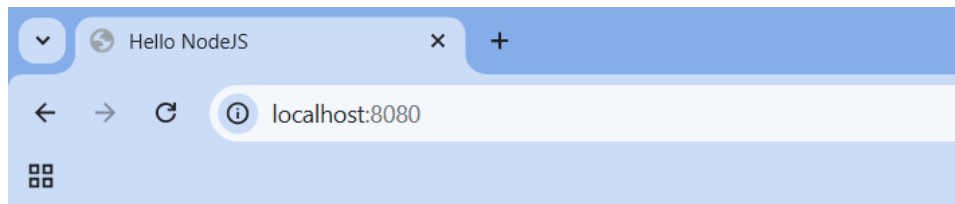


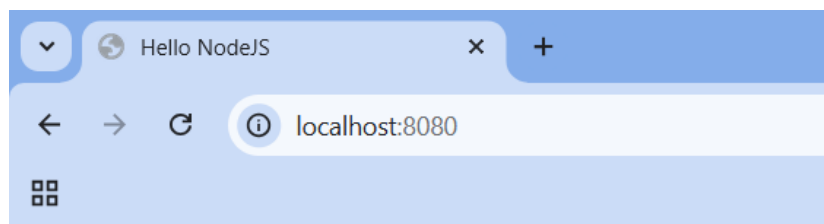
1. Develop nodejs application with following requirements:

- 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.)

Output:-**Before the Message:-**

Welcome to NodeJS App

Get Hello Message

After the Message:-

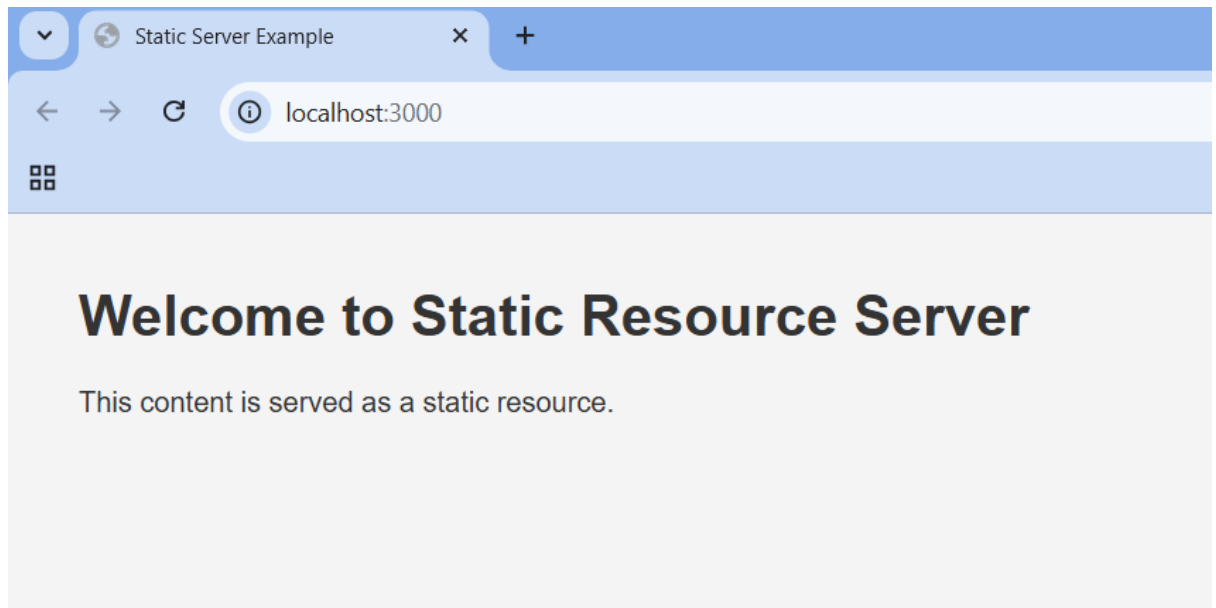
Welcome to NodeJS App

Get Hello Message

Hello NodeJS!!

