

Practical Assignment-1

Name: Yuti . A . Naik

Roll No.: 68

Sem: 7

Div: A

Question 1:

Welcome to NodeJS App

Get Hello Message

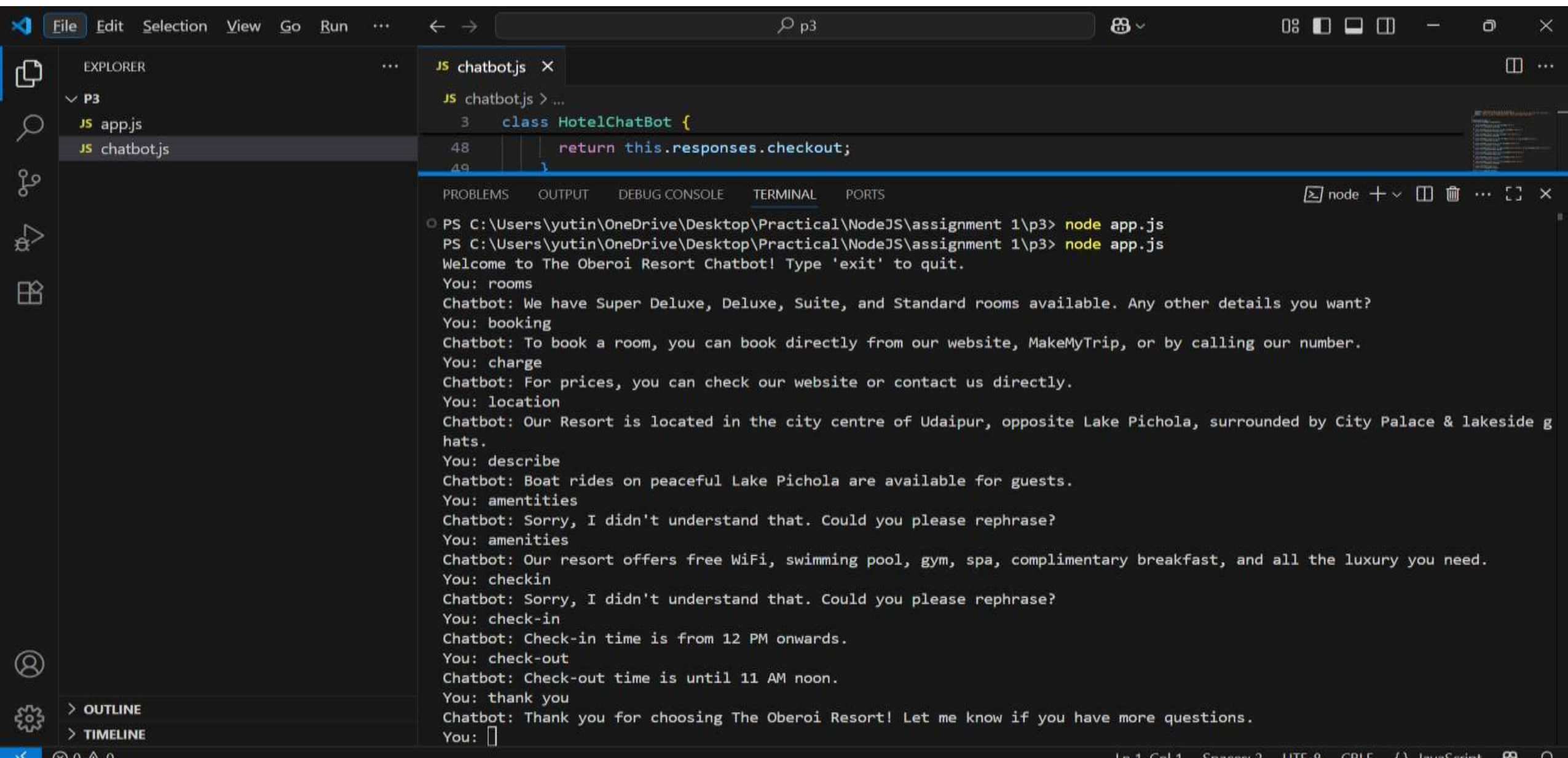
Hello NodeJS!!

