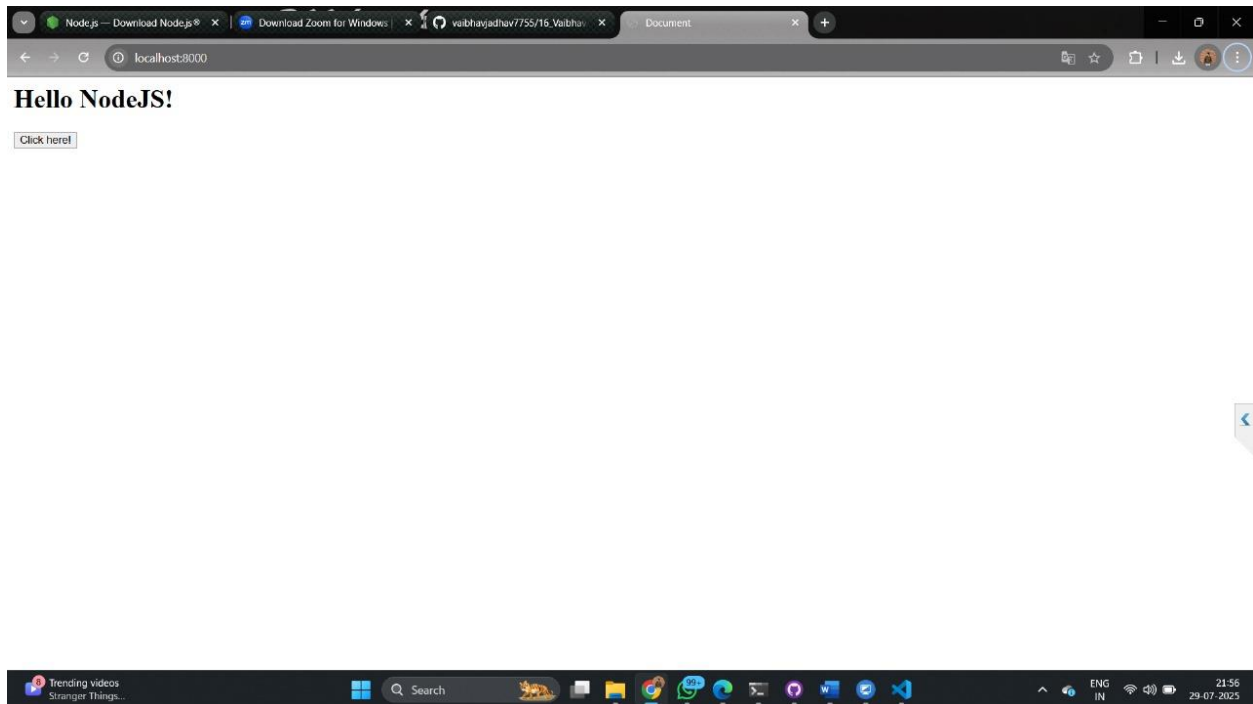


1. Develop nodejs application with following requirements:

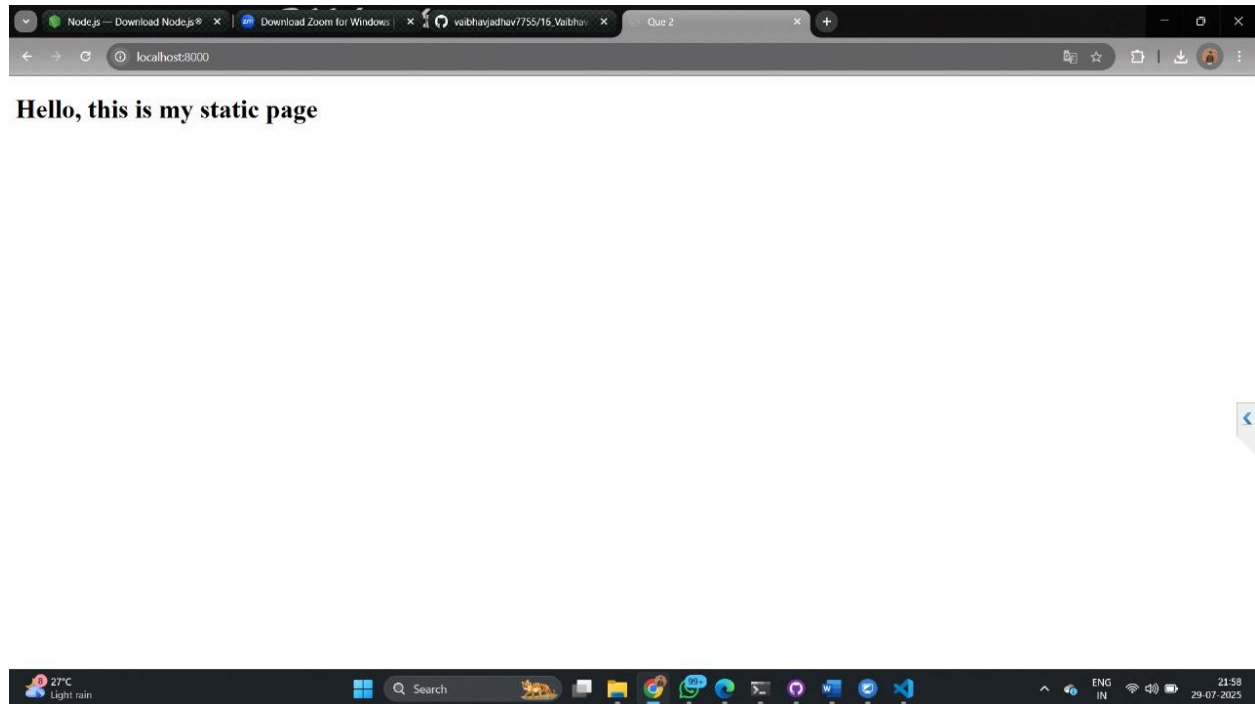
- Develop a route `"/gethello"` with GET method. It displays `"Hello NodeJS!!"` as response.
