

Name : Patel Sharthik H.

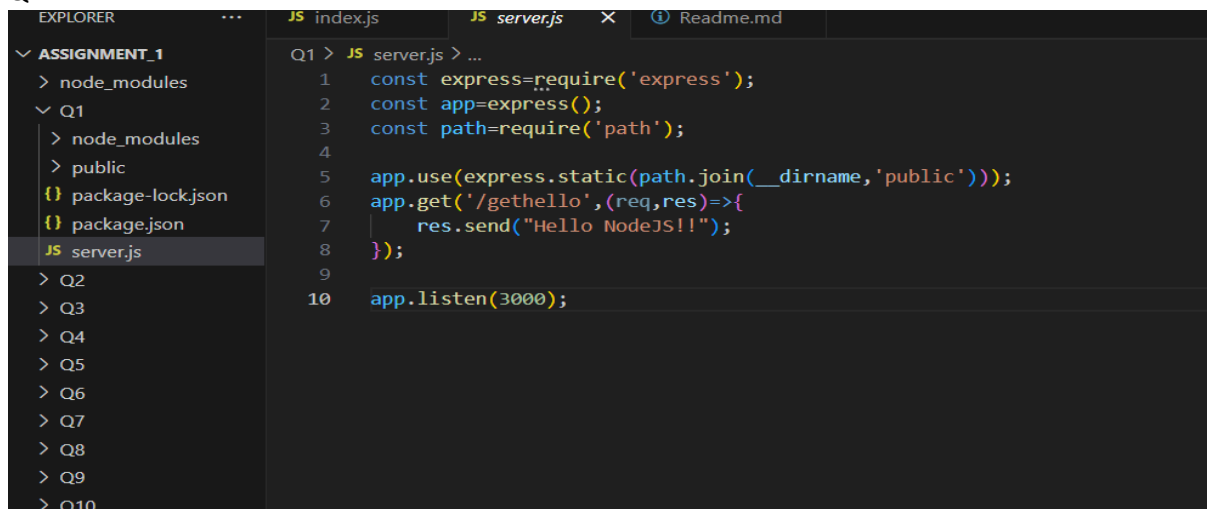
Roll No : 92

Division : b

Subject : Nodejs+

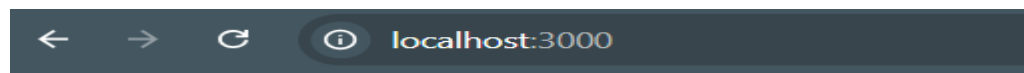
Assignment : Pratical – 1

Q1.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar is open, showing a file tree under 'ASSIGNMENT_1'. The tree includes 'node_modules', 'Q1' (expanded), 'node_modules', 'public', 'package-lock.json', 'package.json', and 'JS server.js' (selected). The main editor area displays the code for 'server.js'. The code is as follows:

```
Q1 > JS server.js > ...
1  const express=require('express');
2  const app=express();
3  const path=require('path');
4
5  app.use(express.static(path.join(__dirname,'public')));
6  app.get('/gethello',(req,res)=>{
7    res.send("Hello NodeJS!!");
8  });
9
10 app.listen(3000);
```



NodeJS AJAX Example

Get Hello



NodeJS AJAX Example

Get Hello

Hello NodeJS!!

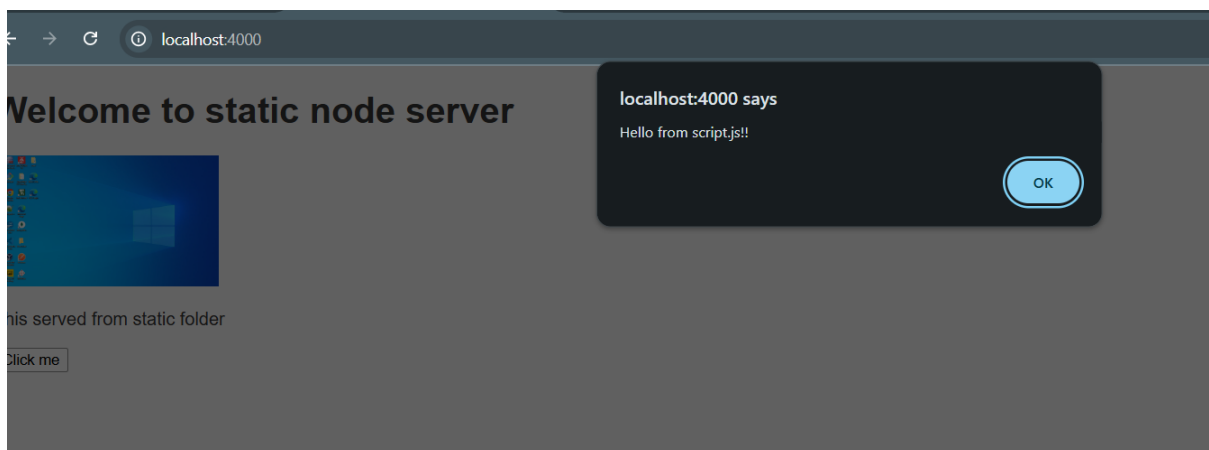
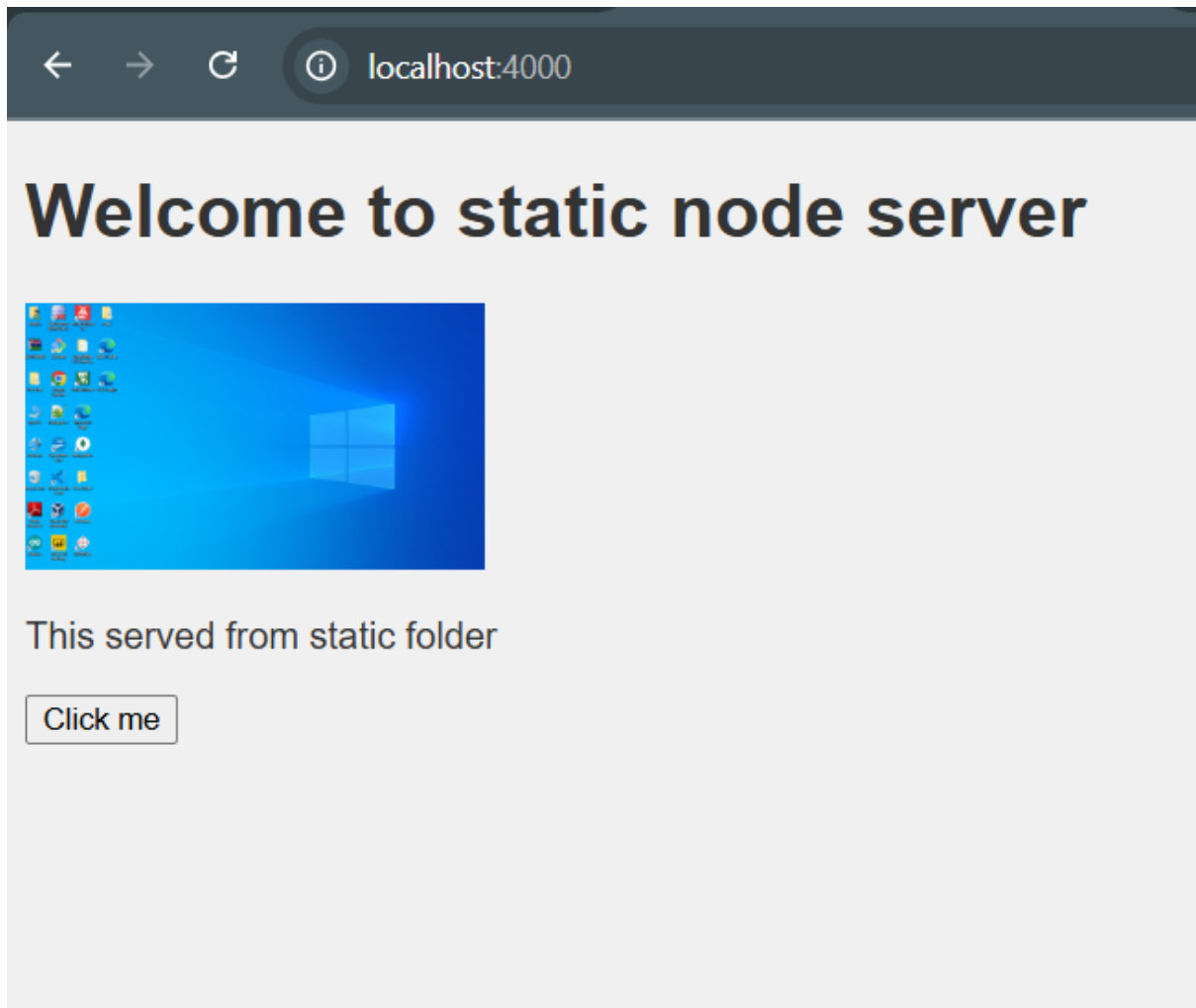
Q2

