**Task-1: URL Parsing and Manipulation**

**Aim:**

**1. Write a program that accepts a URL as user input and uses the url module to parse it. Display the protocol, host, path, and query parameters separately.**

**2. Implement a function that takes a base URL and a relative path as input, and uses the url module to resolve and display the absolute URL.**

**Code 1:**

const url = require('url');

const readline = require('readline');

const rl = readline.createInterface({

  input: process.stdin,

  output: process.stdout

});

rl.question('Enter a URL: ', (urlString) => {

  // Parse the URL

  const parsedUrl = new URL(urlString);

  // Extract the protocol, host, path, and query parameters

  const protocol = parsedUrl.protocol;

  const host = parsedUrl.host;

  const path = parsedUrl.pathname;

  const query = parsedUrl.searchParams;

  // Display the parsed components

  console.log('Protocol:', protocol);

  console.log('Host:', host);

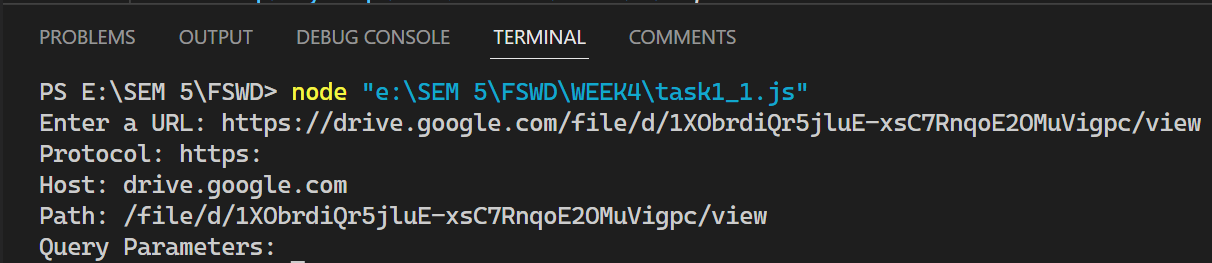
  console.log('Path:', path);

  console.log('Query Parameters:', query.toString());

  rl.close();

});

**Output 1:**

****

**Code 2:**

const url = require('url');

const readline = require('readline');

const rl = readline.createInterface({

  input: process.stdin,

  output: process.stdout

});

function resolveAbsoluteURL(baseURL, relativePath) {

  // Using the url.resolve() method to combine the base URL and relative path

  const absoluteURL = url.resolve(baseURL, relativePath);

  console.log('Absolute URL:', absoluteURL);

  rl.close();

}

// Get user input for the base URL and relative path

rl.question('Enter the base URL: ', (baseURL) => {

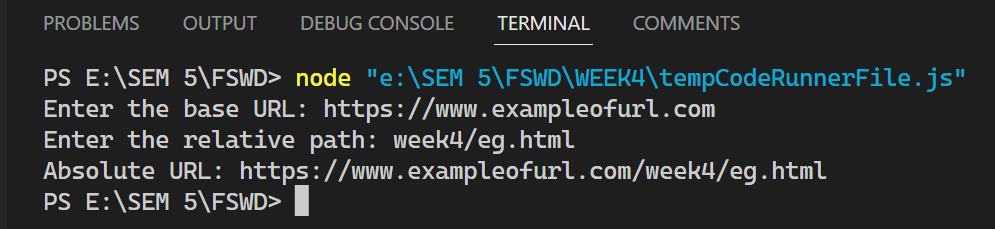
  rl.question('Enter the relative path: ', (relativePath) => {

    resolveAbsoluteURL(baseURL, relativePath);

  });

});

**Output 2:**

****

**Task 2: Query String Operation**

**Aim:**

**Write a Node.js program that takes a URL with a query string as input and extracts the key-value pairs from the query string using the querystring module. The program should display the extracted key-value pairs as output.**

**Code:**

const fs = require('fs');

const readline = require('readline');

const rl = readline.createInterface({

  input: process.stdin,

  output: process.stdout

});

function checkFileExists(filePath) {

  fs.access(filePath, fs.constants.F\_OK, (err) => {

    if (err) {

      console.error(`Error: File '${filePath}' does not exist.`);

    } else {

      console.log(`Success: File '${filePath}' exists.`);

    }

    rl.close();

  });

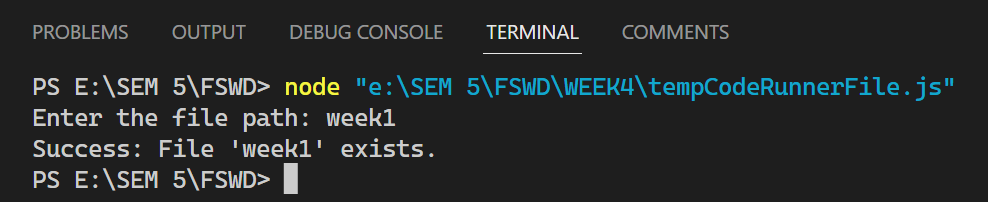
}

rl.question('Enter the file path: ', (filePath) => {

  checkFileExists(filePath);

});

**Output:**

****

**Task 3: Path Operations**

**Aim:**

**1. Create a program that accepts two file paths as input and uses the path module to determine if they refer to the same file.**

**2. Implement a function that accepts a file path as input and uses the path module to extract the file extension. Display the extracted extension to the user.**

**Code 1:**

const path = require('path');

const readline = require('readline');

const fs = require('fs').promises;

const rl = readline.createInterface({

  input: process.stdin,

  output: process.stdout

});

async function areFilesSame(file1, file2) {

  try {

    // Get the absolute paths of the files

    const absolutePath1 = path.resolve(file1);

    const absolutePath2 = path.resolve(file2);

