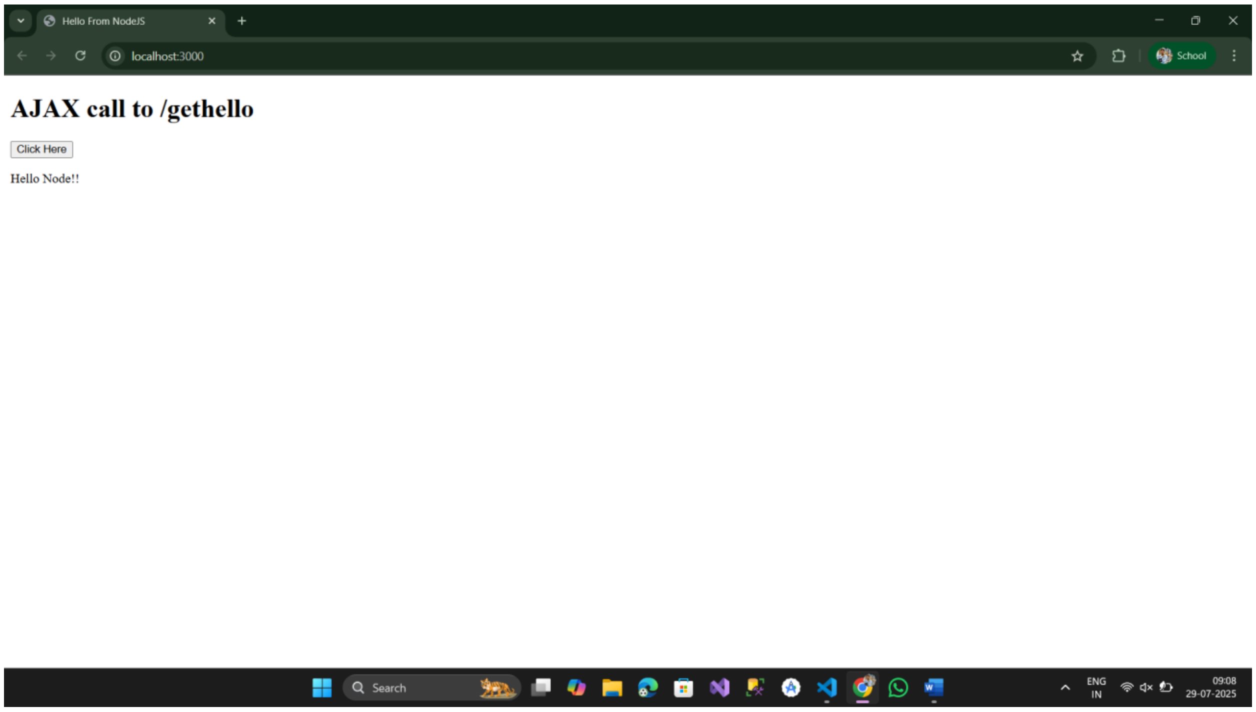
Roll NO:30
Name:Mistri Vatsal Rakeshbhai
Div:ICT