Question 2:

Hello from static server



Question 3:



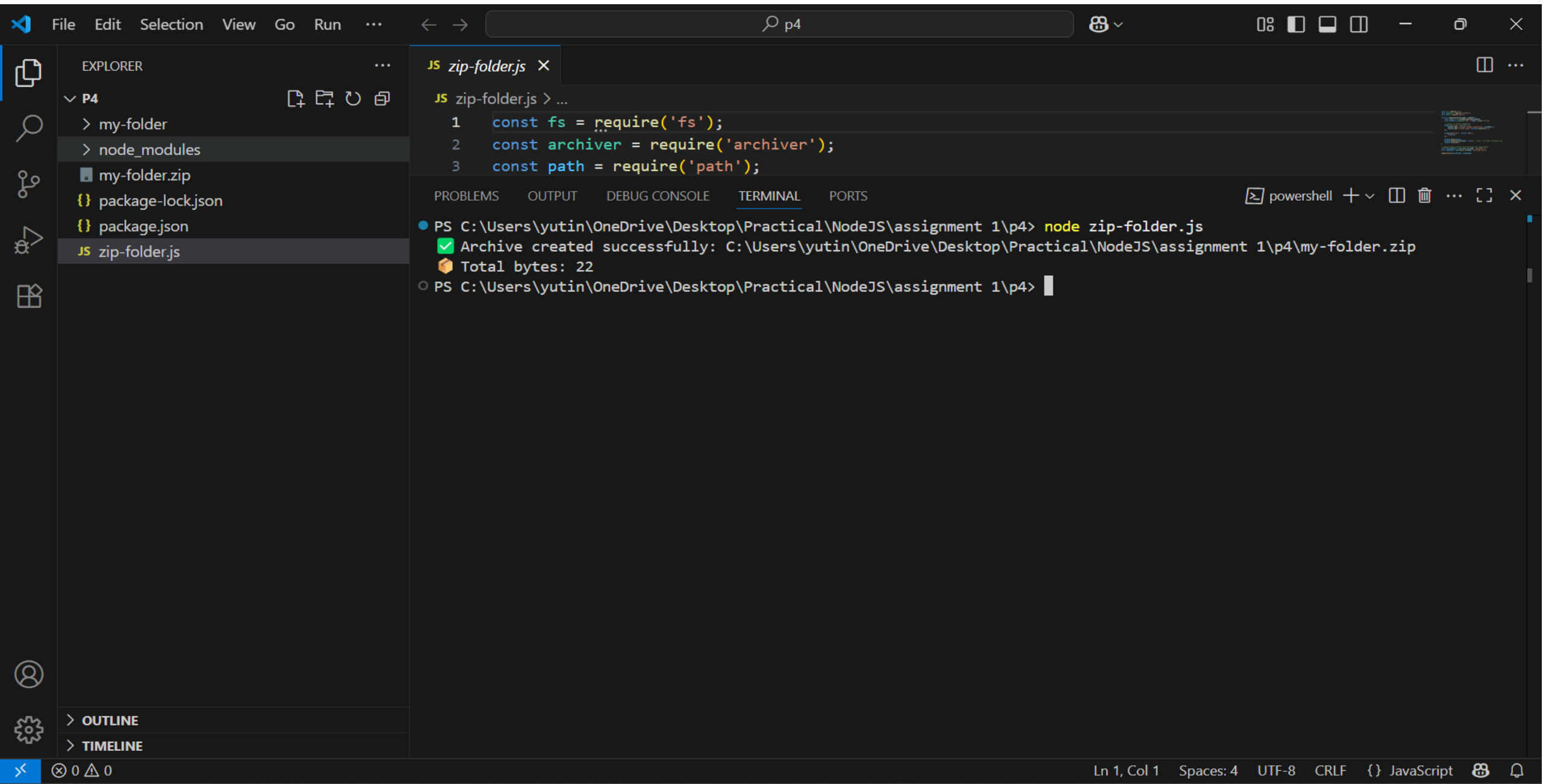
The screenshot shows the Visual Studio Code editor with a project named 'P3'. The Explorer sidebar on the left shows the file structure with 'app.js' and 'chatbot.js'. The 'chatbot.js' file is open in the editor, showing a JavaScript class 'HotelChatBot' with a 'return this.responses.checkout;' statement on line 48. The bottom panel displays the 'TERMINAL' tab, showing the execution of 'node app.js' and a series of user and chatbot interactions. The chatbot provides information about room types, booking methods, prices, location, and amenities, and handles user input like 'rooms', 'booking', 'charge', 'location', 'describe', 'amenities', 'checkin', 'check-out', and 'thank you'.

```
JS chatbot.js > ...
3 class HotelChatBot {
48     return this.responses.checkout;
49 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p3> node app.js
PS C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p3> node app.js
Welcome to The Oberoi Resort Chatbot! Type 'exit' to quit.
You: rooms
Chatbot: We have Super Deluxe, Deluxe, Suite, and Standard rooms available. Any other details you want?
You: booking
Chatbot: To book a room, you can book directly from our website, MakeMyTrip, or by calling our number.
You: charge
Chatbot: For prices, you can check our website or contact us directly.
You: location
Chatbot: Our Resort is located in the city centre of Udaipur, opposite Lake Pichola, surrounded by City Palace & lakeside ghats.
You: describe
Chatbot: Boat rides on peaceful Lake Pichola are available for guests.
You: amenities
Chatbot: Sorry, I didn't understand that. Could you please rephrase?
You: amenities
Chatbot: Our resort offers free WiFi, swimming pool, gym, spa, complimentary breakfast, and all the luxury you need.
You: checkin
Chatbot: Sorry, I didn't understand that. Could you please rephrase?
You: check-in
Chatbot: Check-in time is from 12 PM onwards.
You: check-out
Chatbot: Check-out time is until 11 AM noon.
You: thank you
Chatbot: Thank you for choosing The Oberoi Resort! Let me know if you have more questions.
You:

Question 4:



The screenshot shows the Visual Studio Code interface with the following components:

- Explorer Panel:** Displays the file structure of a project named 'P4'. It includes a folder 'my-folder' and a file 'my-folder.zip'. Other files shown are 'package-lock.json' and 'package.json'. The file 'zip-folder.js' is selected.
- Editor Panel:** Shows the content of 'zip-folder.js' with the following code:

