Node JS

Important Tasks

Task-1

Write a node js script to write the text "You are creating a file" to help.txt file. After that append the text "you are appending data" to same help.txt file. After that read the file and print file contents on console. After finishing read operation, print the line "Thanks for using my program" on console. write, append, read sequence must be maintained. all read, write and append operations are asynchronous.

```
const fs = require('fs');
fs.writeFile('help.txt', 'You are creating a file', () => {
  fs.appendFile('help.txt', '\nyou are appending data', () => {
     fs.readFile('help.txt', 'utf-8', (err, data) => {
        console.log(data);
        console.log('Thanks for using my program');
        });
    });
 });
```

Task-2

Write a function 'ArrayToObject' which takes in an array of arrays, and returns an object with each pair of elements in the array as a key-value pair and store the result in one arraytoobject.txt file.

Input=[['Country', India'], ['State', 'Gujarat'], ['City', 'Ahmedabad']]

Output= { Country : ' India ', State : ' Gujarat ', City : 'Ahmedabad' }

```
function ArrayToObject(a) {
    var temp = {};
   temp[a[0][0]] = a[0][1]; // First key-value pair
   temp[a[1][0]] = a[1][1]; // Second key-value pair
   temp[a[2][0]] = a[2][1]; // Third key-value pair
   return temp;
}

var data = [['Country', 'India'], ['State', 'Gujarat'], ['City', 'Ahmedabad']];
var output = ArrayToObject(data);
console.log(output);

// Store the output in a file
const fs = require('fs');
fs.writeFileSync('arraytoobject.txt', JSON.stringify(output));
```

Task-3

Write node js script designed for laptop having 6GB of RAM which require to monitor system memory usage and emit a custom event named "Threshold". When the memory usage exeeds specified threshold of 50% there will be a message "Memory Threshold Exceeded" along with used data should be displayed on console after every 1 second of Interval.

```
const os = require("os");
const EventEmitter = require("events");
const eventEmitter = new EventEmitter();


eventEmitter.on("threshold", () => {
setInterval(() => {
   const tm = 6
   const fm = os.freemem()/1024/1024/1024;
   const um = tm - fm;
   const up = (um / tm) * 100;
   console.log(up)

   if (up > 50) {
      console.log("Used Memory"+um+" Memory threshold exceeded!");
   }
}, 1000);

});

eventEmitter.emit("threshold");
```

Task -4

Create HTTP webpage on which Home page display "Welcome to Log in page" in blue color and font size must be 32px, Login page shows one HTML file from static URL having Form with detail for Username, Password, submit and reset button, Gallery page reflect one Image "hello.jpg" and any other page shows "Page Not found". Write all necessary files to perform task. (Image already exist in same folder)

login.js

```
var h = require("http");
var url = require("url");
var fs = require("fs");
var addr="http://localhost:5051/login.html";
var server = h.createServer(function (req, res) {
   if (req.url == "/") {
      res.writeHead(200, { "Content-Type": "text/html" });
      res.write("<h1 style='color:blue;font-size:32px;'>Welcome to login page</h1>");
      res.end();
   } else if (req.url == "/login") {
```

```
    var q=url.parse(addr,true);
    data=fs.readFileSync("."+q.pathname);
     res.writeHead(200, { "Content-Type": "text/html" });
     res.write(data);
     res.end();
  } else if (req.url == "/gallery") {
     img = fs.readFileSync("1.png");
     res.writeHead(200, { "Content-Type": "image/png" });
     res.end(img); // Sending image data

  } else {
     res.writeHead(404, { "Content-Type": "text/plain" });
     res.write("Page not found");
     res.end("\nPlease check the URL");
  }
});

server.listen(5051)
```

login.html

```
<html>
<body>
   <h2>Login Form</h2>
   <form action="#" method="POST">
      <label for="username">Username:</label>
      <input type="text"  required><br><br>

      <label for="password">Password:</label>
      <input type="password" required><br><br>

      <button type="submit">Submit</button>
      <button type="reset">Reset</button>
   </form>
</body>
</html>
```