- Make an HTML page and display.
- Call `"/gethello"` route from HTML page using AJAX call. (Any frontend AJAX call API can be used.)

ScreenShot



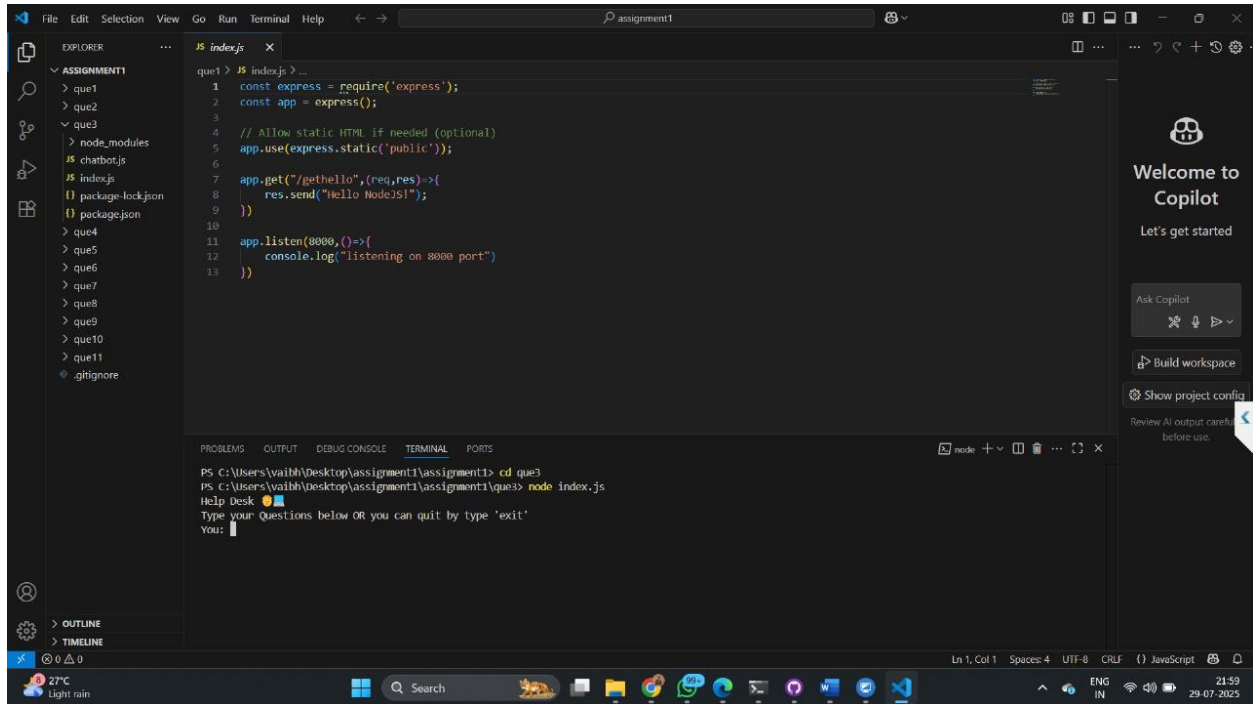
2. Develop a web server which serves static resources.

