

Q1.

The screenshot shows a web browser window with the title "Node.js AJAX Example". The address bar shows "localhost:3000". The page content includes a heading "Node.js AJAX Call", a paragraph "Click the button below to fetch a message from the Node.js server.", a button labeled "Fetch Hello Message", and the text "Hello NodeJS!!".

The browser's developer tools are open to the "Network" tab. A request to "http://localhost:3000/gethello" is selected. The "Headers" sub-tab is active, showing the following details:

- General:**
  - Request URL: http://localhost:3000/gethello
  - Request Method: GET
  - Status Code: 304 Not Modified
  - Remote Address: [::1]:3000
  - Referrer Policy: strict-origin-when-cross-origin
- Response Headers:**
  - Connection: keep-alive
  - Date: Sat, 19 Jul 2025 03:19:36 GMT
  - Etag: W/"e-06Zx9q+hUe0yQdApH7vRZi7l+M"
  - Keep-Alive: timeout=5
  - X-Powered-By: Express

The bottom status bar indicates "2 requests 442 B transferred 1.5 k".

Q2.

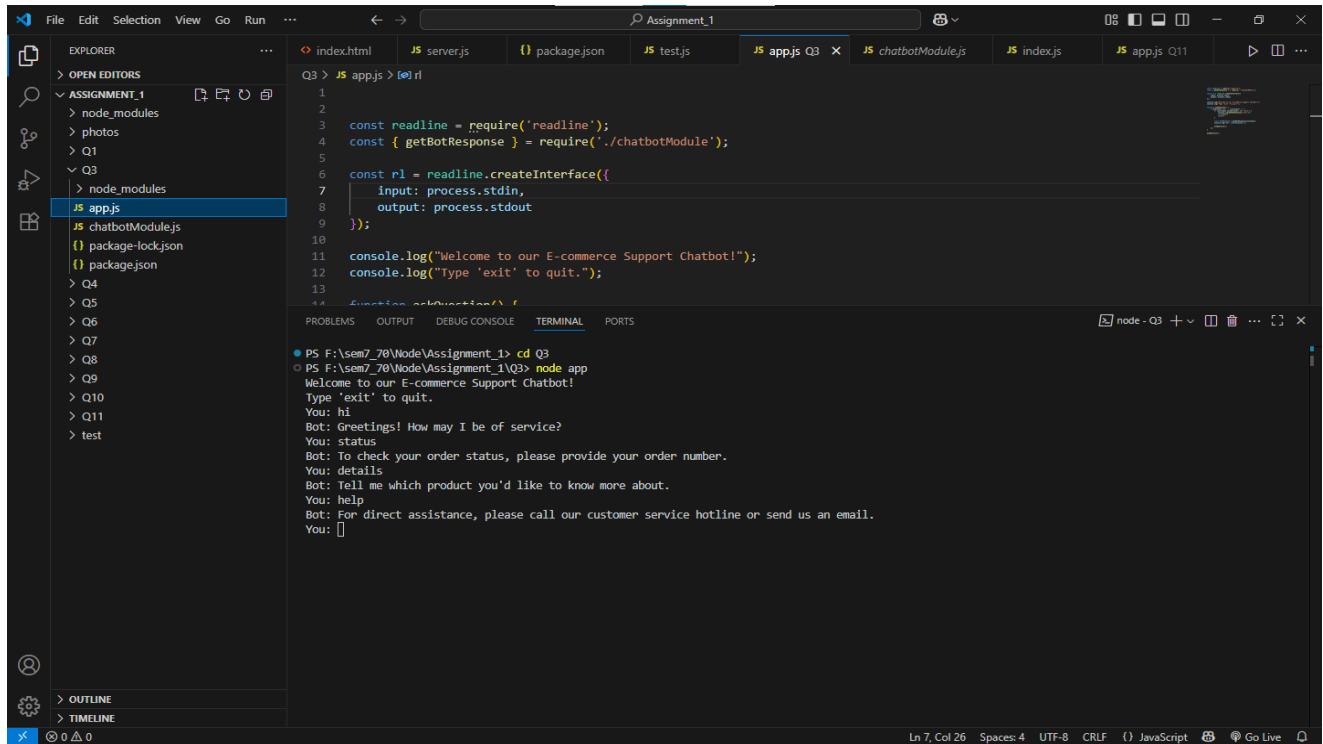
The screenshot shows a web browser window with the title "contact us". The address bar shows "localhost:3000/contactus.html". The page content includes the heading "contact us".

