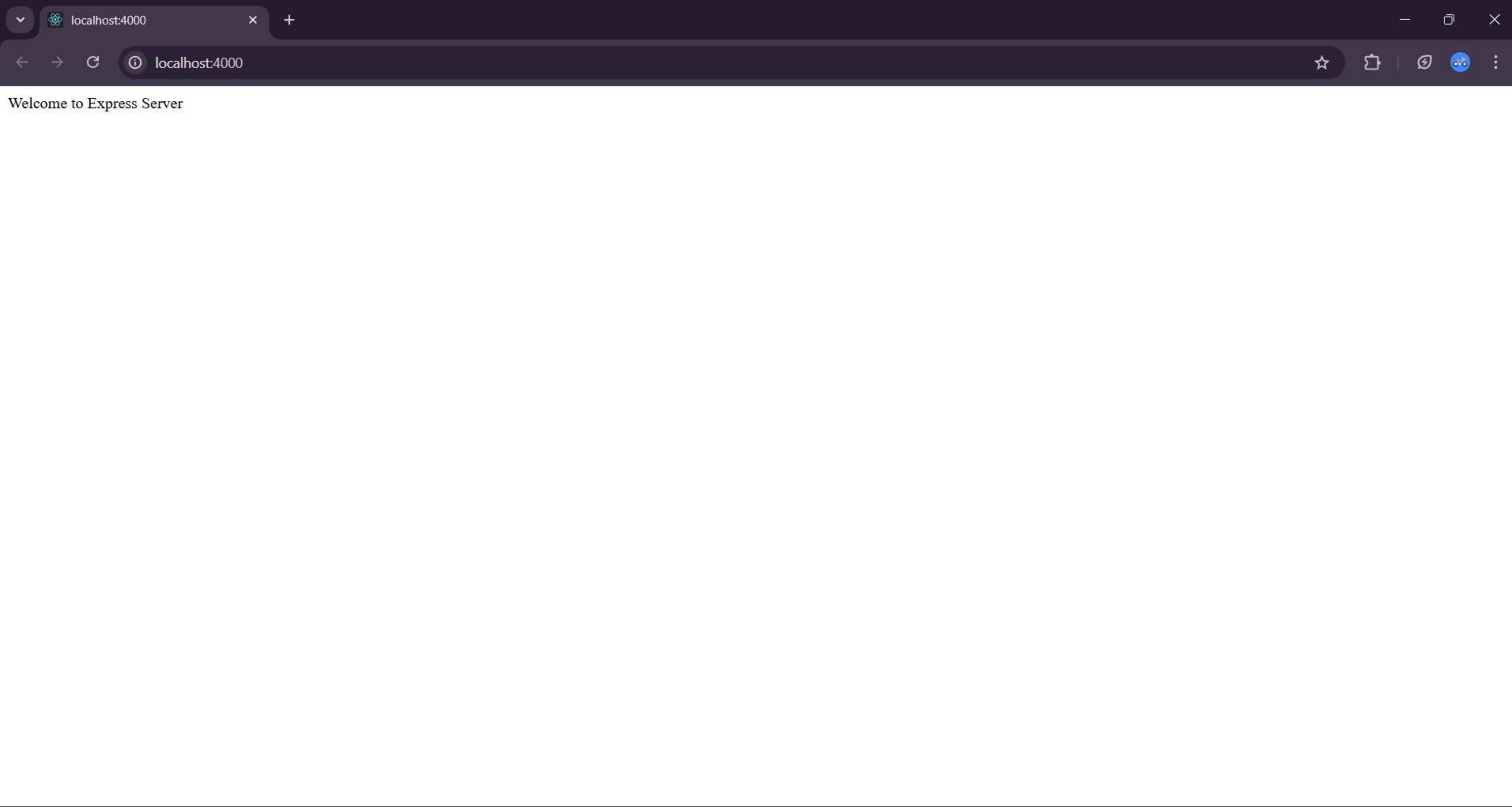
```
Node.js v25.1.0
PS C:\Users\kushw\OneDrive\Desktop\Wipro NGA - MERN\Day19\Day19_assignment> node server.js
```