ScreenShot



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

ScreenShot



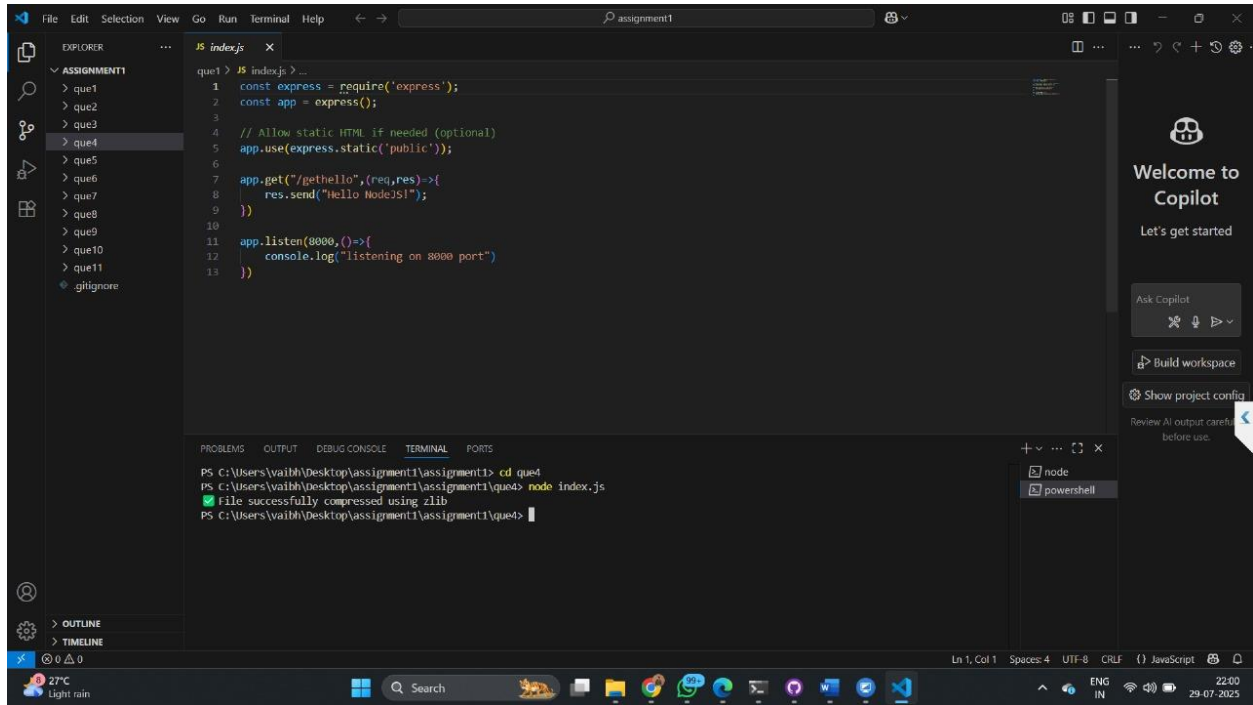
The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a project structure for 'ASSIGNMENT1' with files like 'que1', 'que2', 'que3', 'node_modules', 'chatbot.js', 'index.js', 'package-lock.json', and 'package.json'. The main editor area shows the content of 'index.js', which is a simple Express.js web server. The terminal at the bottom shows the command 'node index.js' being executed, resulting in a 'Hello NodeJS!' message and a confirmation that the server is listening on port 8080.

```
1 const express = require('express');
2 const app = express();
3
4 // Allow static HTML if needed (optional)
5 app.use(express.static('public'));
6
7 app.get("/gethello", (req, res) => {
8   res.send("Hello NodeJS!");
9 })
10
11 app.listen(8080, () => {
12   console.log("listening on 8080 port")
13 })
```

```
PS C:\Users\vaibh\Desktop\assignment1\assignment1> cd que3
PS C:\Users\vaibh\Desktop\assignment1\assignment1\que3> node index.js
Help Desk 🟡🟢🔵
Type your questions below OR you can quit by type 'exit'
You: 
```

