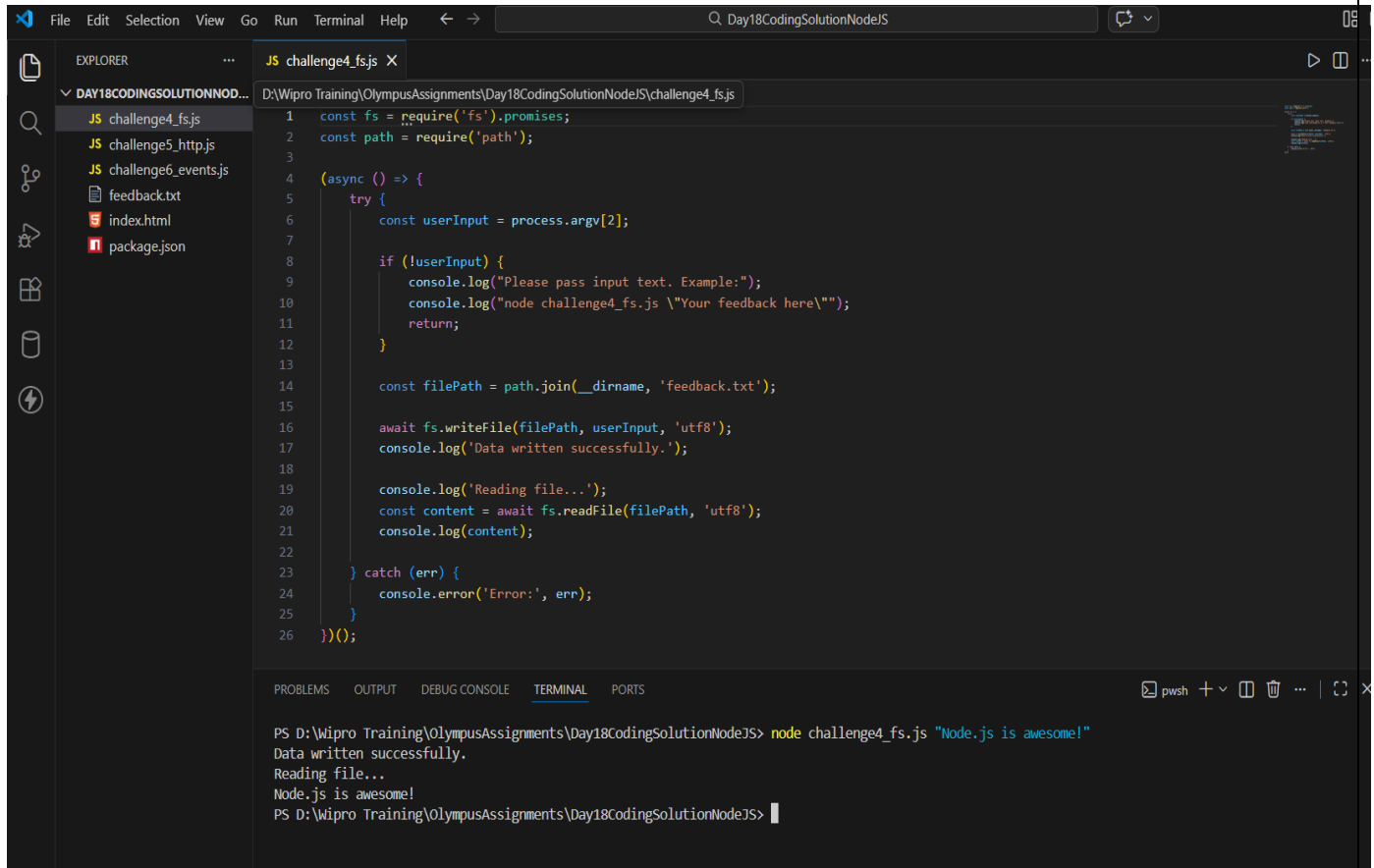


## Day 18 - Node.js Core Modules

### Challenge 4: File System (fs) Module

In this challenge, the Node.js fs module with promises is used to take user input from the command line and store it in a file named feedback.txt. The same file is then read asynchronously and its contents are displayed on the console to confirm successful file operations.

#### Output:



```
1  const fs = require('fs').promises;
2  const path = require('path');
3
4  (async () => {
5    try {
6      const userInput = process.argv[2];
7
8      if (!userInput) {
9        console.log("Please pass input text. Example:");
10       console.log("node challenge4_fs.js \"Your feedback here\"");
11       return;
12     }
13
14     const filePath = path.join(__dirname, 'feedback.txt');
15
16     await fs.writeFile(filePath, userInput, 'utf8');
17     console.log('Data written successfully.');

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS



```
PS D:\wipro Training\OlympusAssignments\Day18CodingSolutionNodeJS> node challenge4_fs.js "Node.js is awesome!"
Data written successfully.
Reading file...
Node.js is awesome!
PS D:\wipro Training\OlympusAssignments\Day18CodingSolutionNodeJS>
```


```

## Challenge 5: HTTP Module

This challenge demonstrates creating a basic HTTP server using Node.js without Express. The server handles multiple routes and serves static HTML content when available, while also supporting graceful shutdown.

### Output:



**Hello from Node.js Server (static HTML)**

This index.html was served as static content.