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

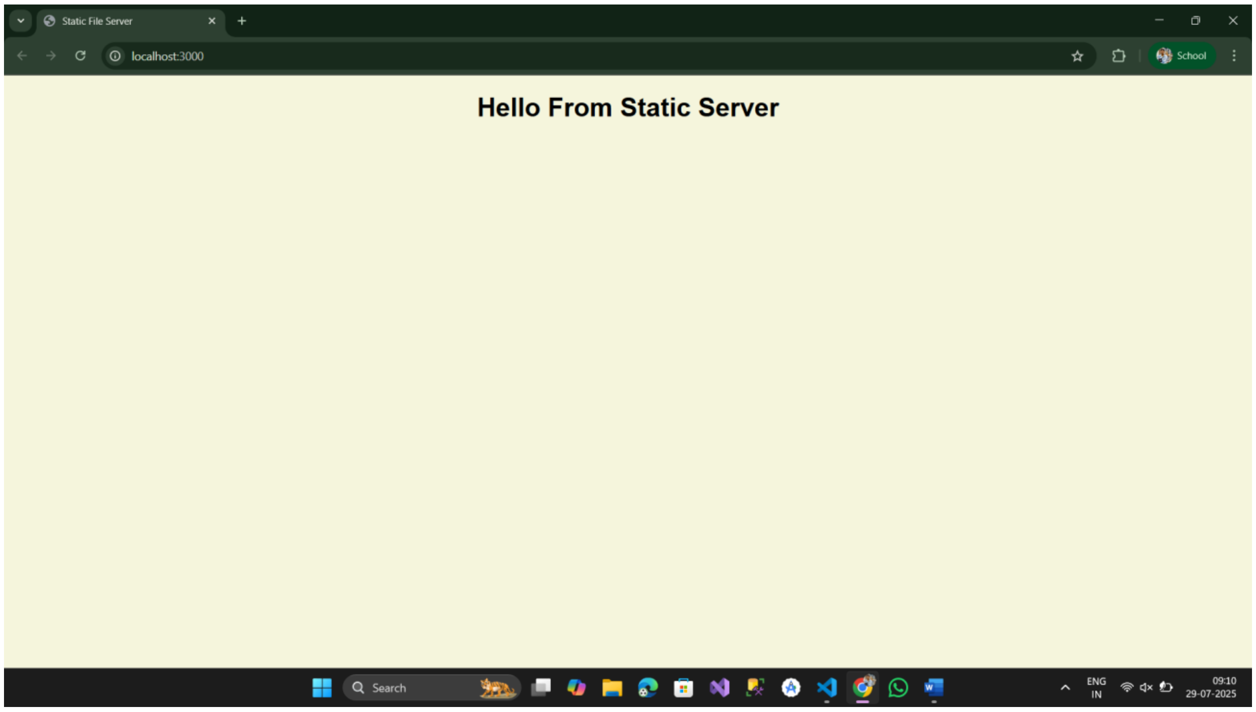


AJAX call to /gethello

Hello Node!!

2) Develop a web server which serves static resources.