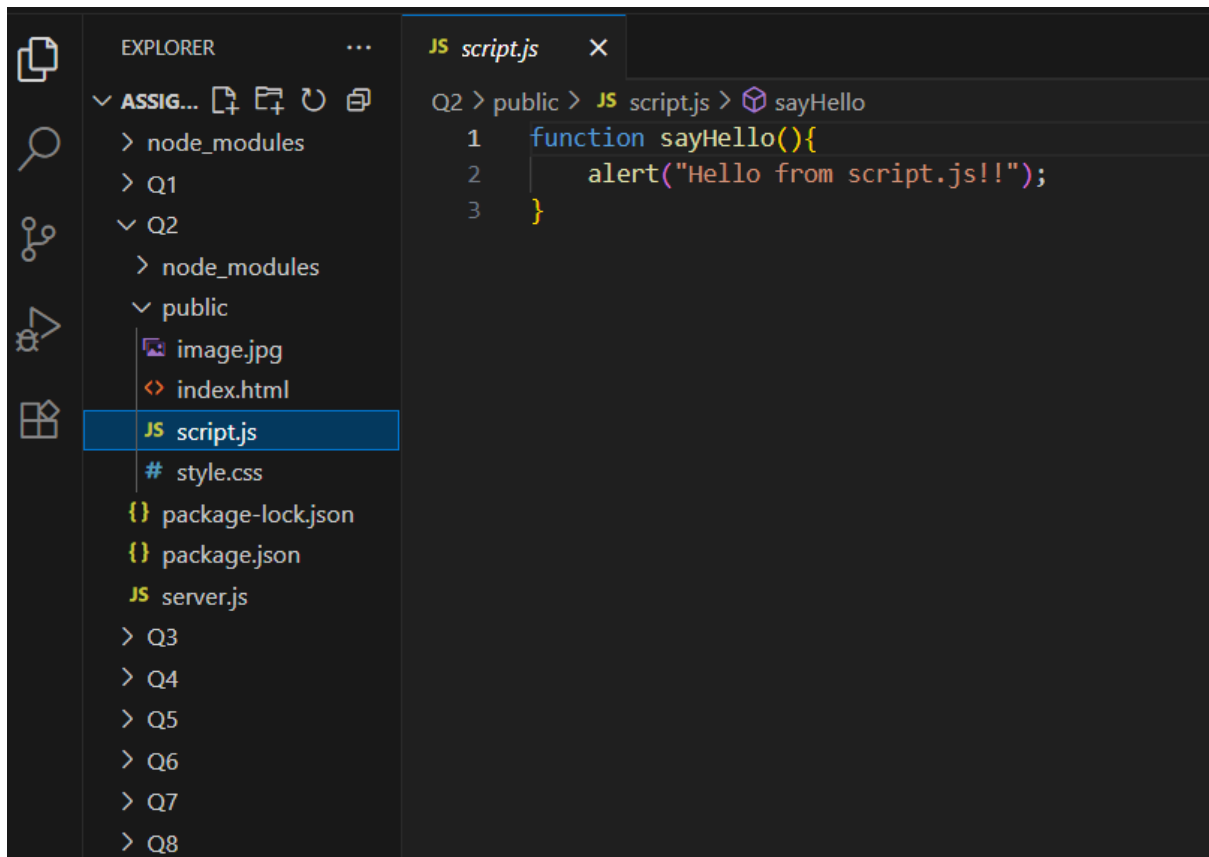
The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a project structure with folders 'node_modules' and 'public', and files 'package-lock.json' and 'package.json'. A file named 'server.js' is selected. The main editor area shows the content of 'server.js' with the following code:

```
1 const express=require('express');
2 const app=express();
3 const path=require('path');
4 app.use(express.static(path.join(__dirname,'public')));
5
6 app.listen(4000);
```