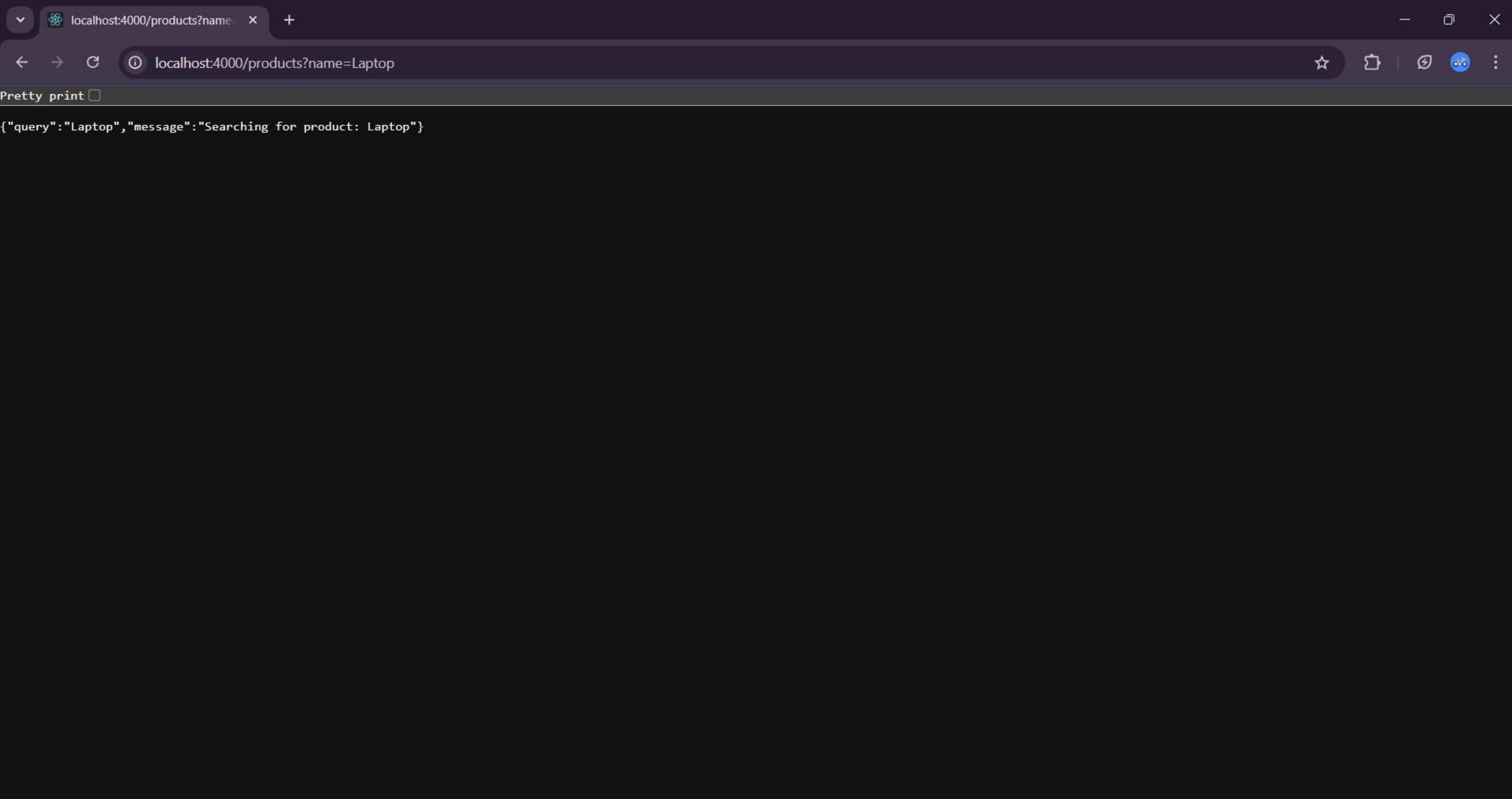
Then, the user runs `npm install express`:

```
PS C:\Users\kushw\OneDrive\Desktop\Wipro NGA - MERN\Day19\Day19_assignment> npm install express
added 103 packages, and audited 104 packages in 7s
18 packages are looking for funding
  run `npm fund` for details
```

Finally, the user runs `node server.js` again, and the terminal shows the server is listening on port 4000:

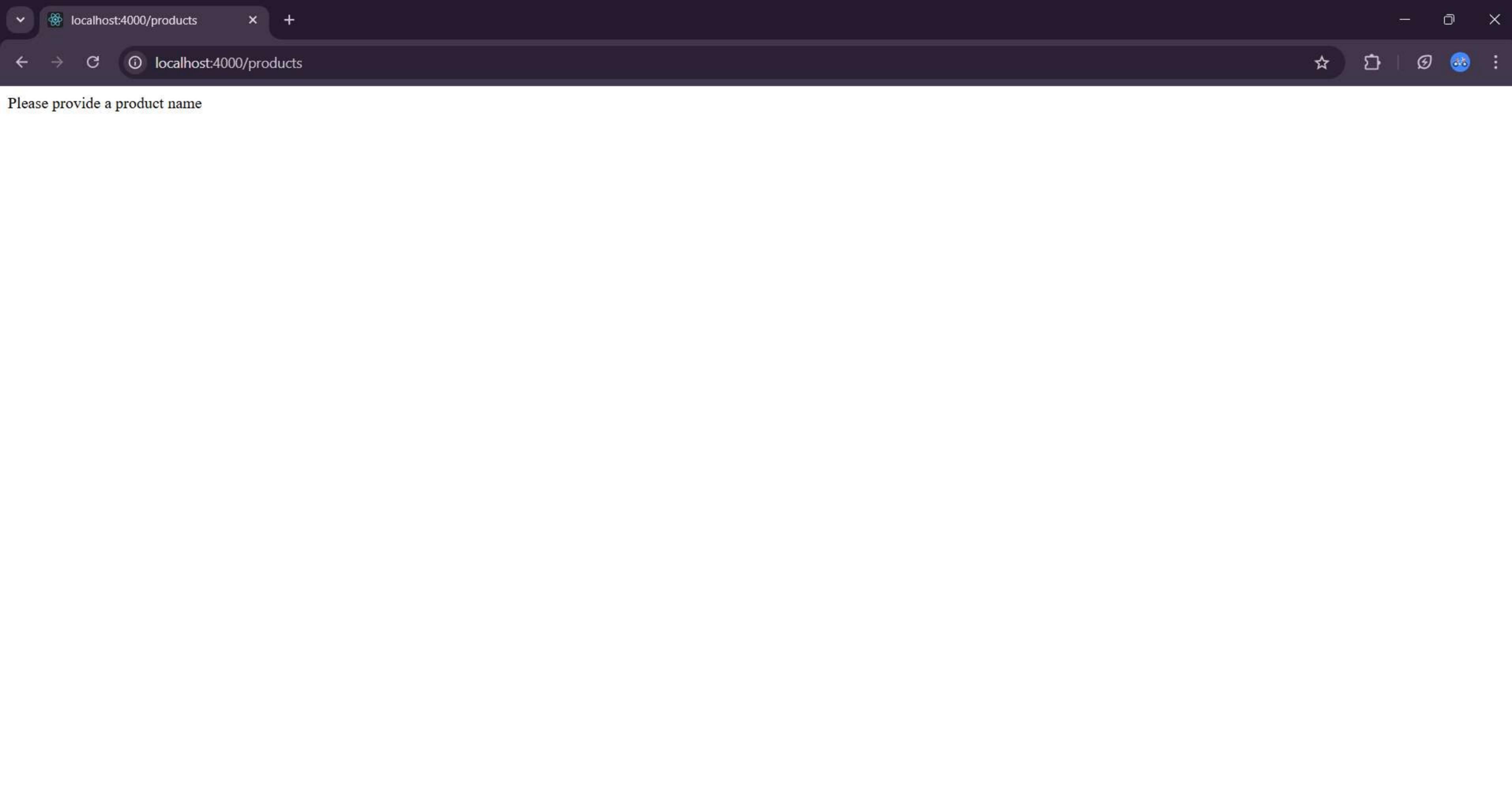
```
PS C:\Users\kushw\OneDrive\Desktop\Wipro NGA - MERN\Day19\Day19_assignment> node server.js
Server listening on http://localhost:4000
[2025-11-15T17:11:01.738Z] [GET] /
```

