

Name : Kajavadra Om S.

Sem. : 7

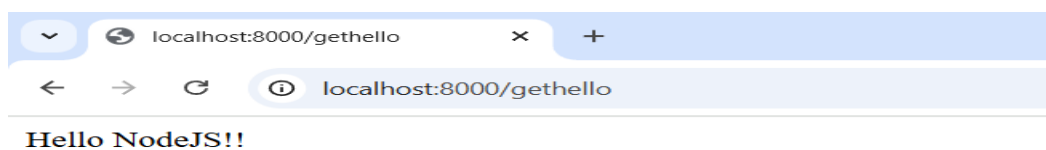
Roll No. : 41

Div. : A

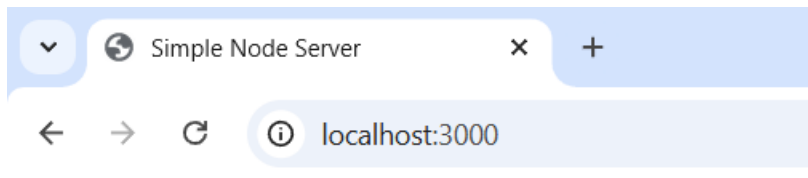
---

Q1. Develop nodejs application with following requirements:

- Develop a route `"/gethello"` with GET method. It displays "Hello NodeJS!!" as response.
- Make an HTML page and display.
- Call `"/gethello"` route from HTML page using AJAX call. (Any frontend AJAX call API can be used.)



Q2. Develop a web server which serves static resources.

