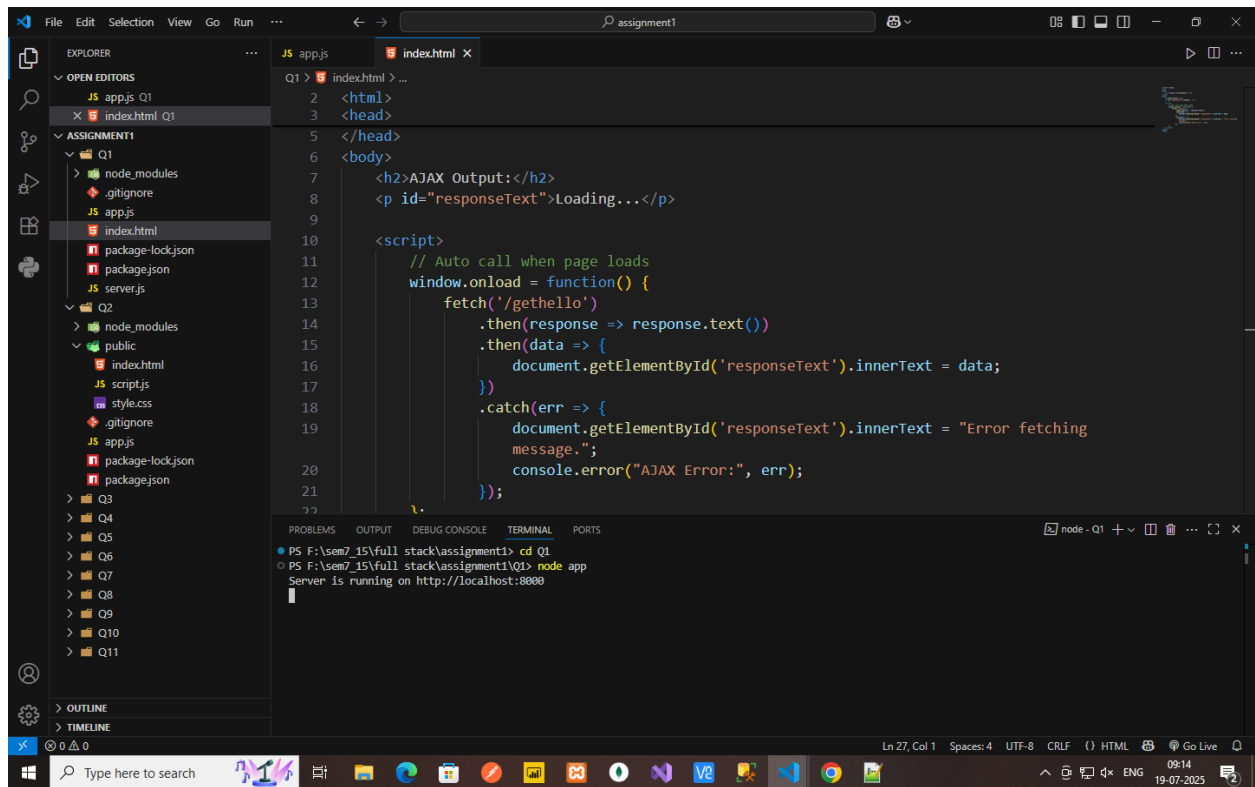


Q1)



The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left displays the file structure of a project named 'assignment1'. The 'Q1' folder is expanded, showing files like 'index.html', 'package-lock.json', 'package.json', and 'server.js'. The main editor area shows the 'index.html' file with the following content:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4
5 </head>
6 <body>
7   <h2>AJAX Output:</h2>
8   <p id="responseText">Loading...</p>
9
10  <script>
11    // Auto call when page loads
12    window.onload = function() {
13      fetch('/gethello')
14        .then(response => response.text())
15        .then(data => {
16          document.getElementById('responseText').innerText = data;
17        })
18        .catch(err => {
19          document.getElementById('responseText').innerText = "Error fetching
20            message.";
21          console.error("AJAX Error:", err);
22        });
23    };
24  </script>
25 </body>
26 </html>
```

Below the editor, the TERMINAL panel is open, showing the command prompt output:

```
PS F:\sem7_15\full stack\assignment1> cd Q1
PS F:\sem7_15\full stack\assignment1\Q1> node app
Server is running on http://localhost:8000
```



AJAX Output:

Hello NodeJS!!



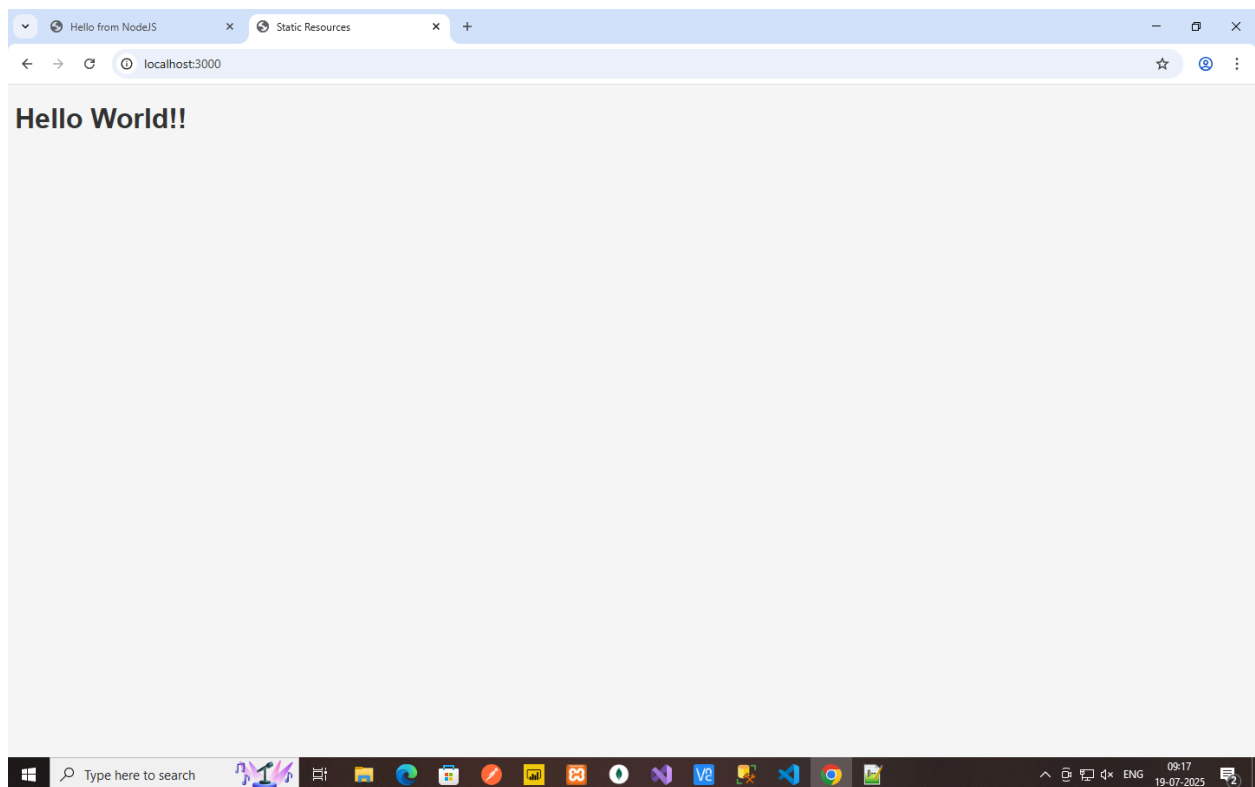
Q2)

