

**Name : Vaishnavi Gosavi**

**Roll No. : 24**

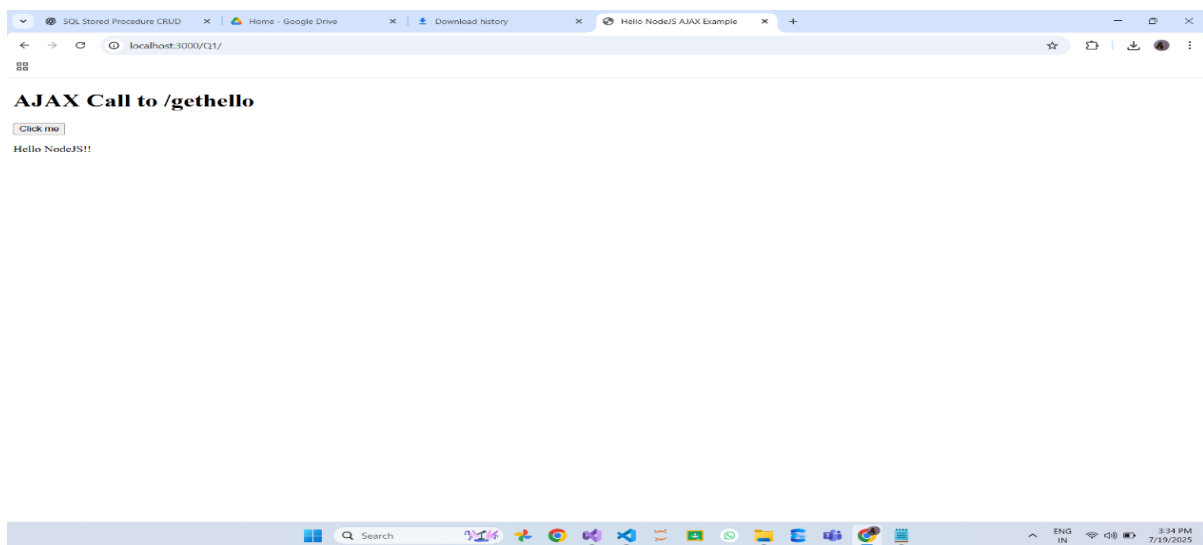
**Div : A**

**Sem : 7<sup>th</sup>**

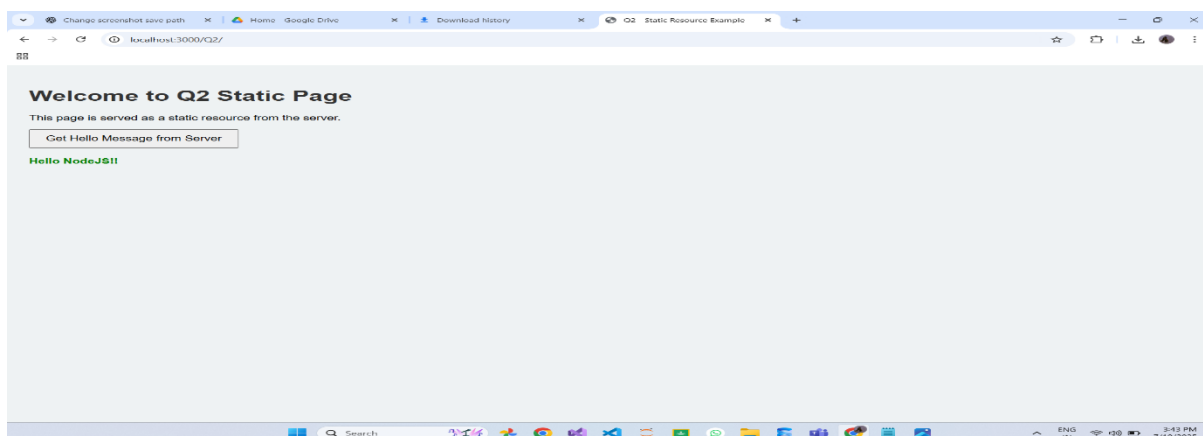
**Subject : 701**

**Practical – Assignment : 1**

**Question 1:**



**Question 2:**



### Question 3:

```
public > Q3 > # chatbot.js > WeatherChatbot > constructor > will it rain tomorrow?
1 // chatbot.js
2 class WeatherChatbot {
3   constructor() {
4     this.responses = {
5       'hello': 'Hello! How can I help you with weather information?',
6       'how is the weather today?': 'Today, it is sunny with a high of 25°C.',
7       'will it rain tomorrow?': 'No, there is no rain forecast for tomorrow.',
8       'bye': 'Goodbye! Have a nice day.',
9     };
10  }
11
12  getResponse(input) {
13    input = input.toLowerCase().trim();
14    return this.responses[input] || "Sorry, I don't understand that. Please ask about the weather.";
15  }
16
17  module.exports = WeatherChatbot;
18
19
```