The browser's developer tools are open to the "Network" tab. A request to "contactus.html" is selected. The "Headers" sub-tab is active, showing the following details:

Name	Status	Type	Initiator	Size	Time	Fulfilled by
contactus.html	200	document	Other	531 B	2 ms	

The bottom status bar indicates "1 requests 531 B transferred 236 B resources Finish: 2 ms DOMContentLoaded: 19 ms Load: 20 ms".

Q3.

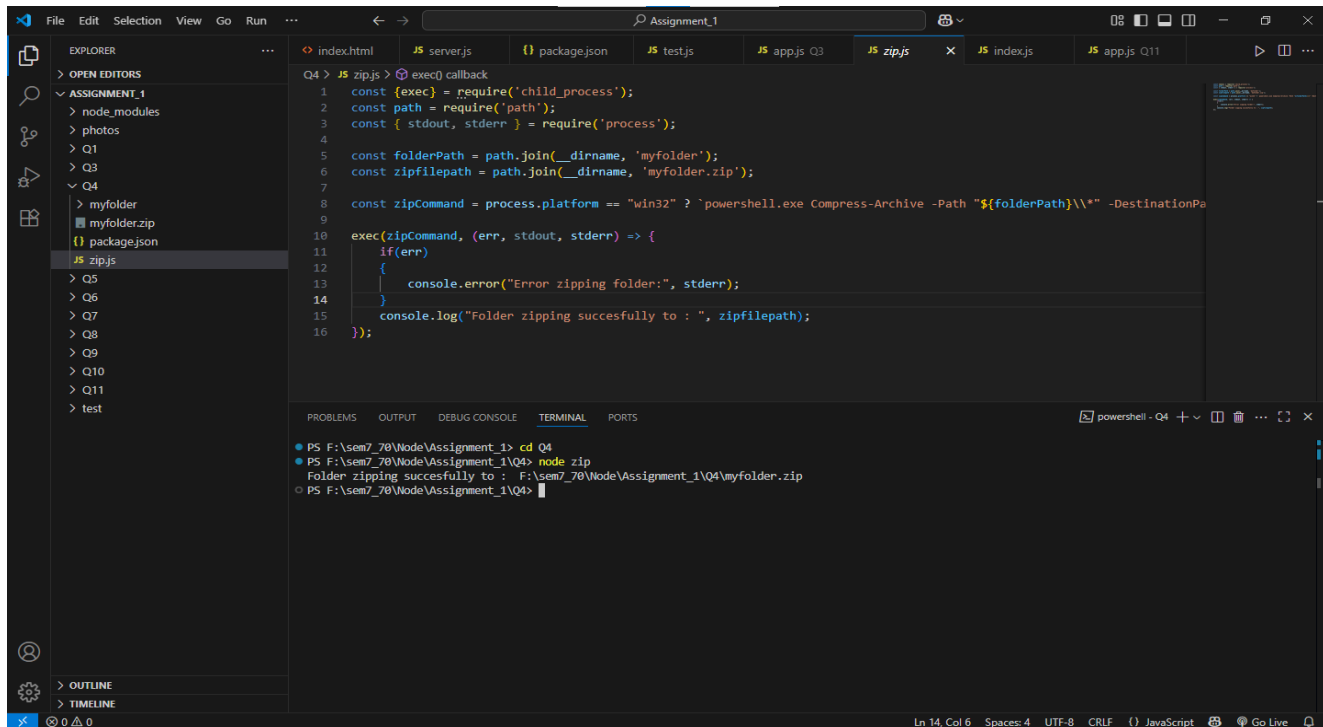


```
Q3 > JS app.js > @ |
1
2
3 const readline = require('readline');
4 const { getBotResponse } = require('./chatbotModule');
5
6 const rl = readline.createInterface({
7   input: process.stdin,
8   output: process.stdout
9 });
10
11 console.log("Welcome to our E-commerce Support Chatbot!");
12 console.log("Type 'exit' to quit.");
13
14 function askQuestion() {
15   rl.question("Enter your question: ", (answer) => {
16     const response = getBotResponse(answer);
17     console.log(response);
18     askQuestion();
19   });
20 }
21 askQuestion();
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\sem7_70\Node\Assignment_1> cd Q3
PS F:\sem7_70\Node\Assignment_1\Q3> node app
Welcome to our E-commerce Support Chatbot!
Type 'exit' to quit.
You: hi
Bot: Greetings! How may I be of service?
You: status
Bot: To check your order status, please provide your order number.
You: details
Bot: Tell me which product you'd like to know more about.
You: help
Bot: For direct assistance, please call our customer service hotline or send us an email.
You: 
```

