

Roll NO:56

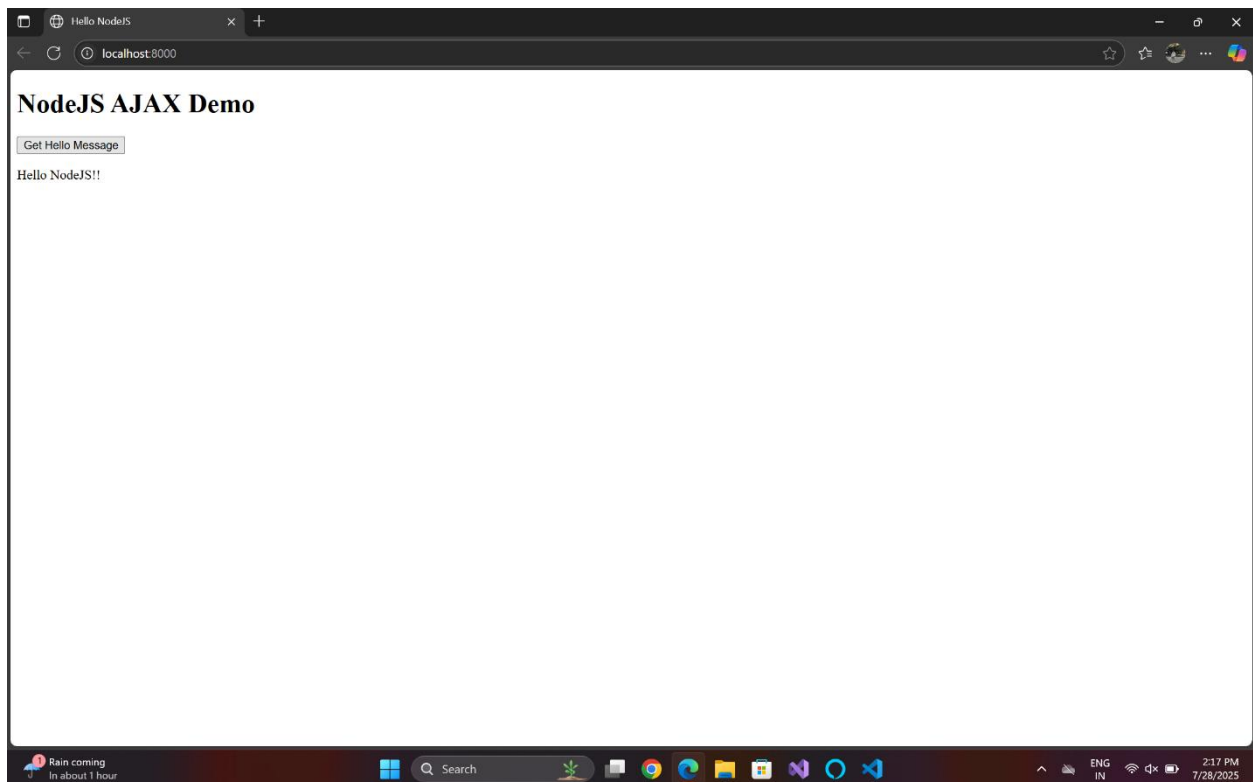
Name:Pratham Kiran Sali

Div:ICT

1) Develop nodejs application with following requirements:

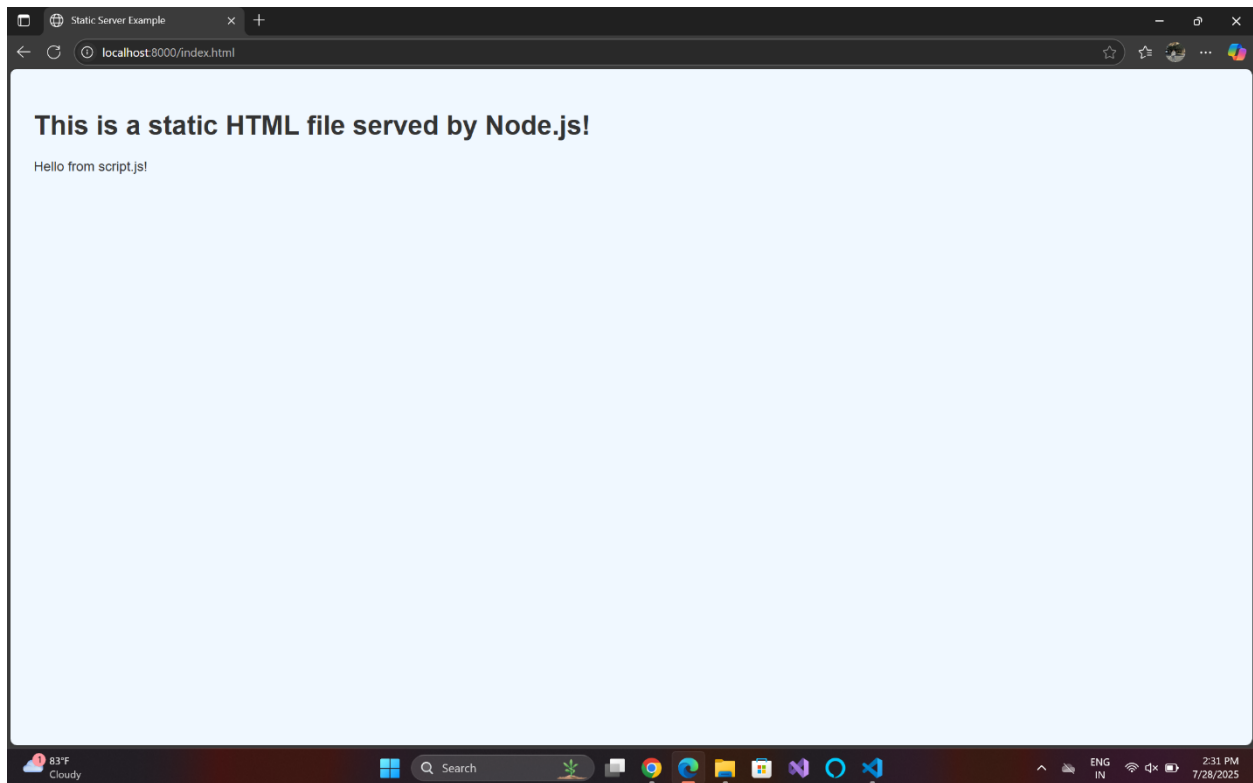
- **Develop a route `"/gethello"` with GET method. It displays `"Hello NodeJS!!"` as response.**
- **Make an HTML page and display.**
- **Call `"/gethello"` route from HTML page using AJAX call.**
(Any frontend AJAX call API can be used.)