PS C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\Q3> node app.js  
Weather Chatbot started! Ask me about the weather or type 'bye' to exit.  
You: hello  
Bot: Hello! How can I help you with weather information?  
You: how is the weather today?  
Bot: Today, it is sunny with a high of 25°C.  
You: will it rain tomorrow?  
Bot: No, there is no rain forecast for tomorrow.  
You: bye  
Bot: Goodbye! Have a nice day.  
Chatbot session ended.  
PS C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\Q3>

### Question 4:

```
public > Q4 > # zipfolder.js >
1 const fs = require('fs');
2 const path = require('path');
3 const archiver = require('archiver');
4
5 // Path to folder to zip - one level up then into public
6 const folderToZip = path.join(__dirname, '..', 'public');
7
8 // Output zip file path - you can save it inside Q4 or anywhere else you want
9 const outputZip = path.join(__dirname, 'public.zip');
10
11 const output = fs.createWriteStream(outputZip);
12 const archive = archiver('zip', {
13   zlib: { level: 9 }
14 });
15
16 output.on('close', () => {
17   console.log('Archive created successfully, total bytes: ${archive.pointer()}');
18 });
19
20 archive.on('error', (err) => {
21   throw err;
22 });
23
24 archive.pipe(output);
25
26 archive.directory(folderToZip, false);
```

PS C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\Q4> node zipfolder.js  
Archive created successfully, total bytes: 22  
PS C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\Q4>

## Question 5:

The screenshot shows a VS Code editor with a file explorer on the left containing a project named 'Assignment-1'. The file explorer shows a directory structure with files like 'node\_modules', 'public', 'Q1', 'Q2', 'Q3', 'Q4', 'public.zip', 'zipfolder.js', 'Q5', 'extractZip.js', 'Q6', 'Q7', 'Q8', 'Q9', 'Q10', 'Q11', 'package-lock.json', and 'server.js'. The main editor area shows the 'extractZip.js' file with the following code:

```
1 const fs = require('fs');
2 const unzipper = require('unzipper');
3 const path = require('path');
4
```

The terminal output shows the command `node extractZip.js` being executed, resulting in the message: `Extraction complete into q5\extracted!`. The status bar at the bottom indicates the file is at line 5, column 114, with 2 spaces, UTF-8 encoding, and CRLF line endings.

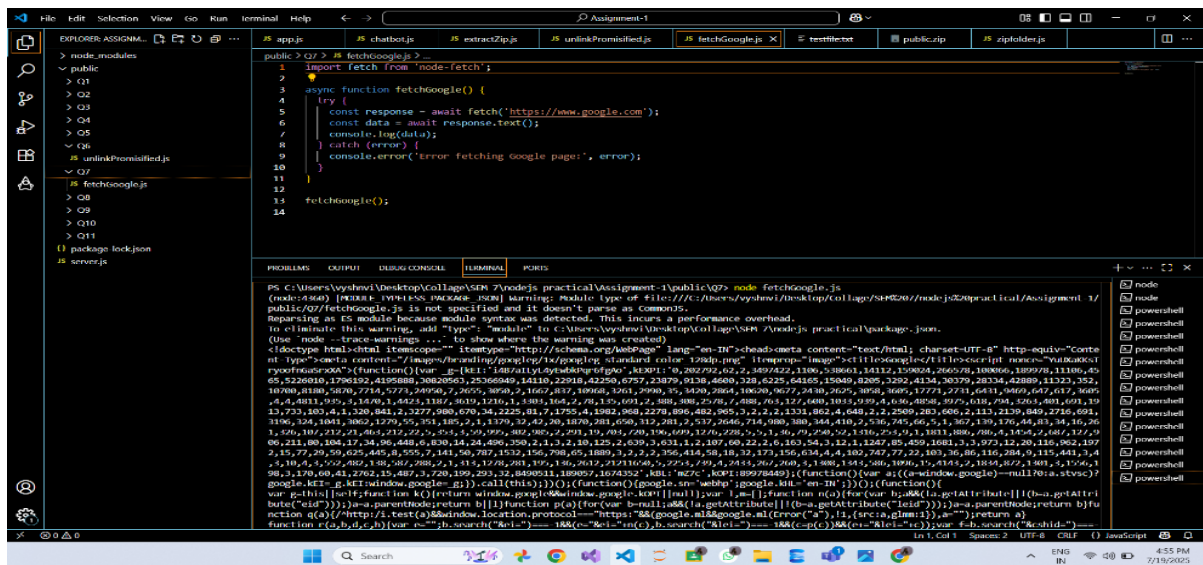
## Question 6:

The screenshot shows a VS Code editor with a file explorer on the left containing a project named 'Assignment-1'. The file explorer shows a directory structure with files like 'node\_modules', 'public', 'Q1', 'Q2', 'Q3', 'Q4', 'Q5', 'unlinkPromisified.js', 'Q7', 'Q8', 'Q9', 'Q10', 'Q11', 'package-lock.json', and 'server.js'. The main editor area shows the 'unlinkPromisified.js' file with the following code:

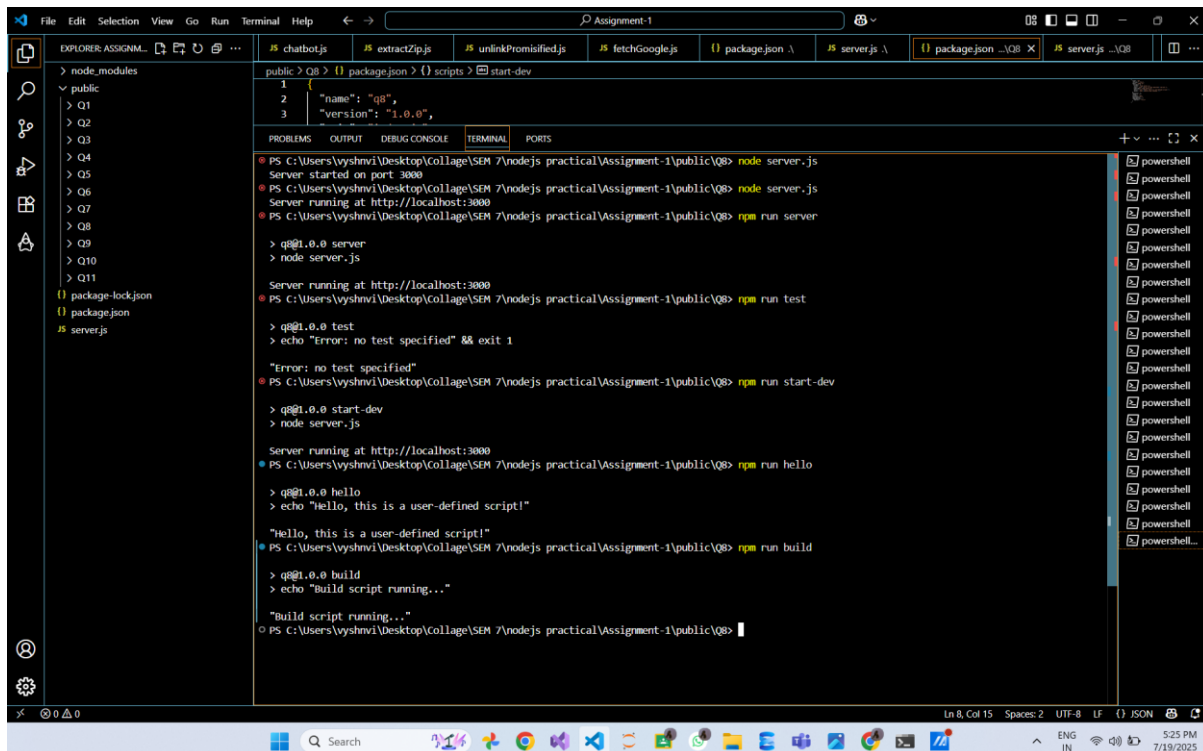
```
1 const util = require('util');
2
3 // Promisify the unlink function
4 const unlinkAsync = util.promisify(fs.unlink);
5
6 // file to delete (update path accordingly)
7 const filePath = './testfile.txt'; // Replace with your actual file path
8
9 // using async/await
10 async function deleteFile() {
11   try {
12     await unlinkAsync(filePath);
13     console.log(`File "${filePath}" deleted successfully.`);
14   } catch (err) {
15     console.error(`Error deleting file "${filePath}":`, err.message);
16   }
17 }
18
```