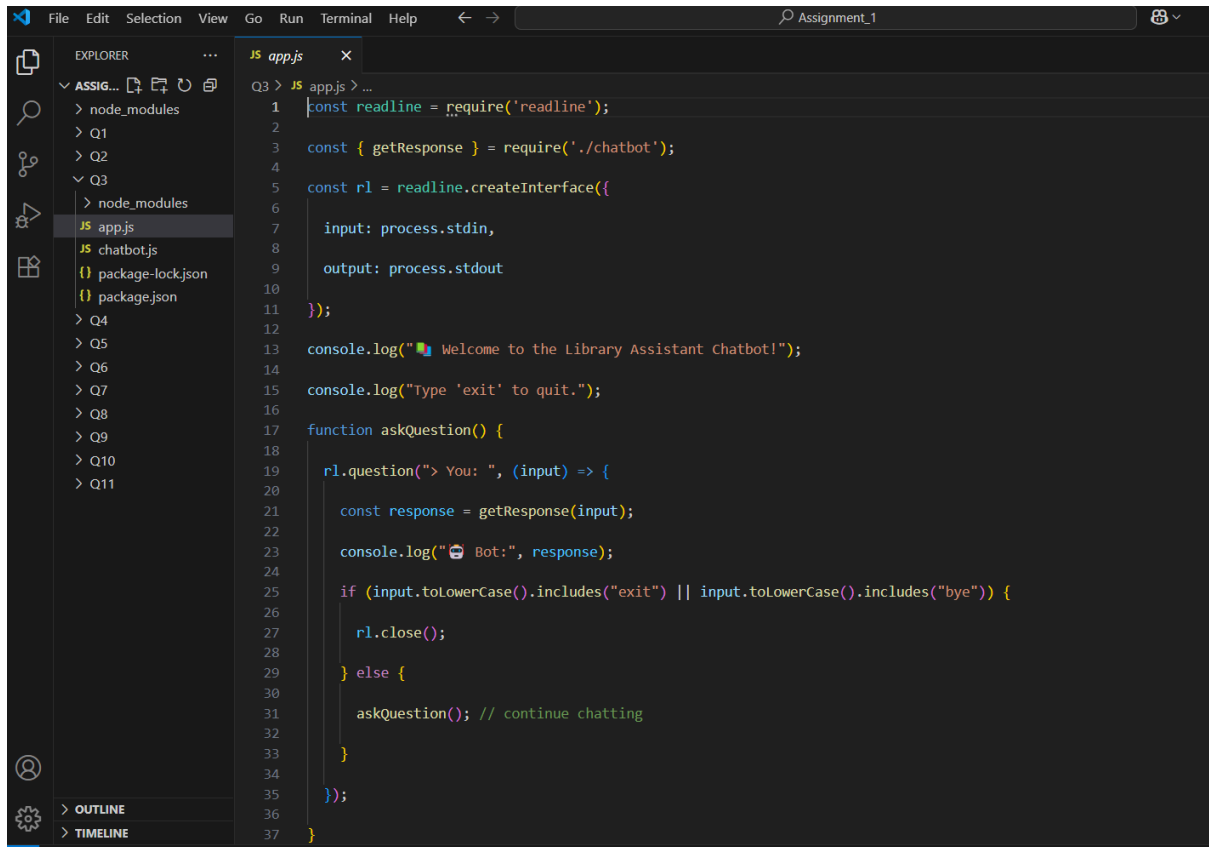
The bottom panel contains the TERMINAL view, which shows the following output:

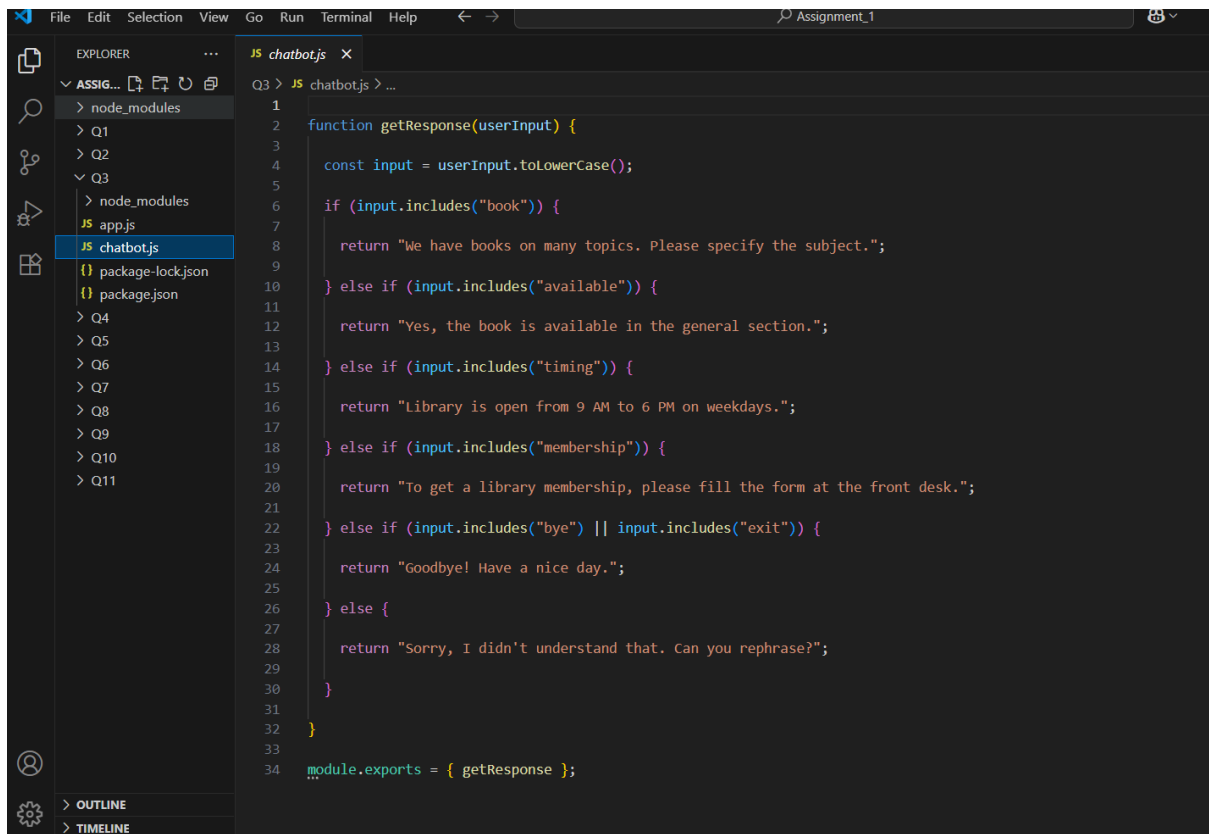
```
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
● PS D:\Hari\sem7\Assignment_1\Q1> cd ..
● PS D:\Hari\sem7\Assignment_1> cd Q2
○ PS D:\Hari\sem7\Assignment_1\Q2> nodemon server.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node server.js`
```