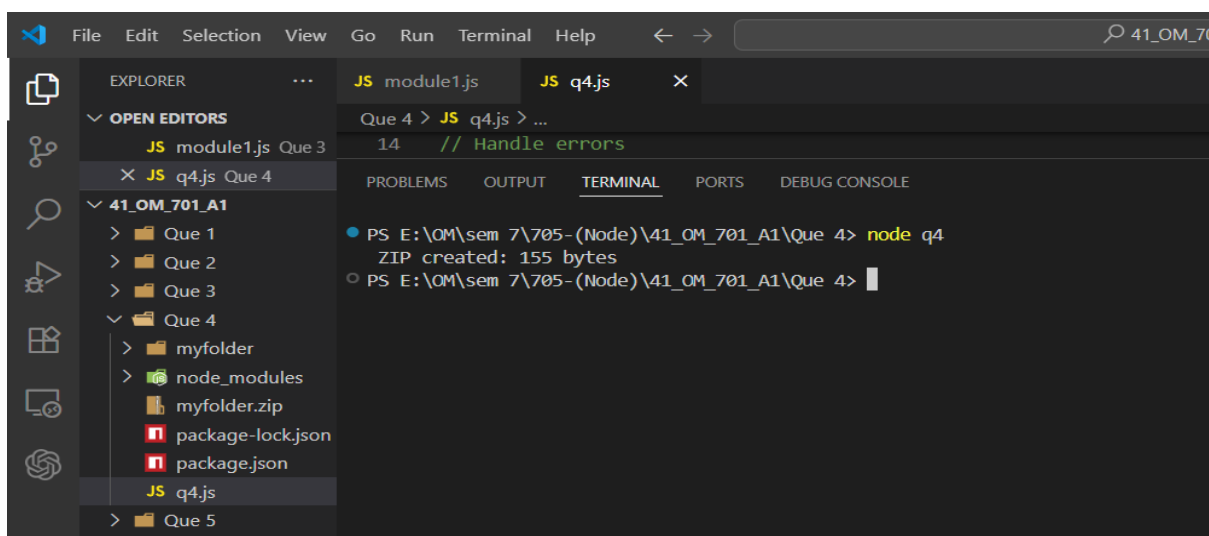


# Hello from Node.js!

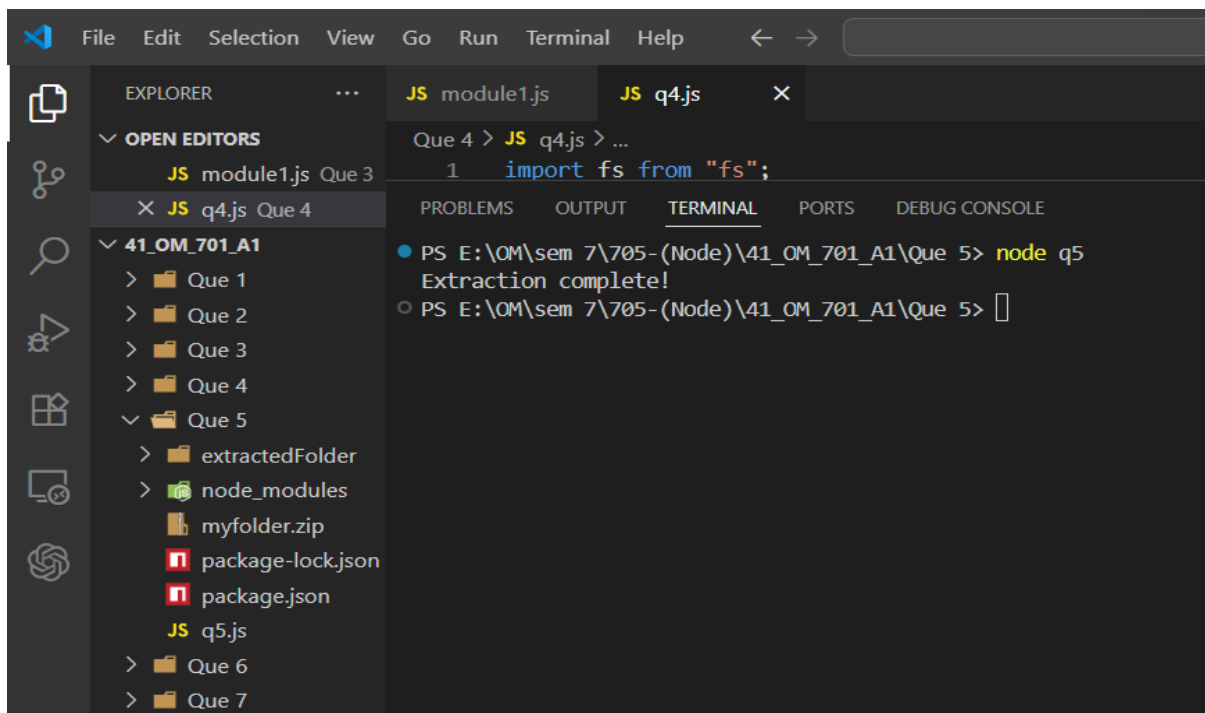
Q3. Develop a module for domain specific chatbot and use it in a command line application.

```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS E:\OM\sem 7\705-(Node)\41_OM_701_A1\Que 3> node q3
You ==> Hi
TravelBot ==> Hello! Ready to explore the world?
You ==> What is your name
TravelBot ==> I'm TravelBot, your travel companion!
You ==> How many year old your company
TravelBot ==> I'm 3 years old – young but well-traveled!
You ==> where are you from
TravelBot ==> I'm based in Mumbai, but I know places all over the globe!
You ==> please recommend place to travel
TravelBot ==> I recommend visiting Kyoto for culture, Bali for beaches, and Iceland for adventure!
You ==> what is advice and tip from you to travel
TravelBot ==> Always carry a power bank, learn a few local phrases, and respect local customs!
You ==> []
```

Q4. Write a program to create a compressed zip file for a folder.



Q5. Write a program to extract a zip file.