ScreenShot



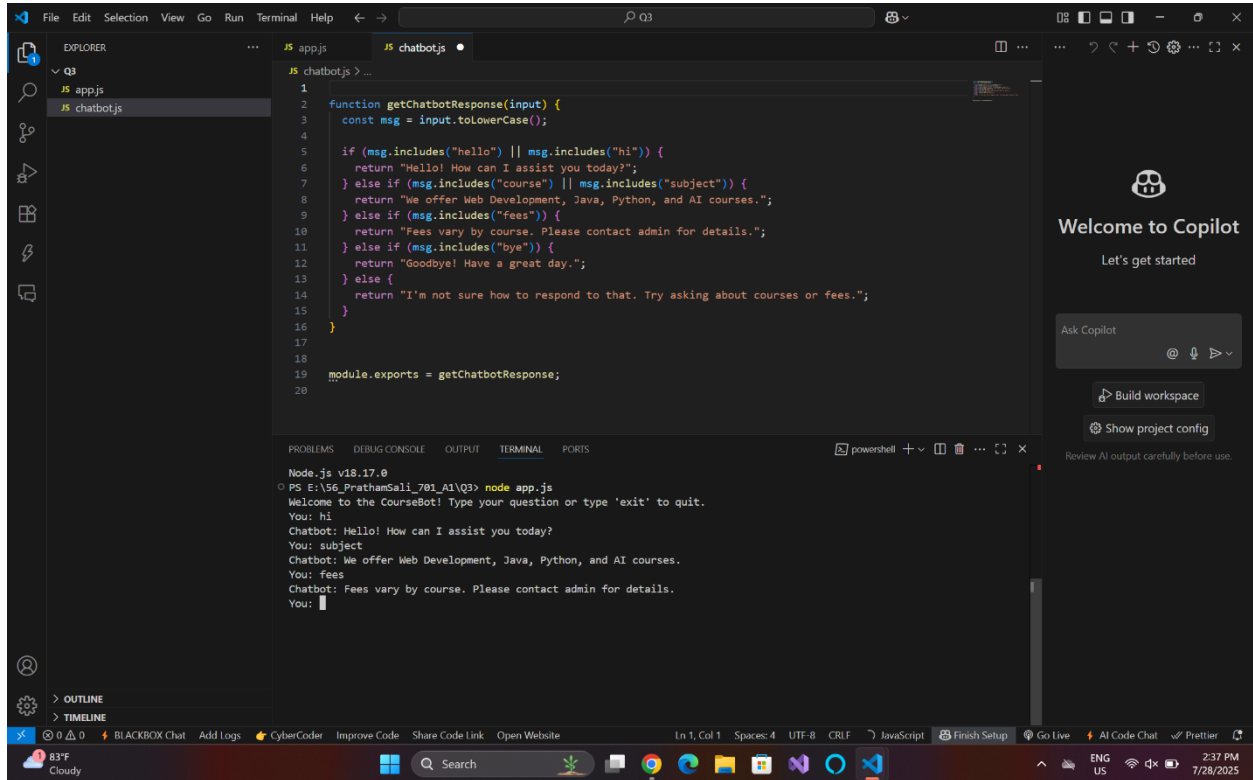
2) Develop a web server which serves static resources.

