

you t (BS) K localh expre How t Atlas k x

localhost:3000

Home Page

Express

JS index.js

process.env

my-express-app > JS index.js > ...

```
6 }
9
10 app.use(logRequests); // Apply it globally
11
12 app.get('/', (req, res) => {
13   res.send('Home Page');
14 });
15
16 app.get('/about', (req, res) => {
17   res.send('About Page');
18 });
19
20 app.listen(3000, () => {
21   console.log('Server running on port 3000');
22 });
23
```

PROBLEMS OUTPUT TERMINAL

node - my-express-app

```
PS C:\Users\ishit\OneDrive\Desktop\Express> cd my-express-app
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server running on port 3000
GET request to /redirect at 4:53:20 pm
GET request to / at 4:53:26 pm
GET request to /favicon.ico at 4:53:26 pm
[]
```

OUTLINE

TIMELINE

Ln 23, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live

Browser tabs: you tub, (85) Kal, localhost, express, How to, loc x

Address bar: localhost:3000/about

Page title: About Page

Page content: (Empty)

Meeting title: Full Stack Web-D Interns Evening Session Day-8

Time: 26:57

Controls: Pop out, Chat, Raise, React, More, Camera, Mic, Share

Participants: VR, VV, AS, RM, AG (On hold), [Profile Picture]

Code Editor (VS Code):

```
index.js
1 // Importing the required modules
2 const express = require('express');
3 const app = express();
4
5 // Setting the port number
6 const PORT = 3000;
7
8 // Creating the server
9 const server = app.listen(PORT, () => {
10   console.log(`Server running on port ${PORT}`);
11 });
12
13 // Exporting the server
14 module.exports = server;
```

Terminal:

```
PS C:\Users\mohamed> node index.js
Server running on port 3000
PS C:\Users\mohamed> node index.js
Server running on port 3000
```

Vishal Rai (Guest)

localhost:3000

Something went wrong!

EXPLORE

- my-express-app
  - node\_modules
  - index.js
  - package-lock.json
  - package.json
  - process.env

```
1 const express = require('express');
2 const app = express();
3
4 // Sample route to trigger an error
5 app.get('/', (req, res, next) => {
6   const error = new Error('This is a test error');
7   error.status = 500;
8   next(error);
9 });
10
11 // Error-handling middleware function
12 function errorHandler(err, req, res, next) {
13   console.error(err.message); // Log the error message
14   res.status(err.status || 500).send('Something went w
15 }
16
17 // Apply error-handling middleware
```