Q3



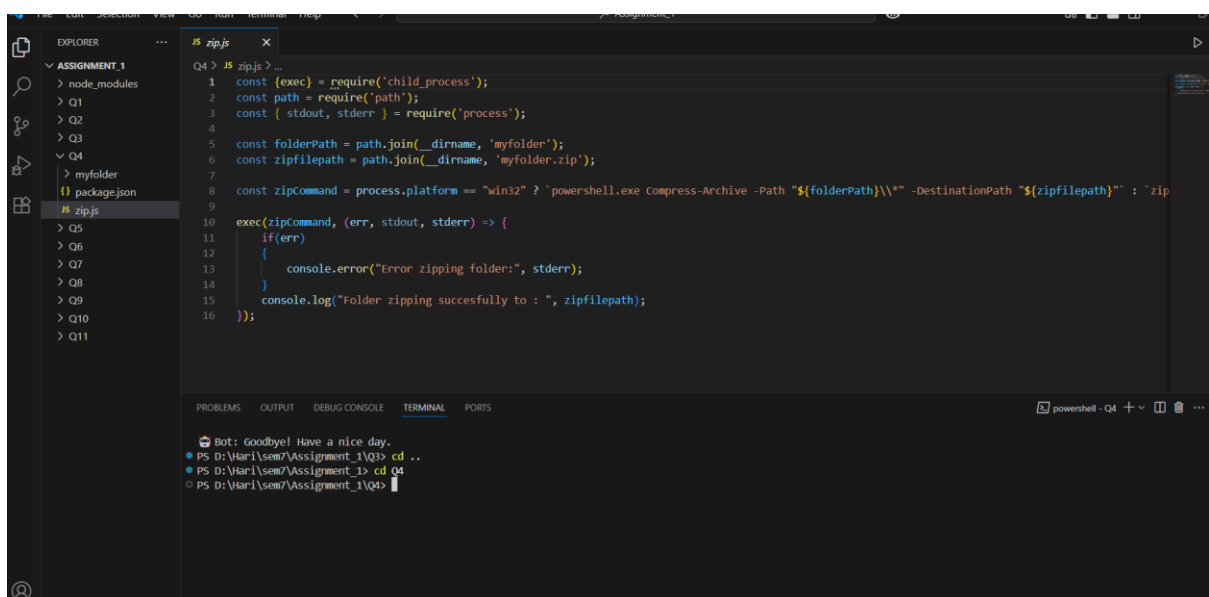


The image shows a screenshot of the Visual Studio Code (VS Code) editor interface. The top menu bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. The Explorer sidebar on the left shows a project structure with a folder named 'ASSIG...' containing subfolders 'node_modules', 'Q1', 'Q2', 'Q3', 'Q4', 'Q5', 'Q6', 'Q7', 'Q8', 'Q9', 'Q10', and 'Q11'. The 'Q3' folder is expanded, showing files 'node_modules', 'app.js', and 'chatbot.js'. The 'chatbot.js' file is selected and its content is displayed in the main editor window. The code is a JavaScript function 'getResponse' that takes 'userInput' as an argument and returns a response based on the input. The function uses 'toLowerCase()' and 'includes()' methods. The responses are: 'We have books on many topics. Please specify the subject.' for 'book', 'Yes, the book is available in the general section.' for 'available', 'Library is open from 9 AM to 6 PM on weekdays.' for 'timing', 'To get a library membership, please fill the form at the front desk.' for 'membership', 'Goodbye! Have a nice day.' for 'bye' or 'exit', and 'Sorry, I didn't understand that. Can you rephrase?' for any other input. The function is exported as 'module.exports'.

```
1
2 function getResponse(userInput) {
3
4     const input = userInput.toLowerCase();
5
6     if (input.includes("book")) {
7
8         return "We have books on many topics. Please specify the subject.";
9     }
10    } else if (input.includes("available")) {
11
12        return "Yes, the book is available in the general section.";
13    }
14    } else if (input.includes("timing")) {
15
16        return "Library is open from 9 AM to 6 PM on weekdays.";
17    }
18    } else if (input.includes("membership")) {
19
20        return "To get a library membership, please fill the form at the front desk.";
21    }
22    } else if (input.includes("bye") || input.includes("exit")) {
23
24        return "Goodbye! Have a nice day.";
25    }
26    } else {
27
28        return "Sorry, I didn't understand that. Can you rephrase?";
29    }
30 }
31
32
33
34 module.exports = { getResponse };
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS D:\Hari\sem7\Assignment_1> cd Q3
○ PS D:\Hari\sem7\Assignment_1\Q3> nodemon app.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node app.js`
👋 Welcome to the Library Assistant Chatbot!
Type 'exit' to quit.
> You: book
book
🤖 Bot: We have books on many topics. Please specify the subject.
> You: available
available
🤖 Bot: Yes, the book is available in the general section.
> You: timing
timing
timing
🤖 Bot: Library is open from 9 AM to 6 PM on weekdays.
> You: membership
membership
🤖 Bot: To get a library membership, please fill the form at the front desk.
membership
🤖 Bot: To get a library membership, please fill the form at the front desk.
🤖 Bot: To get a library membership, please fill the form at the front desk.
> You: bye
bye
🤖 Bot: Goodbye! Have a nice day.
█
```