```
1 const fs = require('fs');  
2 const archiver = require('archiver');  
3 const path = require('path');
```
- Terminal Panel:** Displays the output of running the command 'node zip-folder.js'. The output shows a successful archive creation for 'C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p4\my-folder.zip' with a total size of 22 bytes.
- Terminal Output:**

```
PS C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p4> node zip-folder.js  
✓ Archive created successfully: C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p4\my-folder.zip  
📦 Total bytes: 22  
PS C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p4>
```
- Status Bar:** At the bottom, it indicates the current line and column (Ln 1, Col 1), the number of errors and warnings (0 errors, 0 warnings), the encoding (UTF-8), the line ending (CRLF), and the language (JavaScript).

Question 5:

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Displays the file structure of a project named 'p5'. It includes folders 'extracted-folder', 'my-folder', and 'node_modules', and files 'extract-zip.js', 'my-folder.zip', 'package-lock.json', and 'package.json'.
- Editor:** Shows the content of 'extract-zip.js' with the following code:

```
1 const fs = require('fs');
2 const unzipper = require('unzipper');
3 const path = require('path');
4
5 function extractZip(zipPath, extractTo) {
6     fs.createReadStream(zipPath)
7       .pipe(unzipper.Extract({ path: extractTo }))
8       .on('close', () => {
9         console.log(`Extraction complete to folder: ${extractTo}`);
10      })
11       .on('error', (err) => {
12         console.error('Extraction error:', err);
13      });
14 }
15
16 const zipFile = path.join(__dirname, 'my-folder.zip');
17 const outputFolder = path.join(__dirname, 'extracted-folder');
```
- TERMINAL:** Shows the command prompt output for running the script:

```
PS C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p5> node extract-zip.js
Extraction complete to folder: C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p5\extracted-folder
PS C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p5>
```

The status bar at the bottom indicates the current file is 'extract-zip.js' at line 1, column 1, using UTF-8 encoding with CRLF line endings.

Question 6:

The screenshot shows the Visual Studio Code interface with a dark theme. The Explorer sidebar on the left shows a project named 'p6' containing a file 'fs-exmp.js'. The main editor area displays the content of 'fs-exmp.js' with line numbers 1 through 18. The code uses ES6 features like 'const', 'require', 'promisify', and 'async/await' to delete a file. The bottom panel shows the 'TERMINAL' tab with a PowerShell session where the command 'node fs-exmp.js' has been executed, resulting in the message './test.txt was deleted successfully.'.