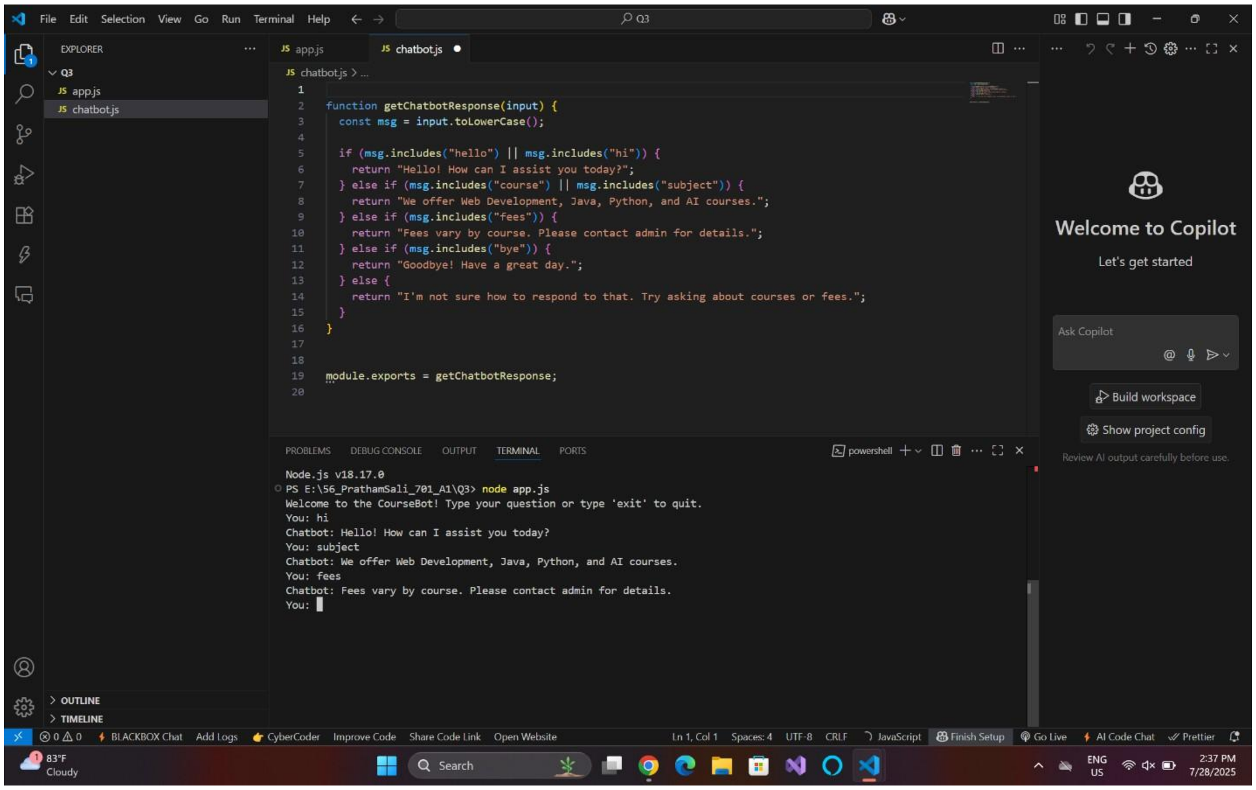
Screeshot



Hello From Static Server

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

ScreenShot

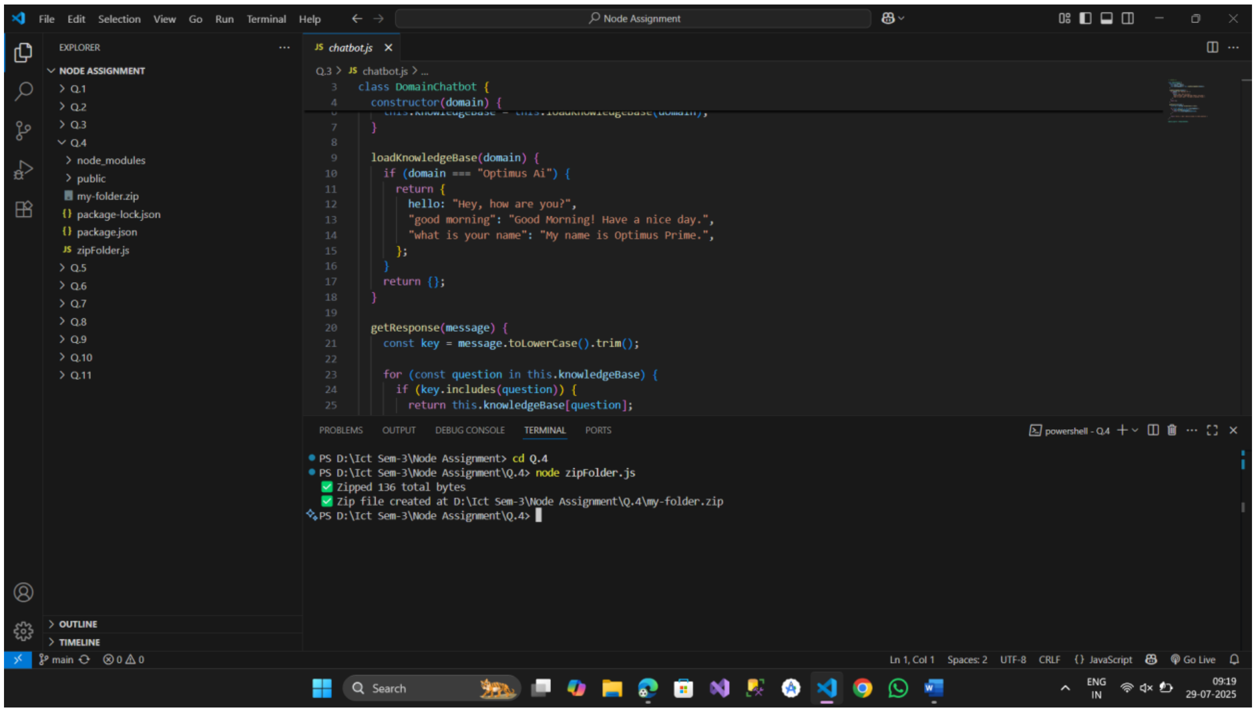


```javascript

function getChatbotResponse(input) {
  const msg = input.toLowerCase();

  if (msg.includes("hello") || msg.includes("hi")) {
    return "Hello! How can I assist you today?";
  } else if (msg.includes("course") || msg.includes("subject")) {
    return "We offer Web Development, Java, Python, and AI courses.";
  } else if (msg.includes("fees")) {
    return "Fees vary by course. Please contact admin for details.";
  } else if (msg.includes("bye")) {
    return "Goodbye! Have a great day.";
  } else {
    return "I'm not sure how to respond to that. Try asking about courses or fees.";
  }
}


module.exports = getChatbotResponse;

```

```
Node.js v18.17.0
PS E:\56_PrathamSali_701_A1\Q3> node app.js
Welcome to the CourseBot! Type your question or type 'exit' to quit.
You: hi
Chatbot: Hello! How can I assist you today?
You: subject
Chatbot: We offer Web Development, Java, Python, and AI courses.
You: fees
Chatbot: Fees vary by course. Please contact admin for details.
You: 
```