Screenshot



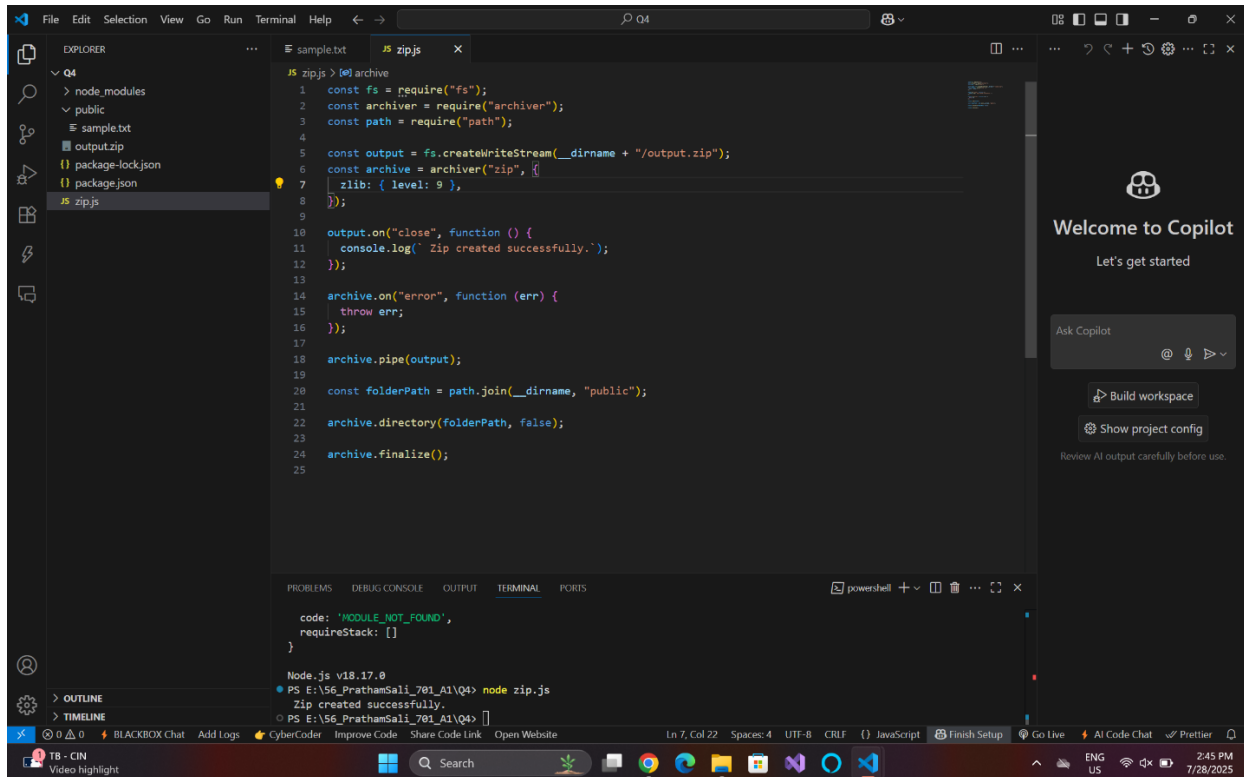
3) Develop a module for domain specific chatbot and use it in a command line application.

ScreenShot



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

Screenshot



5) Write a program to extract a zip file.

Screenshot

The screenshot displays the Visual Studio Code interface. The Explorer sidebar on the left shows a project structure with a folder named 'Q5' containing files like 'extractFolder', 'sample.txt', 'node_modules', 'output.zip', 'package-lock.json', and 'package.json'. The file 'unzipFile.js' is selected and open in the editor. The code in the editor is as follows:

```
1 const fs = require("fs");
2 const unzipper = require("unzipper");
3 const path = require("path");
4
5 const zipFilePath = path.join(__dirname, "output.zip");
6
7 const outputDir = path.join(__dirname, "extractFolder");
8
9 fs.createReadStream(zipFilePath)
10 .pipe(unzipper.Extract({ path: outputDir }))
11 .on("close", () => console.log("Extraction completed successfully."))
12 .on("error", (err) => console.error("Extraction failed:", err));
13
```