Task-5

Write a nodeJS script to fire an event named calculate which calculates the total marks of 5 subjects about of 25 marks and displays the total marks on console as an output. The calculate event fires another event name percentage which takes total marks as argument and percentage should get displayed in console.

```
const EventEmitter = require("events");
const ee = new EventEmitter();

// Event to calculate total marks
ee.on("calculate", () => {
   const marks = [22, 18, 20, 25, 24];
```

```
    let total = 0;
    for (let i = 0; i < marks.length; i++) {
        total += marks[i];
    }
    console.log("Total Marks:", total);
    ee.emit("percentage", total);
});

ee.on("percentage", (total) => {
    const totalMaxMarks = 5 * 25;
    const percentage = (total / totalMaxMarks) * 100;
    console.log("Percentage:", percentage + "%");
});

ee.emit("calculate");
```

Task-6

Create HTTP webpages where Admin page displays “Sufficient Memory:” in bold blue color with font size of 24px along with available memory in GB with font size 32px and red color if available physical memory of the system is greater than 1 GB.
Else it shows “Not Sufficient Memory” in simple text.
For any other page requested then shows “You are not admin” message.

```
const http = require('http');
const os = require('os');

const server = http.createServer((req, res) => {
    if (req.url === "/admin") {
        const freeMem = os.freemem()/1024/1024/1024; // Convert bytes to GB

        if (freeMem > 1) {
            res.writeHead(200, { 'Content-Type': 'text/html' });
            res.end(`
                <h1 style="color: blue; font-size: 24px; font-weight: bold;">Sufficient
Memory:</h1>
                <h2 style="color: red; font-size: 32px;">${freeMem} GB</h2>
            `);
        } else {
            res.writeHead(200, { 'Content-Type': 'text/plain' });
            res.end("Not Sufficient Memory");
        }
    } else {
        res.writeHead(403, { 'Content-Type': 'text/plain' });
        res.end("You are not admin");
```

```
    }
});
server.listen(3000, () => { console.log("Server started");});
```

Task-7

Write node js script and json to perform below tasks.

1. **Write below object in txt file named s2.txt**
   {d:{a:10,b:20,c:[30,10]}}
2. **Read data from the same file and perform the below tasks.**
   a. **addition of a and b.**
   b. **subtraction of 2nd element of c and b. (Must be positive value)**
   c. **multiplication of elements of c.**
3. **Add the Output of addition, subtraction and multiplication below the object in s2.txt file.**

```
const fs = require('fs');
const filePath = 's2.txt';

const data = { d: { a: 10, b: 20, c: [30, 10] } };
fs.writeFileSync(filePath, JSON.stringify(data), 'utf8');
const fileData = JSON.parse(fs.readFileSync(filePath, 'utf8'));
const { a, b, c } = fileData.d;
const addition = a + b;
const subtraction = Math.abs(c[1] - b);
const multiplication = c[0] * c[1];

fs.appendFileSync(filePath, `\nAddition: ${addition}\nSubtraction:
${subtraction}\nMultiplication: ${multiplication}`, 'utf8');
```

Task-8

Write a Node.Js program for following action

1. Write a file having five random elements separated by white space in .txt file.

2. append sorted array of these 5 elements in same file along with message : “Sorted array:” in new line.

3. Find maximum number from that and append with message “maximum number=” in same file.

```
const fs = require('fs');
fs.writeFileSync('num.txt', "20 30 13 7 22");
var d=fs.readFileSync("num.txt","utf-8");
d1= d.split(" ")
const sn = d1.sort((a, b) => a - b)
fs.appendFileSync('num.txt', `\nSorted array: ${sn}`);
const max = Math.max(...sn);

fs.appendFileSync('num.txt', `\nmaximum number= ${max}`);
```

Task -9

Write node.js script to print “Welcome to Home Page” with two links containing two pages named as “About Us” and “Contact Us” on home page of server. If user request for About Us page it should display “Welcome to LJ University” in bold font-style with blue color and if user request for Contact Us page it should display “Email:abc@ljinstitutes.edu.in” in italic font-style with red color if any other request is requested it shows “Page not found” message in plaintext.

```
const http = require('http');

const server = http.createServer((req, res) => {
   res.writeHead(200, { 'Content-Type': 'text/html' });

   if (req.url === "/") {
      res.end(`
         <h1>Welcome to Home Page</h1>
         <a href="/about">About Us</a>   <a href="/contact">Contact Us</a>
      `);
   } else if (req.url === "/about") {
      res.end(`<h1 style="color: blue; font-weight: bold;">Welcome to LJ University</h1>`);
   } else if (req.url === "/contact") {
      res.end(`<h2 style="color: red; font-style: italic;">Email: abc@ljinstitutes.edu.in</h2>`);
   } else {
      res.writeHead(404, { 'Content-Type': 'text/plain' });
      res.end("Page not found");
   }
});

server.listen(3000, () => {console.log("Server started");});
```