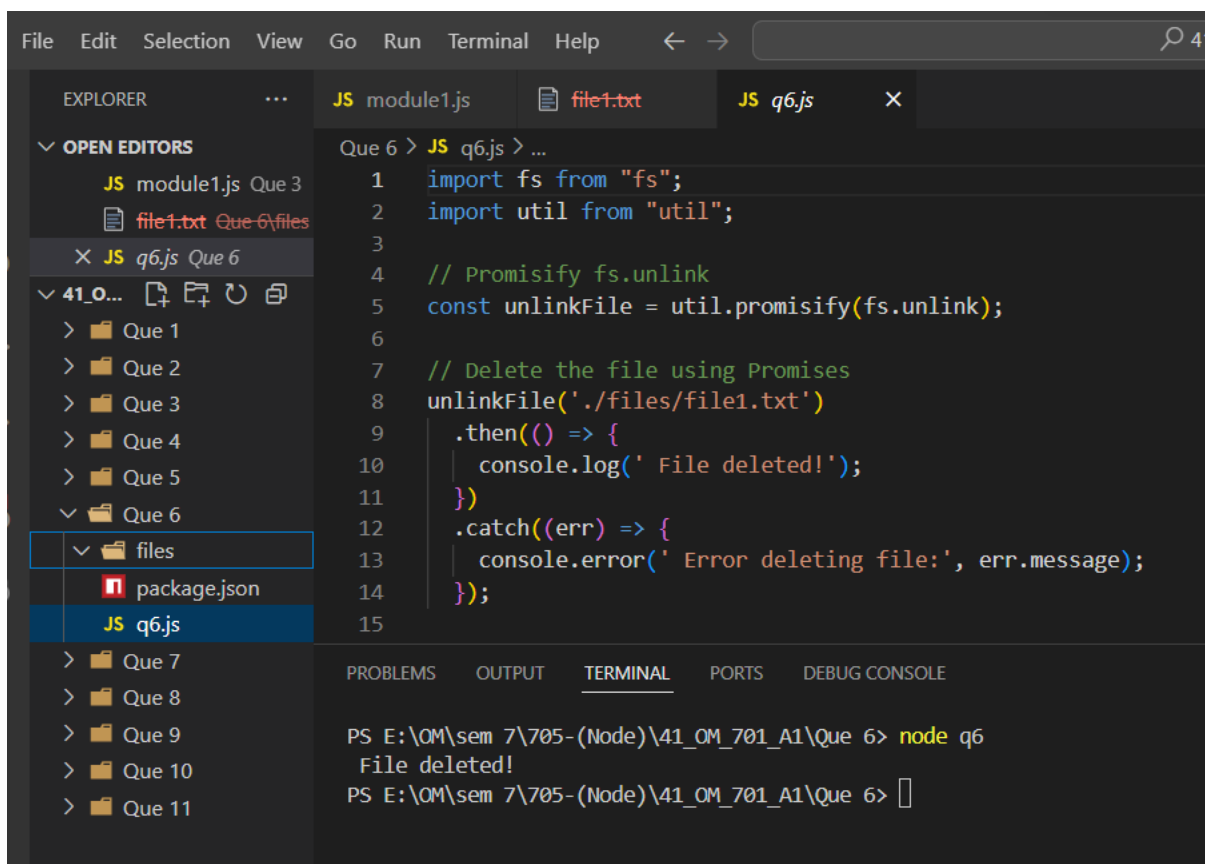
The screenshot shows the VS Code interface with the Explorer sidebar on the left. The '41\_OM\_701\_A1' folder is expanded, showing subfolders 'Que 1' through 'Que 7', and files 'extractedFolder', 'node\_modules', 'myfolder.zip', 'package-lock.json', 'package.json', and 'q5.js'. The main editor area shows the 'q4.js' file with the following code:

```
1 import fs from "fs";
```

The TERMINAL panel at the bottom shows the command prompt output:

```
PS E:\OM\sem 7\705-(Node)\41_OM_701_A1\Que 5> node q5
Extraction complete!
PS E:\OM\sem 7\705-(Node)\41_OM_701_A1\Que 5>
```

Q6. Write a program to promisify fs.unlink function and call it.