Q4



The screenshot shows the Visual Studio Code editor with a file named `zip.js` open. The file explorer on the left shows the project structure: `ASSIGNMENT_1` containing `node_modules`, `Q1`, `Q2`, `Q3`, `Q4`, `myfolder`, and `package.json`. The `zip.js` file is selected in the editor. The code in `zip.js` is as follows:

```
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 "${folderPath}\\*" -DestinationPath "${zipfilePath}"` : `zip
9
10 exec(zipCommand, (err, stdout, stderr) => {
11   if(err)
12   {
13     console.error("Error zipping folder:", stderr);
14   }
15   console.log("Folder zipping successfully to : ", zipfilePath);
16 });
```

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

```
PS D:\Hari\sem7\Assignment_1\Q3> cd ..
PS D:\Hari\sem7\Assignment_1> cd Q4
PS D:\Hari\sem7\Assignment_1\Q4>
```

The terminal also shows the output of the `zip.js` script, which is "Folder zipping successfully to : ", zipfilePath".

```

🤖 Bot: Goodbye! Have a nice day.
PS D:\Hari\sem7\Assignment_1\Q3> cd ..
PS D:\Hari\sem7\Assignment_1> cd Q4
PS D:\Hari\sem7\Assignment_1\Q4> nodemon zip.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node zip.js`
Folder zipping succesfully to : D:\Hari\sem7\Assignment_1\Q4\myfolder.zip
[nodemon] clean exit - waiting for changes before restart

```

Q5

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar shows a project structure with folders Q1 through Q11. Q5 is expanded, showing 'extracted_folder' and 'myfolder.zip'. The file 'JS unzip.js' is selected. The main editor displays the code for 'unzip.js'. The terminal at the bottom shows the command history and the output of running 'nodemon unzip.js' in the Q5 directory, which successfully extracts the contents of 'myfolder.zip' into 'extracted_folder'.

```

// unzip.js
1
2 const { exec } = require('child_process');
3 const path = require('path');
4
5 // Path to your zip file and destination folder
6 const zipFilePath = path.join(__dirname, 'myfolder.zip');
7 const extractTo = path.join(__dirname, 'extracted_folder');
8
9 const unzipCommand = process.platform === 'win32'
10   ? `powershell.exe Expand-Archive -Path "${zipFilePath}" -DestinationPath "${extractTo}"`
11   : `unzip -o "${zipFilePath}" -d "${extractTo}"`;
12
13 exec(unzipCommand, (err, stdout, stderr) => {
14   if (err) {
15     console.error('Extraction error:', stderr);
16     return;
17   }
18   console.log('Zip extracted successfully to:', extractTo);
19 });
20

```

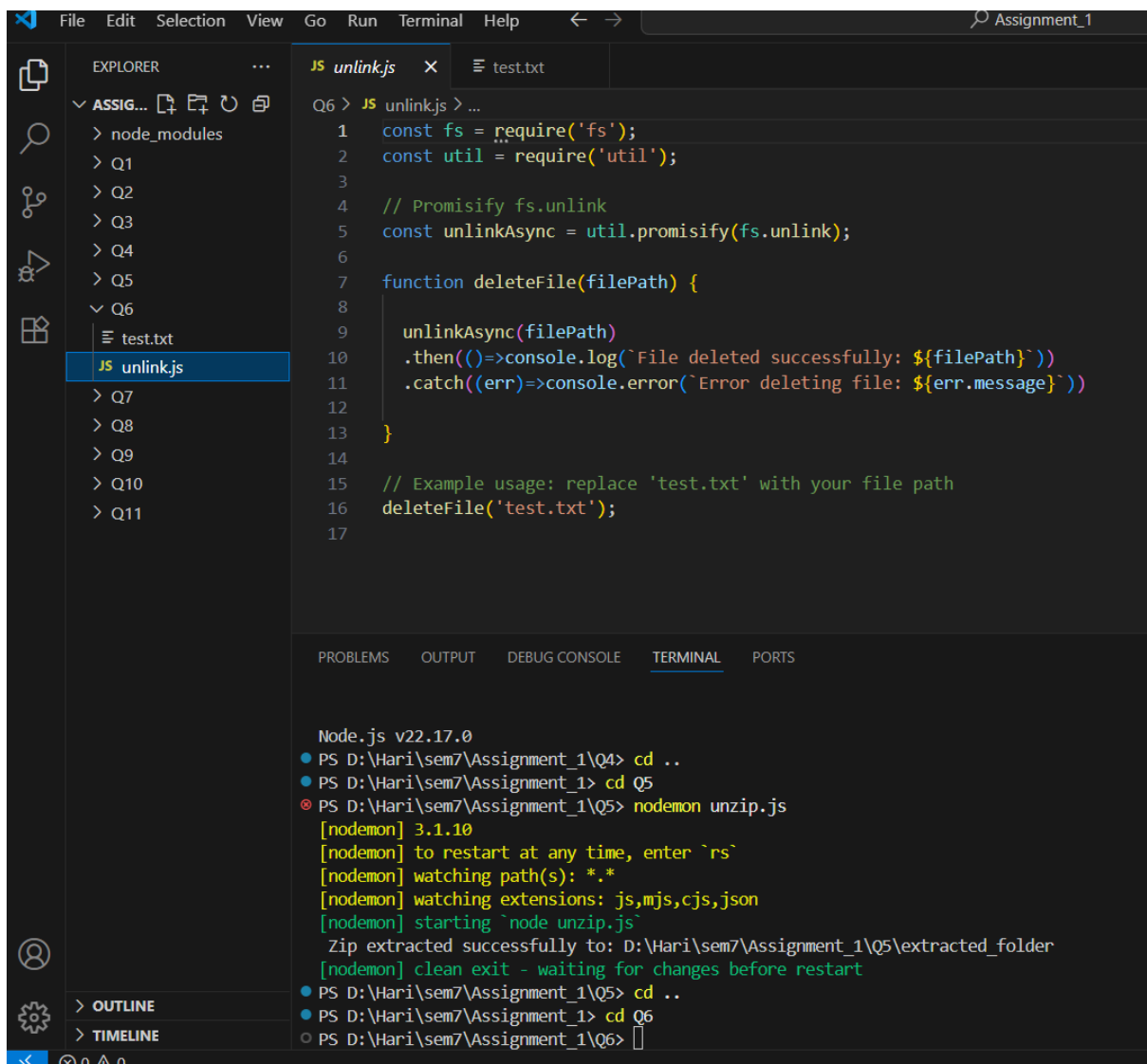
```

requireStack: []
}

Node.js v22.17.0
PS D:\Hari\sem7\Assignment_1\Q4> cd ..
PS D:\Hari\sem7\Assignment_1> cd Q5
PS D:\Hari\sem7\Assignment_1\Q5> nodemon unzip.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node unzip.js`
Zip extracted successfully to: D:\Hari\sem7\Assignment_1\Q5\extracted_folder
[nodemon] clean exit - waiting for changes before restart