Miscellaneous Tasks

Task-1

Write node js script to fetch values from url given below and display output as asked.
"https://www.google.com/exam.txt?c1=Hello&c2=FSD2 T1 Test&c3=Welcome to LJU#AllTheBest"
1) Data must be written as below in file named "exam.txt". File name must be fetched from the url given above.
Output:
Hello!
Welcome to LJU FSD2 T1 Test
#AllTheBest
2) Read content from file "exam.txt" and send response to server and display data in "/" page in same format as above but in H1 tag and in red color.
3) If any other page is requested it shows "Page not found" message in plain text.

```
const http = require("http");
const fs = require("fs");
const url = require("url");

const url1 =
"https://www.google.com/exam1.txt?c1=Hello&c2=FSD2+T1+Test&c3=Welcome+to+LJU#A
llTheBest";

const parsedUrl = url.parse(url1,true);
console.log(parsedUrl.pathname)

const c1 = parsedUrl.query.c1;
const c2 = parsedUrl.query.c2;
const c3 =  parsedUrl.query.c3;
const hash =  parsedUrl.hash;
const filename = "."+parsedUrl.pathname

const fileContent = c1+"!\n"+c3+" "+c2+"\n"+hash;
fs.writeFileSync(filename, fileContent);

const server = http.createServer((req, res) => {
   if (req.url === "/") {
      data= fs.readFileSync(filename, "utf-8")
      res.writeHead(500, { "Content-Type": "text/html" });
      res.end("<h1 style='color:red'><pre>"+data+"<pre></h1>");
   }
   else {
      res.writeHead(404, { "Content-Type": "text/plain" });
      res.end("Page not found");
   }});
server.listen(3000)
```

Task-2

Create JSON object in file which contains array of objects. Calculate perimeter of square and perimeter of circle by using side value and diameter value respectively. And append final answers in file.

```
const shape =   [ { name: "circle", diameter: 8}, {name: "square", side: 10}]

var ps=require("fs");

ps.writeFileSync("shape.txt",JSON.stringify(shape));
data=ps.readFileSync("shape.txt","utf-8");

b=JSON.parse(data);

if( b[0].name == 'circle'){
   var perimeter = (b[0].diameter/2) * 3.14 * 2 ;
   console.log(perimeter);
}
if ( b[1].name == 'square'){
   var peri = (b[1].side) *4  ;
   console.log(peri);
}
ps.appendFileSync("shape.txt","\nPerimeter of circle = "+ JSON.stringify(perimeter)+
"\nPerimeter of square = "+JSON.stringify(peri));

Output:
[{"name":"circle","diameter":8},{"name":"square","side":10}]
Perimeter of circle = 25.12
Perimeter of square = 40
```

Task:3

Write a NodeJS code to create a folder named tempReports and inside it, create a file called daily.txt containing the initial text "Report Start". Then, it should add three entries to the file: "Entry 1: Success", "Entry 2: Warning", and "Entry 3: Success". After writing, the program must read the file and count how many times the word "Success" appears. If the word "Success" appears three or more times, the program should delete the tempReports folder. Ensure all file operations are asynchronous.

```
const fs = require('fs');
const folderName =  'tempReports';
const fileName = folderName + '/daily.txt';
```

```
fs.mkdir("tempReports", (err) => {
  if (err) return console.log('Folder creation failed:', err);
  const initialText = 'Report Start\n----------\n';
  fs.writeFile("tempReports/daily.txt", initialText, (err) => {
    if (err) return console.log('Write failed:', err);
     const logs = 'Entry 1: Success\nEntry 2: Warning\nEntry 3: Success\n';
    fs.appendFile("tempReports/daily.txt", logs, (err) => {
      if (err) return console.log('Append failed:', err);
          fs.readFile("tempReports/daily.txt", 'utf-8', (err, data) => {
        if (err) return console.log('Read failed:', err);
        const count = data.split('Success').length - 1;
        console.log('Success Count:', count);

        if (count >= 3) {
          fs.unlink("tempReports/daily.txt", (err) => {
            if (err) return console.log('File deletion failed:', err);

            fs.rmdir("tempReports", (err) => {
              if (err) return console.log('Folder deletion failed:', err);
              console.log('File and folder deleted.');
            })        });
        } else {
          console.log('Condition not met. Nothing deleted.');
        }     });
    })  });
```