```
JS fs-exmp.js X
JS fs-exmp.js > ...
1  const fs = require('fs');
2  const util = require('util');
3
4  const unlinkAsync = util.promisify(fs.unlink);
5
6
7  const filePath = './test.txt';
8
9  async function deleteFile() {
10   try {
11     await unlinkAsync(filePath);
12     console.log(`${filePath} was deleted successfully.`);
13   } catch (err) {
14     console.error(`Error deleting file: ${err.message}`);
15   }
16 }
17
18 deleteFile();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + - [] [X] [] [X]

- PS C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p6> node fs-exmp.js
./test.txt was deleted successfully.
- PS C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p6>

> OUTLINE
> TIMELINE

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript

Question 7:

The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left shows a project named 'P7' with a 'node_modules' directory and two files: 'package-lock.json' and 'package.json'. The main editor area displays the file 'async.js' with the following code:

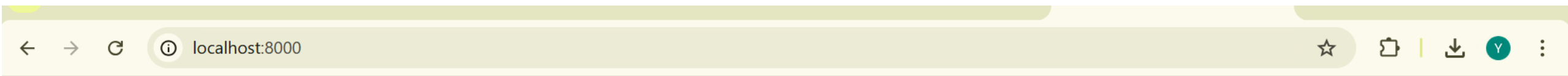
```
1 const fetch = require('node-fetch');
2
3 async function fetchGoogleHomepage() {
4   try {
5     const response = await fetch('https://www.google.com');
6     const body = await response.text();
7     console.log(body);
8   } catch (error) {
9     console.error('Error fetching Google homepage:', error);
10  }
11 }
12
13 fetchGoogleHomepage();
14
```

The bottom panel shows the TERMINAL output, which displays the command 'node async.js' and the resulting HTML content of the Google homepage:

```
PS C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p7> node async.js
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN"><head><meta content="text/html; charset=UTF-8" http-equiv="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="_6c4oJJ-tiqFJTo2uk34uw">(function(){var _g={kEI: 'AEB9aPWcBJSTseMPk4LuqAk',kEXPI: '0,202791,63,2,3497418,675,435,523563,15098,48791,30022,16105,344796,290044,11106,5219175,11400,472,32768462,4043709,25366949,14110,8944,3003,10971,42256,6751,23878,9140,4599,328,6225,1116,63049,15049,8205,878,1,2,2411,4134,30379,28334,54211,353,2268,9513,7099,5870,7714,5773,8977,11027,7608,3050,2,1667,11805,3261,2990,35,1153,3,5,4,4948,171,661,9959,12107,3705,1978,3605,17771,1515,1216,1848,2,11,4570,9470,646,4222,4,4,4801,2418,3436,438,1737,1857,3,1959,1016,1,3302,161,2,82,826,2,388,29,2777,88,7,1370,600,1033,946,5,627,4859,379,2321,3,1890,3168,7,1974,519,157,259,1711,103,4,1,321,840,2,980,2289,988,670,34,60,2246,7,3741,968,3174,1447,3,2,2,2,856,475,4027,283,606,3,112,340,1443,1,4,352,848,2716,691,3196,324,1041,3061,1280,54,295,242,2,1,1379,32,42,20,623,389,226,632,285,36,615,306,822,1844,802,713,979,1017,5,113,2,535,344,402,66,5,1,367,139,175,43,85,11,31,8,178,82,327,107,173,21,503,210,7,5,393,52,981,282,6,561,290,2,170,2,272,717,320,3,4,2,585,2189,142,136,114,53,60,1026,228,254,9,1,116,1695,884,505,283,4,1179,271,2,691,119,295,618,171,8,113,104,17,34,96,448,6,832,12,25,495,493,2,639,3,638,
```

The status bar at the bottom indicates the current line and column (Ln 1, Col 1), the number of spaces (Spaces: 2), the encoding (UTF-8), the line ending (CRLF), and the language (JavaScript).

Question 8: Server Page

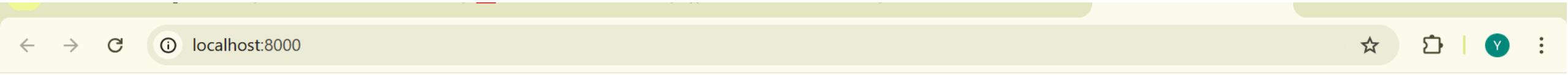


Server is running!