    // Check if the files exist

    const exists1 = await fs.access(absolutePath1).then(() => true).catch(() => false);

    const exists2 = await fs.access(absolutePath2).then(() => true).catch(() => false);

    if (!exists1) {

      console.log(`File '${file1}' does not exist.`);

      return;

    }

    if (!exists2) {

      console.log(`File '${file2}' does not exist.`);

      return;

    }

    // Check if the paths are the same

    const isSame = absolutePath1 === absolutePath2;

    console.log(`The files ${file1} and ${file2} ${isSame ? 'refer to the same file.' : 'do not refer to the same file.'}`);

  } catch (error) {

    console.error('An error occurred:', error.message);

  }

  rl.close();

}

rl.question('Enter the first file path: ', async (file1) => {

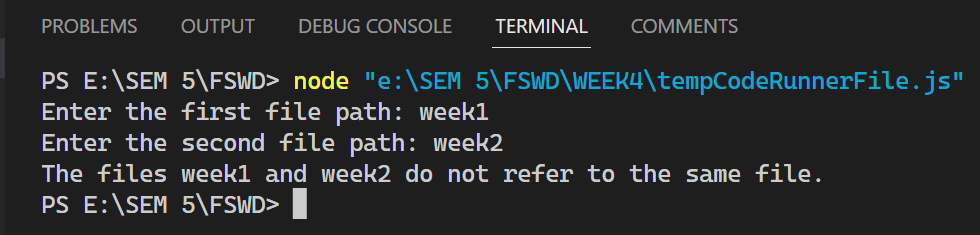
  rl.question('Enter the second file path: ', async (file2) => {

    await areFilesSame(file1, file2);

  });

});

**Output 1:**

****

**Code 2:**

const path = require('path');

const readline = require('readline');

const rl = readline.createInterface({

  input: process.stdin,

  output: process.stdout

});

function extractFileExtension(filePath) {

  // Using the path.extname() method to get the file extension

  const extension = path.extname(filePath);

  return extension;

}

rl.question('Enter the file path: ', (filePath) => {

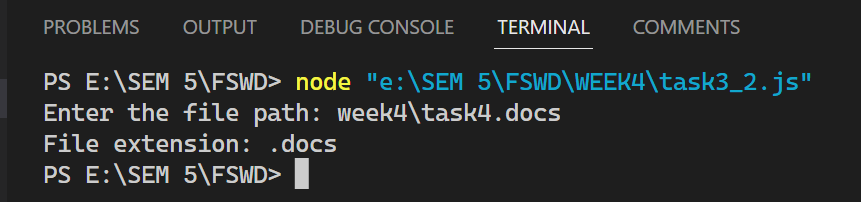
  const extension = extractFileExtension(filePath);

  console.log('File extension:', extension);

  rl.close();

});

**Output 2:**

****

**Task 4: File Paths and Operations**

**Aim:**

**1. Implement a program that accepts a file path as input and uses the path module to extract the directory name and base name. Display the extracted values separately.**

**2. Write a function that uses the fs module to check if a given file path exists. Display a success message if the file exists, or an error message if it doesn't.**

**Code 1:**

const path = require('path');

const readline = require('readline');

const rl = readline.createInterface({

  input: process.stdin,

  output: process.stdout

});

function extractDirectoryAndBaseName(filePath) {

  // Using the path.dirname() method to get the directory name

  const directoryName = path.dirname(filePath);

  // Using the path.basename() method to get the base name

  const baseName = path.basename(filePath);

  return { directoryName, baseName };

}

rl.question('Enter the file path: ', (filePath) => {

  const { directoryName, baseName } = extractDirectoryAndBaseName(filePath);

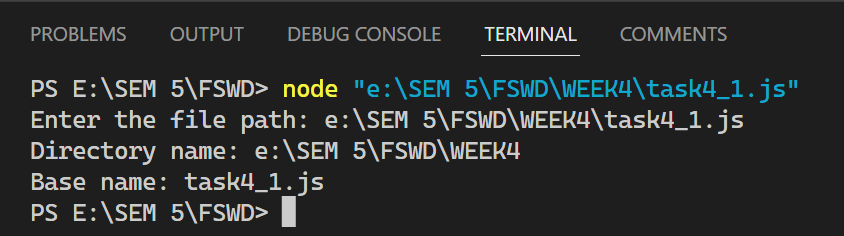
  console.log('Directory name:', directoryName);

  console.log('Base name:', baseName);

  rl.close();

});

**Output 1:**

****

**Code 2:**

const fs = require('fs');

const readline = require('readline');

const rl = readline.createInterface({

  input: process.stdin,

  output: process.stdout

});

function checkFileExists(filePath) {

  fs.access(filePath, fs.constants.F\_OK, (err) => {

    if (err) {

      console.error(`Error: File '${filePath}' does not exist.`);

    } else {

      console.log(`Success: File '${filePath}' exists.`);

    }

    rl.close();

  });

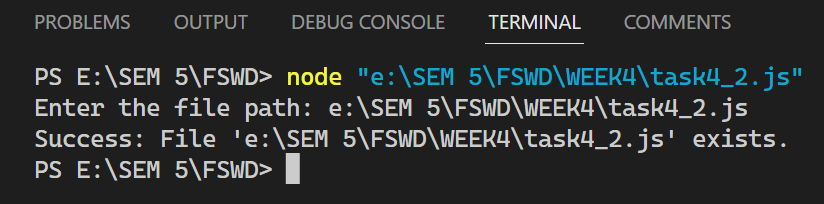
}

rl.question('Enter the file path: ', (filePath) => {

  checkFileExists(filePath);

});

**Output 2:**

****