```


Q6

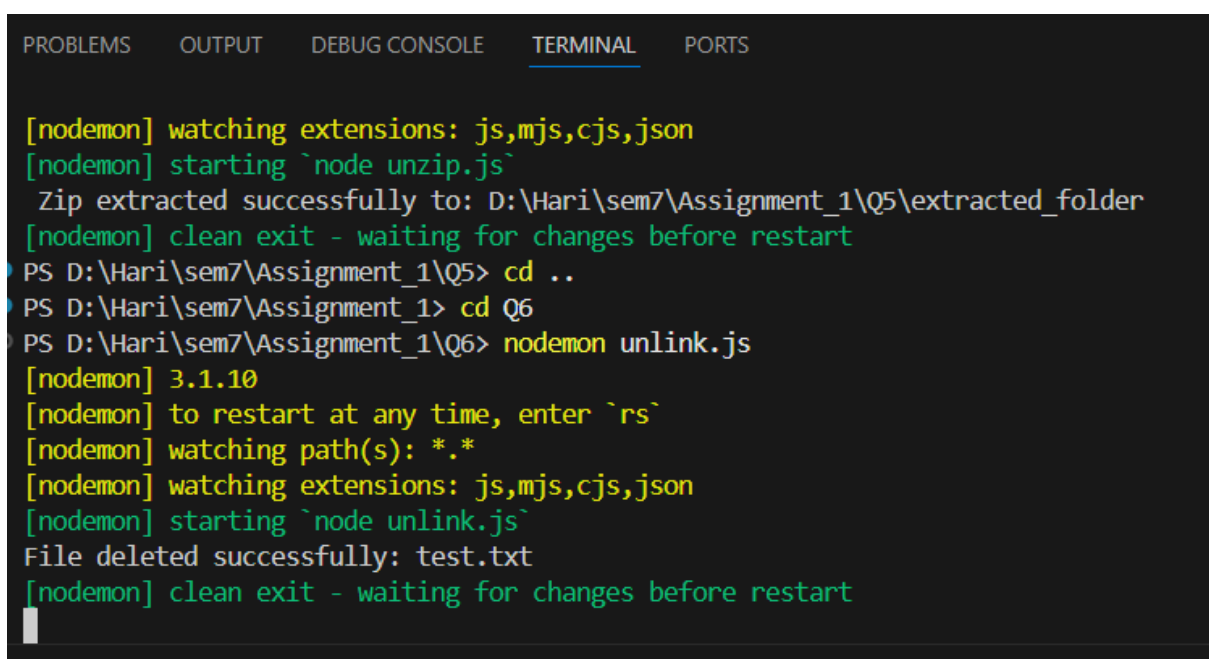


The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left shows a project structure with folders Q1 through Q11 under 'ASSIGNMENT_1'. The file 'test.txt' is selected. The main editor area shows the content of 'JS unlink.js'.

```
1 const fs = require('fs');
2 const util = require('util');
3
4 // Promisify fs.unlink
5 const unlinkAsync = util.promisify(fs.unlink);
6
7 function deleteFile(filePath) {
8
9     unlinkAsync(filePath)
10     .then(()=>console.log(`File deleted successfully: ${filePath}`))
11     .catch((err)=>console.error(`Error deleting file: ${err.message}`))
12 }
13
14
15 // Example usage: replace 'test.txt' with your file path
16 deleteFile('test.txt');
```

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

```
Node.js v22.17.0
PS D:\Hari\sem7\Assignment_1\Q4> cd ..
PS D:\Hari\sem7\Assignment_1> cd Q5
PS D:\Hari\sem7\Assignment_1\Q5> nodemon unzip.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node unzip.js`
Zip extracted successfully to: D:\Hari\sem7\Assignment_1\Q5\extracted_folder
[nodemon] clean exit - waiting for changes before restart
PS D:\Hari\sem7\Assignment_1\Q5> cd ..
PS D:\Hari\sem7\Assignment_1> cd Q6
PS D:\Hari\sem7\Assignment_1\Q6>
```



The terminal window continues with the following commands and output:

```
PS D:\Hari\sem7\Assignment_1\Q6> nodemon unlink.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node unlink.js`
File deleted successfully: test.txt
[nodemon] clean exit - waiting for changes before restart
```