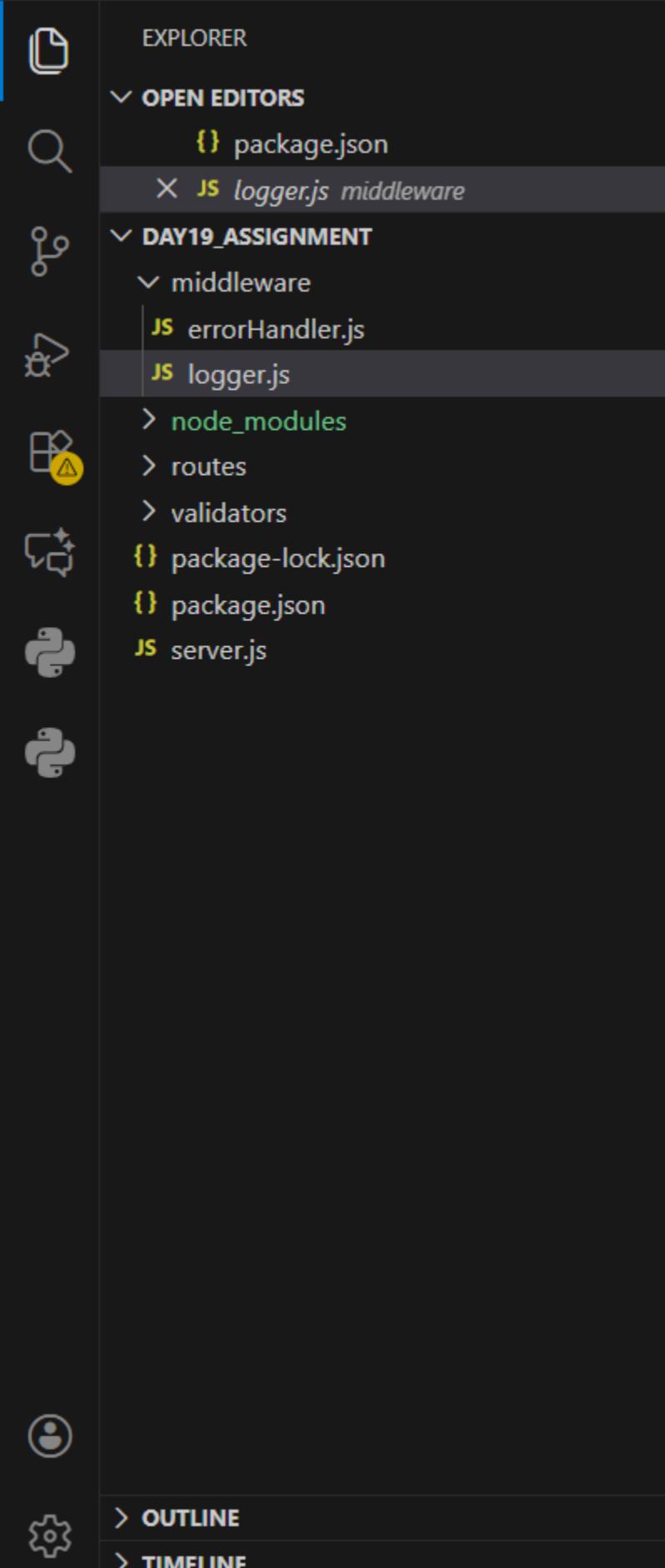




Pretty print

```
{"query": "Laptop", "message": "Searching for product: Laptop"}
```





PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kushw\OneDrive\Desktop\Wipro NGA - MERN\Day19\Day19_assignment> node server.js
[2025-11-15T17:11:01.738Z] [GET] /
④ PS C:\Users\kushw\OneDrive\Desktop\Wipro NGA - MERN\Day19\Day19_assignment> node server.js
Server listening on http://localhost:4000
[2025-11-15T17:14:37.553Z] [GET] /products?name=Laptop
[2025-11-15T17:15:39.906Z] [GET] /products
[2025-11-15T17:19:49.914Z] [GET] /.well-known/appspecific/com.chrome.devtools.json
[2025-11-15T17:19:54.102Z] [GET] /products
[2025-11-15T17:19:54.167Z] [GET] /.well-known/appspecific/com.chrome.devtools.json
[2025-11-15T17:19:54.239Z] [GET] /favicon.ico
❖ PS C:\Users\kushw\OneDrive\Desktop\Wipro NGA - MERN\Day19\Day19_assignment> node server.js
Server listening on http://localhost:4000
[2025-11-15T17:21:58.643Z] [GET] /products
[2025-11-15T17:21:58.681Z] [GET] /.well-known/appspecific/com.chrome.devtools.json
```

localhost:4000/books

localhost:4000/books

Pretty print

```
[  
  {  
    "id": 1,  
    "title": "1984",  
    "author": "Orwell"  
  },  
  {  
    "id": 2,  
    "title": "The Alchemist",  
    "author": "Coelho"  
  }]
```