Task -4

Write a NodeJS code to calculate and display the profit or loss incurred by an investor who purchased 100 shares of a company's stock at Rs.50 per share and later sold them at Rs.60 per share. The script should utilize the HTTP module to create an HTTP server that serves the profit or loss details. If there is a profit, the message should be displayed in green color, and if there is a loss, it should be displayed in red color on server.

```
const http = require('http');

const quantity = 100;
const buyPrice = 50;
const sellPrice = 60;

const investment = quantity * buyPrice;
const returnAmount = quantity * sellPrice;
const result = returnAmount - investment;

const server = http.createServer((req, res) => {
```

```
  res.writeHead(200, { 'Content-Type': 'text/html' });

  let message = '';
  let color = '';

  if (result > 0) {
    message = `You made a profit of Rs.${result}`;
    color = 'green';
  } else if (result < 0) {
    message = `You incurred a loss of Rs.${Math.abs(result)}`;
    color = 'red';
  } else {
    message = 'No profit, no loss.';
    color = 'black';
  }

  res.write(`<h1 style="color: ${color};">${message}</h1>`);
  res.end();
});
server.listen(3000, () => {
  console.log('Server is running at http://localhost:3000');
});
```

Task-5

Create HTTP webpage on which home page will fetch json data, about page shows centrally aligned message “Hello from class” in red colour and cyan background using internal css styling. And any other page shows “404 error”.
(Render Response & Routing)

```
const http=require("http")
http.createServer((req,res)=>{

if(req.url=="/"){
const data={"name":"ABC","age":30}
res.writeHead(200,{"content-Type":"application/json"})
res.write(JSON.stringify(data))
res.end()
}
else if(req.url=="/about"){
res.writeHead(200,{"content-Type":"text/html"})

res.write(`<html>
                    <head>
                    <style>
```

```
                body { background-color: cyan;text-align: center;}
                h1 {color: red;}
                </style> </head>
        <body>
        <h1>Hello from class</h1>
        </body>
        </html>`)
res.end()
}
else{
res.writeHead(404,{"content-Type":"text/html"});
res.end("<h1>404 error</h1>")
}
}).listen(3000)
console.log("server running")
```

Task-6

Write a nodejs program which fetch filename from requested url and print that file's data on http web server. (Async)

```
var h=require("http");
var ps=require("fs");
var u=require("url");
var server=h.createServer(
   function(req,res)
   {
      var q=u.parse(req.url,true);
     Filename ="."+q.pathname;
     ps.readFile(Filename,function(err,data)
     {
      if(err){
         res.writeHead(404,{"content-type":"text/hplain"})
         res.end("Error che")
      }
      else{
         res.writeHead(200,{"content-type":"text/html"}); //text/plain gives program
         console.log(data)
         console.log("Hello")
         res.write(data);
         res.end()

        // return(res.end())
      }      });
     console.log("Hello")
});
```

```
server.listen(6050);
console.log("Hi") //
```

Task-7: Write a nodejs script to print query string of url on console as well as on file using ES6 callback.

```
var u=require("url");
var ps=require("fs");
var
adr1="https://www.google.com/search?q=url+module+in+node+js&rlz=1C1YTUH_enIN1039
IN1039&oq=url+modu&aqs=chrome.0.0i512j69i57j0i512l4j0i390l4.2814j0j7&sourceid=chro
me&ie=UTF-8";

var q1=u.parse(adr1,true);
var qdata=q1.query;
  console.log(qdata);
ps.writeFile("fsd2.txt",qdata.q,(err)=>
{
  console.log("completed");
});
```

Output:

```
[Object: null prototype] {
  q: 'url module in node js',
  rlz: '1C1YTUH_enIN1039IN1039',
  oq: 'url modu',
  aqs: 'chrome.0.0i512j69i57j0i512l4j0i390l4.2814j0j7',
  sourceid: 'chrome',
  ie: 'UTF-8'
}

completed
```