Q4.

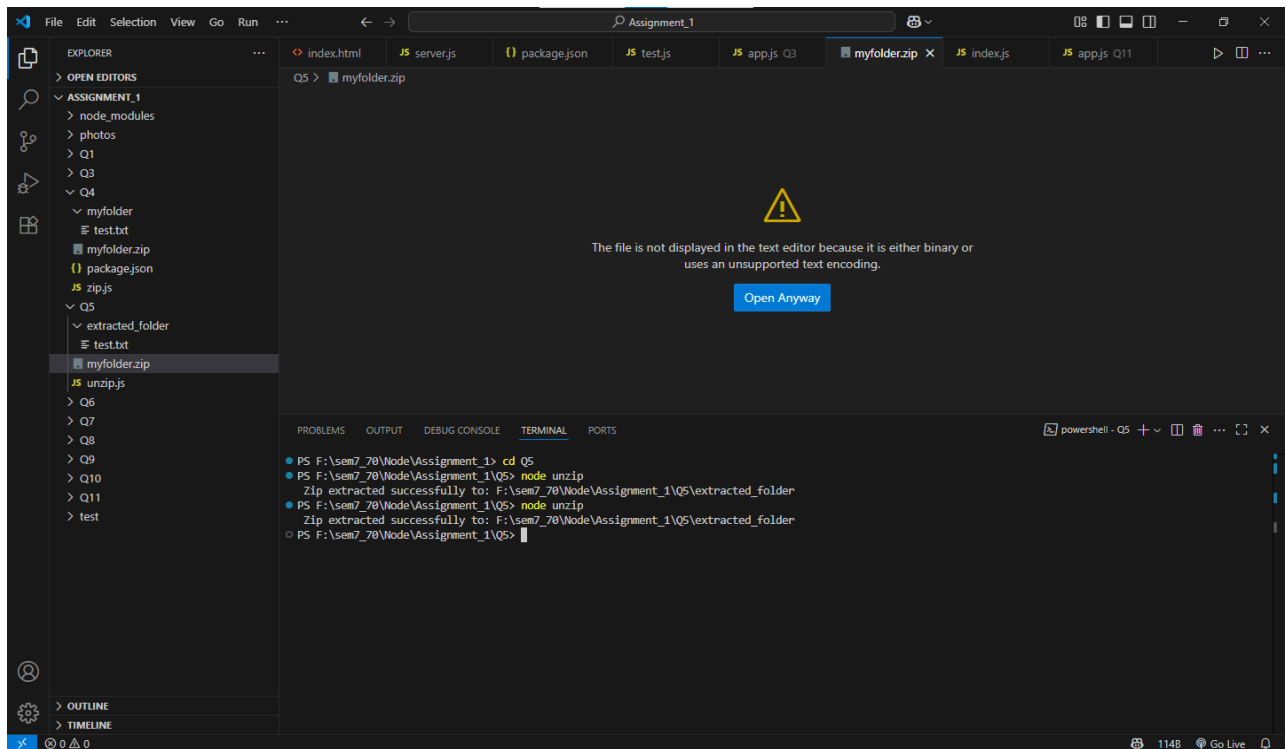


```
Q4 > JS zip.js > exec() callback
1 const {exec} = require('child_process');
2 const path = require('path');
3 const { stdout, stderr } = require('process');
4
5 const folderPath = path.join(__dirname, 'myfolder');
6 const zipfilePath = path.join(__dirname, 'myfolder.zip');
7
8 const zipCommand = process.platform == "win32" ? `powershell.exe Compress-Archive -Path "${folderPath}\\*" -DestinationPa
9
10 exec(zipCommand, (err, stdout, stderr) => {
11   if(err)
12     {
13       console.error("Error zipping folder:", stderr);
14     }
15   console.log("Folder zipping succesfully to : ", zipfilePath);
16 });
```