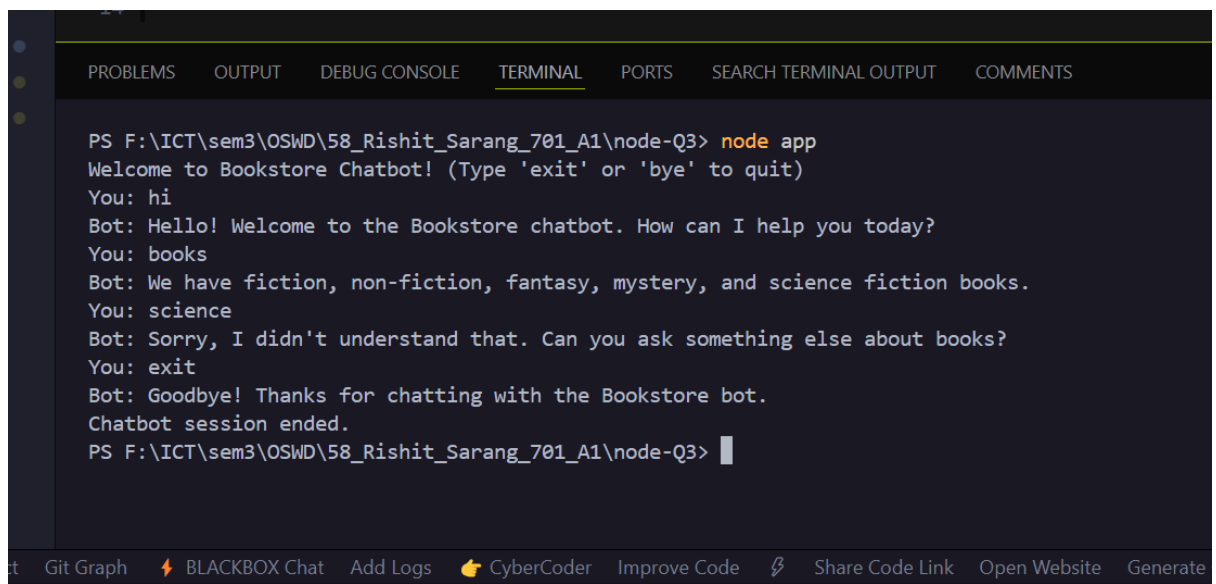
2. Develop a web server which serves static resources.

Output:-



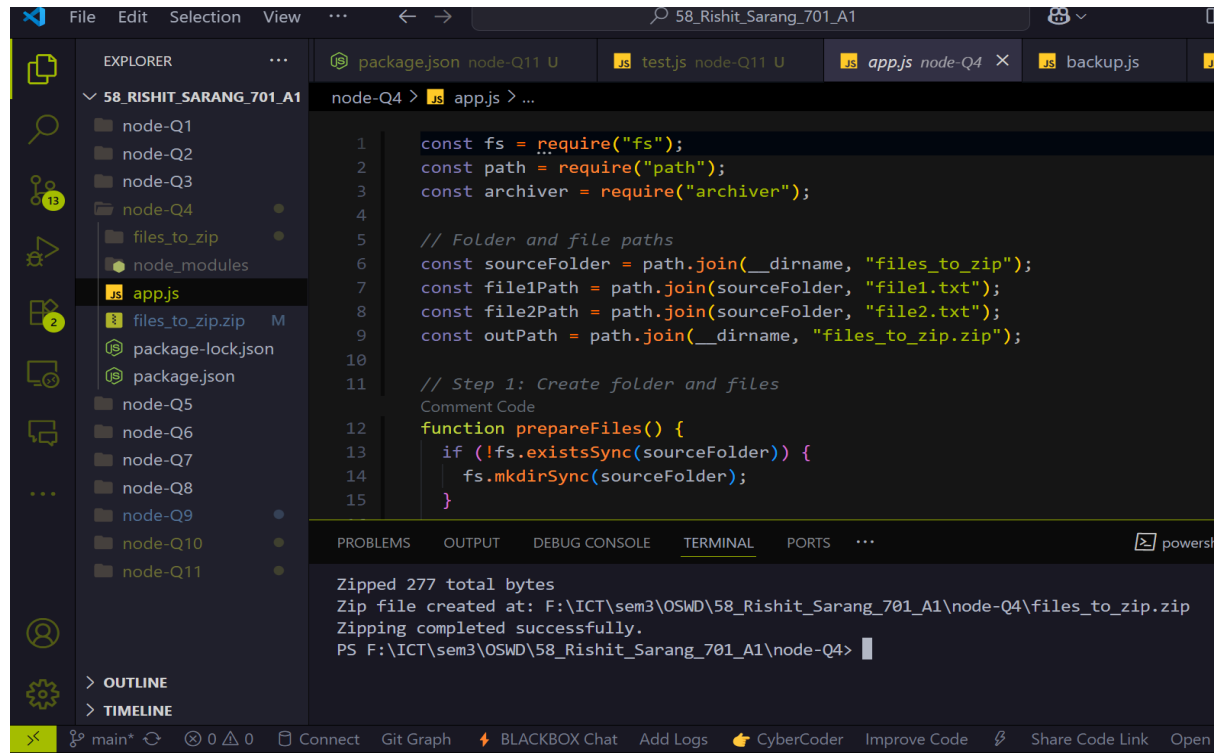
3. Develop a module for domain specific chatbot and use it in a command line application.