```
Q2 > JS app.js > ...
1  const express = require('express');
2  const path = require('path');
3  const app = express();
4  const PORT = 3000;
5
6  // Serve static files from the 'public' directory
7  app.use(express.static(path.join(__dirname, 'public')));
8
9  // Start the server
10 app.listen(PORT, () => {
11   console.log(`Server is running at http://localhost:${PORT}`);
12 });
13
```

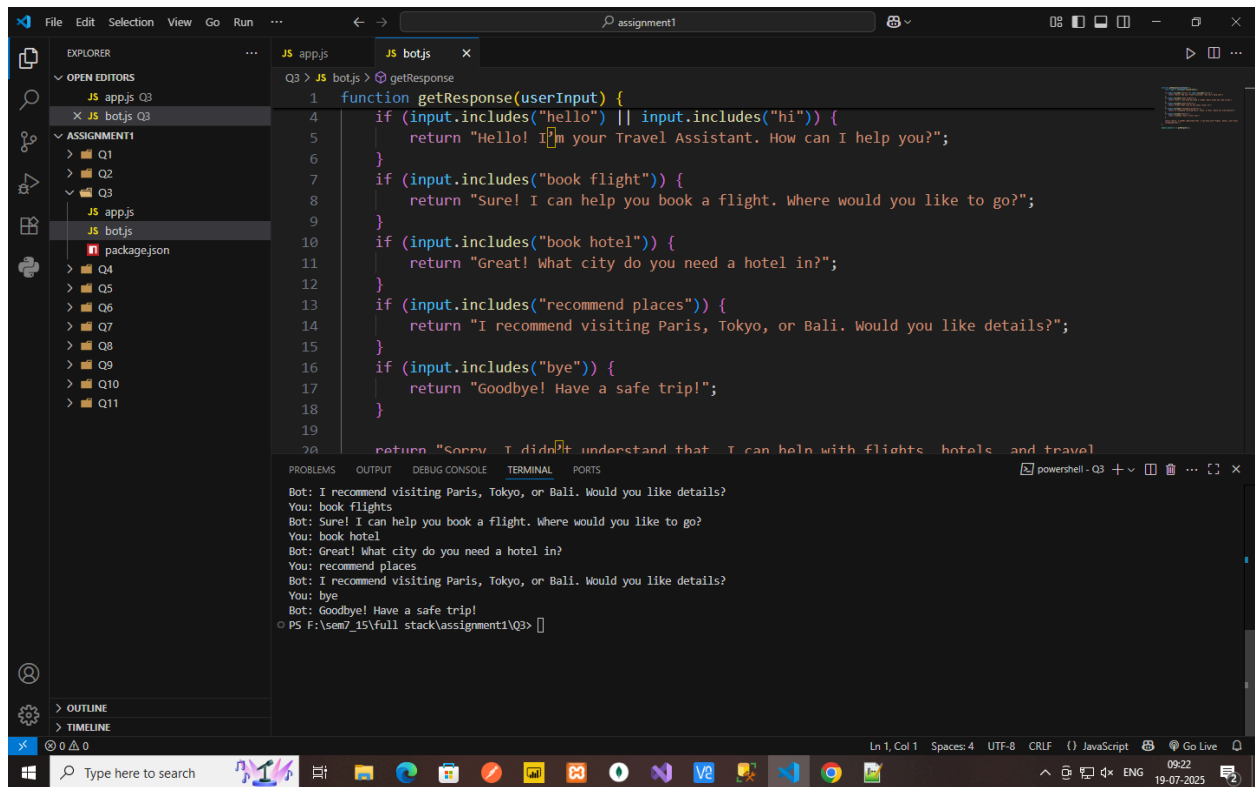
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS F:\sem7_15\full stack\assignment1> cd Q1
PS F:\sem7_15\full stack\assignment1\Q1> node app
Server is running on http://localhost:8080
PS F:\sem7_15\full stack\assignment1\Q1> cd ..
PS F:\sem7_15\full stack\assignment1> cd Q2
PS F:\sem7_15\full stack\assignment1\Q2> npm app
Unknown command: "app"

To see a list of supported npm commands, run:
npm help
PS F:\sem7_15\full stack\assignment1\Q2> node app
Server is running at http://localhost:3000
```