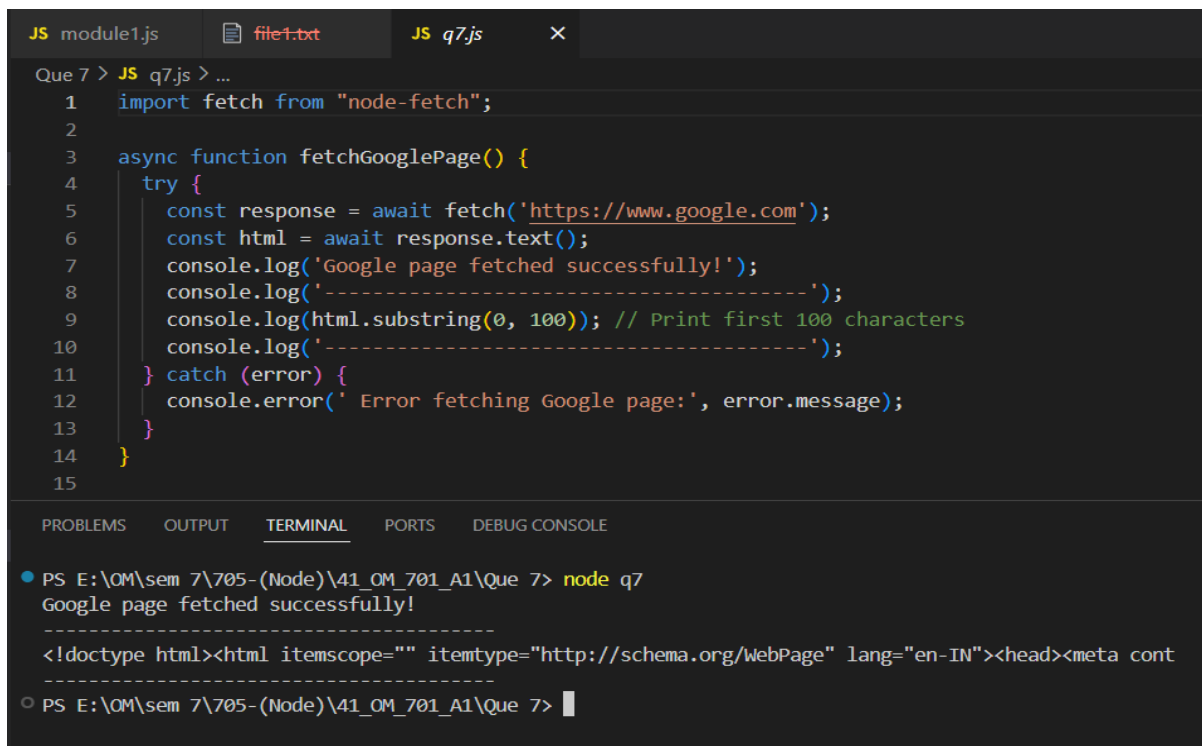
The screenshot shows the VS Code interface with the Explorer sidebar on the left. The '41\_OM\_701\_A1' folder is expanded, showing subfolders 'Que 1' through 'Que 11', and files 'package.json' and 'q6.js'. The main editor area shows the 'q6.js' file with the following code:

```
1 import fs from "fs";
2 import util from "util";
3
4 // Promisify fs.unlink
5 const unlinkFile = util.promisify(fs.unlink);
6
7 // Delete the file using Promises
8 unlinkFile('./files/file1.txt')
9   .then(() => {
10     console.log(' File deleted!');
11   })
12   .catch((err) => {
13     console.error(' Error deleting file:', err.message);
14   });
15
```

The TERMINAL panel at the bottom shows the command prompt output:

```
PS E:\OM\sem 7\705-(Node)\41_OM_701_A1\Que 6> node q6
File deleted!
PS E:\OM\sem 7\705-(Node)\41_OM_701_A1\Que 6>
```

Q7. Fetch data of google page using node-fetch using async-await model.