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

Screenshot

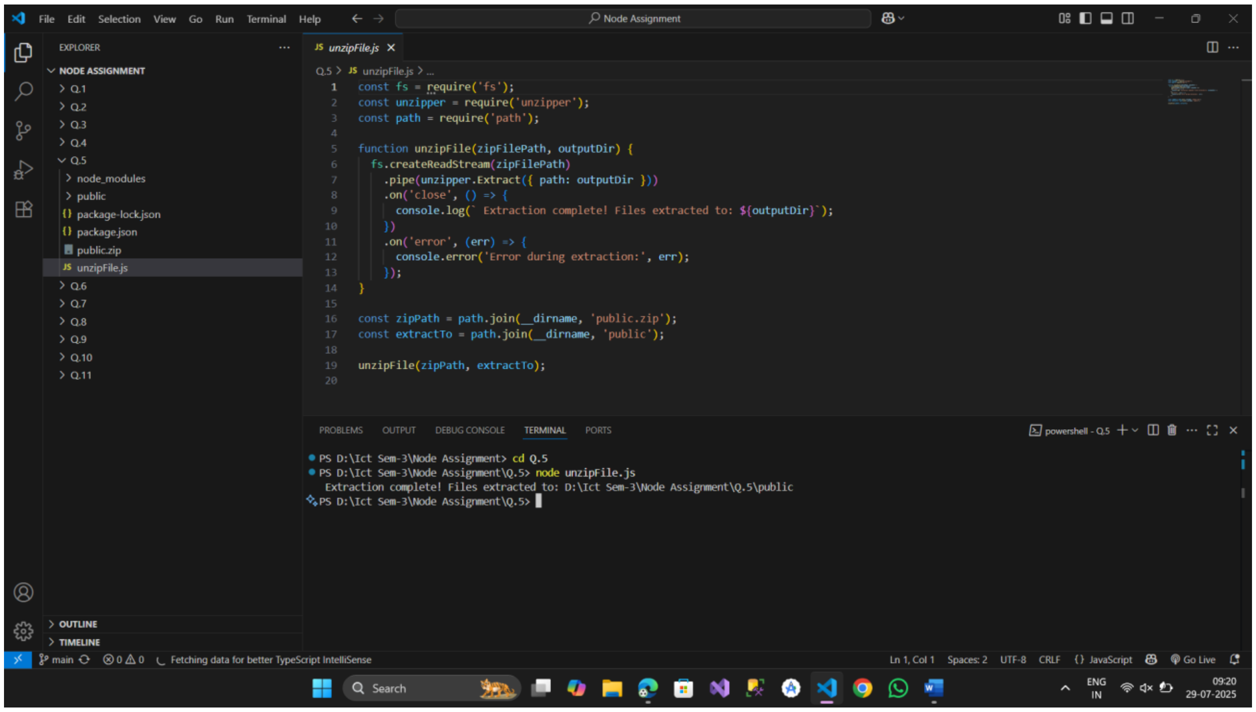


```javascript
class DomainChatbot {
  constructor(domain) {
    this.knowledgeBase = this.loadKnowledgeBase(domain);
  }

  loadKnowledgeBase(domain) {
    if (domain === "Optimus Ai") {
      return {
        hello: "Hey, how are you?",
        "good morning": "Good Morning! Have a nice day.",
        "what is your name": "My name is Optimus Prime.",
      };
    }
    return {};
  }

  getResponse(message) {
    const key = message.toLowerCase().trim();

    for (const question in this.knowledgeBase) {
      if (key.includes(question)) {
        return this.knowledgeBase[question];
```

```
PS D:\Ict Sem-3\Node Assignment> cd Q.4
PS D:\Ict Sem-3\Node Assignment\Q.4> node zipFolder.js
✅ Zipped 136 total bytes
✅ Zip file created at D:\Ict Sem-3\Node Assignment\Q.4\my-folder.zip
PS D:\Ict Sem-3\Node Assignment\Q.4>
```

5) Write a program to extract a zip file.

Screnshot

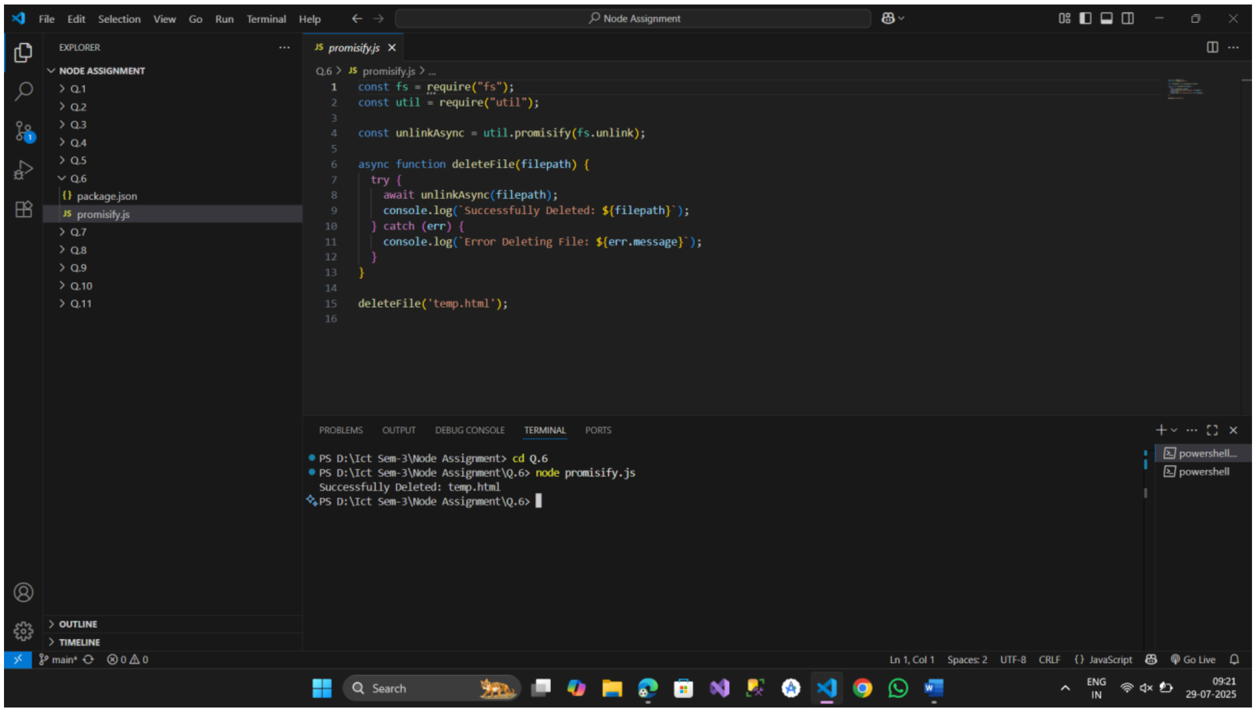


```javascript
const fs = require('fs');
const unzipper = require('unzipper');
const path = require('path');

function unzipFile(zipFilePath, outputDir) {
  fs.createReadStream(zipFilePath)
    .pipe(unzipper.Extract({ path: outputDir }))
    .on('close', () => {
      console.log(` Extraction complete! Files extracted to: ${outputDir}`);
    })
    .on('error', (err) => {
      console.error('Error during extraction:', err);
    });
}

const zipPath = path.join(__dirname, 'public.zip');
const extractTo = path.join(__dirname, 'public');

unzipFile(zipPath, extractTo);
```

```
PS D:\Ict Sem-3\Node Assignment> cd Q.5
PS D:\Ict Sem-3\Node Assignment\Q.5> node unzipFile.js
 Extraction complete! Files extracted to: D:\Ict Sem-3\Node Assignment\Q.5\public
PS D:\Ict Sem-3\Node Assignment\Q.5>
```