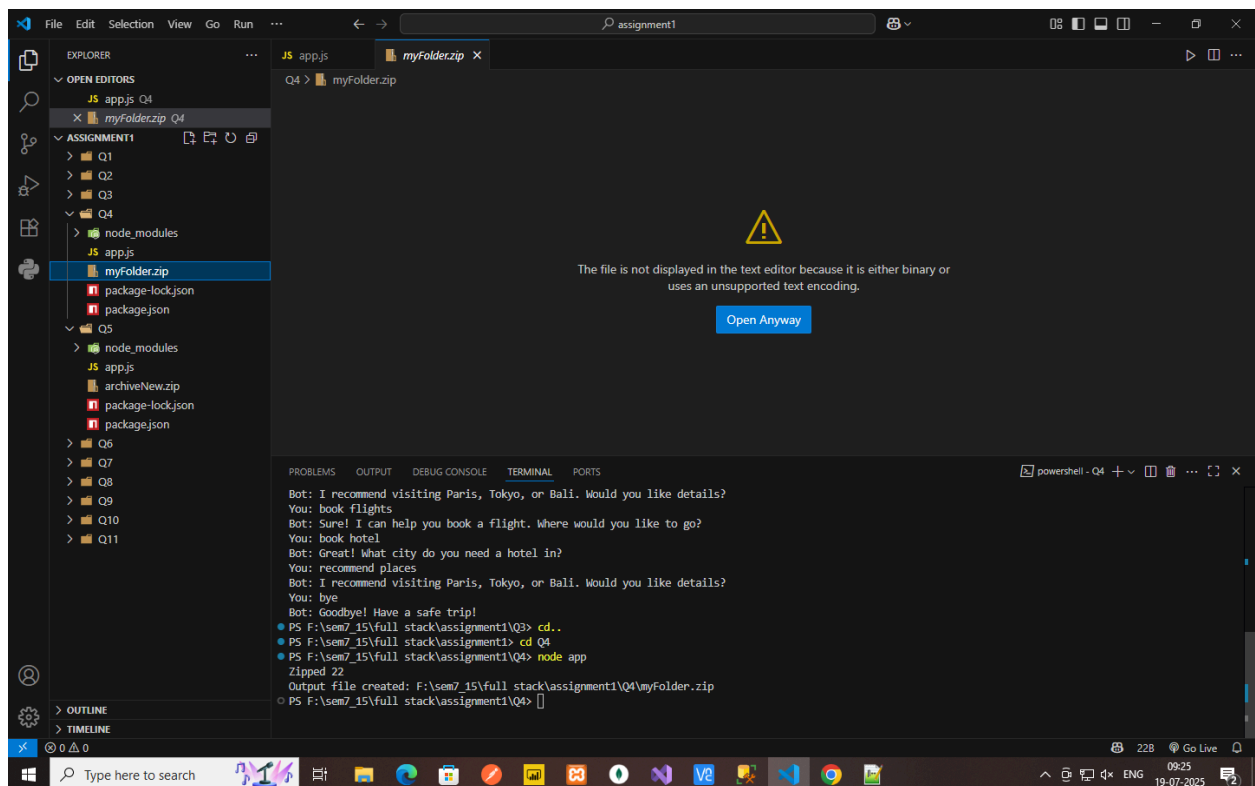


Q3)



```
Q3 > JS botjs > getResponse
1 function getResponse(userInput) {
2   if (input.includes("hello") || input.includes("hi")) {
3     return "Hello! I'm your Travel Assistant. How can I help you?";
4   }
5   if (input.includes("book flight")) {
6     return "Sure! I can help you book a flight. Where would you like to go?";
7   }
8   if (input.includes("book hotel")) {
9     return "Great! What city do you need a hotel in?";
10  }
11  if (input.includes("recommend places")) {
12    return "I recommend visiting Paris, Tokyo, or Bali. Would you like details?";
13  }
14  if (input.includes("bye")) {
15    return "Goodbye! Have a safe trip!";
16  }
17  return "Sorry, I didn't understand that. I can help with flights, hotels, and travel."
18 }
19
20 Bot: I recommend visiting Paris, Tokyo, or Bali. Would you like details?
You: book flights
Bot: Sure! I can help you book a flight. Where would you like to go?
You: book hotel
Bot: Great! What city do you need a hotel in?
You: recommend places
Bot: I recommend visiting Paris, Tokyo, or Bali. Would you like details?
You: bye
Bot: Goodbye! Have a safe trip!
PS F:\sem7_15\full stack\assignment1\Q3>
```