Q7

```
EXPLORER  ... JS fetch.js x
ASSIGNMENT_1
  > node_modules
  > Q1
  > Q2
  > Q3
  > Q4
  > Q5
  > Q6
  > Q7
    > node_modules
    JS fetch.js
    {} package-lock.json
    {} package.json
  > Q8
  > Q9
  > Q10
  > Q11

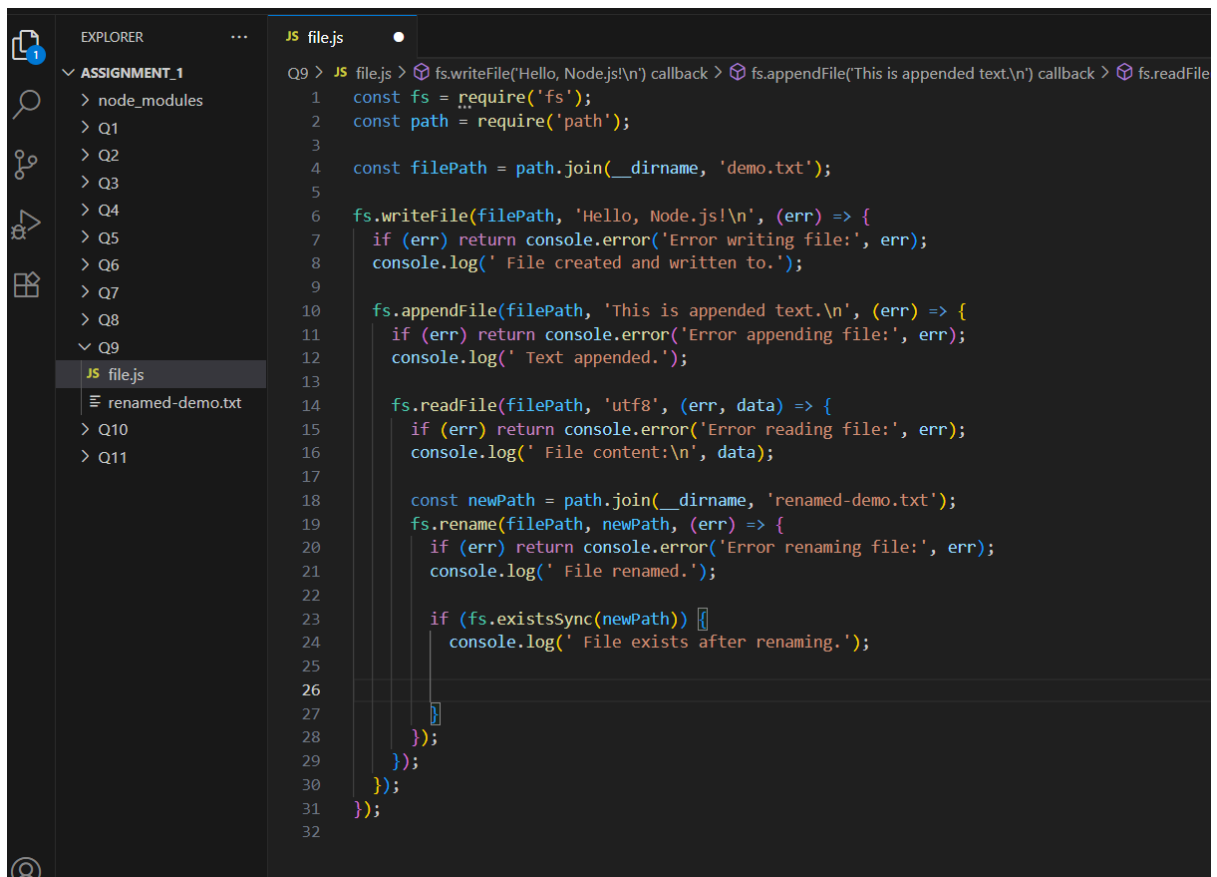
JS fetch.js
1  const fetch = require('node-fetch');
2
3  async function fetchGoogle() {
4    try {
5      const response = await fetch('https://www.google.com');
6      if (!response.ok) {
7        throw new Error(`HTTP error! status: ${response.status}`);
8      }
9      const body = await response.text();
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
18 fetchGoogle();
19
```

```
PS D:\Hari\sem7\Assignment 1> cd Q7
PS D:\Hari\sem7\Assignment 1\Q7> nodeemon fetch.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node fetch.js'
Fetched Google homepage HTML length: 19658
<!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="/image
s/branding/google/1x/google_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="hnK08TUK00Q-5tyG_SE9ka">(function(){var g={key:'d588amhC1LMisQPka.M
wV',keyPI:'0,202792,25,37,2,3497435,658,435,633579,344796,270475,19569,11106,4558,5226018,462,36812180,25275056,91893,14110,8458,3489,45184,8040,6754,23878,9139,4600,328,6225,56
576,3613,9976,6747,8302,48,8157,3292,4134,30379,28334,48287,617,4050,1258,352,18880,5870,7714,5773,8976,6053,12583,3050,2,13472,3261,2990,35,6284,10610,2820,9288,2604,3079,3605,5
90,6737,10444,2731,1848,2,11,4569,8881,490,646,678,3544,4,4,4811,935,3,1470,5612,276,5,3537,1016,1,3463,2,82,1216,2423,463,7,487,891,601,1032,939,6,634,4858,4239,115,239,3694,145
5,935,1711,103,4,1,321,209,980,3010,670,34,2306,7,785,7099,481,965,3,2,2,2,601,730,359,1,322,3352,276,611,4,106,1806,1,4,329,850,514,1488,712,691,3520,1041,729,5,149,18,1873,287,
1158,122,55,282,254,2,1,747,632,32,42,20,1238,632,285,646,311,822,1976,2,668,44,409,264,976,1134,2,1282,65,5,1,367,315,44,83,311,327,107,211,21,674,7,5,445,981,304,2,1005,2,990,5
15,216,2113,251,53,60,1025,229,253,9,1,302,1509,886,786,4,2143,1032,292,104,10,42,95,449,8,841,25,495,552,120,462,3,244,354,5,34,1,2,101,56,251,3,12,1,1248,84,66,37,3016,12,20,11
5,791,122,49,2064,21,692,454,722,2040,3,309,58,12,84,797,67,136,387,350,1,3,2,538,470,3,2,2,2,26,296,4,444,107,174,462,4,906,374,138,89,114,293,115,399,9,4,31,475,4,581,139,352,2
33,605,221,757,4,87,143,346,21214593,5,2253,739,4,2433,267,260,3,1307,1344,586,1096,15,4143,2,1771,2236,3,1553,371,62,41,1668,1095,15,372,122,3,42,2,633,36,199,488,456,8489892,28
515',keyL:'IjMl',keyPI:89978449});(function(){var a;((a=window.google)==null?0:a.stvsc)?google.keyL=g.keyL:window.google=g;}).call(this));})(function(){google.sn='webhp';google.k
L='en-IN'}});(function(){function k(){return window.google&&window.google.koPI||null;var l,m=[];function n(a){for(var b;a&&(la.getAttribute(l)||b=a.getAttribute("eid")));a=a.parentNode;
return b||l}function p(a){for(var b=null;a&&(la.getAttribute(l)||b=a.getAttribute("leid")));a=a.parentNode;return b}function q(a){/^http/i.test(a)&&window.location.protocol=="h
ttps"&&(google.ml&&google.ml(Error("a"),l1,{src:a,glmm:1}),a="");return a}
function r(a,b,d,c,h){var e="";b.search("&ei=")===-1&&(e="&ei="+n(c),b.search("&lei=")===-1&&(c=p(c))&&(e="&lei="+c));var f=b.search("&csid=")===-1&&a!=""&&"slh";c="&zx="+Date.now
().toString();g.csid&&f&&(c+="&csid="+g.csid);(d=d())&&(c+="&opi="+d);return"/"+(h||"gen_204")+"?atyp=i&ct="+String(a)+"&cad="+b+(e+c));l=google.keyI;google.getEI=n;google.ge
tLEI=p;google.ml=function(){return null;google.log=function(a,b,d,c,h,e){e===void 0?k:e;d||(d=r(a,b,e,c,h));if(d=q(d)){a=new Image;var f=m.length;m[f]=a;a.onerror=a.onload=a.d
ehabort=function(){delete m[f];a.src=d}};google.logUrl=function(a,b){b===void 0?k:b;return r("","a,b));}).call(this);(function(){google.y={};google.sy={};function e(a,b,c){if(a)
var d=a.id;else{do d=Math.random();while(c[d])}c[d]=[a,b]}var f;(f=google.x)||f.x=function(a,b){e(a,b,google.y)}};var g;(g=google).sx||(g.sx=function(a,b){e(a,b,google.sy)});goc
gle.lm=[];var h;(h=google).plm||(h.plm=function(a){google.lm.push.apply(google.lm,a)});google.lq=[];var k;(k=google).load||(k.load=function(a,b,c){google.lq.push([a,b,c]);var
l;(l=google).loadAll||(l.loadAll=function(a,b){google.lq.push([a,b]);});google.lx=[];var m;(m=google).lx||(m.lx=function(l){var n=[];p;(p=google).fce||(p.fce=function(a,b,c,d){
n.push([a,b,c,d]);});google.qce=n;google.adl=[];}).call(this);google.f={};(function(){
document.documentElement.addEventListener("submit",function(b){var a;if(a=b.target){var c=a.getAttribute("data-submitfalse");a=c=="1"?[c=="q"&&la.elements.q.value?l0:l1]else a=
1;a&&(b.preventDefault(),b.stopPropagation());l0=document.documentElement.addEventListener("click",function(b){var a;a={for(a=b.target;a&&a!=document.documentElement;a=pare
ntElement)if(a.tagName=="A"){a=a.getAttribute("data-nohref")=="1"?break a:a=l1}}a&&b.preventDefault(),l0);}).call(this);</script><style>#ghar,#user{font-size:13px;padding-top:
1px !important;#ghar{height:22px;#user{padding-bottom:7px !important;text-align:right}#ghh,#ghd{border-top:1px solid #c9d7f1;font-size:1px}#ghh{height:0;position:absolute;top:2
4px;width:100%;@media all{.gb1{height:22px;margin-right:.5em;vertical-align:top}#ghar{float:left}a.gb1,a.gb4{text-decoration:underline !important}a.gb1,a.gb4{color:#00c !important
```