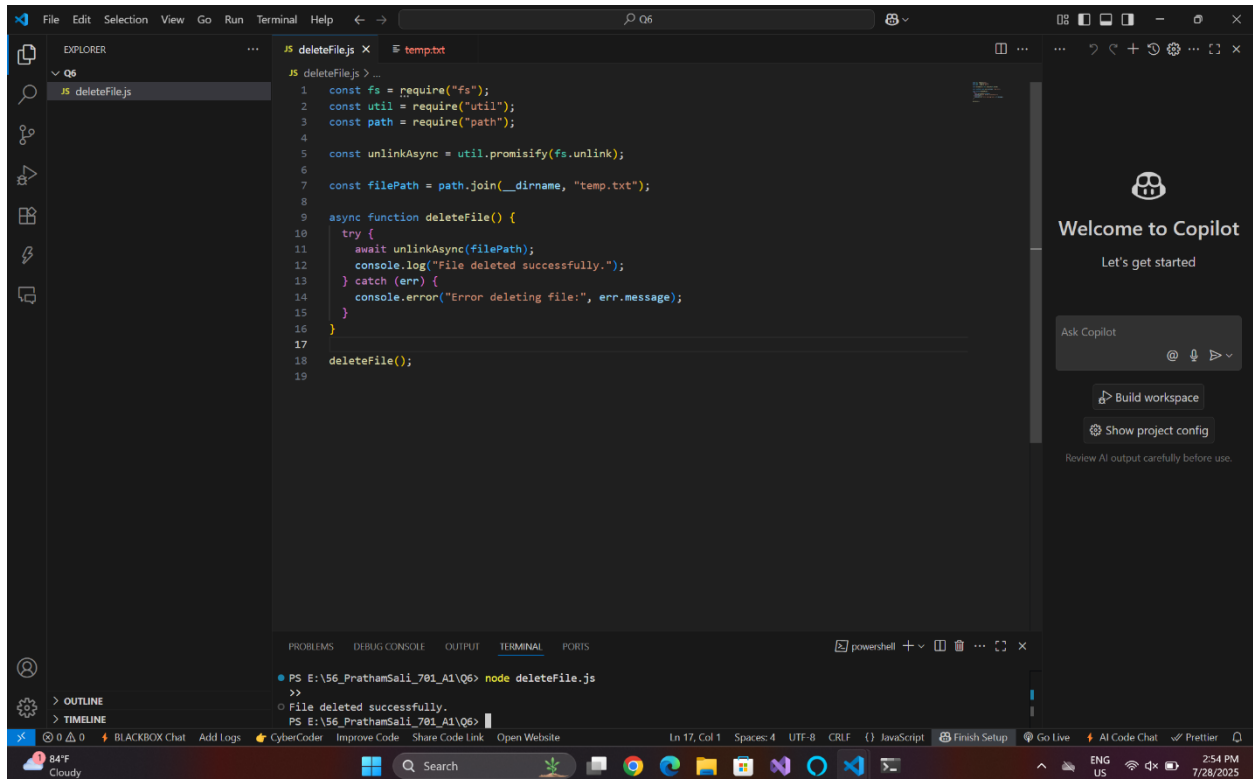
The bottom panel shows the 'TERMINAL' tab with a PowerShell session. The output indicates that 16 packages were added and audited in 6 seconds, followed by the successful execution of the 'node unzipFile.js' command, which resulted in the message 'Extraction completed successfully.'

```
added 16 packages, and audited 17 packages in 6s
found 0 vulnerabilities
PS E:\56_PrathamSali_701_A1\Q5> node unzipFile.js
>>
Extraction completed successfully.
PS E:\56_PrathamSali_701_A1\Q5>
```

The right sidebar features the 'Welcome to Copilot' panel with options to 'Ask Copilot', 'Build workspace', and 'Show project config'.

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 shows a file named `deleteFile.js`. The main editor area displays the code for `deleteFile.js`:

```
1 const fs = require("fs");
2 const util = require("util");
3 const path = require("path");
4
5 const unlinkAsync = util.promisify(fs.unlink);
6
7 const filePath = path.join(__dirname, "temp.txt");
8
9 async function deleteFile() {
10   try {
11     await unlinkAsync(filePath);
12     console.log("File deleted successfully.");
13   } catch (err) {
14     console.error("Error deleting file:", err.message);
15   }
16 }
17
18 deleteFile();
19
```

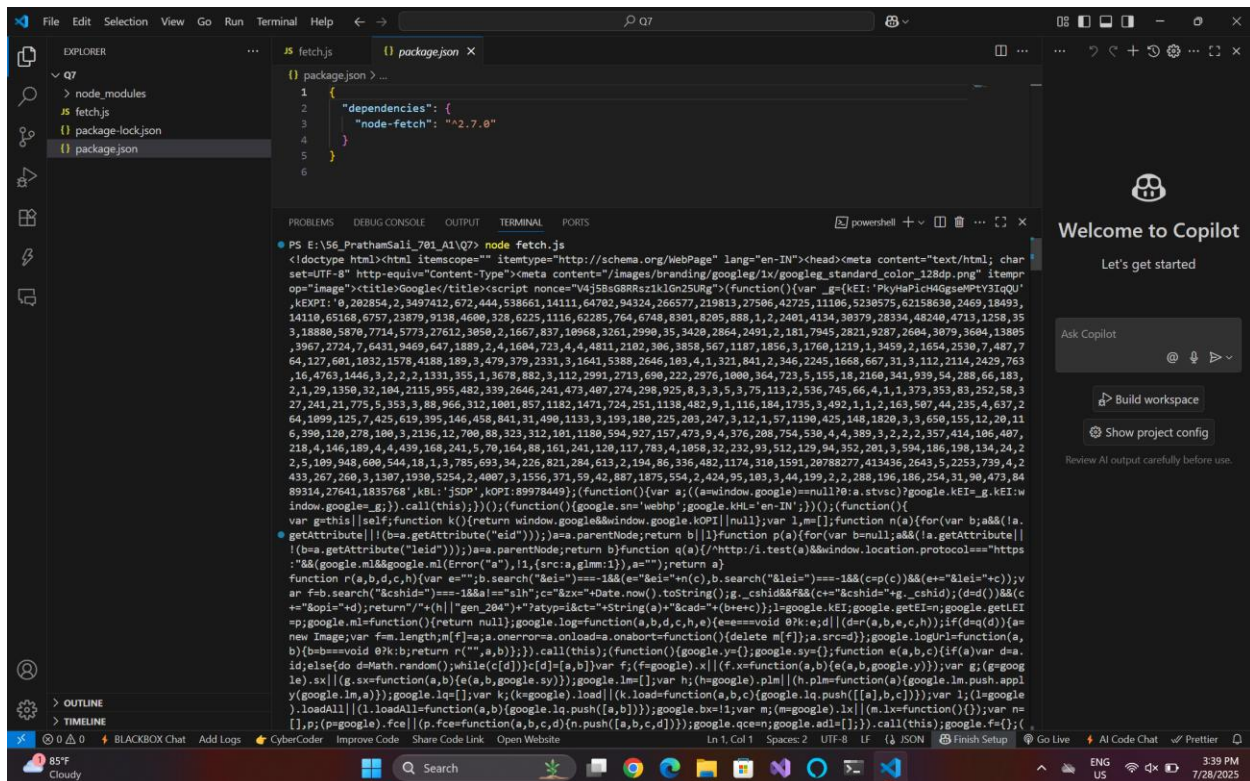
The bottom panel shows the TERMINAL output:

```
PS E:\56_PrathamSali_701_A1\Q6> node deleteFile.js
>>
File deleted successfully.
PS E:\56_PrathamSali_701_A1\Q6>
```

