

**Name: Patel Happy Sureshbhai**

**Semester: 7                      Div: B**

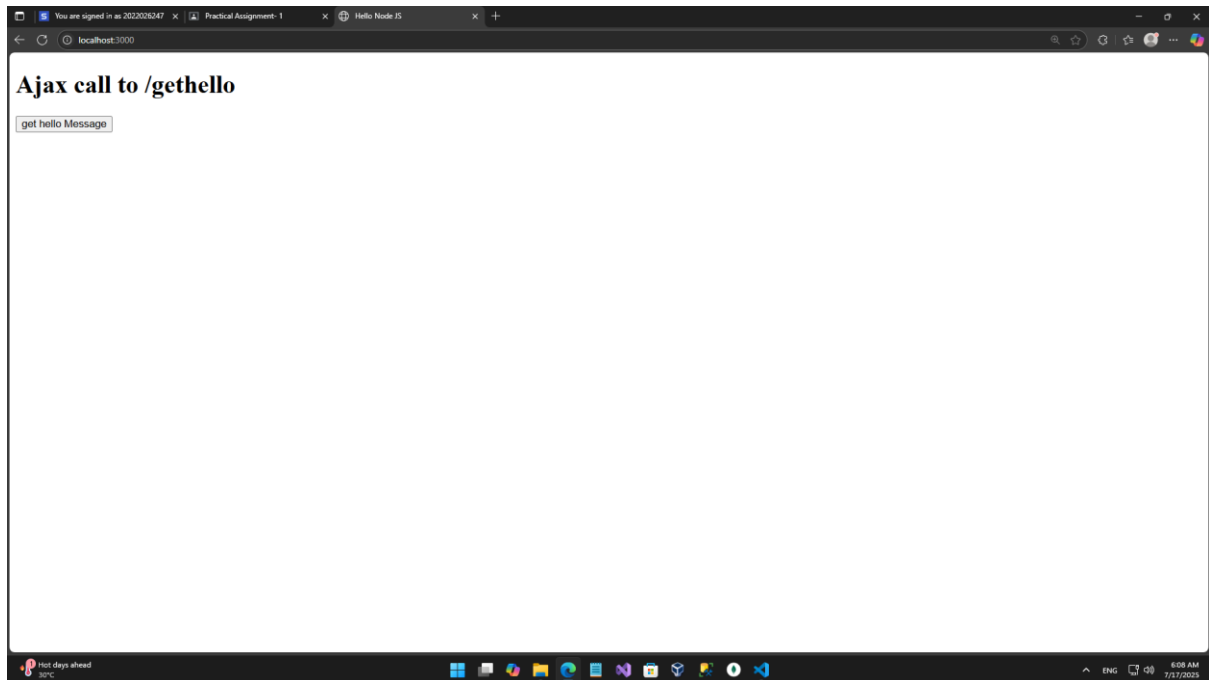
**Roll No.: 78**

**Subject: 701- Application Development Using Fullstack**

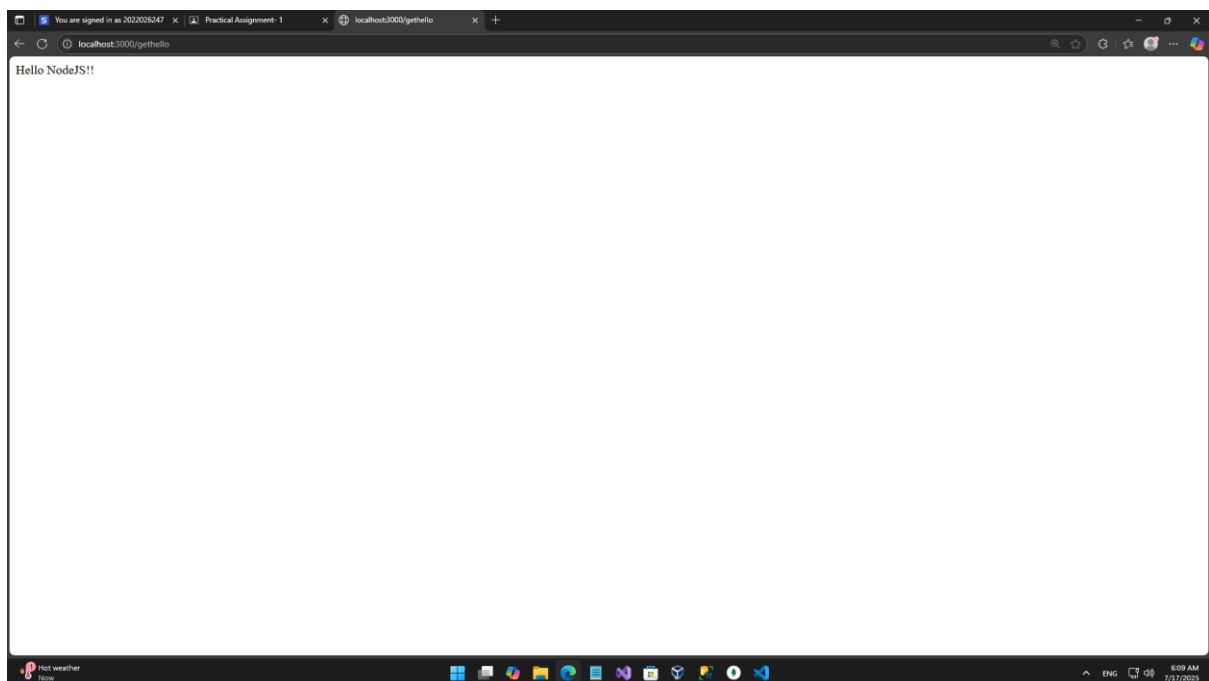
### **Practical Assignment – 1**

**Q1.