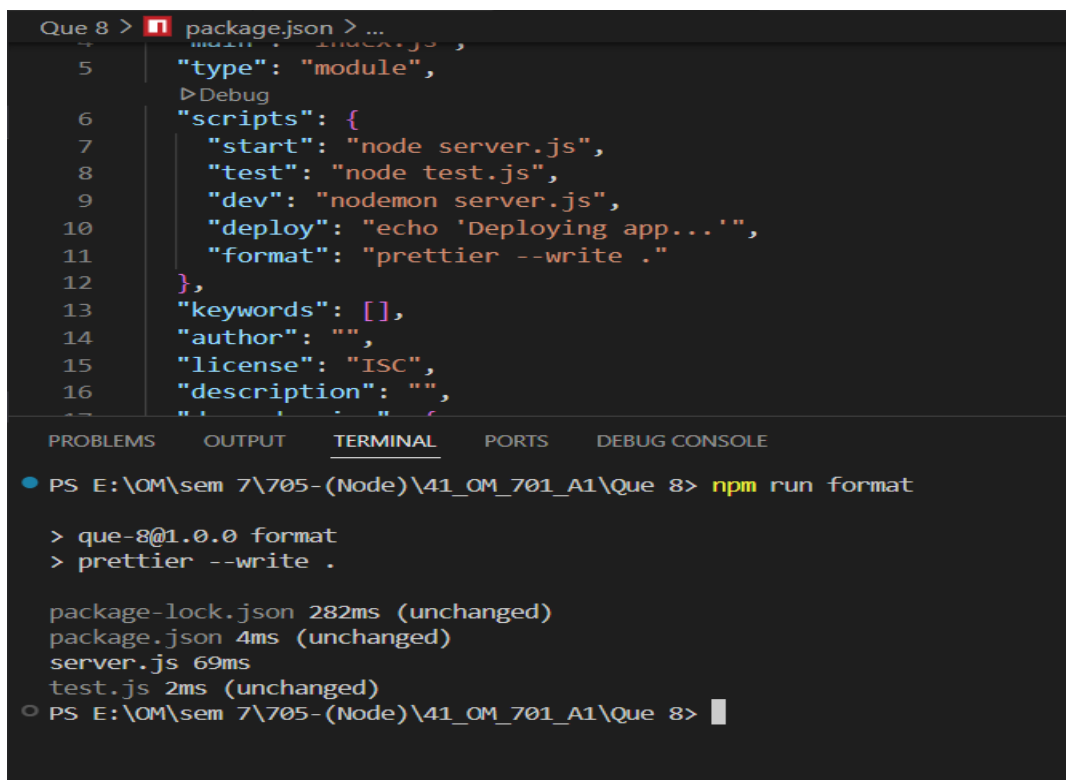


```
JS module1.js  file1.txt  JS q7.js  X
Que 7 > JS q7.js > ...
1  import fetch from "node-fetch";
2
3  async function fetchGooglePage() {
4      try {
5          const response = await fetch('https://www.google.com');
6          const html = await response.text();
7          console.log('Google page fetched successfully!');
8          console.log('-----');
9          console.log(html.substring(0, 100)); // Print first 100 characters
10         console.log('-----');
11     } catch (error) {
12         console.error(' Error fetching Google page:', error.message);
13     }
14 }
15
```

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
● PS E:\OM\sem 7\705-(Node)\41_OM_701_A1\Que 7> node q7
Google page fetched successfully!
-----
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN"><head><meta cont
-----
○ PS E:\OM\sem 7\705-(Node)\41_OM_701_A1\Que 7> |
```

Q8. Set a server script, a test script and 3 user defined scripts in package.json file in your nodejs application.