On the right side of the editor, the Copilot sidebar is visible with the text "Welcome to Copilot" and "Let's get started".

7) Fetch data of google page using note-fetch using async-await model.

ScreenShot

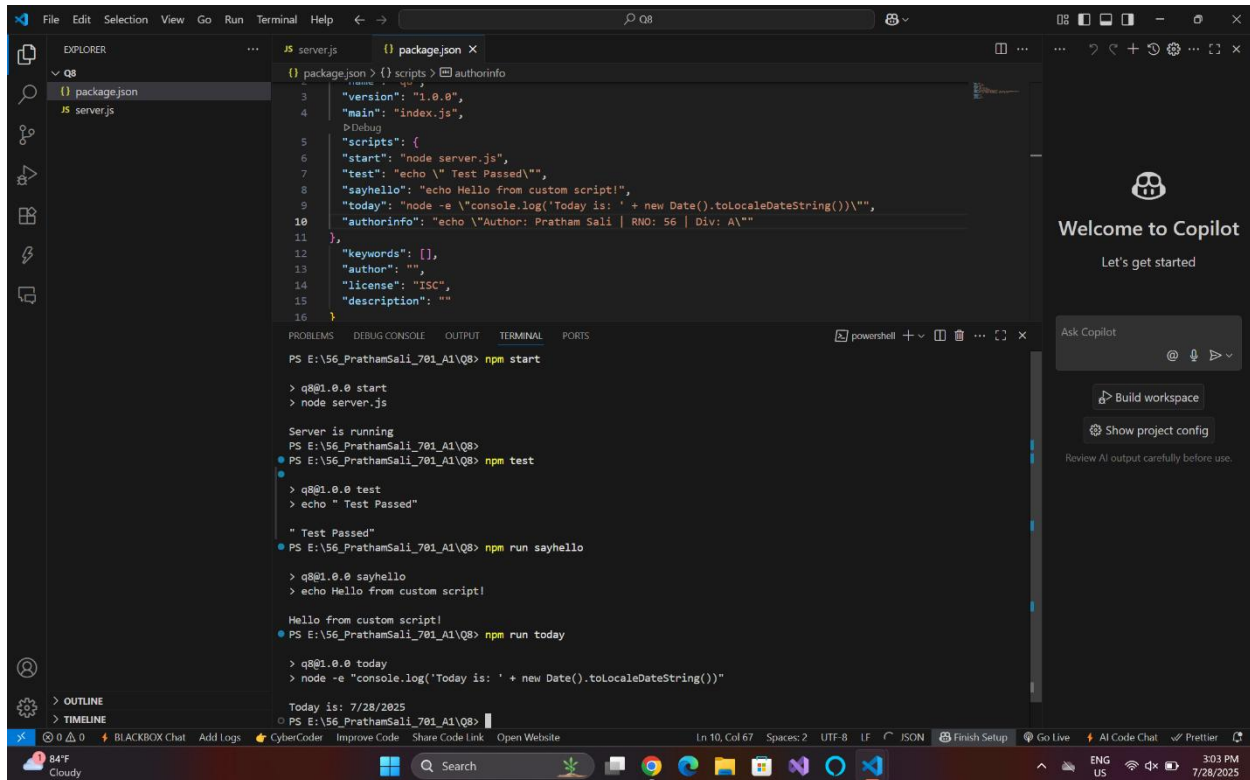


```
1 {
2   "dependencies": {
3     "node-fetch": "^2.7.0"
4   }
5 }
6
```

```
PS E:\S6_PrathamSali_701_A1\Q7> node fetch.js
<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/google_standard_color_128dp.png" itemprop="image"><title>Google</title><script nonce="VgJS8u88Rrzi2kIGn25Ug">(function(){var _ga=({kEI: 'PkYwP1CHWdgs0PEy312QU',kEPI: '8,202854,2,3497412,672,444,538661,24111,64702,94324,266577,219813,27506,42725,11106,5230575,62158638,2469,18493,14110,65168,6757,23879,9138,4600,328,6225,1116,62285,764,6748,8301,8205,888,1,2,2401,4134,30379,28334,48240,4713,1258,35,3,18880,5870,7714,5773,27612,3850,2,1667,837,10968,3261,2990,35,3420,2864,2491,2,181,7945,2821,9287,2604,3079,3604,13805,3967,2724,7,6431,9469,647,1889,2,4,1604,723,4,4,4811,2102,306,3858,567,1187,1856,3,1760,1219,1,3459,2,1654,2530,7,487,7,64,127,601,1032,1578,4188,109,3,479,379,2331,3,1641,5388,2646,103,4,1,321,841,2,346,2245,1668,667,31,3,112,2114,2429,763,16,4763,1446,3,2,2,2,1331,355,1,3678,882,3,112,2991,2713,690,222,2976,1000,364,723,5,155,18,2160,341,939,54,288,66,183,2,1,29,1350,32,104,2115,955,482,339,2646,241,473,407,274,298,925,8,3,3,5,3,75,113,2,536,745,66,4,1,1,373,353,83,252,58,3,27,241,21,775,5,353,3,88,966,312,1001,857,1182,1471,724,251,1138,482,9,1,116,184,1735,3,492,1,1,2,163,507,44,235,4,637,2,64,1099,125,7,425,619,395,146,458,841,31,490,1133,3,193,180,225,203,247,3,12,1,57,1190,425,148,1820,3,3,650,155,12,20,11,6,390,128,278,100,3,2136,12,700,88,323,312,101,1180,594,927,157,473,9,4,376,208,754,530,4,4,389,3,2,2,2,357,414,106,407,218,4,146,109,4,4,409,108,241,5,70,164,80,161,241,120,117,783,4,1088,32,232,93,612,129,94,352,201,3,594,186,190,114,24,2,5,109,948,600,544,10,1,3,785,693,34,226,821,284,613,2,194,86,336,482,1174,310,1591,20788277,413496,2643,5,2253,739,4,2,433,267,260,3,1307,1930,5254,2,4007,3,1556,371,59,42,887,1875,554,2,424,95,103,3,44,109,2,2,288,196,186,254,31,90,473,84,89314,27641,1835768',kBL:'JSOP',kOPI:89978449});function(){var a;((a=window.google)==null?0:a.stvc)?google.kEI=g.kEI:window.google=g;}).call(this));(function(){google.sn='webhp';google.kHL='en-IN';})();function(){var g=this||self;function k(){(return window.google&&window.google.kOPI||null);var