The terminal output shows the command `node unlinkPromisified.js` being executed, resulting in an error: `Error: Cannot find module 'C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\q5\unlinkPromisified.js'`.` The status bar at the bottom indicates the file is at line 18, column 2, with 2 spaces, UTF-8 encoding, and CRLF line endings.

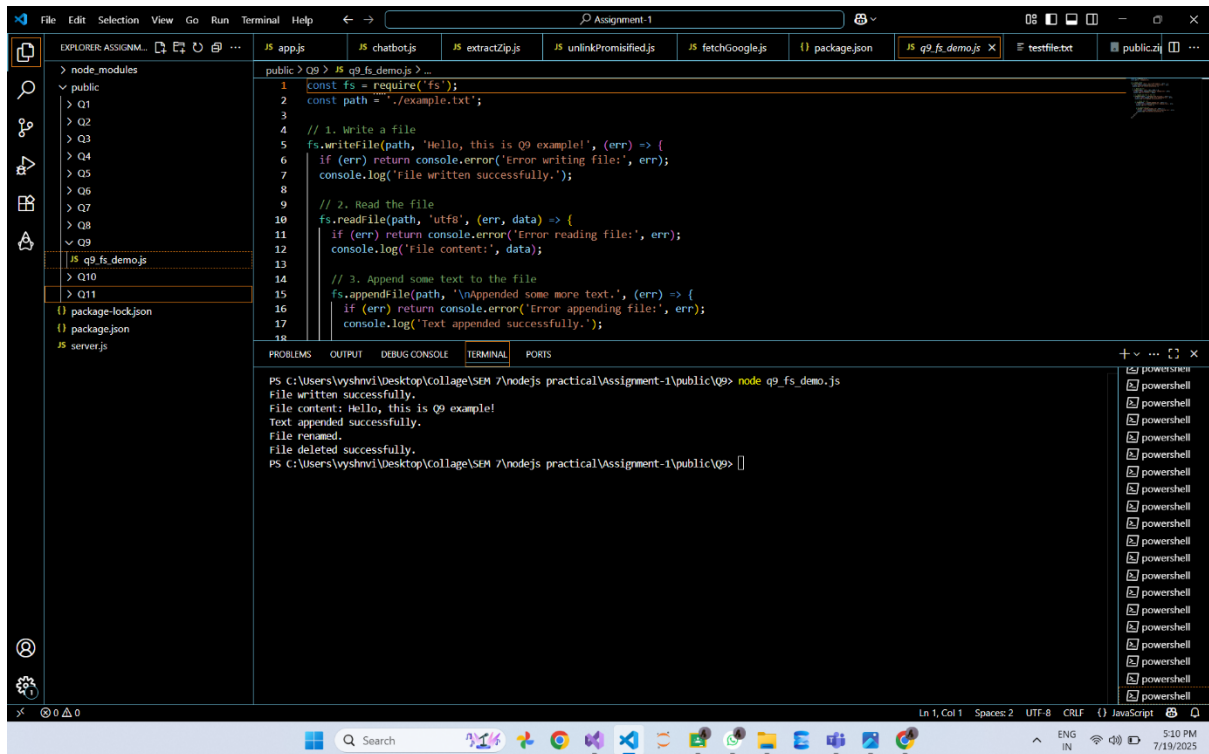
Question 7:



Question 8:



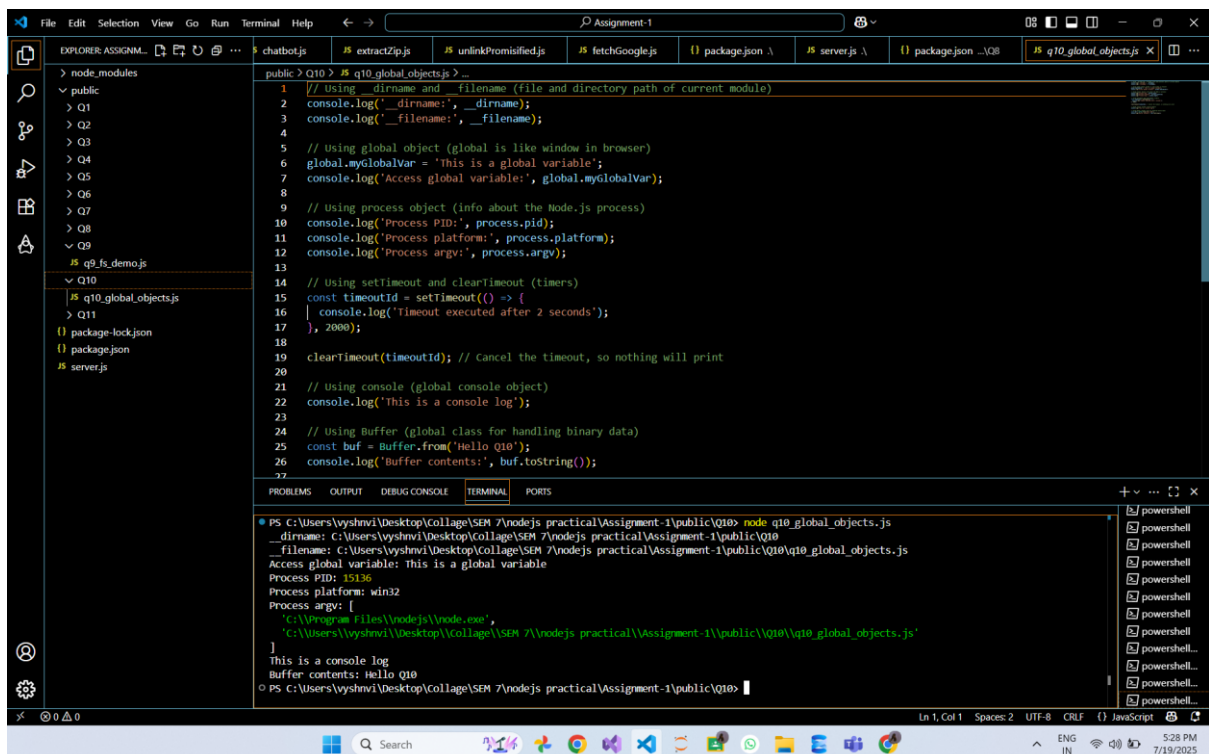
## Question 9:



```
1 const fs = require('fs');
2 const path = './example.txt';
3
4 // 1. Write a file
5 fs.writeFile(path, 'Hello, this is Q9 example!', (err) => {
6   if (err) return console.error('Error writing file:', err);
7   console.log('File written successfully.');
```

```
PS C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\Q9> node q9_fs_demo.js
File written successfully.
File content: Hello, this is Q9 example!
Text appended successfully.
File renamed.
File deleted successfully.
PS C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\Q9>
```