```
Que 8 > package.json > ...
5  "type": "module",
    > Debug
6  "scripts": {
7      "start": "node server.js",
8      "test": "node test.js",
9      "dev": "nodemon server.js",
10     "deploy": "echo 'Deploying app...'",
11     "format": "prettier --write ."
12 },
13 "keywords": [],
14 "author": "",
15 "license": "ISC",
16 "description": "",
17
```

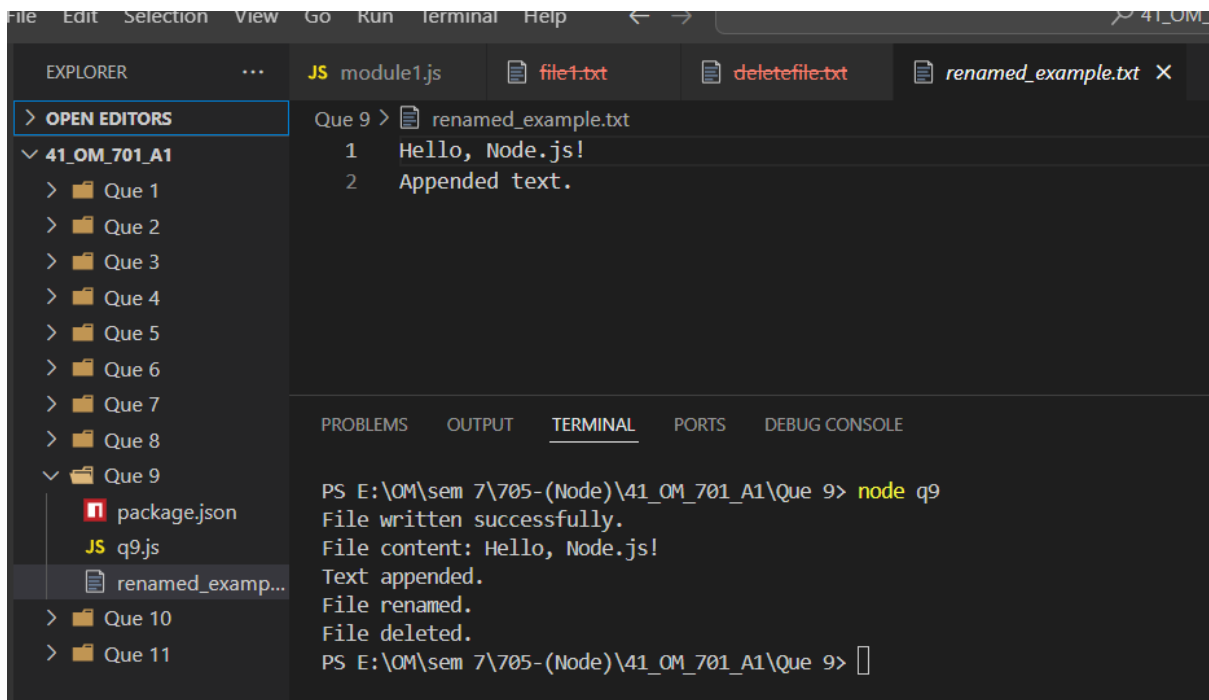
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
● PS E:\OM\sem 7\705-(Node)\41_OM_701_A1\Que 8> npm run format

> que-8@1.0.0 format
> prettier --write .

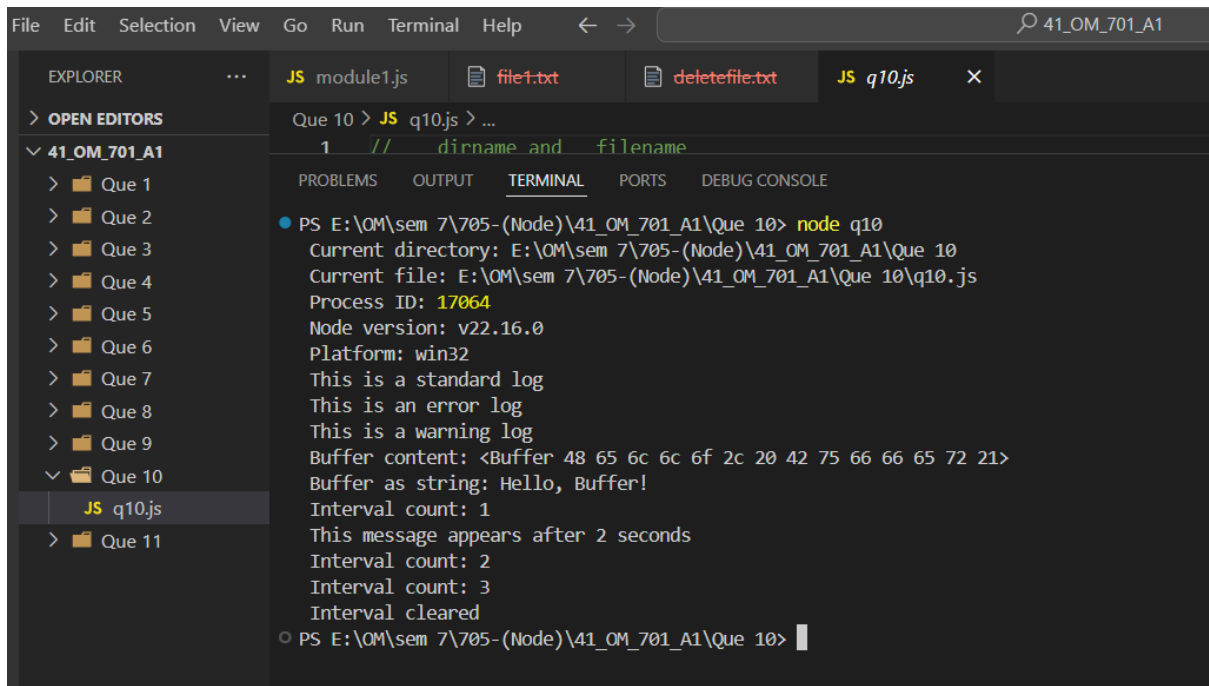
package-lock.json 282ms (unchanged)
package.json 4ms (unchanged)
server.js 69ms
test.js 2ms (unchanged)
○ PS E:\OM\sem 7\705-(Node)\41_OM_701_A1\Que 8> |
```