Output:-



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

Output:-



The screenshot shows the VS Code editor with a project named '58_Rishit_Sarang_701_A1'. The Explorer panel on the left shows a folder structure with 'node-Q4' containing 'files_to_zip' and 'node_modules'. The 'app.js' file is selected. The code in the editor is as follows:

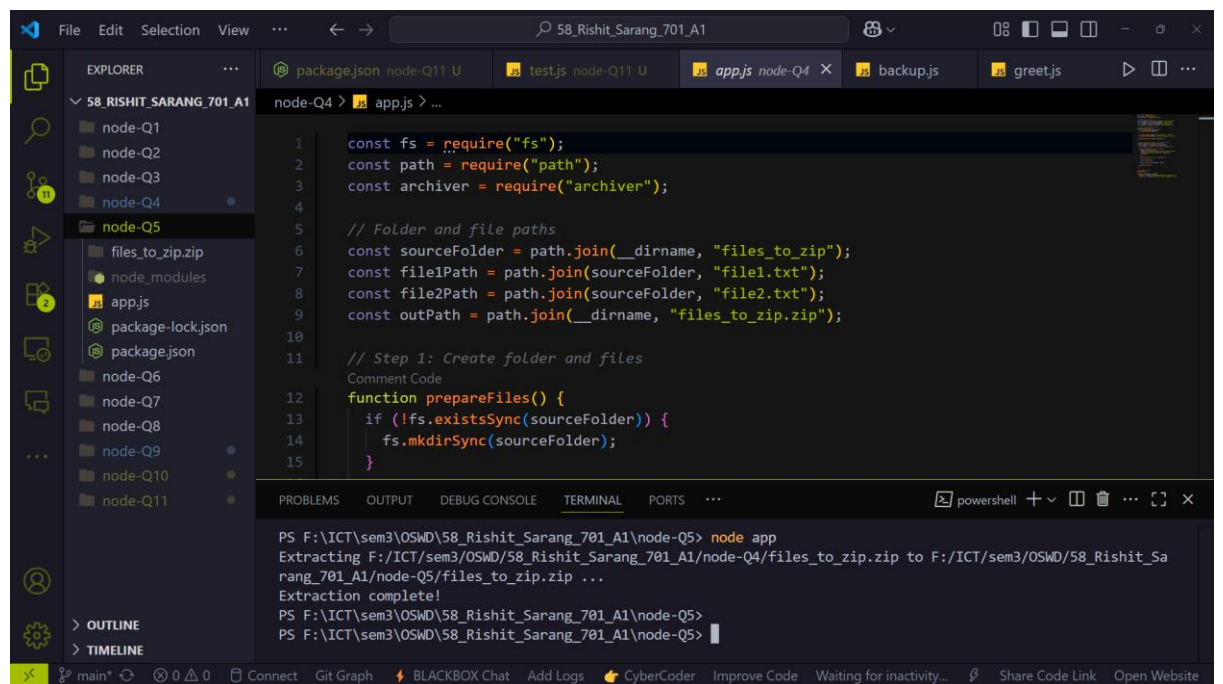
```
1  const fs = require("fs");
2  const path = require("path");
3  const archiver = require("archiver");
4
5  // Folder and file paths
6  const sourceFolder = path.join(__dirname, "files_to_zip");
7  const file1Path = path.join(sourceFolder, "file1.txt");
8  const file2Path = path.join(sourceFolder, "file2.txt");
9  const outputPath = path.join(__dirname, "files_to_zip.zip");
10
11 // Step 1: Create folder and files
12 function prepareFiles() {
13   if (!fs.existsSync(sourceFolder)) {
14     fs.mkdirSync(sourceFolder);
15   }
16 }
```

The terminal at the bottom shows the output of running the program:

```
Zipped 277 total bytes
Zip file created at: F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q4\files_to_zip.zip
Zipping completed successfully.
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q4>
```