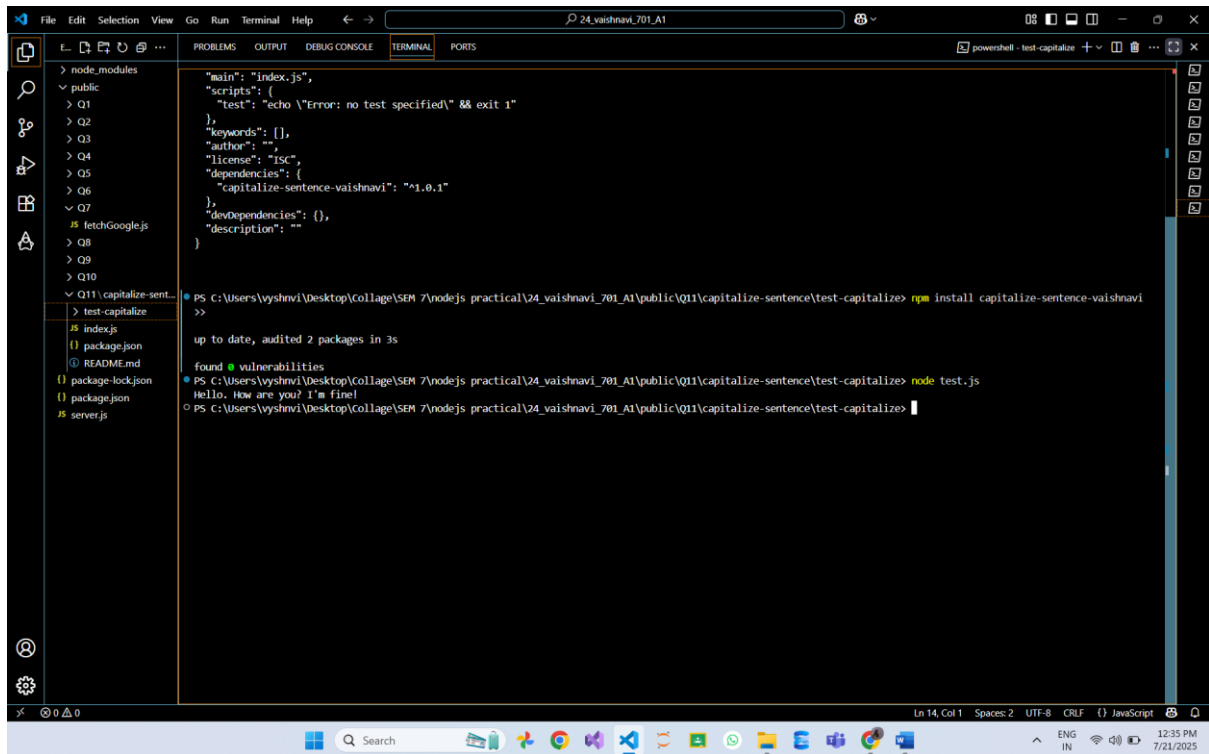
## Question 10:



```
1 // Using __dirname and filename (file and directory path of current module)
2 console.log('__dirname:', __dirname);
3 console.log('__filename:', __filename);
4
5 // Using global object (global is like window in browser)
6 global.myGlobalVar = 'This is a global variable';
7 console.log('Access global variable:', global.myGlobalVar);
8
9 // Using process object (info about the Node.js process)
10 console.log('Process PID:', process.pid);
11 console.log('Process platform:', process.platform);
12 console.log('Process argv:', process.argv);
13
14 // Using setTimeout and clearTimeout (timers)
15 const timeoutId = setTimeout(() => {
16   console.log('Timeout executed after 2 seconds');
17 }, 2000);
18
19 clearTimeout(timeoutId); // Cancel the timeout, so nothing will print
20
21 // Using console (global console object)
22 console.log('This is a console log');
```

```
PS C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\Q10> node q10_global_objects.js
__dirname: C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\Q10
__filename: C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\Q10\q10_global_objects.js
Access global variable: This is a global variable
Process PID: 15136
Process platform: win32
Process argv: [
  'C:\Program Files\nodejs\node.exe',
  'C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\Q10\q10_global_objects.js'
]
This is a console log
Buffer contents: Hello Q10
PS C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\Assignment-1\public\Q10>
```

### Question 11:



The screenshot shows a Visual Studio Code editor with a project structure on the left and a terminal on the right. The project structure includes a `public` directory with subdirectories `Q1` through `Q10`, and a `Q11` directory containing `capitalize-sentence` and `test-capitalize`. The `test-capitalize` directory contains `index.js`, `package.json`, `package-lock.json`, `README.md`, and `server.js`. The terminal shows the following commands and output:

```
PS C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\24_vaishnavi_701_A1\public\Q11\capitalize-sentence\test-capitalize> npm install capitalize-sentence-vashnavi
>>
up to date, audited 2 packages in 3s
found 0 vulnerabilities
PS C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\24_vaishnavi_701_A1\public\Q11\capitalize-sentence\test-capitalize> node test.js
Hello. How are you? I'm fine!
PS C:\Users\vyshnvi\Desktop\Collage\SEM 7\nodejs practical\24_vaishnavi_701_A1\public\Q11\capitalize-sentence\test-capitalize>
```

The `package.json` file in the `test-capitalize` directory is as follows:

```
{
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "capitalize-sentence-vashnavi": "^1.0.1"
  },
  "devDependencies": {},
  "description": ""
}
```