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

ScreenShot



The screenshot shows the Visual Studio Code interface with a project named 'assignment1'. The Explorer sidebar on the left shows a folder structure with 'ASSIGNMENT1' containing subfolders 'que1' through 'que11' and a file '.gitignore'. The main editor displays a file named 'index.js' with the following JavaScript code:

```
1  const express = require('express');
2  const app = express();
3
4  // Allow static HTML if needed (optional)
5  app.use(express.static('public'));
6
7  app.get("/gethello", (req, res) => {
8    res.send("Hello NodeJS!");
9  })
10
11 app.listen(8080, () => {
12   console.log("listening on 8080 port")
13 })
```

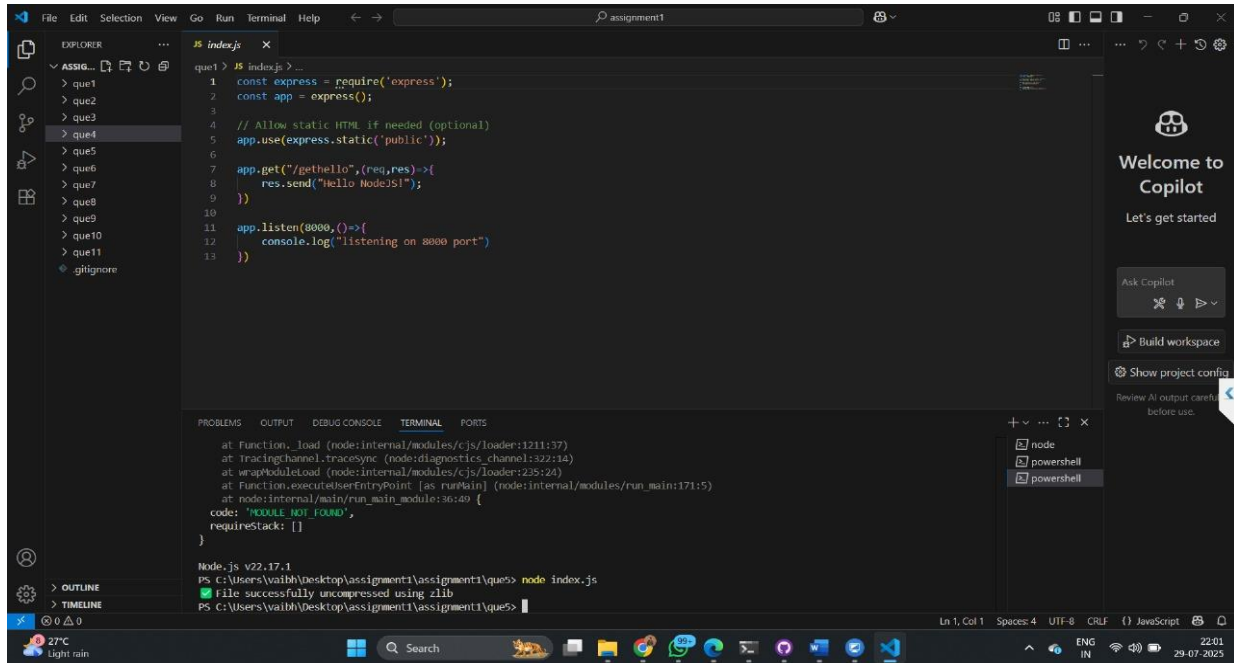
The bottom panel shows the TERMINAL with the following commands and output:

```
PS C:\Users\vaibh\Desktop\assignment1\assignment1> cd que4
PS C:\Users\vaibh\Desktop\assignment1\assignment1\que4> node index.js
[?] file successfully compressed using zlib
PS C:\Users\vaibh\Desktop\assignment1\assignment1\que4> |
```