**

**localhost:3000**

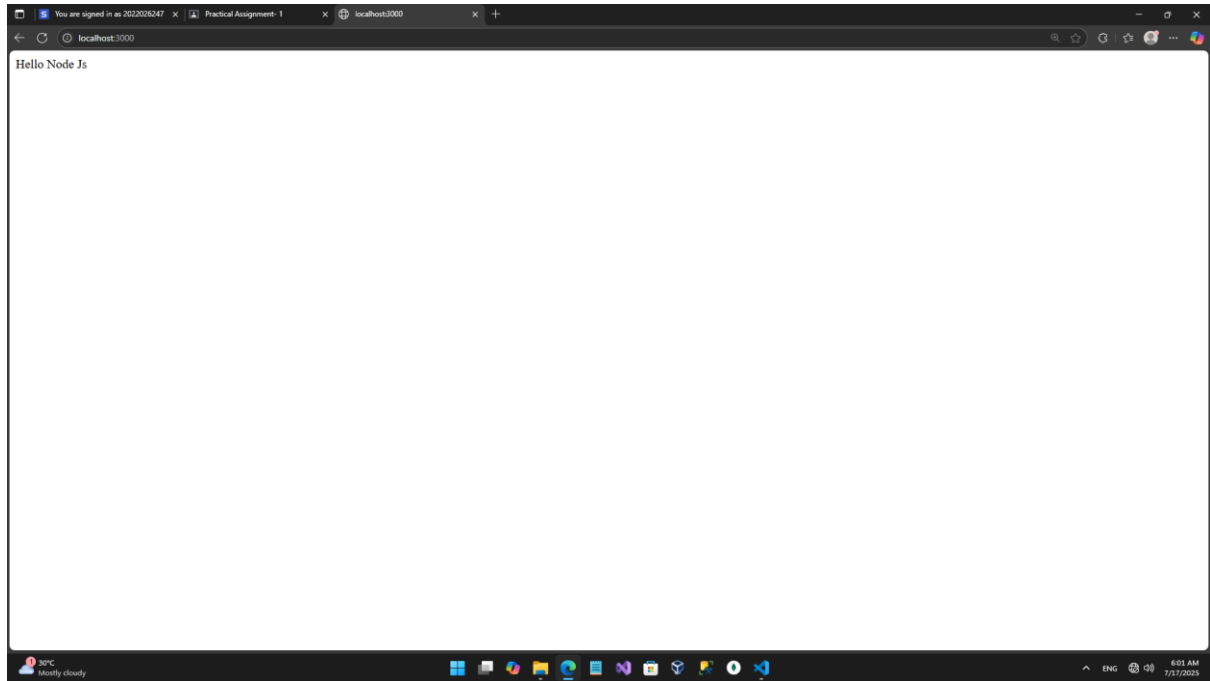


**localhost:3000/gethello**

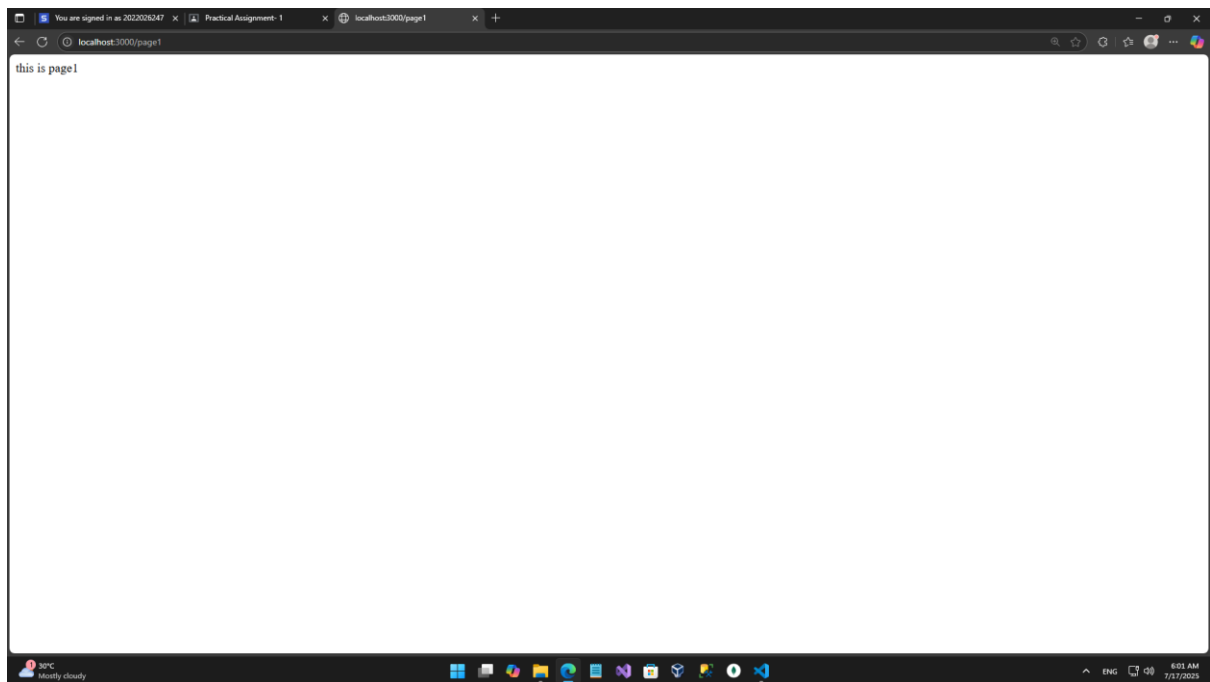


**Q2.**

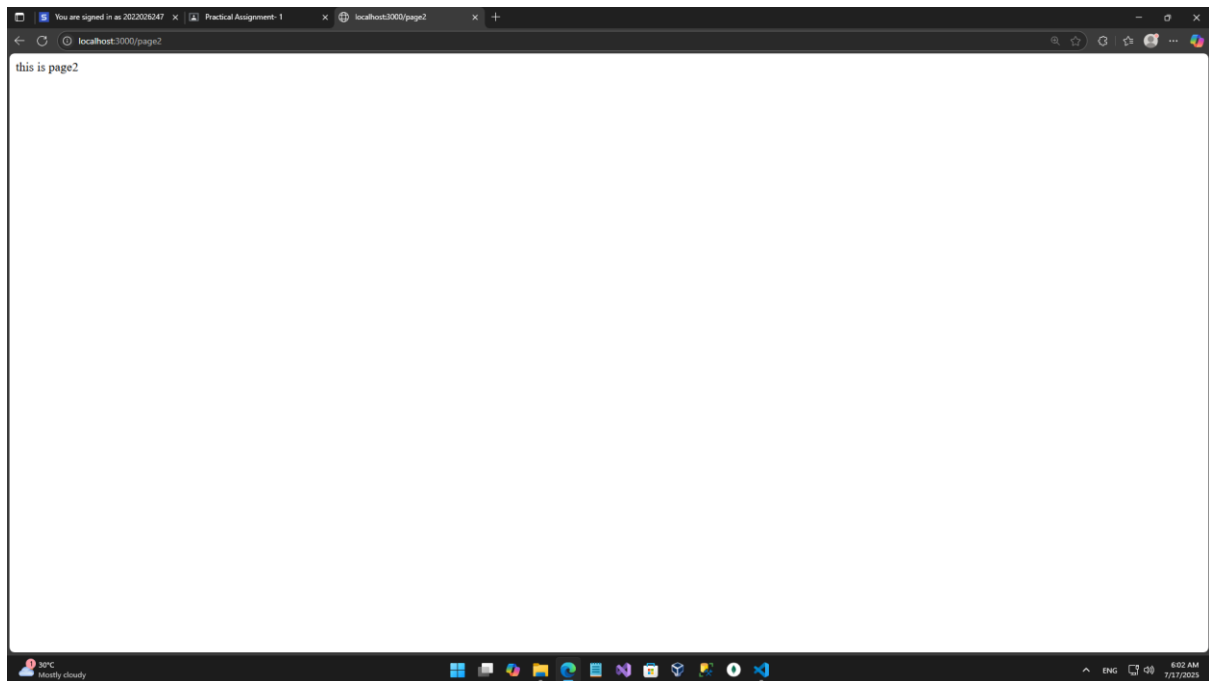
localhost:3000



localhost:3000/page1

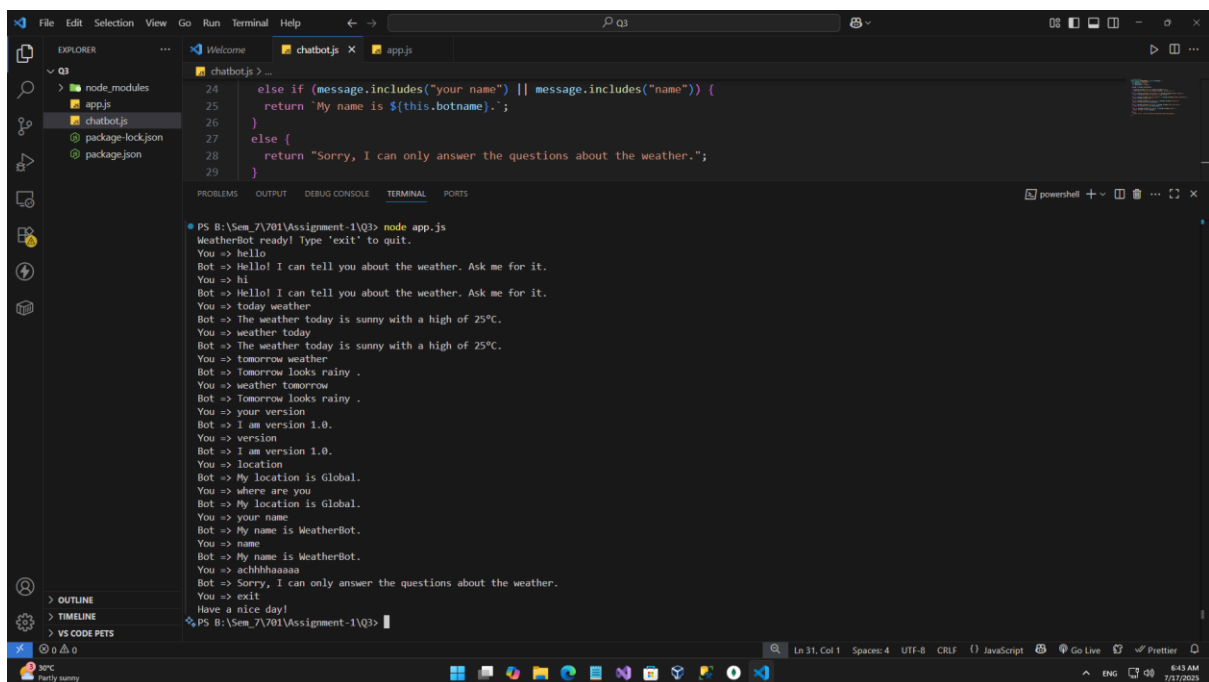


localhost:3000/page1



Q3.