```
{ } package.json X JS books.js
{ } package.json > { } dependencies
1  {
2      "name": "express-day19",
3      "version": "1.0.0",
4      "description": "Day 19 – Express.js",
5      "main": "server.js",
6      "scripts": {
7          "start": "node server.js",
8          "dev": "nodemon server.js"
9      },
10     "dependencies": {
11         "cors": "^2.8.5",
12         "express": "^4.21.2",
13         "express-validator": "^7.0.1"
14     },
15     "devDependencies": {
16         "nodemon": "^3.0.1"
17     },
18     "keywords": [],
19     "author": "",
20     "license": "TSC"
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\kushw\OneDrive\Desktop\Wipro NGA - MERN\Day19\Day19_assignment> node server.js
Server listening on http://localhost:4000
[2025-11-15T17:43:30.292Z] [GET] /books
```



You are using the Lightweight API Client, sign in or create an account to work with collections, environments and **unlock all free features in Postman**.

History

New

Import

POST http://localhost:4000/b

+

...

HTTP http://localhost:4000/books

Save

</>

Today

POST http://localhost:4000/books

POST http://localhost:4000/books

POST

http://localhost:4000/books

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

JSON

Beautify

```
1 {  
2   "title": "Sapiens",  
3   "author": "Harari"  
4 }  
5
```

Body

Cookies

Headers (8)

Test Results



Status: 201 Created

Time: 25 ms

Size: 316 B

Save Response

Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "id": 3,  
3   "title": "Sapiens",  
4   "author": "Harari"  
5 }
```

Create collections in Postman

Use collections to save your requests and share them with others.

Create a Collection



You are using the Lightweight API Client, sign in or create an account to work with collections, environments and unlock all free features in Postman.

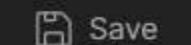
History

New Import

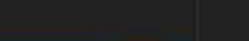
PUT http://localhost:4000/books/1



HTTP http://localhost:4000/books/1



Save



PUT

http://localhost:4000/books/1

Send



Params

Authorization

Headers (8)

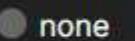
Body

Pre-request Script

Tests

Settings

Cookies



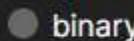
none



form-data



x-www-form-urlencoded



raw



binary



JSON

Beautify

```
1 {  
2   "title": "Nineteen Eighty-Four"  
3 }  
4
```

Body

Cookies

Headers (8)

Test Results



Status: 200 OK Time: 9 ms Size: 324 B

Save Response



Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "id": 1,  
3   "title": "Nineteen Eighty-Four",  
4   "author": "Orwell"  
5 }
```

Create collections in Postman

Use collections to save your requests and share them with others.

Create a Collection



Console

Not connected to a Postman account



localhost:4000/books

localhost:4000/books

Pretty print

```
[  
  {  
    "id": 1,  
    "title": "Nineteen Eighty-Four",  
    "author": "Orwell"  
  },  
  {  
    "id": 2,  
    "title": "The Alchemist",  
    "author": "Coelho"  
  },  
  {  
    "id": 3,  
    "title": "Sapiens",  
    "author": "Harari"  
  }]
```



You are using the Lightweight API Client, sign in or create an account to work with collections, environments and **unlock all free features in Postman**.

History

New Import

DEL http://localhost:4000/books/2



...

HTTP http://localhost:4000/books/2

Save



Today

DEL http://localhost:4000/books/2

PUT http://localhost:4000/books/1

POST http://localhost:4000/books

POST http://localhost:4000/books

DELETE http://localhost:4000/books/2

Send



Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

JSON

Beautify

1

Body Cookies Headers (8) Test Results

Status: 200 OK Time: 20 ms Size: 329 B

Save Response

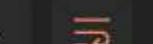
Pretty

Raw

Preview

Visualize

JSON



```
1 {  
2   "deleted": {  
3     "id": 2,  
4     "title": "The Alchemist",  
5     "author": "Coelho"  
6   }  
7 }
```

Create collections in Postman

Use collections to save your requests and share them with others.

Create a Collection

localhost:4000/books

localhost:4000/books

Pretty print

```
[  
  {  
    "id": 1,  
    "title": "Nineteen Eighty-Four",  
    "author": "Orwell"  
  },  
  {  
    "id": 3,  
    "title": "Sapiens",  
    "author": "Harari"  
  }]
```