On the right side of the editor, there is a 'Welcome to Copilot' panel with options to 'Ask Copilot', 'Build workspace', and 'Show project config'. The status bar at the bottom indicates 'Ln 1, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and 'JavaScript'.

5. Write a program to extract a zip file.

ScreenShot



The screenshot shows the Visual Studio Code (VS Code) editor interface. The Explorer panel on the left shows a file named `index.js` under a folder named `assignment1`. The main editor area displays the content of `index.js`, which is a simple Express.js server program. The terminal window at the bottom shows the output of running `node index.js`, indicating that the file was successfully uncompressed using zlib and is now listening on port 8080.

```
1 const express = require('express');
2 const app = express();
3
4 // Allow static HTML if needed (optional)
5 app.use(express.static('public'));
6
7 app.get('/', (req, res) => {
8   res.send('Hello NodeJS!');
9 })
10
11 app.listen(8080, () => {
12   console.log('listening on 8080 port')
13 })
```

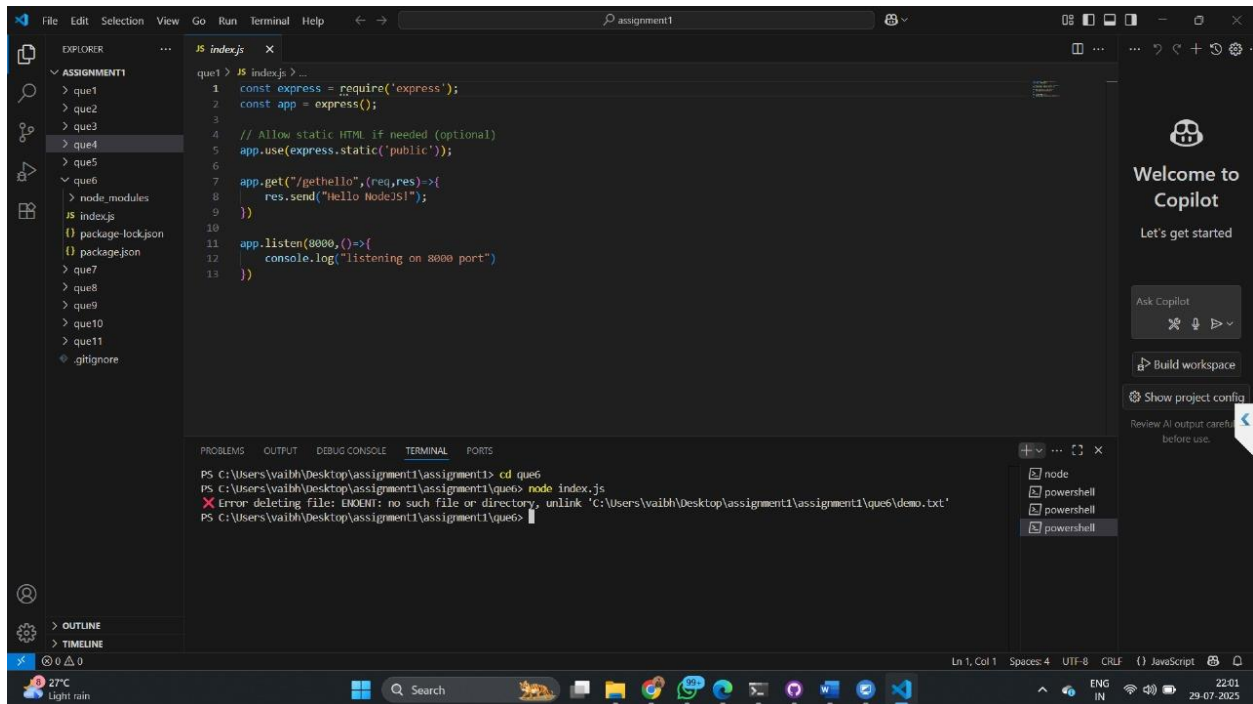
Terminal Output:

```
at Function._load (node:internal/modules/cjs/loader:1211:37)
at tracingChannel.traceSync (node:diagnostics_channel:122:14)
at wrapModuleLoad (node:internal/modules/cjs/loader:1235:24)
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:171:5)
at node:internal/main/run_main_module:36:40 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}

Node.js v22.17.1
PS C:\Users\vaibh\Desktop\assignment1\assignment1\ques> node index.js
[File successfully uncompressed using zlib]
PS C:\Users\vaibh\Desktop\assignment1\assignment1\ques>
```