chatbot.js



# Q4.

## Before zipped

```
1 const fs = require('fs');
2 const archiver = require('archiver');
3 const path = require('path');
4
5 function zipFolder(sourceFolderPath, outputZipPath) {
6   const output = fs.createWriteStream(outputZipPath);
7   const archive = archiver('zip', {
8     zlib: { level: 9 }
9   });
10
11   output.on('close', () => {
12     console.log(`${archive.pointer()} total bytes`);
13     console.log(`Folder "${sourceFolderPath}" has been zipped to "${outputZipPath}"`);
14   });
15
16   archive.on('error', (err) => {
17     throw err;
18   });
19 }
```

PS B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4> node zipFolder.js

22 total bytes

Folder "B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4\my\_folder" has been zipped to "B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4\my\_folder.zip"

PS B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4> node app.js

Extracted "B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4\my\_folder.zip" to "B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4\unzipped\_folder"

PS B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4>

## After zipped

```
1 const fs = require('fs');
2 const archiver = require('archiver');
3 const path = require('path');
4
5 function zipFolder(sourceFolderPath, outputZipPath) {
6   const output = fs.createWriteStream(outputZipPath);
7   const archive = archiver('zip', {
8     zlib: { level: 9 }
9   });
10
11   output.on('close', () => {
12     console.log(`${archive.pointer()} total bytes`);
13     console.log(`Folder "${sourceFolderPath}" has been zipped to "${outputZipPath}"`);
14   });
15
16   archive.on('error', (err) => {
17     throw err;
18   });
19 }
```

PS B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4> node zipFolder.js

22 total bytes

Folder "B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4\my\_folder" has been zipped to "B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4\my\_folder.zip"

PS B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4> node app.js

Extracted "B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4\my\_folder.zip" to "B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4\unzipped\_folder"

PS B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4> node zipFolder.js

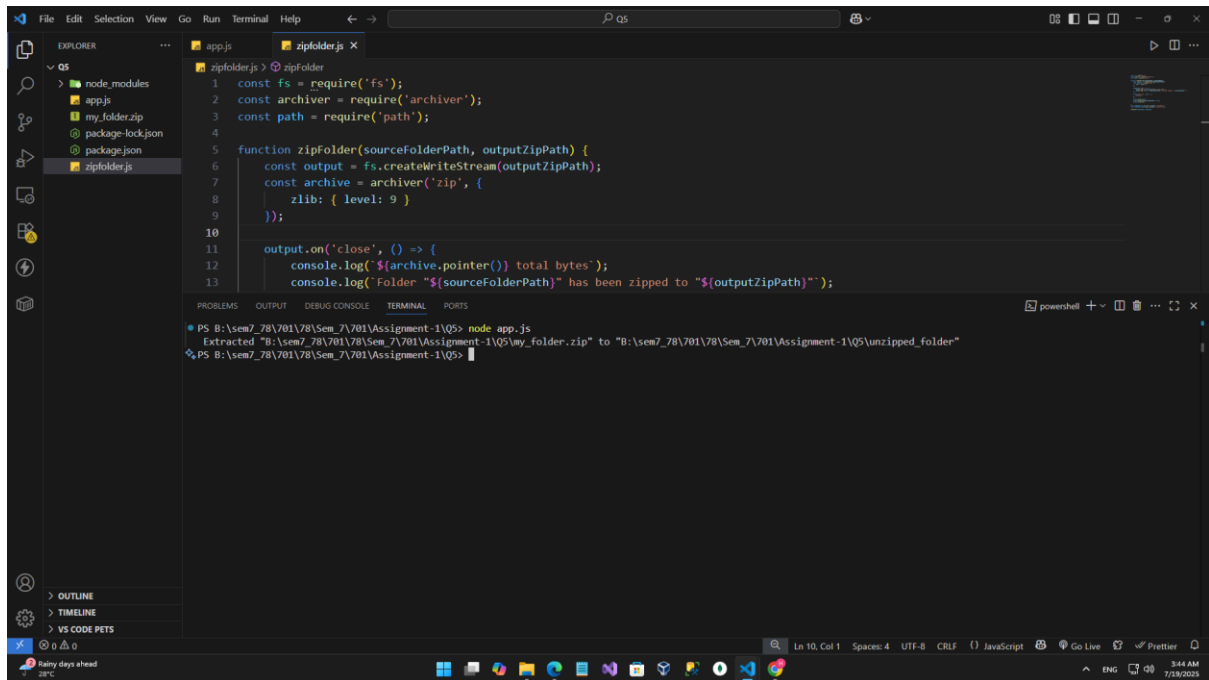
22 total bytes

Folder "B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4\my\_folder" has been zipped to "B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4\my\_folder.zip"

PS B:\sem7\_78\701\Sem\_7\701\Assignment-1\Q4>

Q5.

Unzipped folder



The screenshot shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left shows a project structure with folders 'node\_modules' and 'app.js', and files 'my\_folder.zip', 'package-lock.json', and 'zipFolder.js'. The main editor area displays the 'zipFolder.js' file with the following code:

```
1 const fs = require('fs');
2 const archiver = require('archiver');
3 const path = require('path');
4
5 function zipFolder(sourceFolderPath, outputZipPath) {
6     const output = fs.createWriteStream(outputZipPath);
7     const archive = archiver('zip', {
8         zlib: { level: 9 }
9     });
10
11     output.on('close', () => {
12         console.log(`${archive.pointer()} total bytes`);
13         console.log(`Folder "${sourceFolderPath}" has been zipped to "${outputZipPath}"`);
14     });
15 }
```

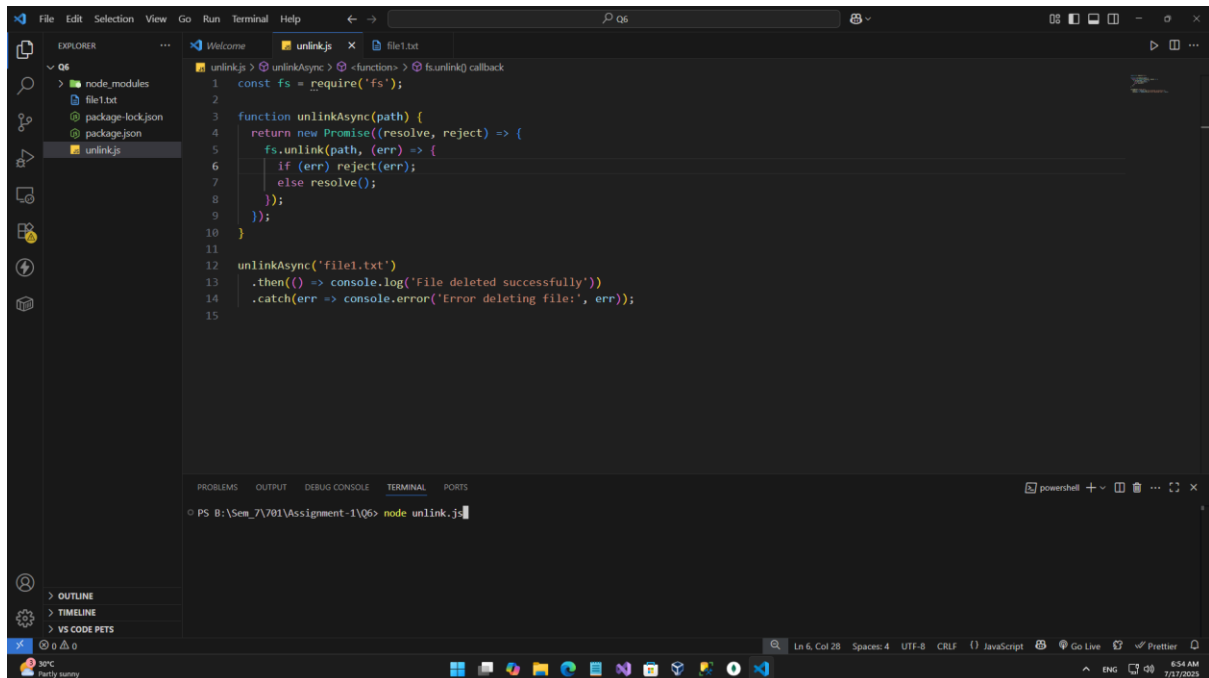
Below the editor, the TERMINAL panel is open, showing the execution of the script. The first command is `node app.js`, which results in the following output:

```
Extracted "B:\sem7_78\701\Sem_7\701\Assignment-1\Q5\my_folder.zip" to "B:\sem7_78\701\Sem_7\701\Assignment-1\Q5\unzipped_folder"
```

The second command in the terminal is `PS B:\sem7_78\701\Sem_7\701\Assignment-1\Q5>`, indicating the prompt is ready for further input.