6) Write a program to promisify fs.unlink function and call it.

Screenshot

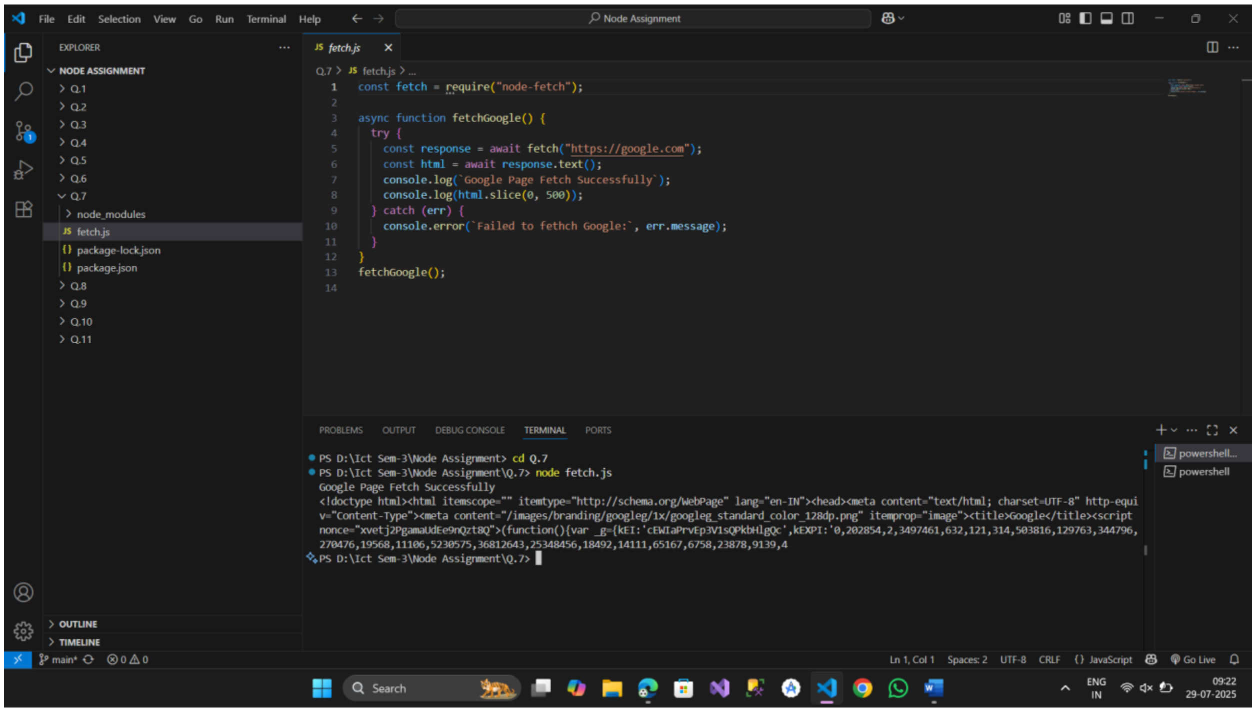


```javascript
const fs = require("fs");
const util = require("util");

const unlinkAsync = util.promisify(fs.unlink);

async function deleteFile(filepath) {
  try {
    await unlinkAsync(filepath);
    console.log(`Successfully Deleted: ${filepath}`);
  } catch (err) {
    console.log(`Error Deleting File: ${err.message}`);
  }
}

deleteFile('temp.html');
```

```
PS D:\Ict Sem-3\Node Assignment> cd Q.6
PS D:\Ict Sem-3\Node Assignment\Q.6> node promisify.js
Successfully Deleted: temp.html
PS D:\Ict Sem-3\Node Assignment\Q.6>
```