Question 8: Page 1

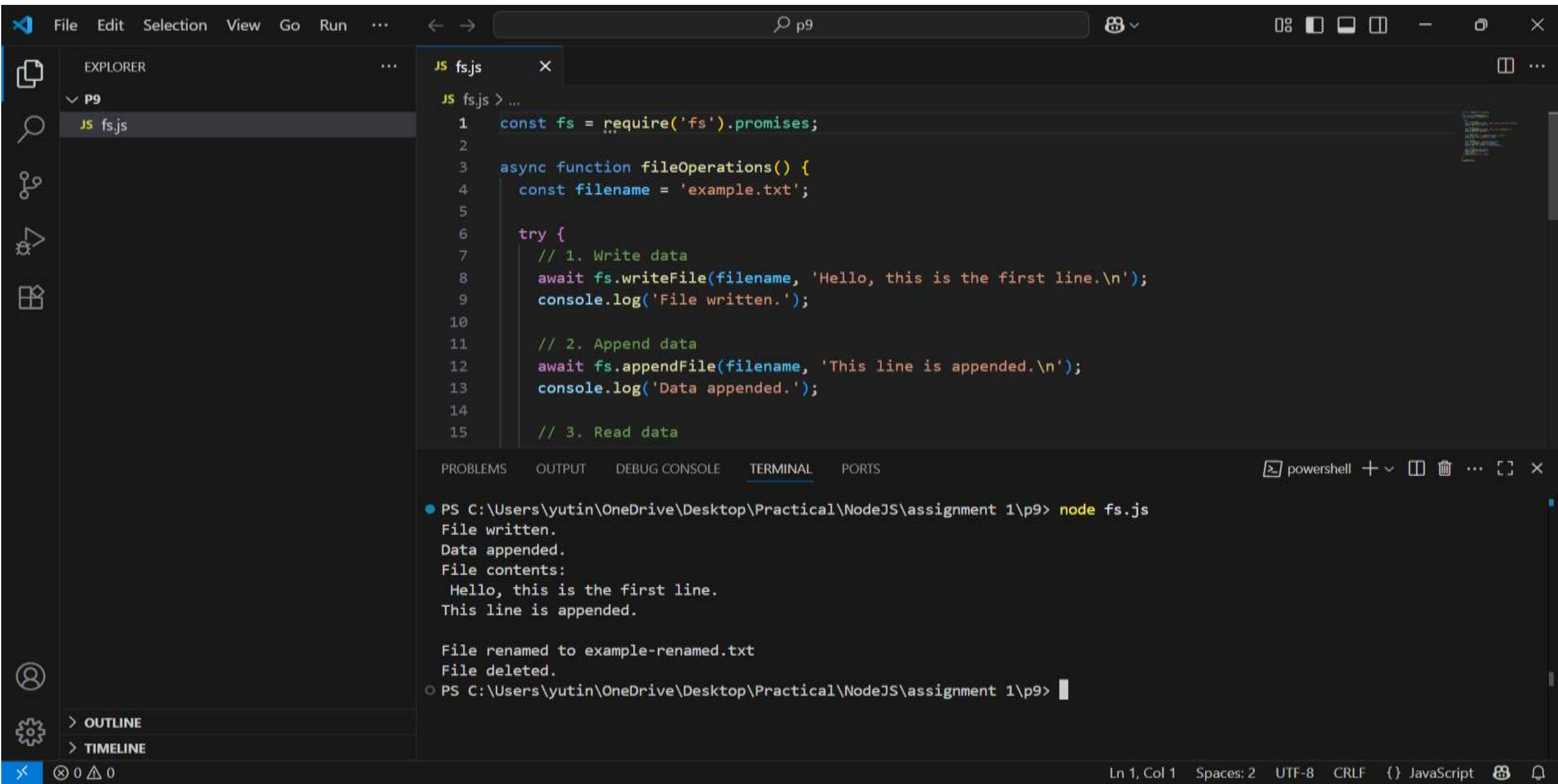
Question 8:Page 2

Question 8:Testing Page



Testing page

Question 9:



The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a file named `fs.js` under a folder named `p9`. The main editor area displays the content of `fs.js`, which is a JavaScript file using the `fs.promises` API to perform file operations. The code includes comments for each step: writing data, appending data, and reading data. The bottom panel shows the TERMINAL view with the output of running `node fs.js`. The output shows the file being written, data being appended, and the final contents of the file being displayed. The status bar at the bottom indicates the current line and column (Ln 1, Col 1), the number of spaces (2), the encoding (UTF-8), the line ending (CRLF), and the language (JavaScript).

```
1  const fs = require('fs').promises;
2
3  async function fileOperations() {
4      const filename = 'example.txt';
5
6      try {
7          // 1. Write data
8          await fs.writeFile(filename, 'Hello, this is the first line.\n');
9          console.log('File written.');

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS



PS C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p9> node fs.js



File written.  
Data appended.  
File contents:  
Hello, this is the first line.  
This line is appended.



File renamed to example-renamed.txt  
File deleted.



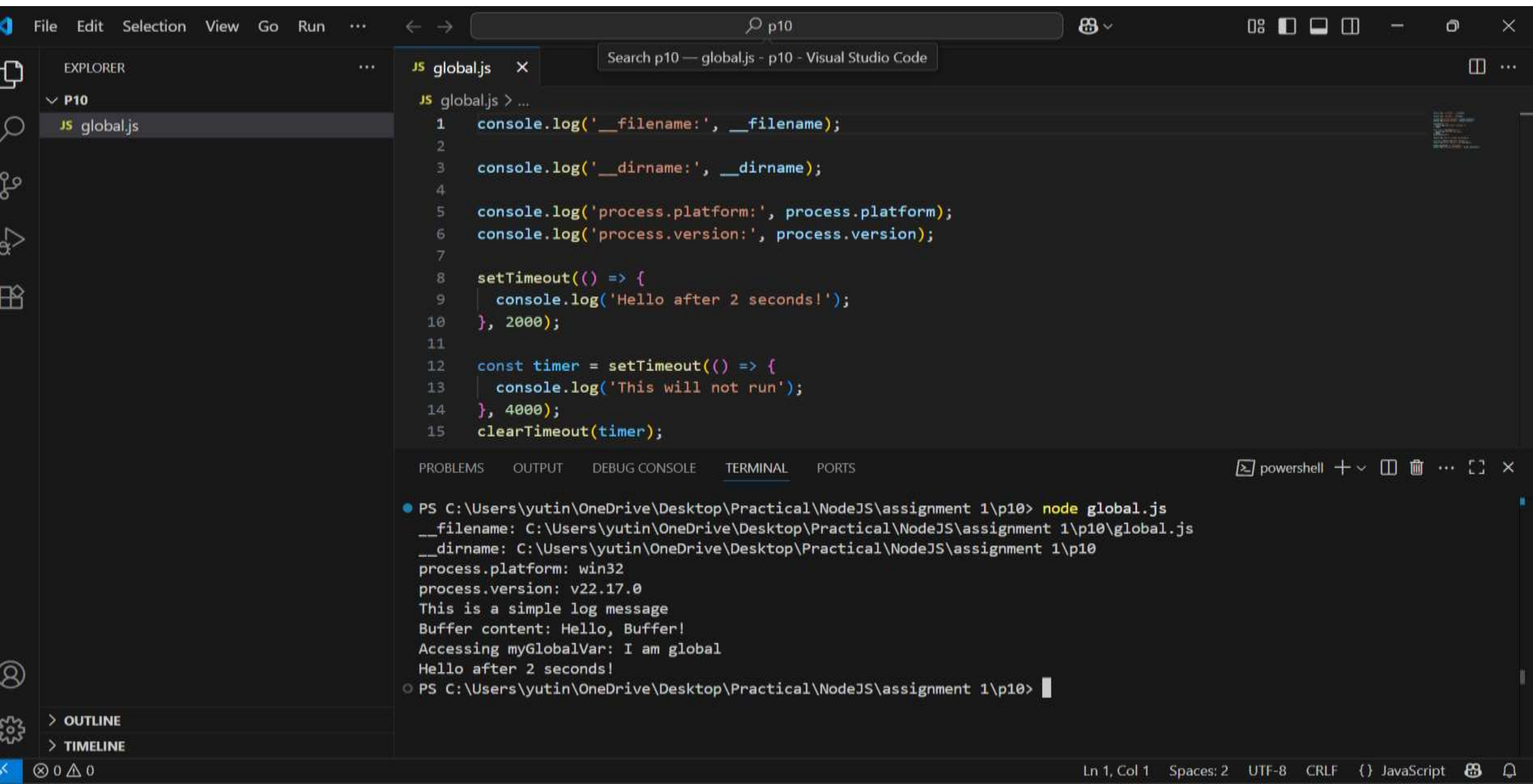
PS C:\Users\yutin\OneDrive\Desktop\Practical\NodeJS\assignment 1\p9>




Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} JavaScript



```



Question 10:





Question 11:


 Pro Teams Pricing Documentation





 Profile

 Packages


 Account

 Billing Info


 Access Tokens

Organizations

None

 Search packages

Search



1 packages

time-mark-now

Returns the current timestamp in YYYY-MM-DD HH:MM:SS format with 'Current Date' and 'Current Time' labels.

yutinaik29 published 1.0.0 • an hour ago