In file: url module in node js

Task -8

Create own Node.js module (t1.js).
1. Parses a given URL to extract query parameters.
2. Validates that all parameters (a, b, c, d) are non-negative integers.
3. Evaluates the mathematical expression: a*c - a/d + b
4. Returns the computed result or an error message if any parameter is negative.
The main script (t2.js) should require the module. Pass a sample URL ("http://example.com/calculate?a=20&b=30&c=40&d=-1") with query parameters (a=20, b=30, c=40, d=-1). Display the evaluation result.

t1.js

```
const url = require('url');
function evaluate(input) {
    const parsedUrl = url.parse(input, true);
    const query = parsedUrl.query; // Extract query parameters
    let a = parseInt(query.a);
    let b = parseInt(query.b);
    let c = parseInt(query.c);
    let d = parseInt(query.d);

    // Check if all parameters are >0
    if (a<0 || b<0 || c<0 || d<0 ) {
        return "Invalid input. Please provide valid numbers";
    }
    const result = a * c - a / d + b;
   console.log(`Result of expression (a*c - a/d + b) = ${result}`);
}
module.exports = evaluate;
```

t2.js

```
 const eval = require('./t1');
const sampleUrl = "http://example.com/calculate?a=20&b=30&c=40&d=-1";
// Get and display the result
console.log(eval(sampleUrl));
```

Task -9

Write code to perform the tasks as asked below.

1. **Add three buttons.**
2. **Increase button to increase the fonts. It should stop increasing the fonts when the font size reaches to 200px or stop button is clicked.**
3. **Stop button to stop increasing or decreasing the fonts.**
4. **Decrease button to decrease the fonts. It should stop decreasing the fonts when the font size reaches to 20px or stop button is clicked.**
5. **Increasing/decreasing interval is of 100 ms and Default font size = 50px**

----------Without clearInterval()----------

```
<body>
    <p id="p1" style="font-size: 50px;">Hello</p>
    <button onclick="changeFont(1)">Increase</button>
    <button onclick="changeFont(-1)">Decrease</button>
    <button onclick="stopped = true">Stop</button>
```

```
   <script>
      let font = 50, stopped = false, direction = 0;
      function changeFont(d) {
         if (stopped) { direction = d; stopped = false; return; }
         if ((d > 0 && font < 100) || (d < 0 && font > 15)) {
            font += d;
            document.getElementById('p1').style.fontSize = font + 'px';
            setTimeout(() => changeFont(d), 100);
         }       }
   </script>
</body>
----------------------------------------------With clearInterval()----------------------------------------------
<html>
   <body>
      <p id="p1" style="font-size: 50px;"> Hello</p>
      <button onclick="inc()">Increase</button>
      <button onclick="dec()">Decrease</button>
      <button onclick="stop()">stop</button>
      <script>
      font=50;
      function fun(font)  {
        document.getElementById('p1').style.fontSize=font+'px';
      }
      function inc(){
         test = setInterval(()=>{if(font<100){ fun(++font);} },100);
      }
      function dec(){
        test =setInterval(()=>{if(font>15){ fun(--font);}},100);
      }
      function stop() { clearInterval(test); }
   </script>
   </body></html>
```

Task - 10

Write a nodejs script to create own module to calculate reverse of a given number. That module should be used to check given number of which square of reverse and reverse of square is same. (ADAM NUMBER)
For Example, 12 (12^2 =144)
21 (21^2=441)
144 = reverse(441)

In rev.js file

```
function reversenum(num)
{
   let rev=0;
   while(num>0)  {
      rev=rev*10+(num%10);
      num=parseInt(num/10);
   }
```

```
    return rev;
}
function square(num1){
    return num1*num1;
}
function checknum(num2){
    a=square(num2);
    b=square(reversenum(num2));
    if(a==reversenum(b))
    {
        console.log("Number is equal")    }
    else
    {
        console.log("Number is not equal")    }}

module.exports.reversenum=reversenum;
module.exports.square=square;
module.exports.checknum=checknum;
```

In another file:

```
var d=require("./rev.js");
d.checknum(12);
```

Output:

Number is equal

Task -11

Write all necessary .js files to create module having a function to check numbers from 2 to 50 are prime number or not.