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

ScreenShot

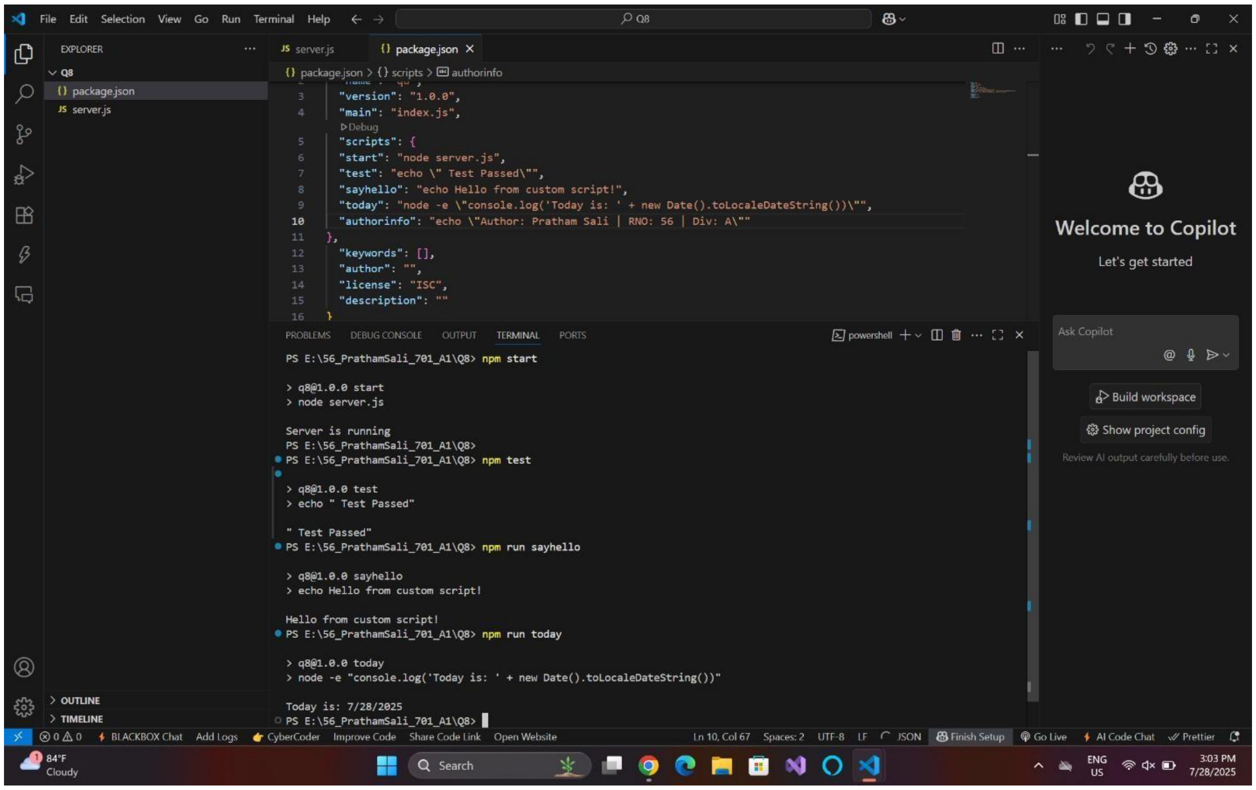


```javascript
const fetch = require("node-fetch");

async function fetchGoogle() {
  try {
    const response = await fetch("https://google.com");
    const html = await response.text();
    console.log(`Google Page Fetch Successfully`);
    console.log(html.slice(0, 500));
  } catch (err) {
    console.error(`Failed to fethch Google:`, err.message);
  }
}
fetchGoogle();
```

```
PS D:\Ict Sem-3\Node Assignment> cd Q.7
PS D:\Ict Sem-3\Node Assignment\Q.7> node fetch.js
Google Page Fetch Successfully
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en-IN"><head><meta content="text/html; charset=UTF-8" http-equi
v="Content-Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_128dp.png" itemprop="image"><title>Google</title><script
nonce="xvetj2PgamaUdEe9nQzt8Q">(function(){var _g={kEI:'cEWIaPrvEp3V1sQPkbHlgQc',kEXPI:'0,202854,2,3497461,632,121,314,503816,129763,344796,
270476,19568,11106,5230575,36812643,25348456,18492,14111,65167,6758,23878,9139,4
PS D:\Ict Sem-3\Node Assignment\Q.7>
```