Q6.

File before unlink

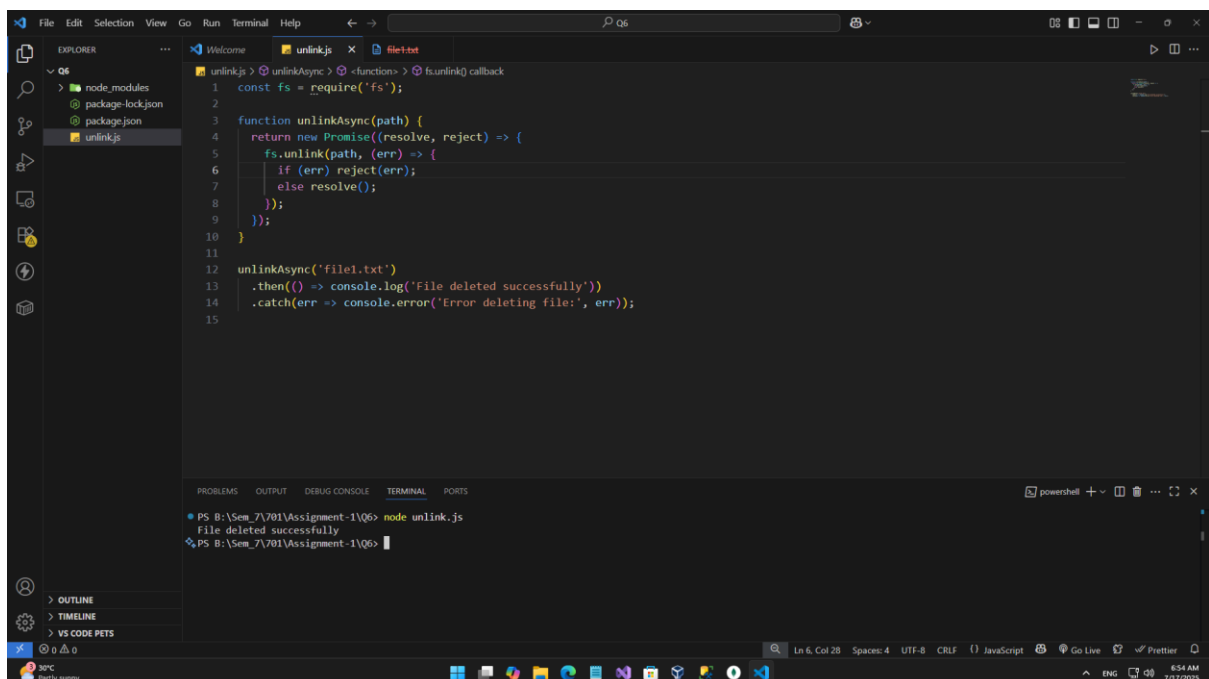


The screenshot shows the Visual Studio Code editor with a project named 'Q6'. The Explorer sidebar on the left shows a file tree with 'node\_modules', 'file1.txt', 'package-lock.json', and 'package.json'. The main editor area displays the 'unlink.js' file with the following JavaScript code:

```
1 const fs = require('fs');
2
3 function unlinkAsync(path) {
4   return new Promise((resolve, reject) => {
5     fs.unlink(path, (err) => {
6       if (err) reject(err);
7       else resolve();
8     });
9   });
10 }
11
12 unlinkAsync('file1.txt')
13   .then(() => console.log('File deleted successfully'))
14   .catch(err => console.error('Error deleting file:', err));
15
```

Below the editor, the TERMINAL panel shows a PowerShell prompt at 'PS B:\Sem\_7\701\Assignment-1\Q6>' with the command 'node unlink.js' entered.

File after unlink

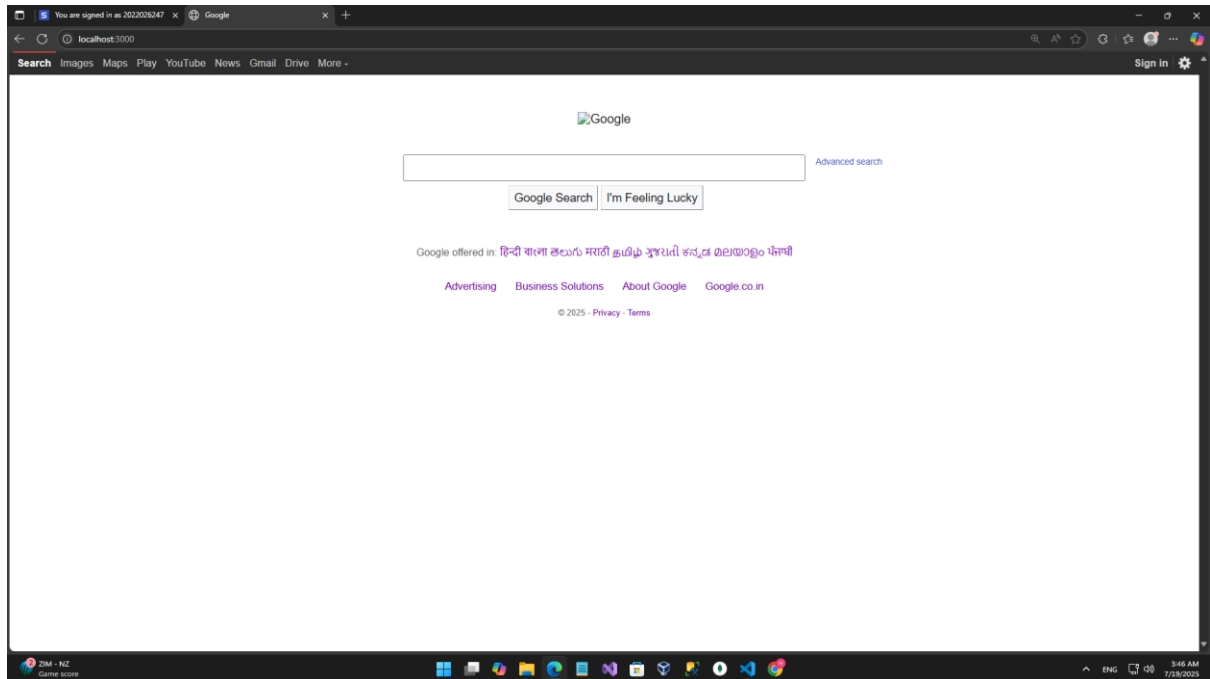


The screenshot shows the same Visual Studio Code editor after running the script. The Explorer sidebar now shows 'file1.txt' as deleted. The main editor area displays the same 'unlink.js' code as in the previous screenshot. The TERMINAL panel now shows the output of the command:

```
PS B:\Sem_7\701\Assignment-1\Q6> node unlink.js
File deleted successfully
PS B:\Sem_7\701\Assignment-1\Q6>
```

Q7.