l=[];function n(a){for(var b;a&&(la.getAttribute||(b=a.getAttribute("eid"))));a=a.parentNode;return b[l]}function p(a){for(var b=null;a&&(la.getAttribute||(b=a.getAttribute("leid"))));a=a.parentNode;return b}function q(a){/^http:\/\/i.test(a)&&window.location.protocol=="https"?:"&&(google.ml&&google.ml(Error("a"),l,{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("&ei=")===-1&&(e=c+p(c))&&(e="&ei="+c));var f=b.search("&csid=")===-1&&(e="&ei="+h);c="&zx="+Date.now().toString().replace(/g_/_/g,"_");c=c+"&csid="+g._csid;f?(d=d())&&(e="&ei="+d);return"/"+h+"/gen_304?&_zatzp1&et="+String(e)+"&cad="+b(e+c)};l=google.kEI;google.getEI=google.getEI?google.ml:function(){(return null);google.log=function(a,b,d,c,h,e){e="";e+=void 0?k:b;d||(d=r(a,b,e,c,h));if(d(d)){a=new Image;var f=m.length;m[f]=a;a.onerror=a.onload=a.onabort=function(){delete m[f];a.src=""};google.logurl=function(a,b){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.le).sx||(g.sx=function(a,b){e(a,b,google.sy)});google.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.bx=1;var m;(m=google).lx||(m.lx=function(){var n=[];p;(p=google).fce||(p.fce=function(a,b,c,d){n.push([a,b,c,d]);google.qcen=google.adl-[[]];}).call(this);google.f=();(
```


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

Screenshot



The screenshot shows a Visual Studio Code editor with a project named 'Q8'. The Explorer sidebar on the left shows a file named 'package.json'. The main editor area displays the contents of 'package.json', which includes a 'scripts' section with the following configuration:

```
{
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "start": "node server.js",
    "test": "echo \\\" Test Passed\\\"",
    "sayhello": "echo Hello from custom script!",
    "today": "node -e \\\"console.log('Today is: ' + new Date().toLocaleDateString())\\\"",
    "authorinfo": "echo \\\"Author: Pratham Sali | RNO: 56 | Div: A\\\""
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

Below the editor, the TERMINAL panel shows the execution of several npm commands in a PowerShell session:

```
PS E:\56_PrathamSali_701_A1\Q8> npm start

> q8@1.0.0 start
> node server.js

Server is running
PS E:\56_PrathamSali_701_A1\Q8>
PS E:\56_PrathamSali_701_A1\Q8> npm test

> q8@1.0.0 test
> echo " Test Passed"

" Test Passed"
PS E:\56_PrathamSali_701_A1\Q8> npm run sayhello

> q8@1.0.0 sayhello
> echo Hello from custom script!