Q9. A program which calls useful functions in fs module.



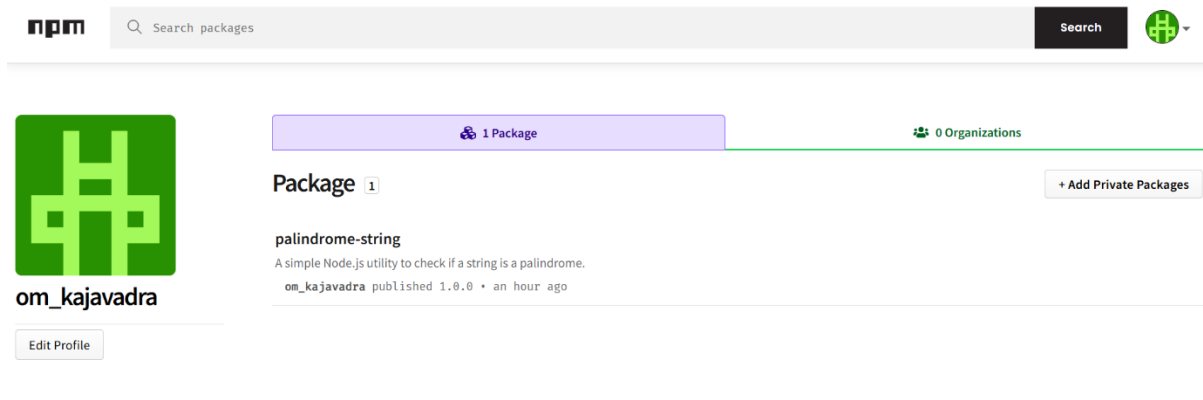
The screenshot shows the VS Code interface with the Explorer sidebar on the left displaying a project structure under '41\_OM\_701\_A1' containing folders 'Que 1' through 'Que 11' and files 'package.json', 'q9.js', and 'renamed\_examp...'. The main editor area shows 'module1.js' with two lines of code: '1 Hello, Node.js!' and '2 Appended text.'. The TERMINAL panel at the bottom shows the command 'node q9' being executed, with the following output: 'File written successfully.', 'File content: Hello, Node.js!', 'Text appended.', 'File renamed.', 'File deleted.', and the prompt 'PS E:\OM\sem 7\705-(Node)\41\_OM\_701\_A1\Que 9>'.

Q10. A program which uses global objects in nodejs.