PROBLEMS TERMINAL

node - my-express-app

at node:internal/main/run\_main\_module:33:47

Node.js v24.1.0

PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js

Server running on port 3000

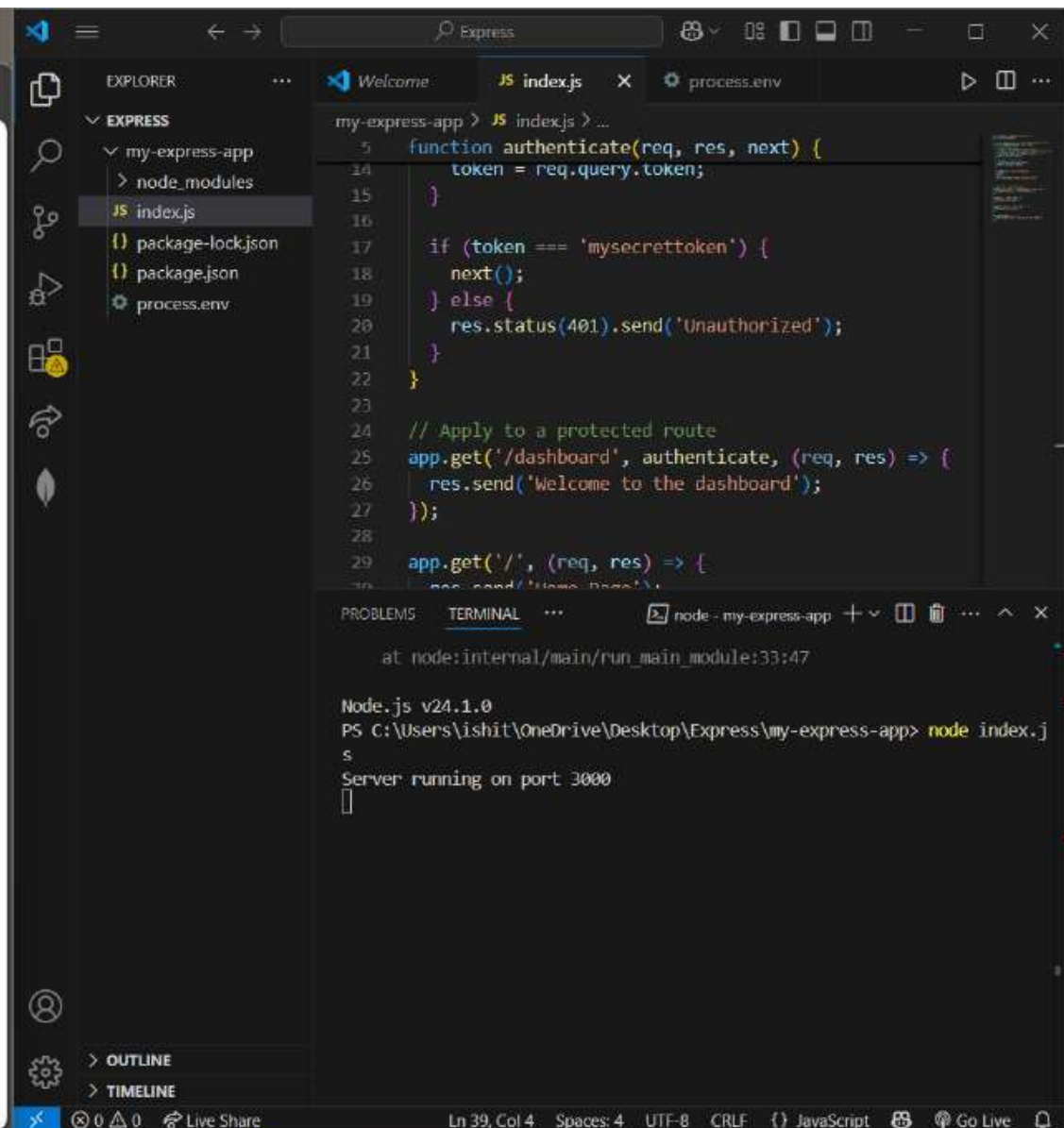
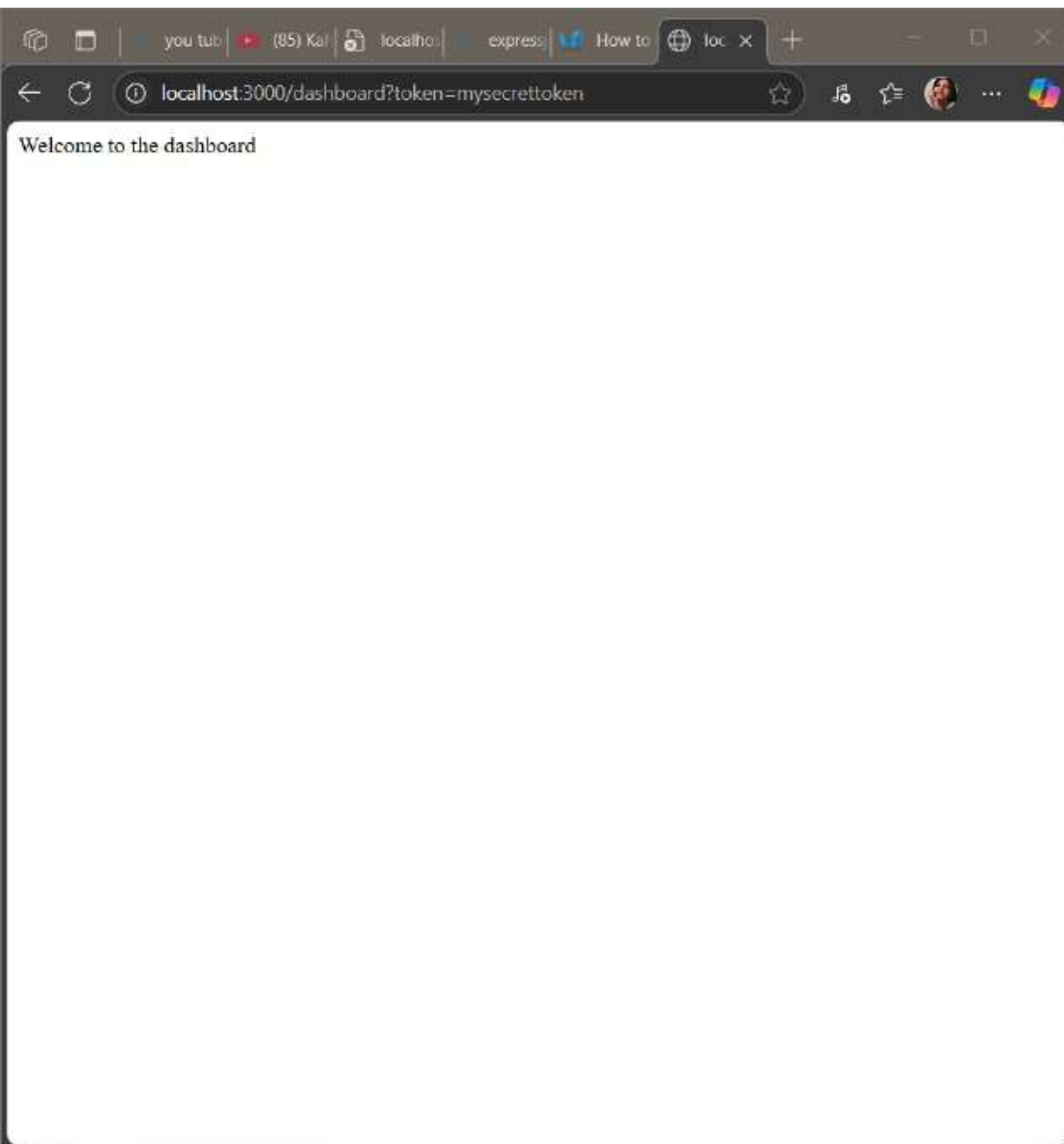
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js