Hello from custom script!
PS E:\56_PrathamSali_701_A1\Q8> npm run today

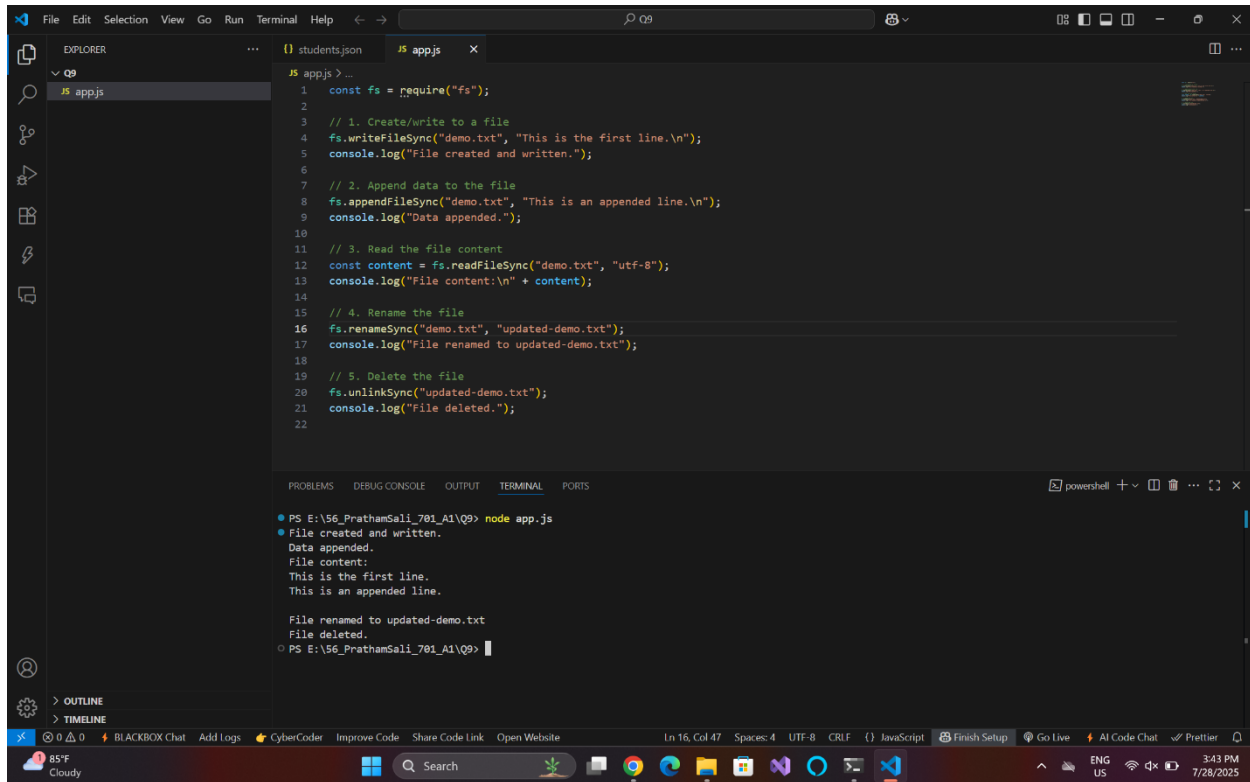
> q8@1.0.0 today
> node -e "console.log('Today is: ' + new Date().toLocaleDateString())"

Today is: 7/28/2025
PS E:\56_PrathamSali_701_A1\Q8>
```

On the right side of the interface, the 'Welcome to Copilot' sidebar is visible, offering options like 'Let's get started', 'Ask Copilot', 'Build workspace', and 'Show project config'.

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

Screenshot



The screenshot displays the Visual Studio Code editor with a JavaScript file named `app.js` open. The code uses the `fs` module to perform five tasks: creating/writing to a file, appending data, reading file content, renaming a file, and deleting a file. The terminal at the bottom shows the command `node app.js` being executed, with the corresponding output messages.