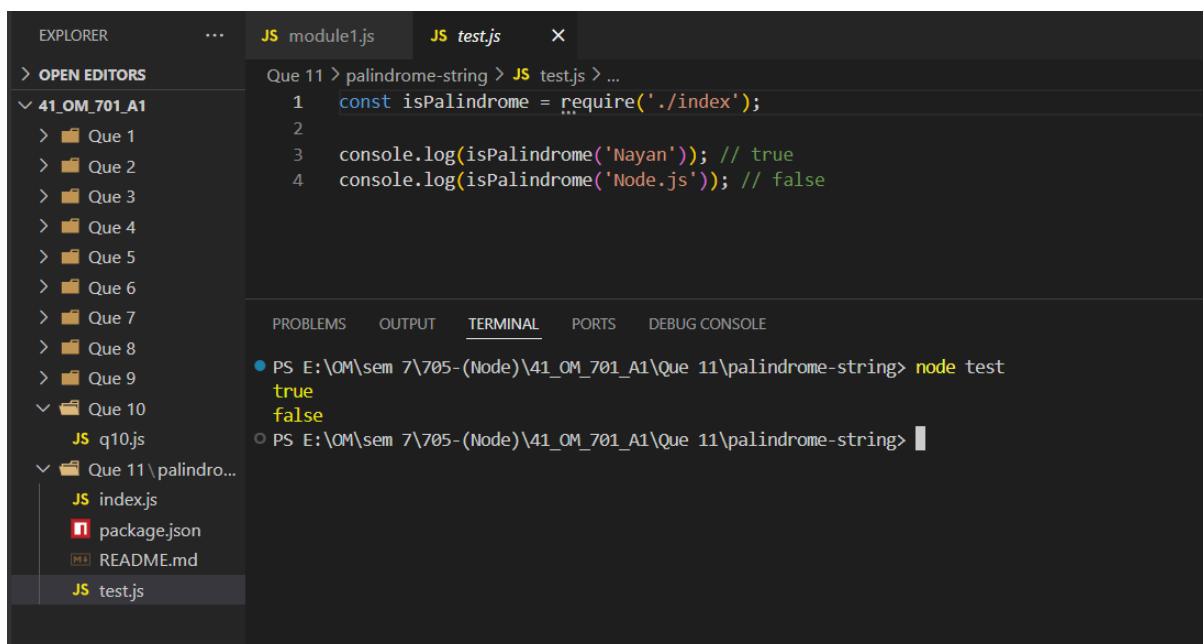


The screenshot shows the VS Code interface with the Explorer sidebar on the left displaying a project structure under '41\_OM\_701\_A1' containing folders 'Que 1' through 'Que 11' and files 'package.json', 'q9.js', and 'q10.js'. The main editor area shows 'module1.js' with two lines of code: '1 // dirname and filename'. The TERMINAL panel at the bottom shows the command 'node q10' being executed, with the following output: 'Current directory: E:\OM\sem 7\705-(Node)\41\_OM\_701\_A1\Que 10', 'Current file: E:\OM\sem 7\705-(Node)\41\_OM\_701\_A1\Que 10\q10.js', 'Process ID: 17064', 'Node version: v22.16.0', 'Platform: win32', 'This is a standard log', 'This is an error log', 'This is a warning log', 'Buffer content: <Buffer 48 65 6c 6c 6f 2c 20 42 75 66 66 65 72 21>', 'Buffer as string: Hello, Buffer!', 'Interval count: 1', 'This message appears after 2 seconds', 'Interval count: 2', 'Interval count: 3', 'Interval cleared', and the prompt 'PS E:\OM\sem 7\705-(Node)\41\_OM\_701\_A1\Que 10>'.

## Q11. Develop a useful package and publish it on npmjs.com



The screenshot shows the npm profile page for the user **om\_kajavadra**. The profile includes a green logo, the username, and an "Edit Profile" button. The "Package" section shows one published package: **palindrome-string**. The package description is "A simple Node.js utility to check if a string is a palindrome." and it was published by **om\_kajavadra** 1.0.0 version, an hour ago. There is a button to "+ Add Private Packages".



The screenshot shows a VS Code editor with the following components:

- EXPLORER:** Shows the project structure with folders "Que 1" through "Que 10" and "Que 11\palindro...". Under "Que 11\palindro...", there are files **index.js**, **package.json**, **README.md**, and **test.js**.
- JS module1.js:** Contains the code for the **isPalindrome** function.
- JS test.js:** Contains the code for testing the **isPalindrome** function.
- TERMINAL:** Shows the command `node test` being executed, resulting in `true` and `false` output.