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

ScreenShot



The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a project structure with a folder named 'ASSIGNMENT1' containing subfolders 'que1' through 'que11' and files 'package-lock.json', 'package.json', and '.gitignore'. The file 'index.js' is selected and open in the editor. The code in 'index.js' is as follows:

```
1  const express = require('express');
2  const app = express();
3
4  // Allow static HTML if needed (optional)
5  app.use(express.static('public'));
6
7  app.get("/gethello", (req, res) => {
8    res.send("Hello NodeJS!");
9  })
10
11 app.listen(8000, () => {
12   console.log("listening on 8000 port")
13 })
```

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

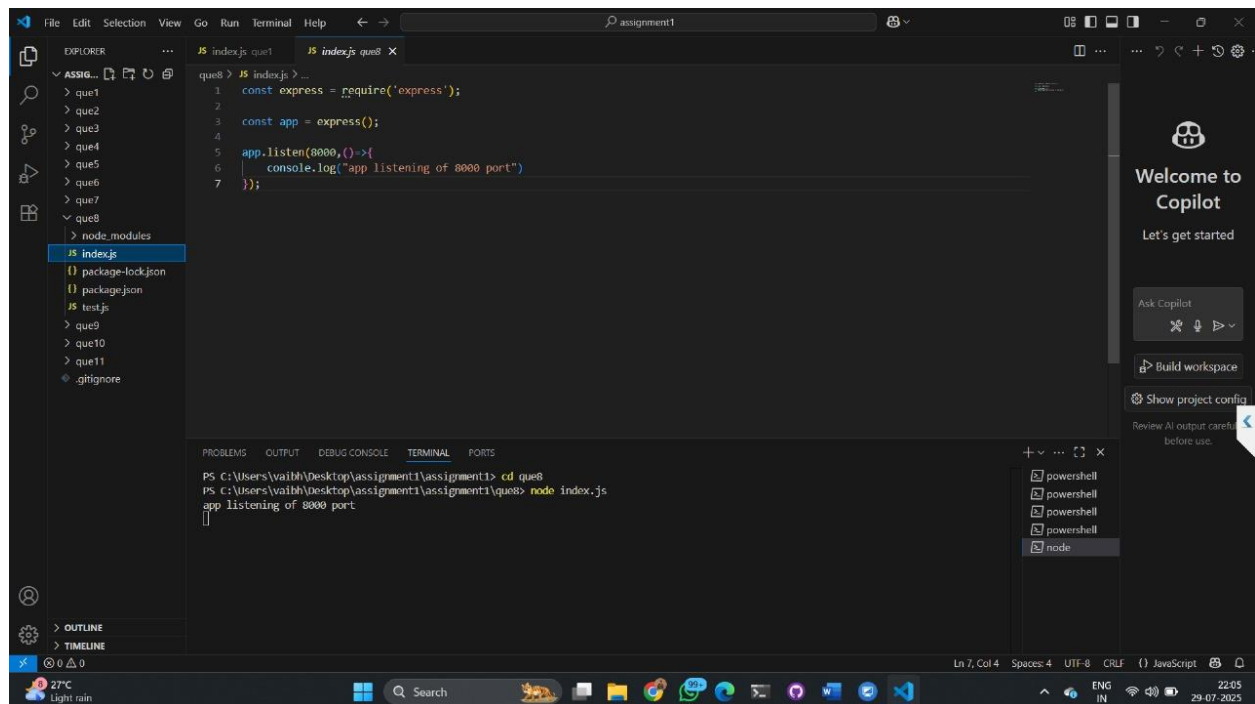
```
PS C:\Users\vaibh\Desktop\assignment1\assignment1> cd que6
PS C:\Users\vaibh\Desktop\assignment1\assignment1\que6> node index.js
Error deleting file: ENOENT: no such file or directory, unlink 'C:\Users\vaibh\Desktop\assignment1\assignment1\que6\demo.txt'
```