Server is running on http://localhost:3000

This is a test error

This is a test error

Ln 24, Col 2 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live



Visual Studio Code interface showing a project named "my-express-app". The Explorer sidebar on the left lists files: `node_modules`, `index.js`, `package-lock.json`, `package.json`, and `process.env`. The main editor displays the `index.js` file with the following code:

```
6 app.get('/', (req, res) => {
7   res.send('welcome to the home Page!');
8 });
9
10
11 // Define a route for the '/about' URL with a GET request
12 app.get('/about', (req, res) => {
13   // Send a response to the client
14   res.send('About Us');
15 });
16
17 // Define a route for the '/contact' URL with a GET request
18 app.get('/contact', (req, res) => {
19   // Send a response to the client
20   res.send('Contact Us');
21 });
22
23 // Start server
24 app.listen(port, () => console.log('Server is running'));
```

The TERMINAL panel at the bottom shows the command prompt output for running the application:

```
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server is running on http://localhost:3000
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server is running on http://localhost:3000
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server is running on http://localhost:3000
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server is running at http://localhost:3000
```

The status bar at the bottom indicates the file is `index.js` at line 24, column 86, using UTF-8 encoding and CRLF line endings.

Web browser window showing the URL `localhost:3000/about`. The page content displays "About Us".



Visual Studio Code interface showing a web application running on localhost:3000.

**EXPLORER**

- EXPRESS
  - my-express-app
    - node\_modules
    - index.js
    - package-lock.json
    - package.json
    - process.env

**JS index.js**

```
1 my-express-app > JS index.js > ...
2 const app = express();
3
4
5
6
7 // Define a route with a parameter for user ID
8 app.get('/user/:id', (req, res) => {
9   // Access the route parameter 'id' from the request
10   const userId = req.params.id;
11
12   // Respond with the user ID
13   res.send('User ID: ${userId}');
14 });
15
16 // Start the server
17 app.listen(3000, () => {
18   console.log('Server is running on http://localhost:3000');
19 });
20
```

**TERMINAL**

```
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server is running on port 3000
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server is running on http://localhost:3000
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server is running on http://localhost:3000
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server is running on http://localhost:3000
```

Web browser window showing the result of the GET request to localhost:3000/user/123.

localhost:3000/user/123

User ID: 123

The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project named 'my-express-app' with files: 'node\_modules', 'index.js', 'package-lock.json', 'package.json', and 'process.env'. The main editor area displays the content of 'index.js'.

```
1 // Require the Express module
2 const express = require('express');
3
4 // Create an Express application
5 const app = express();
6
7 // Define a route that accesses and returns query parameter
8 app.get('/search', (req, res) => {
9   // Access the query parameter 'q' from the request
10   const queryParams = req.query.q;
11
12   // Check if the query parameter exists
13   if (queryParams) {
14     // Respond with the query parameter value
15     res.send(`Search query: ${queryParams}`);
16   } else {
17     // Respond with an error message if 'q' is not provided
18     res.status(400).send('Query parameter "q" is required');
19   }
20 });
21
```

Below the editor is a terminal window titled 'node - my-express-app'. It shows the following commands and output:

```
Node.js v24.1.0
PS C:\Users\ishit\OneDrive\Desktop\Express> cd my-express-app
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server is running on port 3000
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server is running on http://localhost:3000
PS C:\Users\ishit\OneDrive\Desktop\Express\my-express-app> node index.js
Server is running on http://localhost:3000
```

The image shows a web browser window with the address bar set to 'localhost:3000/search'. The page content displays a 400 error message:

400 (Bad Request)

Query parameter "q" is required

The browser's status bar at the bottom shows 'ENG IN' and the time '11:14 10-06-2025'.