5. Write a program to extract a zip file.

Output:-



The screenshot shows the VS Code editor with the same project. The Explorer panel shows 'node-Q5' containing 'files_to_zip.zip'. The 'app.js' file is selected. The code in the editor is as follows:

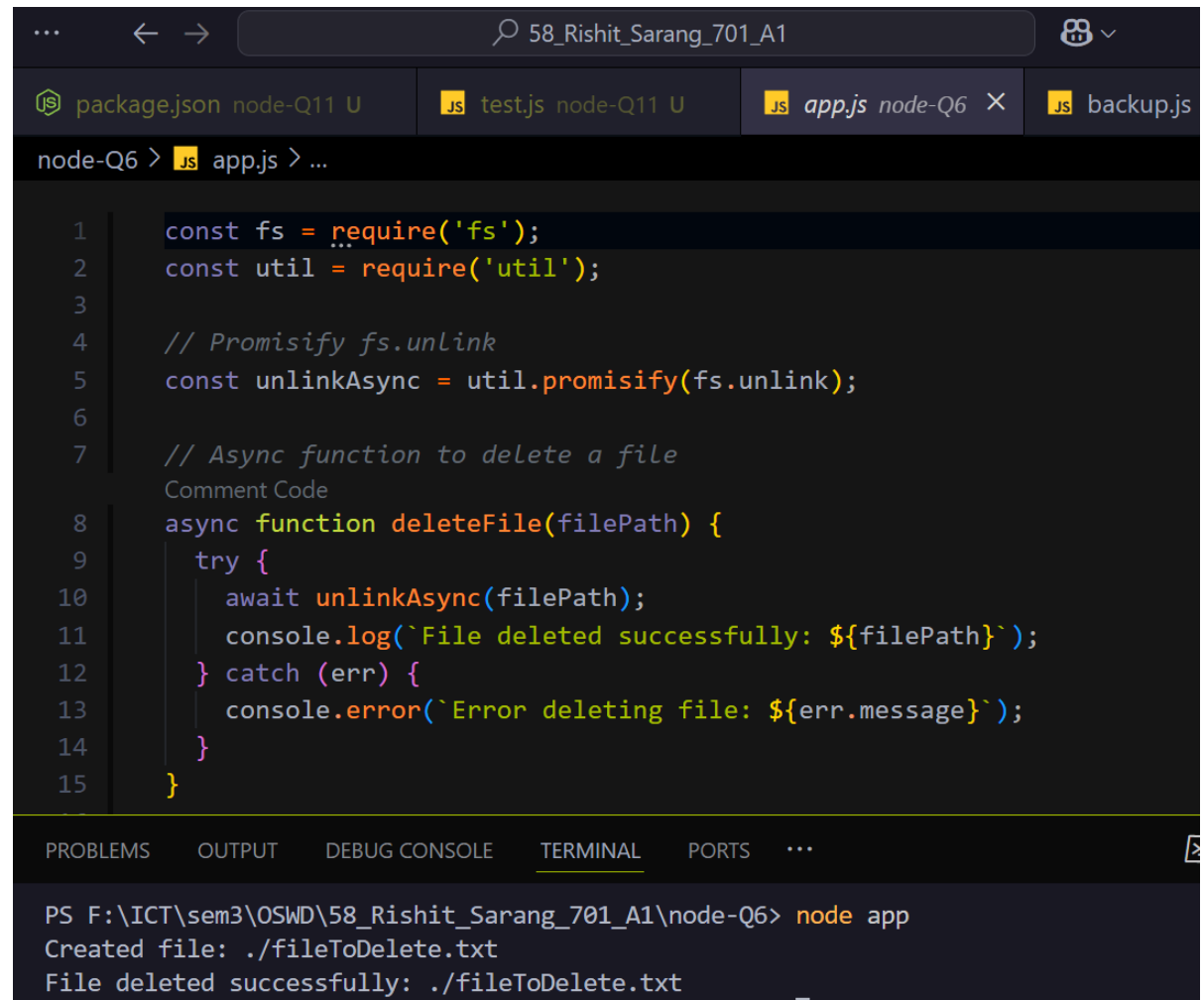
```
1  const fs = require("fs");
2  const path = require("path");
3  const archiver = require("archiver");
4
5  // Folder and file paths
6  const sourceFolder = path.join(__dirname, "files_to_zip");
7  const file1Path = path.join(sourceFolder, "file1.txt");
8  const file2Path = path.join(sourceFolder, "file2.txt");
9  const outputPath = path.join(__dirname, "files_to_zip.zip");
10
11 // Step 1: Create folder and files
12 function prepareFiles() {
13   if (!fs.existsSync(sourceFolder)) {
14     fs.mkdirSync(sourceFolder);
15   }
16 }
```