1.js

```
const PrimeNo = (num) =>{
    let temp = 0
    for(let i=2;i<num;i++)    {
        if(num%i==0)       {
            temp++;
        }
    }
    if(temp==0)    {
        return true;
    }
    else{
```

```
      return false;
   }
}
module.exports=PrimeNo;
```

2.js

```
var PrimeNumber = require("./31.js")
for(i=2;i<=50;i++){
let x=PrimeNumber(i);
if(x==true){
   console.log(i+" Prime Number");
}
else{
   console.log(i+" Not a Prime Number")
}
}
```

Output:

2 Prime Number

3 Prime Number

4 Not a Prime Number

5 Prime Number ……. upto 50

Task -12

Write a node.js script to find all prime no.s between 1-50 using external module having a function checkPrime(). This function returns Boolean value on the basis of a no. is prime or not prime. Write all necessary .js files.

// primeUtils.js

```
function checkPrime(num) {
if (num <= 1) {
return false;
}
for (let i = 2; i <= Math.sqrt(num); i++) {
if (num % i === 0) {
return false;
}
}
return true;
}
module.exports = {
checkPrime,
```

```
};
```

findprime.js

```
const { checkPrime } = require('./primeUtils');
// Find and print prime numbers between 1 and 50
for (let num = 1; num <= 50; num++) {
if (checkPrime(num)) {
console.log(num);
} }
```

Task -13

Create http webpage and display message "Welcome to Khushbu mam's class" in h1 tag after 10 seconds.

----------Client side Delay----------

```
const http = require('http');
const server = http.createServer((req, res) => {
   res.writeHead(200, {'Content-Type': 'text/html'});
  res.write(`
   <p id="demo"></p>
   <script>
     setTimeout(() => {
       document.getElementById("demo").innerHTML= '<h2 style="color:tomato"> Welcome
to Khushbu mam\'s class </h2>'

       },10000)
       </script>`)
   res.end() });
const port = 3000;
server.listen(port, () => {
  console.log(`Server running at http://localhost:${port}/`)});
```

----------Server Side Loading Delay----------

```
const http = require('http');
const server = http.createServer((req, res) => {
  setTimeout(() => {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.end('<h2 style="color:tomato"> Welcome to Khushbu mam\'s class</h2>');
  }, 10000); // 10000 milliseconds delay });
const port = 3000;
server.listen(port, () => {  console.log(`Server running at http://localhost:${port}/`)});
```

Output Will Display after 10 second

Task -14: Write example as asked below

1. Create one CSV(.csv) file with minimum two lines of data and copy the file content in JSON (.json) file. Read the json file data and print the data in console. (For Reference)

2. Write simple html code and create one file named "h1" with .html extension.

3. Write simple JSON string with two properties name and branch to .json file. Read the file data and print the value of name in console.

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

// 1. CSV to JSON conversion
// Sample CSV data
const csvData = `Name,Age,Location
John,30,New York
Jane,25,Los Angeles`;

// Write sample CSV data to a file
fs.writeFileSync("test.csv", csvData);
console.log('CSV file "test.csv" created.');

// CSV to JSON conversion

const [headers, ...lines] = fs.readFileSync("test.csv", "utf-8")
  .trim()
  .split("\n")
  .map(line => line.split(","));

const jsonArray = lines.map(values => Object.fromEntries(headers.map((key, i) => [key, values[i]])));
//Use Object.fromEntries() for cleaner key-value pairing & map() instead of manual looping
fs.writeFileSync("test.json", JSON.stringify(jsonArray, null, 2));
console.log("CSV successfully converted to JSON.");


// 2. Create HTML file
const htmlContent = "<html><body><h1 style='color:red'>Hello</h1></body></html>";
fs.writeFileSync("h1.html", htmlContent);
console.log('HTML file "h1.html" created.');

// 3. Write JSON file
const jsonData = { "name": "LJU", "branch": "CSE" };
fs.writeFileSync("xyz.json", JSON.stringify(jsonData, null, 2));
console.log('JSON file "xyz.json" created.');

// Read and print JSON file data
const jsonDataFromFile = fs.readFileSync("xyz.json", "utf-8");
const parsedData = JSON.parse(jsonDataFromFile);
console.log('Value of "name" in JSON file:', parsedData.name);
```