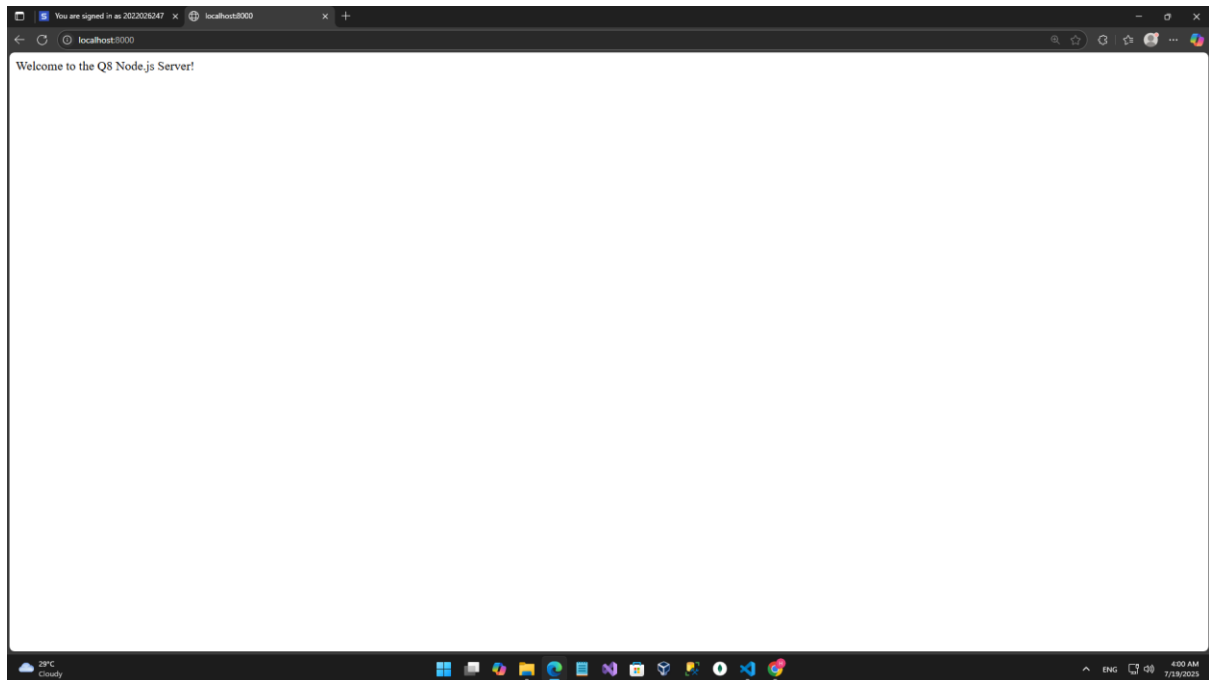
Fetching google page



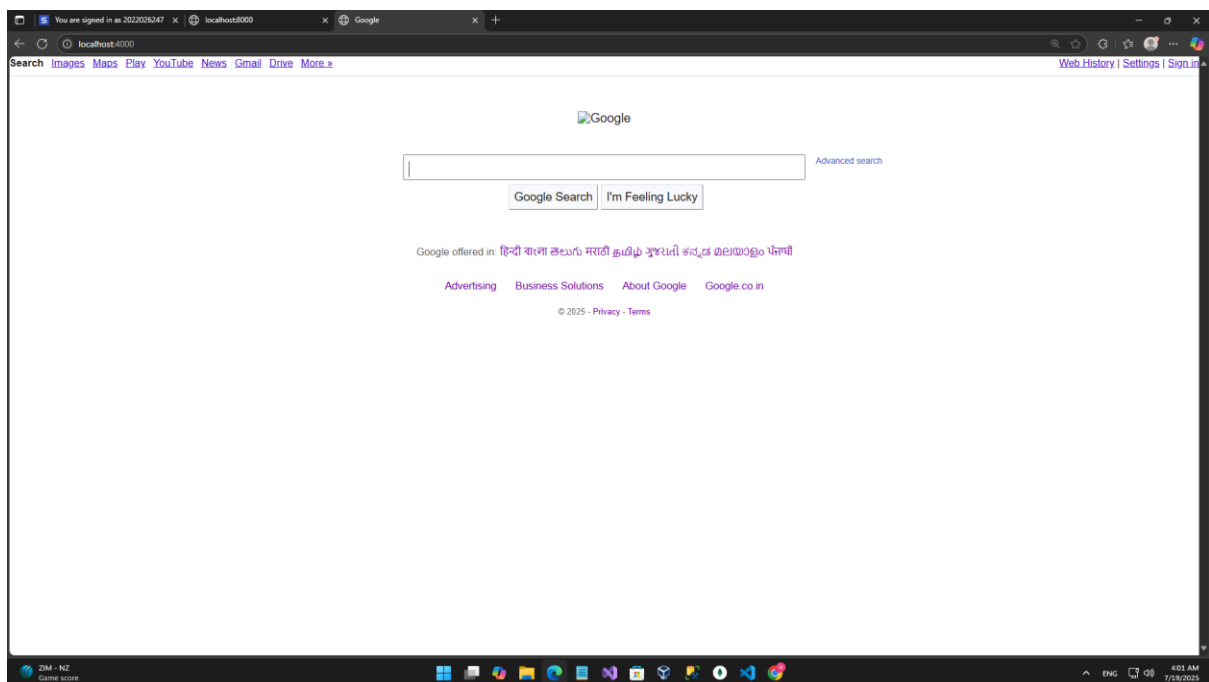


Q8.

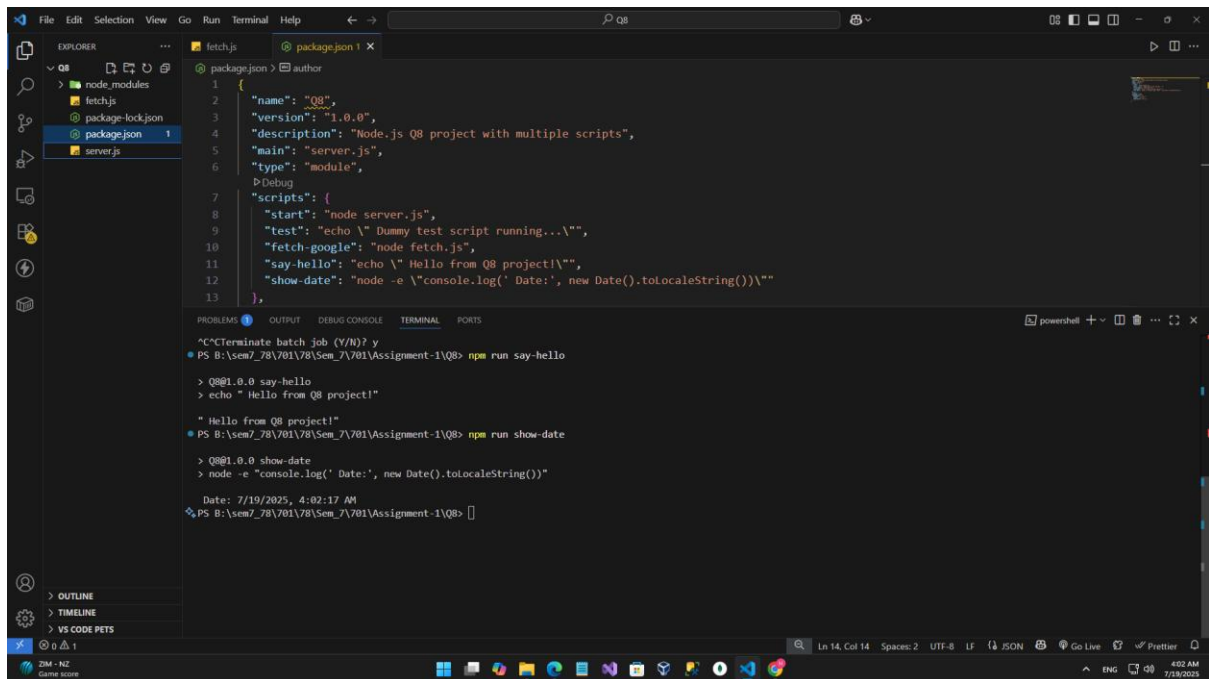
Server script - npm start



npm run fetch-google



## other test scripts



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'Q8' with files 'node\_modules', 'fetch.js', 'package-lock.json', 'package.json', and 'server.js'. The 'package.json' file is open in the editor, showing the following content:

```
1 {
2   "name": "Q8",
3   "version": "1.0.0",
4   "description": "Node.js Q8 project with multiple scripts",
5   "main": "server.js",
6   "type": "module",
7   "scripts": {
8     "start": "node server.js",
9     "test": "echo \\\" Dummy test script running...\\\"",
10    "fetch-google": "node fetch.js",
11    "say-hello": "echo \\\" Hello from Q8 project!\\\"",
12    "show-date": "node -e \\\"console.log(' Date:', new Date().toLocaleString())\\\"",
13  },
14 }
```