The terminal at the bottom shows the output of running the program:

```
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q5> node app
Extracting F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q4\files_to_zip.zip to F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q5\files_to_zip.zip ...
Extraction complete!
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q5>
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q5>
```

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

Output:-



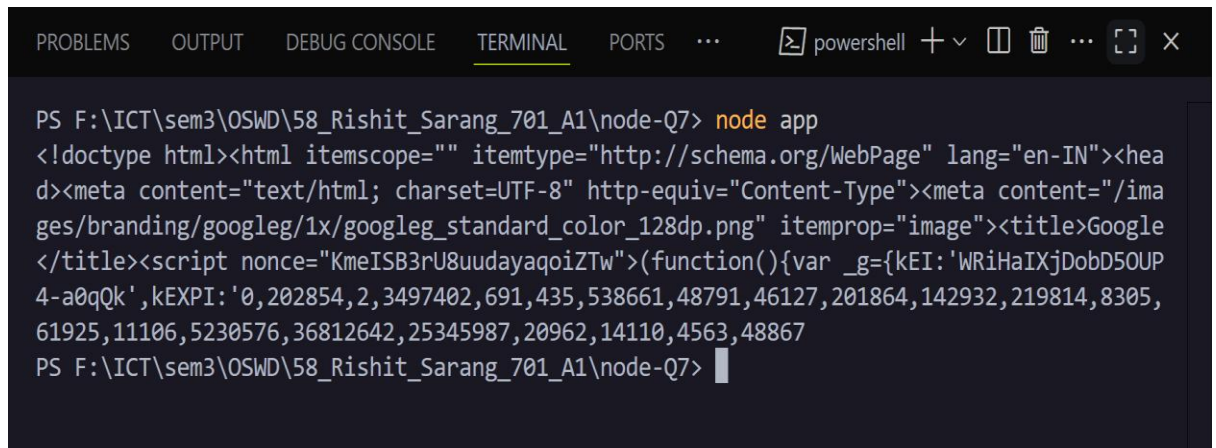
```
... 58_Rishit_Sarang_701_A1
package.json node-Q11 U test.js node-Q11 U app.js node-Q6 X backup.js
node-Q6 > JS app.js > ...

1  const fs = require('fs');
2  const util = require('util');
3
4  // Promisify fs.unlink
5  const unlinkAsync = util.promisify(fs.unlink);
6
7  // Async function to delete a file
8  async function deleteFile(filePath) {
9      try {
10         await unlinkAsync(filePath);
11         console.log(`File deleted successfully: ${filePath}`);
12     } catch (err) {
13         console.error(`Error deleting file: ${err.message}`);
14     }
15 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS ...
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q6> node app
Created file: ./fileToDelete.txt
File deleted successfully: ./fileToDelete.txt
```