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

Screenshot

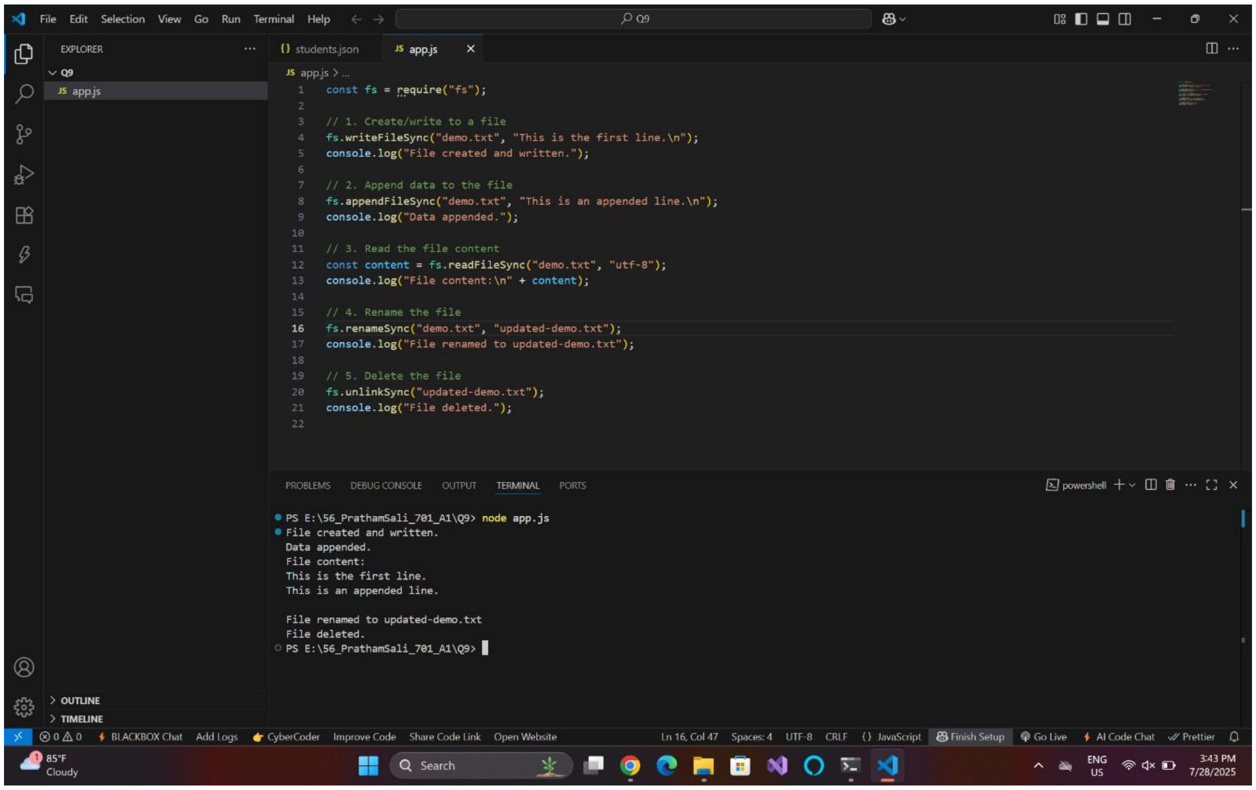


```json
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
  "start": "node server.js",
  "test": "echo \" Test Passed\"",
  "sayhello": "echo Hello from custom script!",
  "today": "node -e \"console.log('Today is: ' + new Date().toLocaleDateString())\"",
  "authorinfo": "echo \"Author: Pratham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

```
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>
```

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

Screenshot

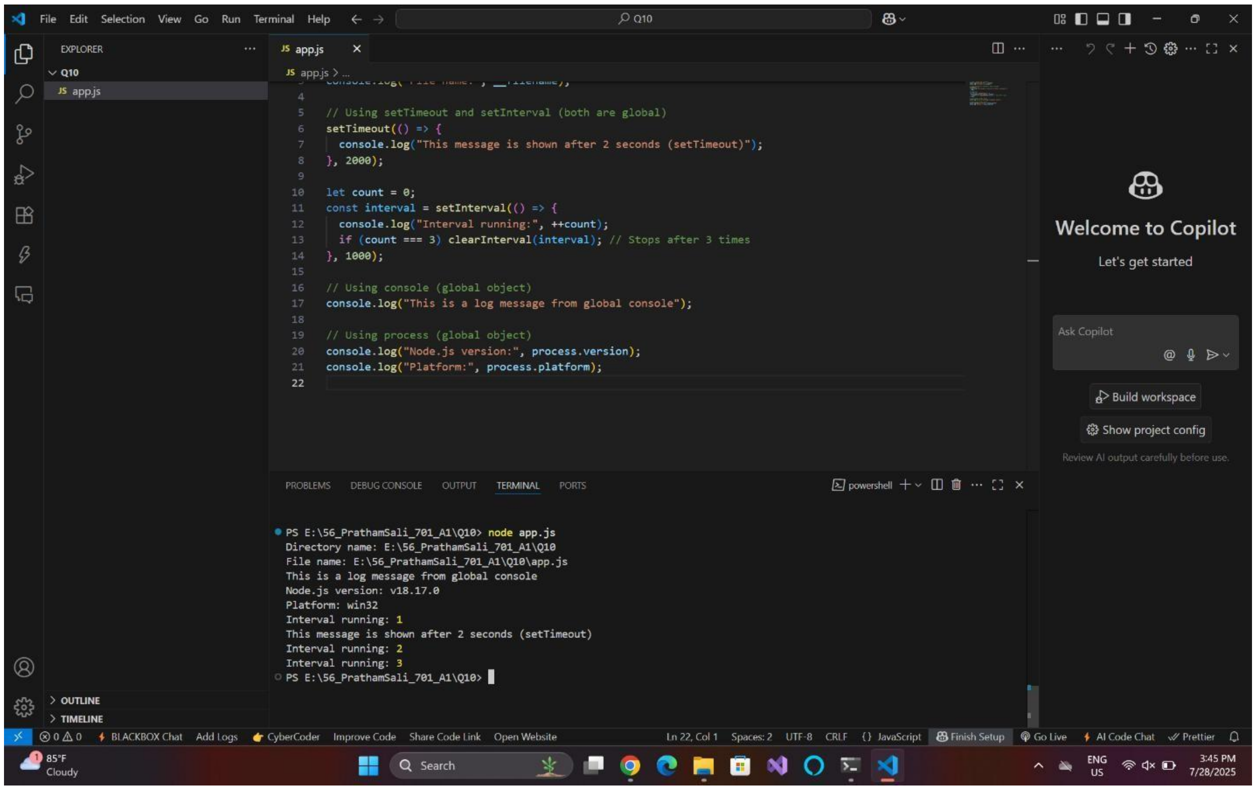