The terminal window at the bottom shows the following commands and output:

```
^C^CTerminate batch job (Y/N)? y
PS B:\sem7_78\701\Sem_7\701\Assignment-1\Q8> npm run say-hello

> Q8@1.0.0 say-hello
> echo " Hello from Q8 project!"

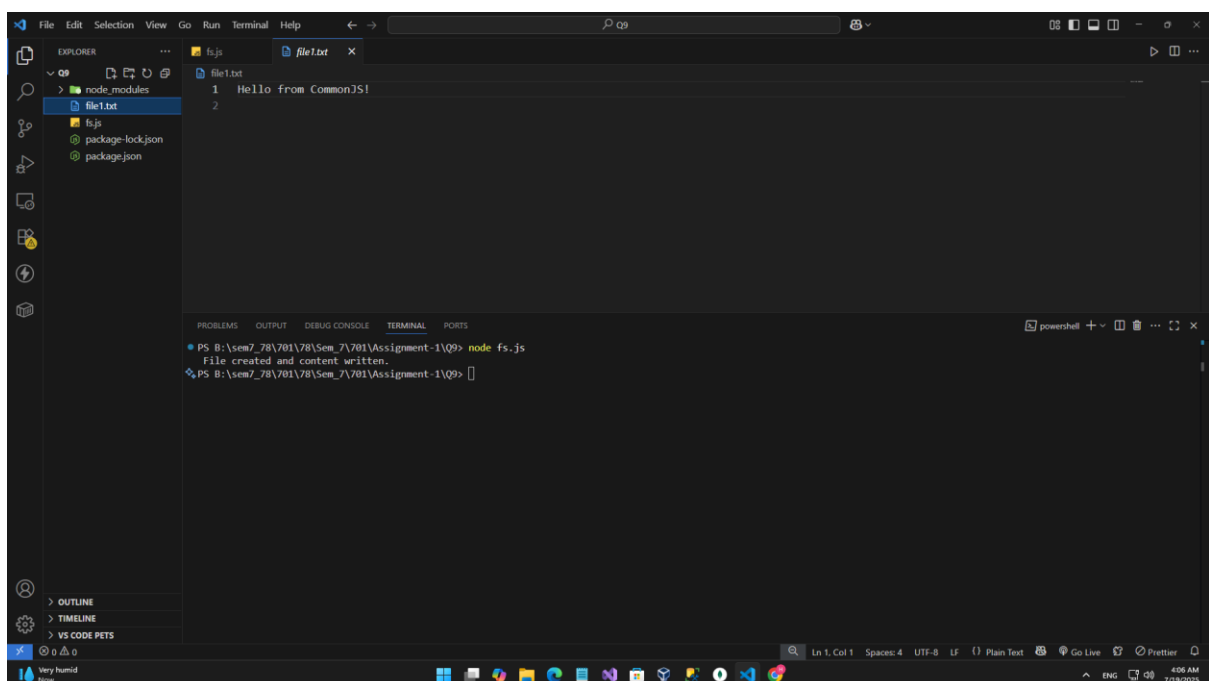
" Hello from Q8 project!"
PS B:\sem7_78\701\Sem_7\701\Assignment-1\Q8> npm run show-date

> Q8@1.0.0 show-date
> node -e "console.log(' Date:', new Date().toLocaleString())"

Date: 7/19/2025, 4:02:17 AM
PS B:\sem7_78\701\Sem_7\701\Assignment-1\Q8>
```

## Q9.

### Create file



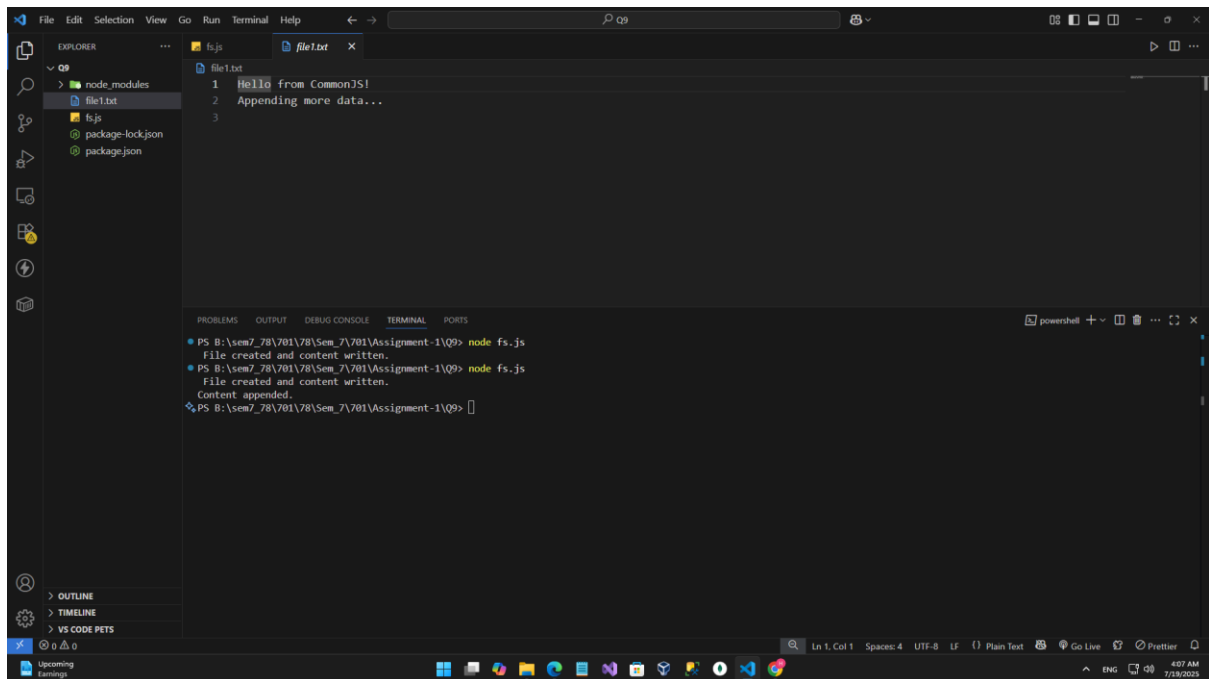
The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows a project named 'Q9' with files 'fs.js', 'file1.txt', 'package-lock.json', and 'package.json'. The 'file1.txt' file is open in the editor, showing the following content:

```
1 Hello from CommonJS!
2
```

The terminal window at the bottom shows the following commands and output:

```
PS B:\sem7_78\701\Sem_7\701\Assignment-1\Q9> node fs.js
File created and content written.
PS B:\sem7_78\701\Sem_7\701\Assignment-1\Q9>
```

## Append file

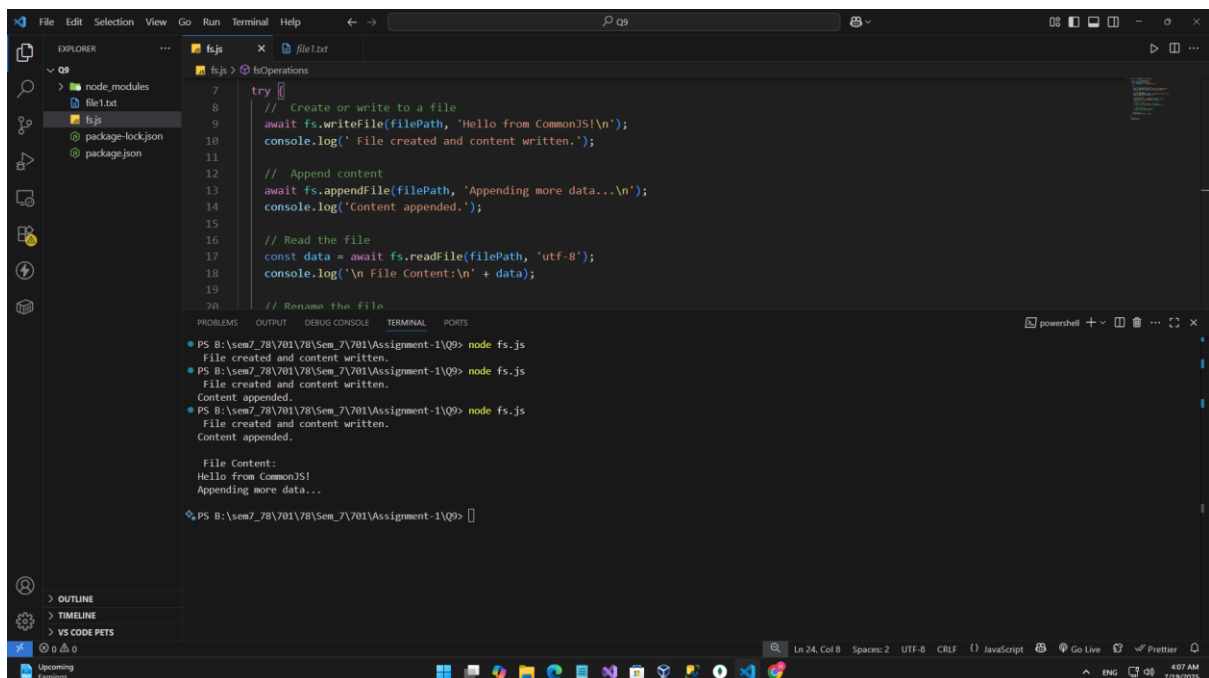


The screenshot shows the Visual Studio Code editor with a file named `file1.txt` open. The file contains two lines of text:

```
1 Hello from CommonJS!  
2 Appending more data...
```

The Explorer sidebar on the left shows the file structure, including `node_modules`, `file1.txt`, `fs.js`, `package-lock.json`, and `package.json`. The Terminal window at the bottom shows the execution of `node fs.js` in a PowerShell shell, with output indicating that the file was created and content was written and appended.