The status bar at the bottom indicates the current file is 'index.js' at line 1, column 1, using UTF-8 encoding and CRLF line endings. The system tray at the very bottom shows the date and time as 22:01 on 29-07-2025.

ScreenShot

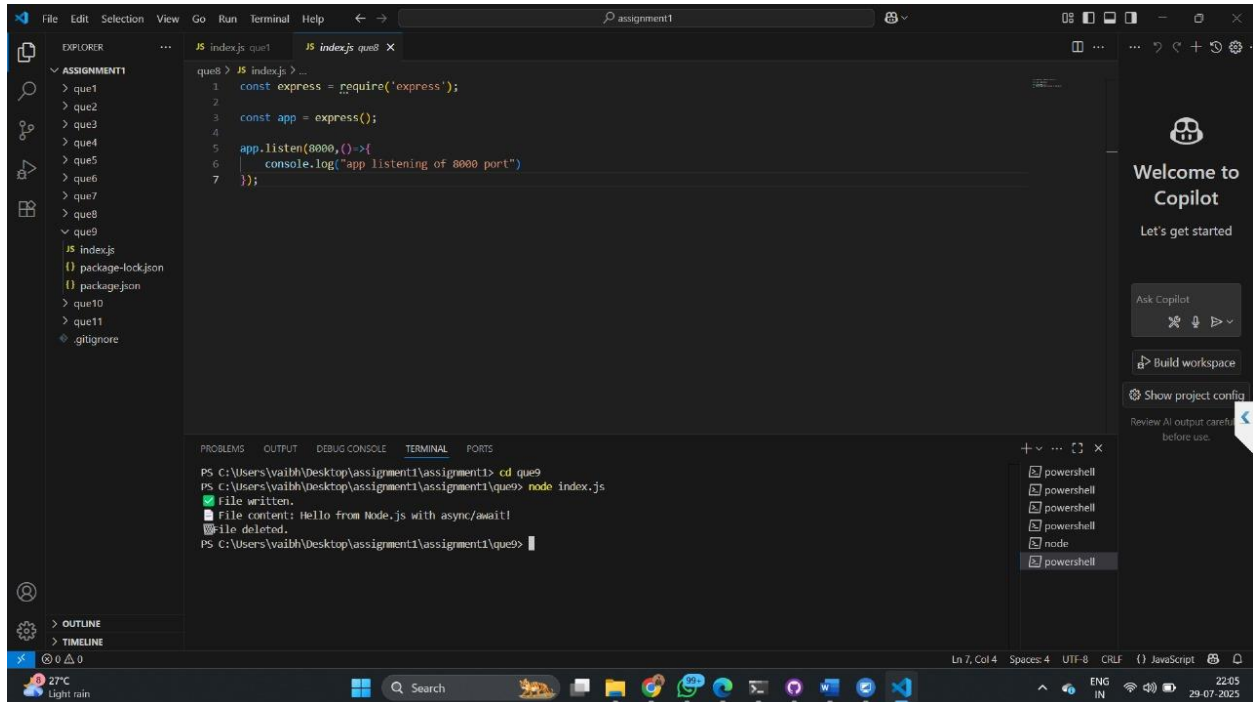
8. Set a server script, a test script and 3 user defined scripts in package.json file in your nodejs application.