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

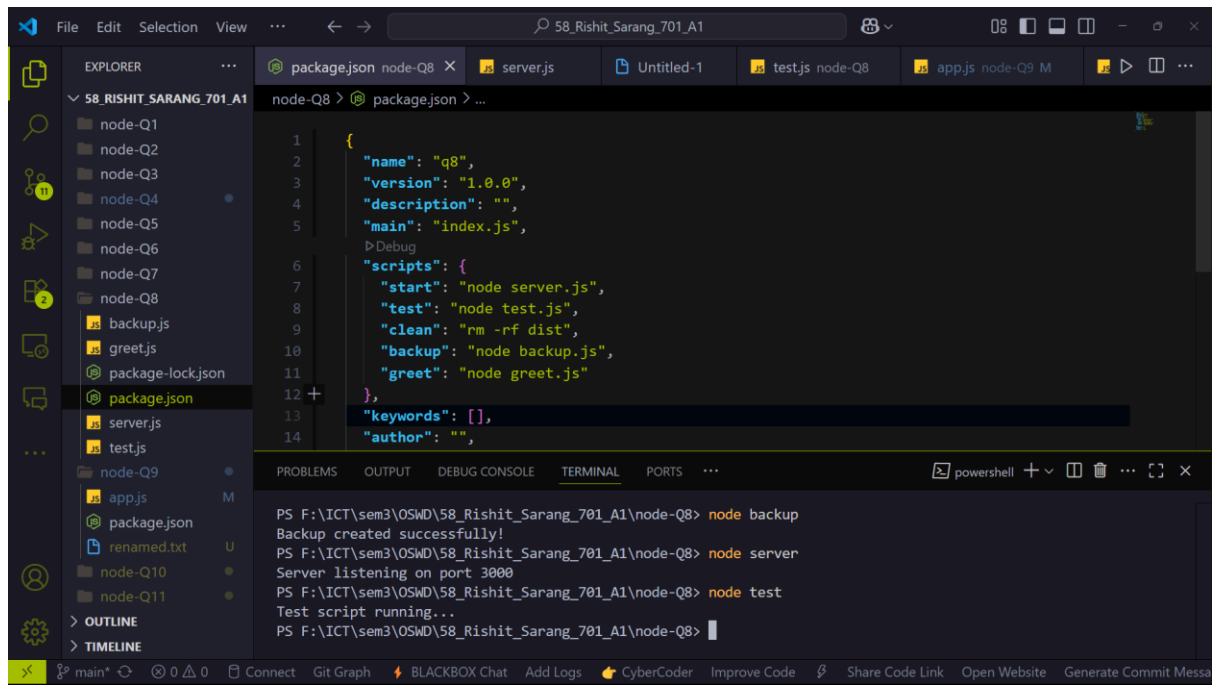
Output:-



```
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q7> node app
<!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="KmeISB3rU8uudayaqoiZTw">(function(){var _g={kEI:'WRiHaIXjDobD5OUP4-a0qQk',kEXPI:'0,202854,2,3497402,691,435,538661,48791,46127,201864,142932,219814,8305,61925,11106,5230576,36812642,25345987,20962,14110,4563,48867
```

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

Output:-

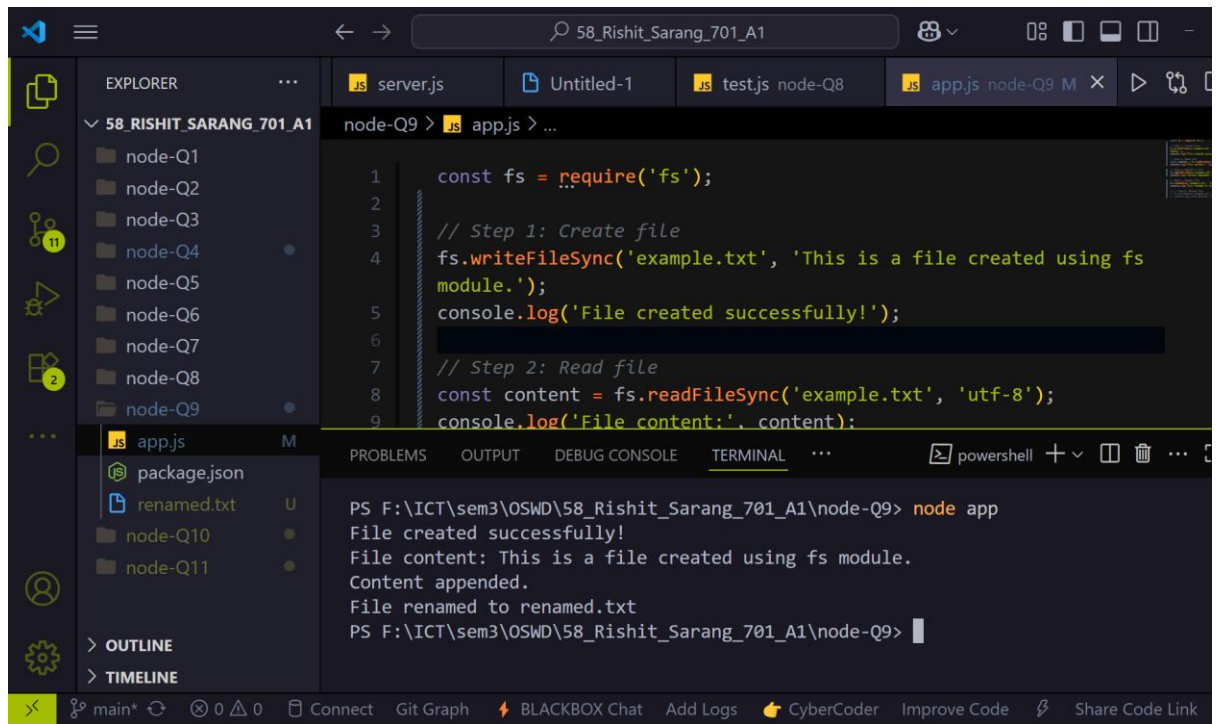


```
1 {
2   "name": "q8",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "start": "node server.js",
8     "test": "node test.js",
9     "clean": "rm -rf dist",
10    "backup": "node backup.js",
11    "greet": "node greet.js"
12  },
13   "keywords": [],
14   "author": ""
15 }
```

```
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q8> node backup
Backup created successfully!
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q8> node server
Server listening on port 3000
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q8> node test
Test script running...
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q8>
```