## Read file

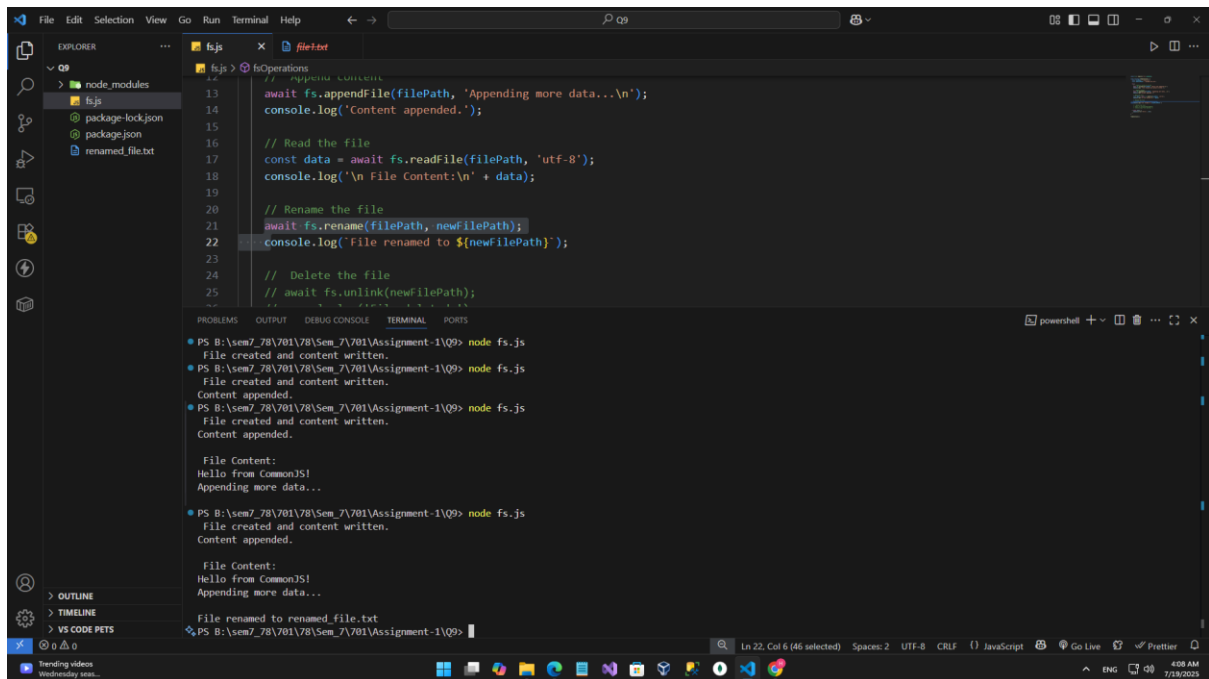


The screenshot shows the Visual Studio Code editor with a file named `fsOperations.js` open. The file contains the following JavaScript code:

```
7 try {  
8   // Create or write to a file  
9   await fs.writeFile(filePath, 'Hello from CommonJS!\n');  
10  console.log(' File created and content written.');
```

The Explorer sidebar on the left shows the file structure, including `node_modules`, `file1.txt`, `fs.js`, `package-lock.json`, and `package.json`. The Terminal window at the bottom shows the execution of `node fs.js` in a PowerShell shell, with output indicating that the file was created and content was written and appended. The output also shows the file content being read and logged.

## Rename file



This screenshot shows the Visual Studio Code editor with a file named `file.txt` open. The file contains JavaScript code for file operations. The Explorer sidebar on the left shows the project structure with `node_modules`, `fs.js`, `package-lock.json`, `package.json`, and `renamed_file.txt`. The code in `file.txt` includes functions for appending content, reading the file, renaming it, and deleting it. The terminal at the bottom shows the output of running `node fs.js`, which includes messages about file creation, content appending, and the successful renaming of the file to `renamed_file.txt`.

```
12 // Append Content
13 await fs.appendFile(filePath, 'Appending more data...\n');
14 console.log('Content appended.');
```

```
16 // Read the file
17 const data = await fs.readFile(filePath, 'utf-8');
18 console.log('\n File Content:\n' + data);
19
20 // Rename the file
21 await fs.rename(filePath, newFilePath);
22 console.log('File renamed to ${newFilePath}');
```

```
24 // Delete the file
25 // await fs.unlink(newFilePath);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

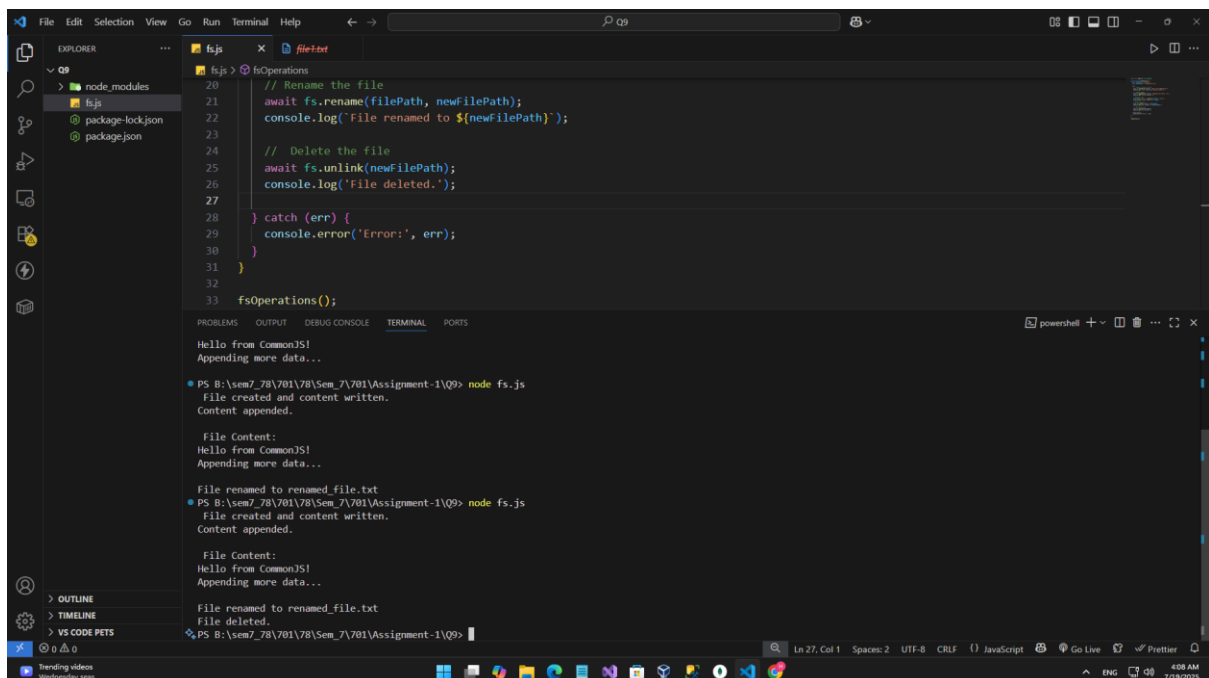
PS B:\sem7\_78\701\78\Sem\_7\701\Assignment-1\Q9> node fs.js  
File created and content written.  
Content appended.  
File Content:  
Hello from CommonJS!  
Appending more data...

PS B:\sem7\_78\701\78\Sem\_7\701\Assignment-1\Q9> node fs.js  
File created and content written.  
Content appended.  
File Content:  
Hello from CommonJS!  
Appending more data...

File renamed to renamed\_file.txt  
PS B:\sem7\_78\701\78\Sem\_7\701\Assignment-1\Q9> node fs.js

Ln 22, Col 6 (46 selected) Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier 4:08 AM 7/19/2023

## Deleting the file



This screenshot shows the Visual Studio Code editor with the same `fs.js` file. The code now includes a `catch` block for errors and a call to `fsOperations()` at the end. The Explorer sidebar shows the project structure with `node_modules`, `fs.js`, `package-lock.json`, and `package.json`. The terminal at the bottom shows the output of running `node fs.js`, which includes messages about file creation, content appending, renaming, and the successful deletion of the file.