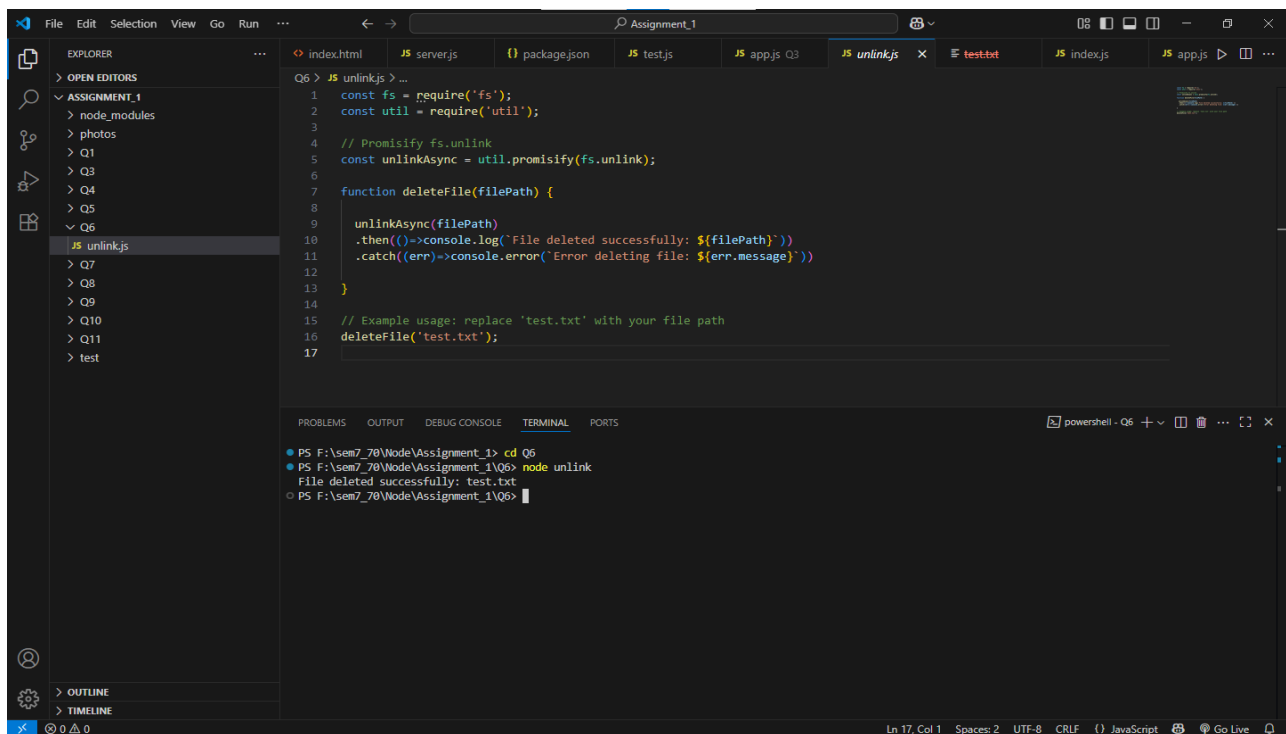
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\sem7_70\Node\Assignment_1> cd Q4
PS F:\sem7_70\Node\Assignment_1\Q4> node zip
Folder zipping succesfully to : F:\sem7_70\Node\Assignment_1\Q4\myfolder.zip
PS F:\sem7_70\Node\Assignment_1\Q4> 
```

Q5.



Q6.



Q7.