```javascript
const fs = require("fs");

// 1. Create/write to a file
fs.writeFileSync("demo.txt", "This is the first line.\n");
console.log("File created and written.");

// 2. Append data to the file
fs.appendFileSync("demo.txt", "This is an appended line.\n");
console.log("Data appended.");

// 3. Read the file content
const content = fs.readFileSync("demo.txt", "utf-8");
console.log("File content:\n" + content);

// 4. Rename the file
fs.renameSync("demo.txt", "updated-demo.txt");
console.log("File renamed to updated-demo.txt");

// 5. Delete the file
fs.unlinkSync("updated-demo.txt");
console.log("File deleted.");

```

```
PS E:\56_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:\56_PrathamSali_701_A1\Q9>
```

10. A program which uses global objects in nodejs.

Screenshot

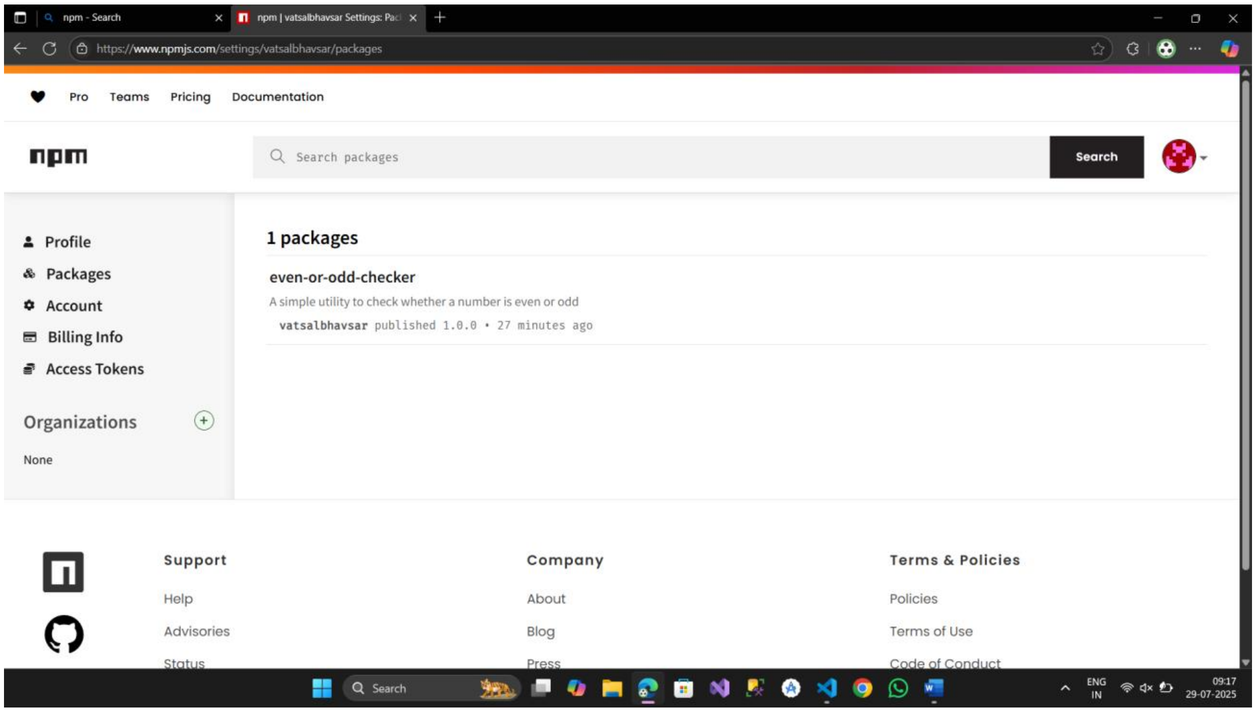


```javascript
// Using setTimeout and setInterval (both are global)
setTimeout(() => {
  console.log("This message is shown after 2 seconds (setTimeout)");
}, 2000);

let count = 0;
const interval = setInterval(() => {
  console.log("Interval running:", ++count);
  if (count === 3) clearInterval(interval); // Stops after 3 times
}, 1000);

// Using console (global object)
console.log("This is a log message from global console");

// Using process (global object)
console.log("Node.js version:", process.version);
console.log("Platform:", process.platform);
```

```
PS E:\56_PrathamSali_701_A1\Q10> node app.js
Directory name: E:\56_PrathamSali_701_A1\Q10
File name: E:\56_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:\56_PrathamSali_701_A1\Q10>
```

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

Screenshot



1 packages

even-or-odd-checker

A simple utility to check whether a number is even or odd

vatsalbhavsar published 1.0.0 • 27 minutes ago