```
1 const fs = require("fs");
2
3 // 1. Create/write to a file
4 fs.writeFileSync("demo.txt", "This is the first line.\n");
5 console.log("File created and written.");
6
7 // 2. Append data to the file
8 fs.appendFileSync("demo.txt", "This is an appended line.\n");
9 console.log("Data appended.");
10
11 // 3. Read the file content
12 const content = fs.readFileSync("demo.txt", "utf-8");
13 console.log("File content:\n" + content);
14
15 // 4. Rename the file
16 fs.renameSync("demo.txt", "updated-demo.txt");
17 console.log("File renamed to updated-demo.txt");
18
19 // 5. Delete the file
20 fs.unlinkSync("updated-demo.txt");
21 console.log("File deleted.");
22
```

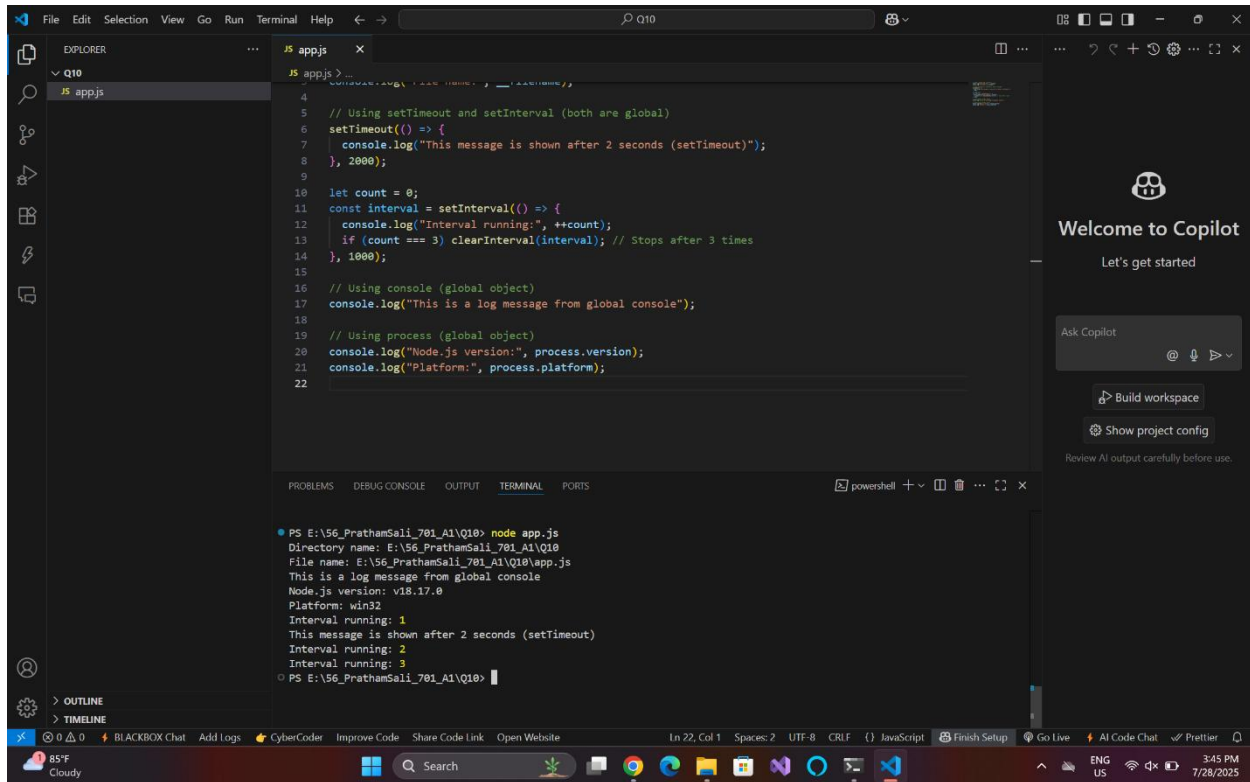
Terminal Output:

```
PS E:\S6_PrathamSali_701_A1\Q9> node app.js
File created and written.
Data appended.
File content:
This is the first line.
This is an appended line.

File renamed to updated-demo.txt
File deleted.
PS E:\S6_PrathamSali_701_A1\Q9>
```

10. A program which uses global objects in nodejs.

Screenshot



The screenshot displays the Visual Studio Code editor with a JavaScript file named `app.js` and its execution output in the terminal.

Code in `app.js`:

```
4 // Using setTimeout and setInterval (both are global)
5
6 setTimeout(() => {
7   console.log("This message is shown after 2 seconds (setTimeout)");
8 }, 2000);
9
10 let count = 0;
11 const interval = setInterval(() => {
12   console.log("Interval running:", ++count);
13   if (count === 3) clearInterval(interval); // Stops after 3 times
14 }, 1000);
15
16 // Using console (global object)
17 console.log("This is a log message from global console");
18
19 // Using process (global object)
20 console.log("Node.js version:", process.version);
21 console.log("Platform:", process.platform);
22
```

Terminal Output:

```
PS E:\S6_PrathamSali_701_A1\Q10> node app.js
Directory name: E:\S6_PrathamSali_701_A1\Q10
File name: E:\S6_PrathamSali_701_A1\Q10\app.js
This is a log message from global console
Node.js version: v18.17.0
Platform: win32
Interval running: 1
This message is shown after 2 seconds (setTimeout)
Interval running: 2
Interval running: 3
PS E:\S6_PrathamSali_701_A1\Q10>
```

The terminal output confirms that the program successfully executed, demonstrating the use of global objects like `setTimeout`, `setInterval`, `console`, and `process` in Node.js.

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

Screenshot