```
Q7 > JS fetchjs > fetchGoogle
1  const fetch = require('node-fetch');
2
3  async function fetchGoogle() {
4    try {
5      const response = await fetch('https://www.google.com');
6      if (!response.ok) {
7        throw new Error('HTTP error! status: ${response.status}');
8      }
9      const body = await response.text();
10     console.log('Fetched Google homepage HTML length:', body.length);
11
12     console.log(body);
13   } catch (error) {
14     console.error('Fetch error:', error);
15   }
16 }
17
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS F:\sem7\_70\Node\Assignment\_1\Q7>

Q8.

```
Q8 > {} package.json > {} dependencies > express
1  {
2    "name": "q8",
3    "version": "1.0.0",
4    "main": "server.js",
5    "scripts": {
6      "test": "echo \\\"Error: no test specified\\\" && exit 1",
7      "start": "node server.js",
8      "build": "echo \\\"build test\\\""}
9  },
10  "keywords": [],
11  "author": "",
12  "license": "ISC",
13  "description": "",
14  "dependencies": {
15    "express": "^5.1.0"
16  }
17 }
18
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS F:\sem7\_70\Node\Assignment\_1\Q8> cd Q8  
PS F:\sem7\_70\Node\Assignment\_1\Q8> npm run build

> q8@1.0.0 build  
> echo "build test"

"build test"

PS F:\sem7\_70\Node\Assignment\_1\Q8> npm run test

> q8@1.0.0 test  
> echo "Error: no test specified" && exit 1

"Error: no test specified"

PS F:\sem7\_70\Node\Assignment\_1\Q8>

Q9.

```
Q9 > JS file.js > ...
1  const fs = require('fs');
2  const path = require('path');
3
4  const filePath = path.join(__dirname, 'demo.txt');
5
6  fs.writeFile(filePath, 'Hello, Node.js!\n', (err) => {
7    if (err) return console.error('Error writing file:', err);
8    console.log(' File created and written to.');
```

```
PS F:\sem7_70\Node\Assignment_1> cd Q9
PS F:\sem7_70\Node\Assignment_1\Q9> node file
File created and written to.
Text appended.
File content:
Hello, Node.js!
This is appended text.

File renamed.
File exists after renaming.
PS F:\sem7_70\Node\Assignment_1\Q9>
```

Q10.

```
Q10 > JS globalVariable.js > ...
1
2  console.log('Current directory (__dirname):', __dirname);
3  console.log('Current file (__filename):', __filename);
4
5  global.myGlobalValue = 'Hello from the global object!';
6  console.log('Global value:', global.myGlobalValue);
7
8
9  setTimeout(() => {
10   console.log('setTimeout called after 1 second');
11 }, 1000);
12
13
14 let count = 0;
15 const intervalId = setInterval(() => {
16   count++;
17   console.log(`setInterval: count = ${count}`);
18   if (count === 3) {
19     clearInterval(intervalId);
```

```
PS F:\sem7_70\Node\Assignment_1> cd Q10
PS F:\sem7_70\Node\Assignment_1\Q10> node globalVariable
Current directory (__dirname): F:\sem7_70\Node\Assignment_1\Q10
Current file (__filename): F:\sem7_70\Node\Assignment_1\Q10\globalVariable.js
Global value: Hello from the global object!
Process PID: 15808
Platform: win32
Node version: v20.18.0
NODE_ENV: not set
setTimeout called after 1 second
setInterval: count = 1
setInterval: count = 2
setInterval: count = 3
Interval cleared
PS F:\sem7_70\Node\Assignment_1\Q10>
```

Q11.

Node.js AJAX Example

@jay-pansuriya/matrix.js - npm

+

←https://www.npmjs.com/package/@jay-pansuriya/matrix.js

ProTeamsPricingDocumentation

npm

Search packages

Search

@jay-pansuriya/matrix.js

1.0.0 • Public • Published a few seconds ago

Readme

CodeBeta

0 Dependencies

0 Dependents

1 Versions

This package does not have a README. [Add a README](#) to your package so that users know how to get started.

Keywords

none

Install

> npm i @jay-pansuriya/matrix.js

Version	License
1.0.0	ISC
Unpacked Size	Total Files
7.01 kB	3
Last publish	a few seconds ago
Collaborators	

jay-pansuriya

Profile

Packages

Account

Billing Info

Access Tokens

+ Add Organization

Sign Out