Q8

```
PS D:\Hari\sem7\Assignment_1\Q8> nodemon server.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node "server.js" server.js`
test
[nodemon] clean exit - waiting for changes before restart
```

Q9



```
Q9 > JS file.js > fs.writeFile('Hello, Node.js!\n') callback > fs.appendFile('This is appended text.\n') callback > fs.readFile
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.');
```

```

[nodemon] to restart at any time, enter `rs`
PS D:\Hari\sem7\Assignment_1\Q9> nodemon file.js
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] 3.1.10
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting `node file.js`
[nodemon] starting `node file.js`
File created and written to.
File created and written to.
Text appended.
File content:
Hello, Node.js!
File content:
Hello, Node.js!
This is appended text.
This is appended text.

File renamed.
File exists after renaming.
File exists after renaming.
[nodemon] clean exit - waiting for changes before restart

```

Q10


```


ASSIGNMENT_1
  > env-checker
  > node_modules
  > Q1
  > Q2
  > Q3
  > Q4
  > Q5
  > Q6
  > Q7
  > Q8
  > Q9
  > Q10
    JS globalVariable.js
      Q10 > JS globalVariable.js > ...
        7   setTimeout(() => {
        8     }, 1000);
        9
        10
        11   let count = 0;
        12   const intervalId = setInterval(() => {
        13     count++;
        14     console.log(`setInterval: count = ${count}`);
        15     if (count === 3) {
        16       clearInterval(intervalId);
        17       console.log('Interval cleared');
        18     }
        19   }, 1000);
        20
        21   console.log('Process PID:', process.pid);
        22   console.log('Platform:', process.platform);
        23   console.log('Node version:', process.version);
        24
        25   console.log('NODE_ENV:', process.env.NODE_ENV || 'not set');
        26

```

Q.11 env-checker

```
PS E:\Sharthik patel\nodejs\Assignment1\env-checker> npm publish -access public
npm notice 
npm notice 📦 sharthik@1.0.0
npm notice Tarball Contents
npm notice 140B Readme.md
npm notice 396B index.js
npm notice 371B package.json
npm notice Tarball Details
npm notice name: sharthik
npm notice version: 1.0.0
npm notice filename: sharthik-1.0.0.tgz
npm notice package size: 566 B
npm notice unpacked size: 907 B
npm notice shasum: f0ee75ce222adaa0fd91c89aa13c4ead6b33db13
npm notice integrity: sha512-1+Qip8baNvHol[...]5ZFiI3Siaa7VQ==
```




sharthik
[Edit Profile](#)

1 Package



0 Organizations

Package 1

sharthik

Simple Node.js utility to detect the current environment.
sharthik published 1.0.0 • a few seconds ago

[+ Add Private Packages](#)

Support
[Help](#)
[Advisories](#)
[Status](#)
[Contact npm](#)

Company
[About](#)
[Blog](#)
[Press](#)

Terms & Policies
[Policies](#)
[Terms of Use](#)
[Code of Conduct](#)
[Privacy](#)