Q4)



```
Q4 > myFolder.zip
The file is not displayed in the text editor because it is either binary or
uses an unsupported text encoding.
Open Anyway

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Bot: I recommend visiting Paris, Tokyo, or Bali. Would you like details?
You: book flights
Bot: Sure! I can help you book a flight. Where would you like to go?
You: book hotel
Bot: Great! What city do you need a hotel in?
You: recommend places
Bot: I recommend visiting Paris, Tokyo, or Bali. Would you like details?
You: bye
Bot: Goodbye! Have a safe trip!
PS F:\sem7_15\full stack\assignment1\Q3> cd..
PS F:\sem7_15\full stack\assignment1> cd Q4
PS F:\sem7_15\full stack\assignment1\Q4> node app
Zipped 22
Output file created: F:\sem7_15\full stack\assignment1\Q4\myFolder.zip
PS F:\sem7_15\full stack\assignment1\Q4>
```

Q5)

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a project structure with folders Q1 through Q11. The main editor area shows a file named 'Weather_Data_1980_2024(hourly).csv' with the following content:

```
1 time,temperature,relative_humidity,dew_point,precipitation (mm),rain (mm),snowfall (cm),pres
2 1980-01-01T00:00,12.7,83,10,0,0,0,1012.8,945.1,1,1,0,0,0.25,7.5,235,0
3 1980-01-01T01:00,12.9,82,9.9,0,0,0,1012.2,944.5,4,4,0,0,0.26,7.9,231,0
4 1980-01-01T02:00,13.2,82,10.2,0,0,0,1012.3,944.7,13,14,0,0,0.27,7.5,235,1
5 1980-01-01T03:00,15.9,78,12,0,0,0,1013.5,946.4,23,26,0,0,0.4,6.6,248,1
6 1980-01-01T04:00,19.4,67,13.1,0,0,0,1014.6,948.2,9,10,0,0,0.75,8.7,265,1
7 1980-01-01T05:00,21.7,56,12.4,0,0,0,1014.9,949.5,6,0,0,1.15,10.4,272,1
8 1980-01-01T06:00,23.5,46,11.4,0,0,0,1014.6,949.1,3,3,0,0,1.55,10.8,274,1
9 1980-01-01T07:00,24.48,12.5,0,0,0,1013.4,948.1,5,5,1,0,1.54,11.2,274,1
10 1980-01-01T08:00,24.8,45,12,0,0,0,1012.3,947.2,2,1,1,0,1.73,11.9,275,1
11 1980-01-01T09:00,25.1,44,12,0,0,0,1011.2,946.3,5,6,0,0,1.78,12.3,277,1
12 1980-01-01T10:00,24.9,45,12.1,0,0,0,1010.6,945.7,8,9,0,0,1.74,14.3,282,1
13 1980-01-01T11:00,24.3,47,12.3,0,0,0,1011.945.9,22,25,0,0,1.61,14.2,279,1
14 1980-01-01T12:00,23.3,52,12.8,0,0,0,1011.2,945.9,30,23,15,0,1.38,11.3,279,1
15 1980-01-01T13:00,21.6,60,13.4,0,0,0,1011.8,946.1,82,56,52,0,1.04,8.5,282,0
16 1980-01-01T14:00,20.4,68,14.2,0,0,0,1012.1,946.1,100,83,73,0,0.78,8.5,282,0
17 1980-01-01T15:00,19.8,74,14.9,0,0,0,1013.2,947,100,98,93,0,0.61,8.8,279,0
18 1980-01-01T16:00,19.4,79,15.6,0,0,0,1013.6,947.3,100,100,97,0,0.48,7.2,273,0
19 1980-01-01T17:00,19.83,16,0,0,0,1013.6,947.2,100,98,67,0,0.38,5.8,266,0
20 1980-01-01T18:00,18.5,87,16.3,0,0,0,1013.4,946.9,86,76,0,0,0.28,4.2,250,0
```