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

Output:-



The screenshot shows the Visual Studio Code editor with a project named '58_Rishit_Sarang_701_A1'. The Explorer sidebar on the left shows a directory structure with folders 'node-Q1' through 'node-Q11' and files 'app.js', 'package.json', and 'renamed.txt'. The main editor window displays the code for 'app.js' in the 'node-Q9' folder. The code uses the 'fs' module to create a file, read its content, and rename it. The output window at the bottom shows the execution results of the 'node app' command.

```
1  const fs = require('fs');
2
3  // Step 1: Create file
4  fs.writeFileSync('example.txt', 'This is a file created using fs
5  module.');
```

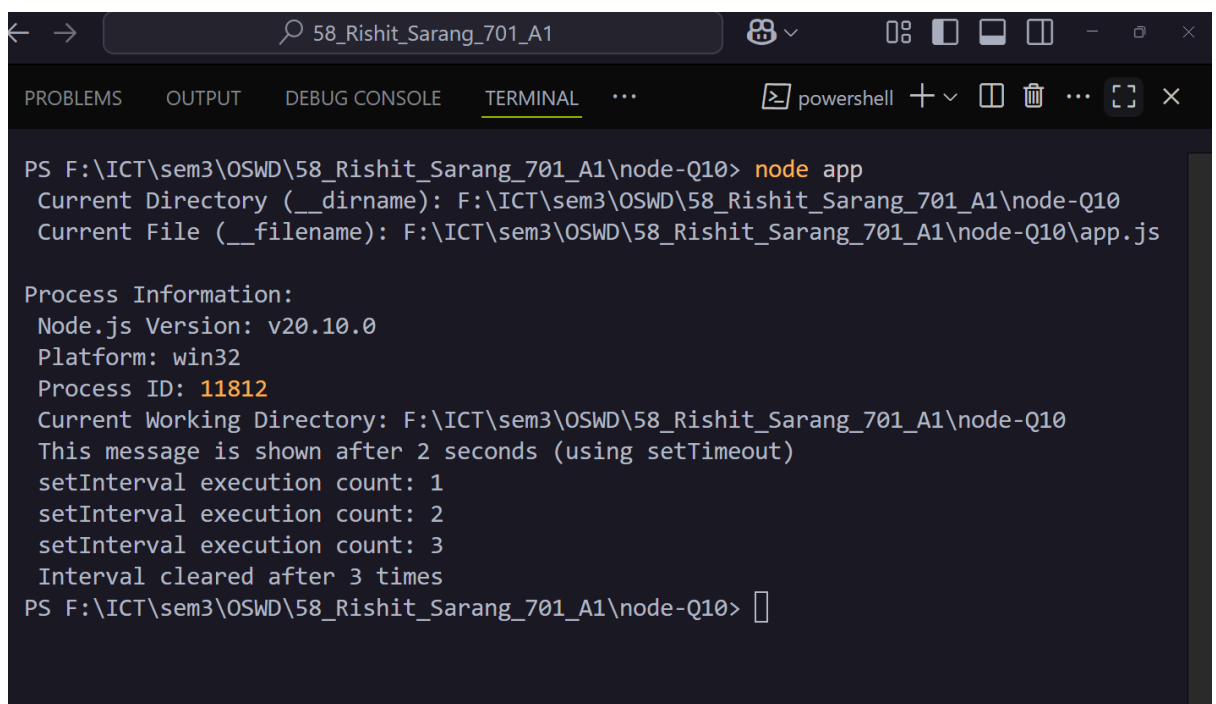
```
6
7  // Step 2: Read file
8  const content = fs.readFileSync('example.txt', 'utf-8');
9  console.log('File content:', content);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q9> node app
File created successfully!
File content: This is a file created using fs module.
Content appended.
File renamed to renamed.txt
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q9>
```

10. A program which uses global objects in nodejs.

Output:-



The screenshot shows the Visual Studio Code editor with a project named '58_Rishit_Sarang_701_A1'. The output window at the bottom shows the execution results of the 'node app' command in the 'node-Q10' folder. The output displays the current directory and file, process information, and the execution of a setTimeout function with a 2-second delay.