```
20 // Rename the file
21 await fs.rename(filePath, newFilePath);
22 console.log('File renamed to ${newFilePath}');
```

```
24 // Delete the file
25 await fs.unlink(newFilePath);
26 console.log('File deleted.');
```

```
27
28 } catch (err) {
29   console.error('Error:', err);
30 }
31
32
33 fsOperations();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Hello from CommonJS!  
Appending more data...

PS B:\sem7\_78\701\78\Sem\_7\701\Assignment-1\Q9> node fs.js  
File created and content written.  
Content appended.  
File Content:  
Hello from CommonJS!  
Appending more data...

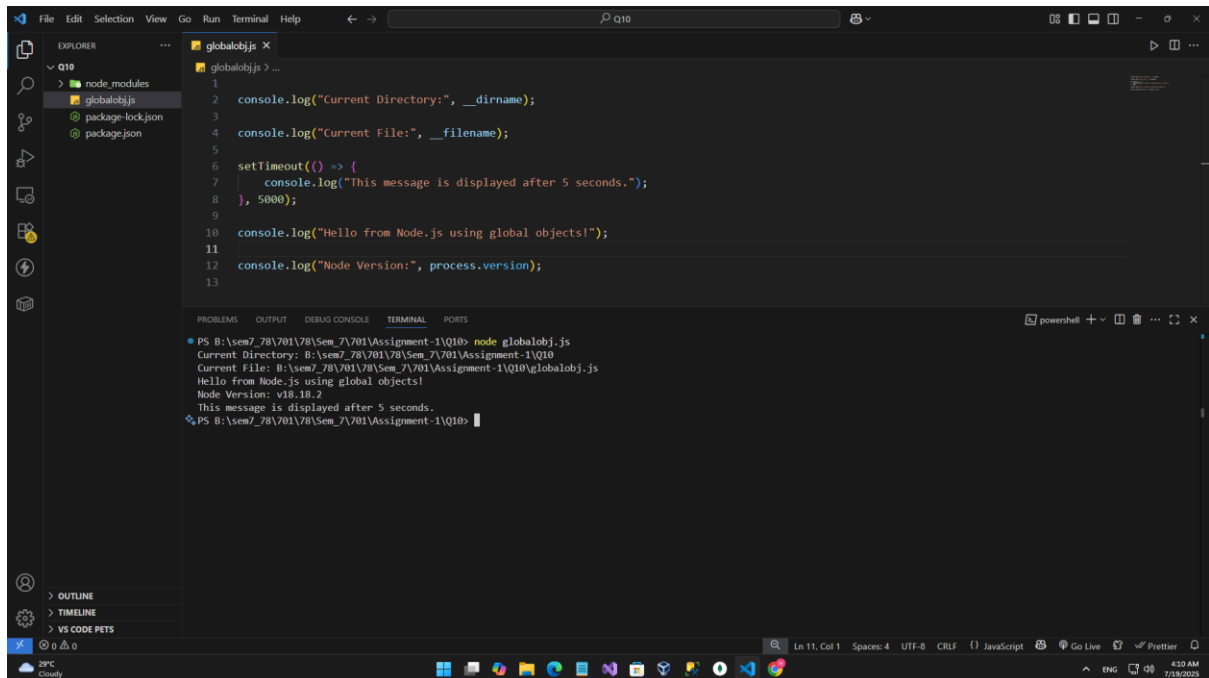
File renamed to renamed\_file.txt  
PS B:\sem7\_78\701\78\Sem\_7\701\Assignment-1\Q9> node fs.js  
File created and content written.  
Content appended.  
File Content:  
Hello from CommonJS!  
Appending more data...

File renamed to renamed\_file.txt  
File deleted.  
PS B:\sem7\_78\701\78\Sem\_7\701\Assignment-1\Q9> node fs.js

Ln 27, Col 1 Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier 4:08 AM 7/19/2023

## Q10.

### Global objects



The screenshot shows a Visual Studio Code editor with a file named `globalobj.js` open. The file contains the following JavaScript code:

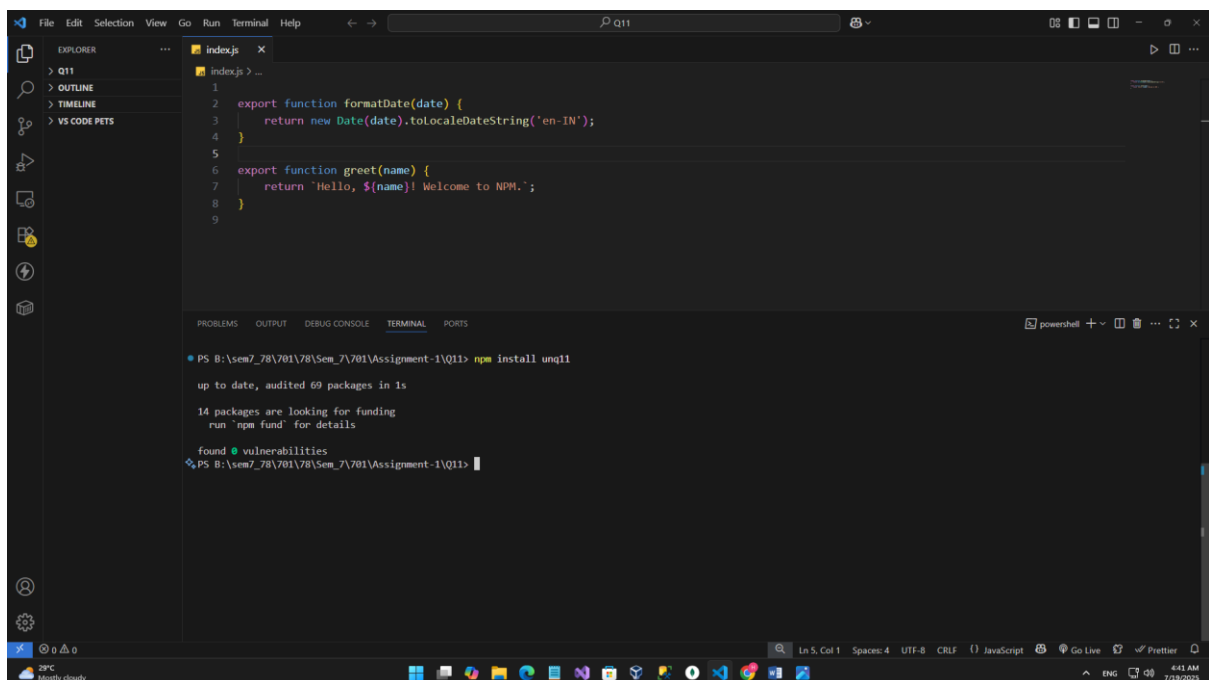
```
1 console.log("Current Directory:", __dirname);
2
3 console.log("Current File:", __filename);
4
5
6 setTimeout(() => {
7   console.log("This message is displayed after 5 seconds.");
8 }, 5000);
9
10 console.log("Hello from Node.js using global objects!");
11
12 console.log("Node Version:", process.version);
13
```

The terminal output shows the execution of the script:

```
PS B:\sem7_78\701\Sem_7\701\Assignment-1\Q10> node globalobj.js
Current Directory: B:\sem7_78\701\Sem_7\701\Assignment-1\Q10
Current File: B:\sem7_78\701\Sem_7\701\Assignment-1\Q10\globalobj.js
Hello from Node.js using global objects!
Node Version: v18.18.2
This message is displayed after 5 seconds.
PS B:\sem7_78\701\Sem_7\701\Assignment-1\Q10>
```

## Q11.

### npm package



The screenshot shows a Visual Studio Code editor with a file named `index.js` open. The file contains the following JavaScript code:

```
1
2 export function formatDate(date) {
3   return new Date(date).toLocaleDateString('en-IN');
4 }
5
6 export function greet(name) {
7   return `Hello, ${name}! Welcome to NPM.`;
8 }
9
```

The terminal output shows the execution of the script:

```
PS B:\sem7_78\701\Sem_7\701\Assignment-1\Q11> npm install unql1
up to date, audited 69 packages in 1s
14 packages are looking for funding
run 'npm fund' for details
found 0 vulnerabilities
PS B:\sem7_78\701\Sem_7\701\Assignment-1\Q11>
```