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

```
PS F:\sem7_15\full stack\assignment1\Q3> cd ..
PS F:\sem7_15\full stack\assignment1> cd Q4
PS F:\sem7_15\full stack\assignment1\Q4> node app
Zipped 22
Output file created: F:\sem7_15\full stack\assignment1\Q4\myFolder.zip
PS F:\sem7_15\full stack\assignment1\Q4> cd ..
PS F:\sem7_15\full stack\assignment1\Q5> cd Q5
PS F:\sem7_15\full stack\assignment1\Q5> node app
Extraction complete to: F:\sem7_15\full stack\assignment1\Q5\archiveNewExtracted
PS F:\sem7_15\full stack\assignment1\Q5>
```

Q6)

The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a project structure with folders Q1 through Q11. The main editor area shows a file named 'sample.txt' with the following content:

```
1 This is a sample file!
```

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

```
PS F:\sem7_15\full stack\assignment1\Q6> node app
Error: Cannot find module 'F:\sem7_15\full stack\assignment1\Q6\Q6'
at Module.resolveFilename (node:internal/modules/cjs/loader:1077:15)
at Module._load (node:internal/modules/cjs/loader:922:27)
at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:86:12)
at node:internal/main/run_main_module:23:47 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}
Node.js v18.18.2
PS F:\sem7_15\full stack\assignment1\Q6> node app
File deleted
PS F:\sem7_15\full stack\assignment1\Q6>
```

Q7)

The screenshot shows the VS Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure with folders Q1 through Q11 and a file sample.txt. The main editor window displays the code for Q7, which is an async function named fetchInfo(). The function uses dynamic import to fetch the node-fetch module and then uses the fetch API to retrieve the HTML content of the Google homepage. The output of the function is logged to the console, showing the first 500 characters of the HTML. The terminal window shows the command to run the app, which successfully fetches the page content.

```
Q7 > JS app.js > ...
1 // Async function to fetch Google page using dynamic import
2 async function fetchInfo() {
3   try {
4     // Dynamically import node-fetch (ESM in CommonJS)
5     const fetch = (await import('node-fetch')).default;
6
7     const response = await fetch('https://www.google.com');
8     const html = await response.text(); // get HTML as string
9     console.log("Fetched!:\n");
10    console.log(html.substring(0, 500)); // print first 500 characters
11  } catch (error) {
12    console.error('Error fetching page:', error.message);
13  }
14 }
15
16 fetchInfo();
17
```

Node.js v18.18.2
PS F:\sem7_15\full stack\assignment1\Q6> node app
File deleted
PS F:\sem7_15\full stack\assignment1\Q6> cd..
PS F:\sem7_15\full stack\assignment1> cd Q7
PS F:\sem7_15\full stack\assignment1\Q7> node app
Fetched!:
<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/google/1x/googleleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="RPyeZVoykp0dC158ojF1-A">(fu
nction(){var _g={kEI:'9xh7aPb8EN2w50UP3V6coAE',kEXPI:'0,202792,62,2,3497403,681,444,538661,14111,64782,368901,226411,2,63631,11106,5238575,5992878,38828565,2
5279871,5,26822,60251,14110,2
PS F:\sem7_15\full stack\assignment1\Q7> |

Q8)

The screenshot shows the VS Code editor with a file explorer on the left and a terminal at the bottom. The file explorer shows a project structure with folders Q1 through Q11 and a file sample.txt. The main editor window displays the code for Q8, which is a package.json file. The package.json file defines the project name as 'q8', version as '1.0.0', and includes a main script 'app.js'. It also defines scripts for start, test, and start-dev, and includes keywords, author, and license information. The terminal window shows the command to run the app, which successfully starts the nodemon process and runs the app.js file.

```
Q8 > package.json > ...
1 {
2   "name": "q8",
3   "version": "1.0.0",
4   "description": "Custom Script Project",
5   "main": "app.js",
6   "scripts": {
7     "start": "node app.js",
8     "test": "node test.js",
9     "hello": "echo 'Hello this is a custom Script'",
10    "start-dev": "nodemon app.js"
11  },
12  "keywords": [],
13  "author": "",
14  "license": "ISC"
15 }
16
```