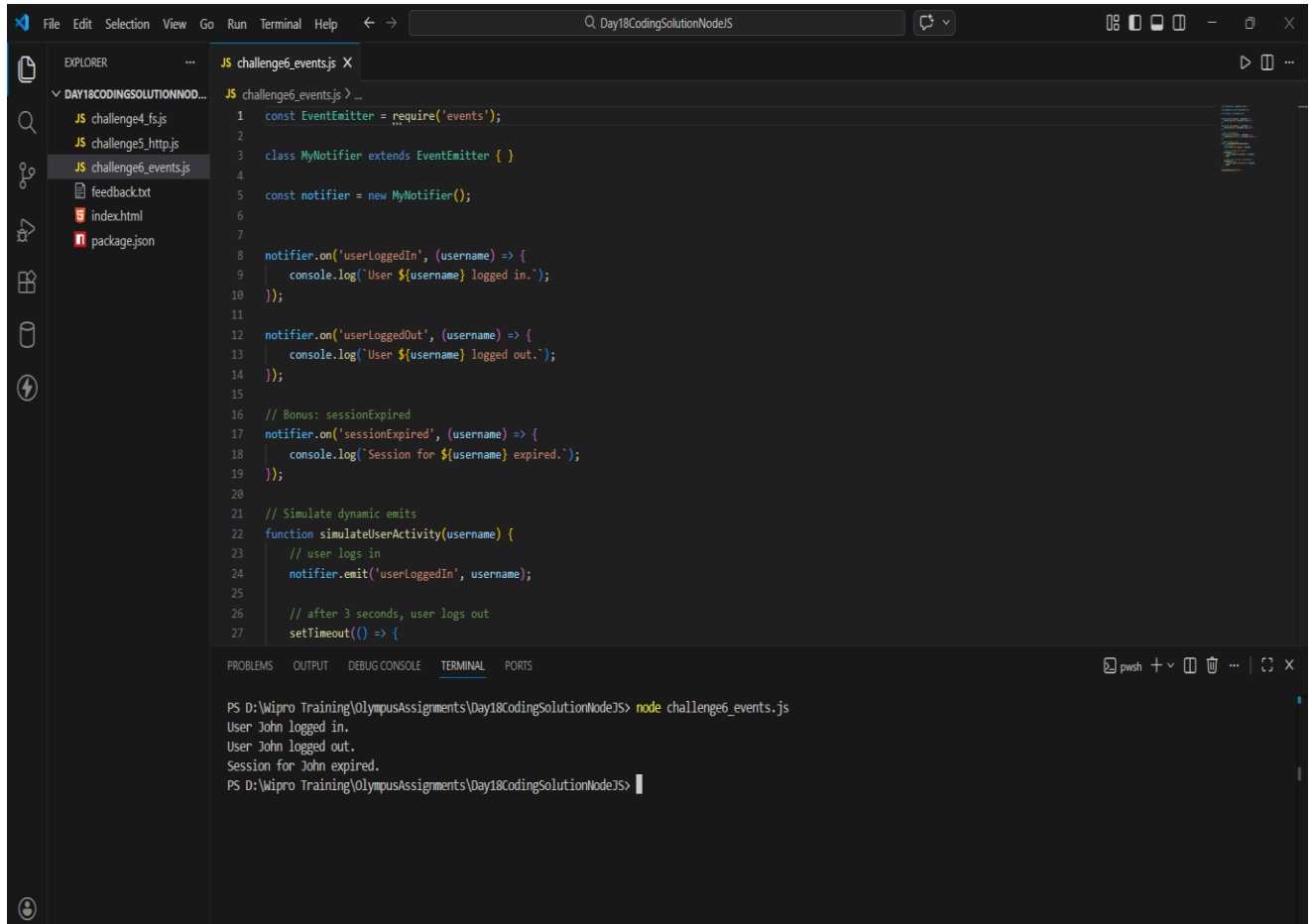


**About Page**

## Challenge 6: Events Module

In this challenge, the EventEmitter class is used to simulate event-driven behavior. Custom events such as user login, logout, and session expiration are emitted dynamically to represent a simple notification system.

### Output:



The screenshot shows a Visual Studio Code editor window with the file `challenge6_events.js` open. The file contains the following code:

```
1 const EventEmitter = require('events');
2
3 class MyNotifier extends EventEmitter { }
4
5 const notifier = new MyNotifier();
6
7
8 notifier.on('userLoggedIn', (username) => {
9   console.log('User ${username} logged in.');
```

```
10 });
11
12 notifier.on('userLoggedOut', (username) => {
13   console.log('User ${username} logged out.');
```

```
14 });
15
16 // Bonus: sessionExpired
17 notifier.on('sessionExpired', (username) => {
18   console.log('Session for ${username} expired.');
```

```
19 });
20
21 // Simulate dynamic emits
22 function simulateUserActivity(username) {
23   // user logs in
24   notifier.emit('userLoggedIn', username);
25
26   // after 3 seconds, user logs out
27   setTimeout(() => {
```

The terminal output shows the execution of the script:

```
PS D:\wipro Training\OlympusAssignments\Day18CodingSolutionNodeJS> node challenge6_events.js
User John logged in.
User John logged out.
Session for John expired.
PS D:\wipro Training\OlympusAssignments\Day18CodingSolutionNodeJS>
```