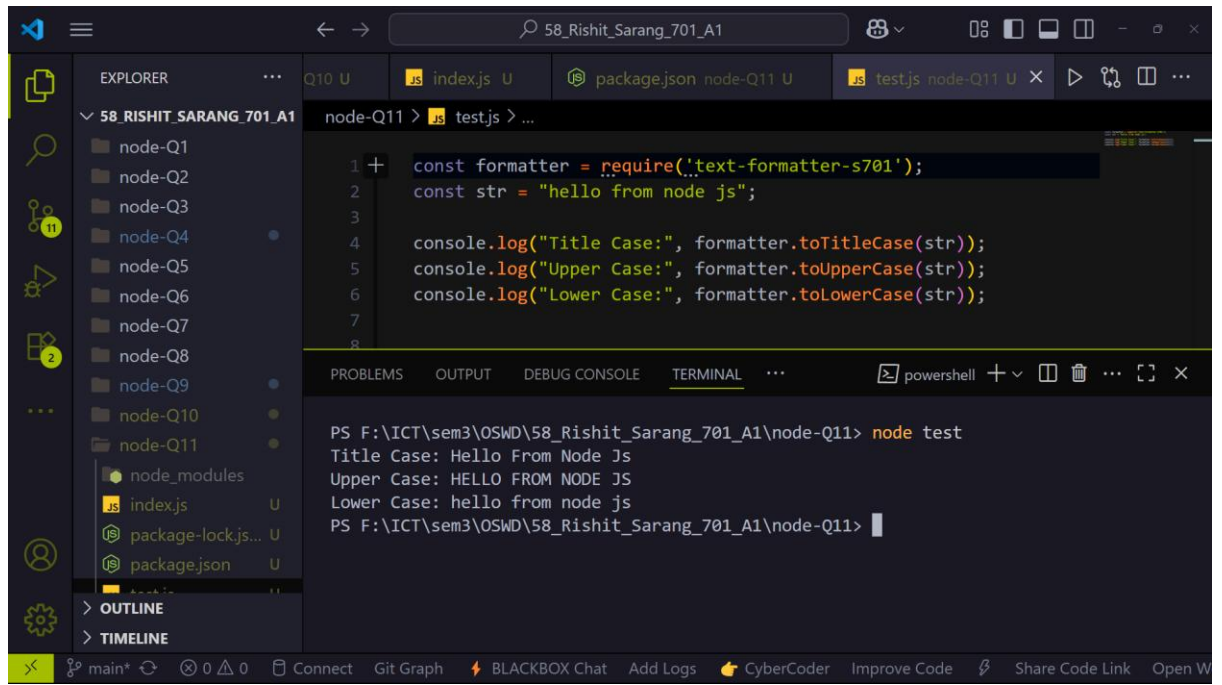
```
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q10> node app
Current Directory (__dirname): F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q10
Current File (__filename): F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q10\app.js

Process Information:
Node.js Version: v20.10.0
Platform: win32
Process ID: 11812
Current Working Directory: F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q10
This message is shown after 2 seconds (using setTimeout)
setInterval execution count: 1
setInterval execution count: 2
setInterval execution count: 3
Interval cleared after 3 times
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q10>
```

11. Develop a useful package and publish it on npmjs.com

Output:-

Test.js Otput:-



The screenshot shows the Visual Studio Code editor with a project named '58_RISHIT_SARANG_701_A1'. The Explorer sidebar on the left shows a directory structure with folders 'node-Q1' through 'node-Q11' and files 'index.js', 'package-lock.js...', and 'package.json'. The active file is 'test.js' in the 'node-Q11' folder. The code in 'test.js' is as follows:

```
1 + const formatter = require('text-formatter-s701');
2   const str = "hello from node js";
3
4   console.log("Title Case:", formatter.toTitleCase(str));
5   console.log("Upper Case:", formatter.toUpperCase(str));
6   console.log("Lower Case:", formatter.toLowerCase(str));
7
8
```

The TERMINAL panel at the bottom shows the command 'node test' being executed in a PowerShell window. The output is:

```
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q11> node test
Title Case: Hello From Node Js
Upper Case: HELLO FROM NODE JS
Lower Case: hello from node js
PS F:\ICT\sem3\OSWD\58_Rishit_Sarang_701_A1\node-Q11>
```

Package in npmjs.com:-