ScreenShot



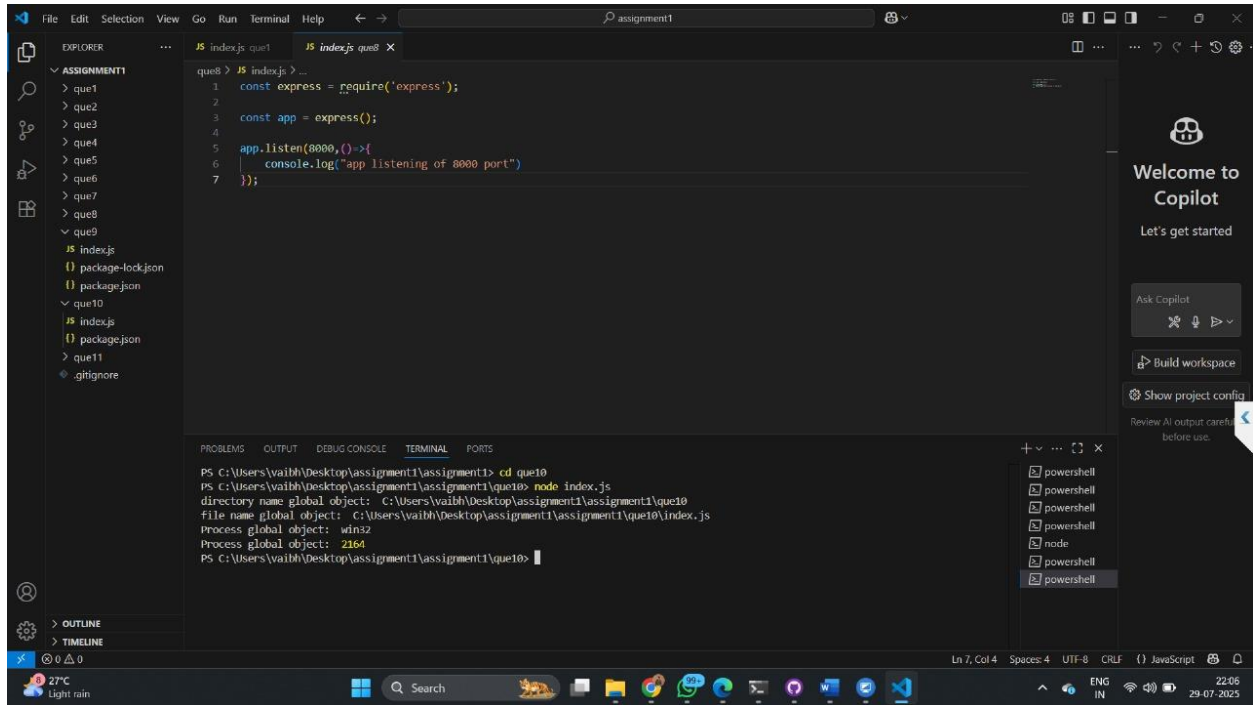
9. A program which calls useful functions in fs module.

ScreenShot



10. A program which uses global objects in nodejs.

ScreenShot



The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays a project structure with a folder named 'ASSIGNMENT1' containing subfolders 'que1' through 'que9' and files 'index.js', 'package-lock.json', and 'package.json'. The main editor window shows a JavaScript file named 'index.js' with the following code:

```
1 const express = require('express');
2
3 const app = express();
4
5 app.listen(8080, () => {
6   console.log("app listening of 8080 port")
7 });
```

Below the editor, the TERMINAL panel is active, showing the execution of the program. The commands and output are as follows:

```
PS C:\Users\vaibh\Desktop\assignment1\assignment1> cd que10
PS C:\Users\vaibh\Desktop\assignment1\assignment1\que10> node index.js
directory name global object: C:\Users\vaibh\Desktop\assignment1\assignment1\que10
file name global object: C:\Users\vaibh\Desktop\assignment1\assignment1\que10\index.js
process global object: win32
Process global object: 2164
PS C:\Users\vaibh\Desktop\assignment1\assignment1\que10>
```

The status bar at the bottom indicates the current file is 'index.js' at line 7, column 4, with 4 spaces, UTF-8 encoding, and CRLF line endings. The system tray at the very bottom shows the date and time as 22:06 on 29-07-2025.

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

ScreenShot