<meta content="/images/branding/google/1x/googleleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="RPyeZVoykp0dC158ojF1-A">(fu
nction(){var _g={kEI:'9xh7aPb8EN2w50UP3V6coAE',kEXPI:'0,202792,62,2,3497403,681,444,538661,14111,64782,368901,226411,2,63631,11106,5238575,5992878,38828565,2
5279871,5,26822,60251,14110,2
PS F:\sem7_15\full stack\assignment1\Q7> cd..
PS F:\sem7_15\full stack\assignment1> cd Q8
PS F:\sem7_15\full stack\assignment1\Q8> nodemon app.js
[nodemon] 3.1.4
[nodemon] restarting at any time, enter 'rs'
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node app.js'
Server is running
[nodemon] clean exit - waiting for changes before restart
node app
PS F:\sem7_15\full stack\assignment1\Q8> node test
Tests passed!
PS F:\sem7_15\full stack\assignment1\Q8> |

Q9)

```
Q9 > JS app.js > ...
1 const fs = require('fs');
2
3 // 1. Create and write to a file
4 fs.writeFileSync('demo.txt', 'Hello, this is the first line.\n');
5 console.log('Created and written');
6
7 // 2. Append content to the file
8 fs.appendFileSync('demo.txt', 'This is an appended line.\n');
9 console.log('Content appended.');
```

```
PS F:\sem7_15\full stack\assignment1\Q8> cd..
PS F:\sem7_15\full stack\assignment1> cd Q9
PS F:\sem7_15\full stack\assignment1\Q9> node app
Created and written.
Content appended.
File content:
Hello, this is the first line.
This is an appended line.

File renamed to renamed.txt
renamed.txt exists.
Directory created.
File moved into "myfolder".
Contents of "myfolder": [ 'renamed.txt' ]
File deleted from "myfolder".
PS F:\sem7_15\full stack\assignment1\Q9>
```

Q10)


```
Q10 > JS app.js > ...
1 // __dirname and __filename
2 console.log("Directory name:", __dirname);
3 console.log("File name:", __filename);
4
5 // setTimeout
6 setTimeout(() => {
7   console.log("This message is printed after 2 seconds");
8 }, 2000);
9
10 // setInterval
11 let count = 0;
12 const interval = setInterval(() => {
13   console.log("Interval count:", ++count);
14   if (count === 3) clearInterval(interval); // Stop after 3 times
15 }, 1000);
16
17 // global object
```

```
Contents of "myfolder": [ 'renamed.txt' ]
File deleted from "myfolder".
PS F:\sem7_15\full stack\assignment1\Q9> cd..
PS F:\sem7_15\full stack\assignment1> cd q10
PS F:\sem7_15\full stack\assignment1\q10> node app
Directory name: F:\sem7_15\full stack\assignment1\q10
File name: F:\sem7_15\full stack\assignment1\q10\app.js
Global variable value: This is a global variable
Process information:
Node version: v18.18.2
Platform: win32
Process ID: 11584
Interval count: 1
This message is printed after 2 seconds
Interval count: 2
Interval count: 3
PS F:\sem7_15\full stack\assignment1\q10>
```

Q11)

cap-converter

1.0.1 • [Public](#) • Published 3 days ago

 [Readme](#)

 [Code](#) Beta

 0 Dependencies


 0 Dependents

name-to-uppercase

A simple package that converts any name (or string) to uppercase.

Installation

```
npm install cap-converter
```

```
C: > Users > ompan > Downloads > 15_SuchishreeyaDas_701_A1 > Q11 >  index.js > ...
```

```
1  function capconvert(name) {  
2    |    return name.toUpperCase();  
3  }  
4  
5  module.exports = {  
6    |    capconvert  
7  };  
8
```