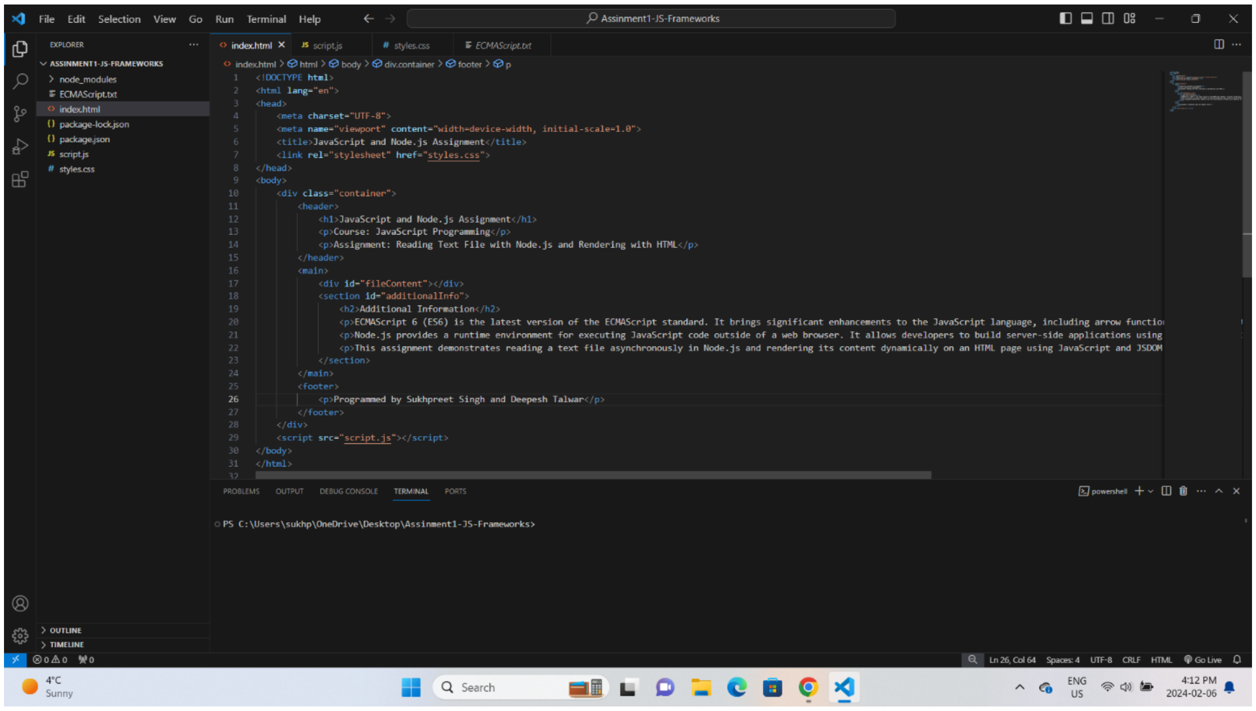
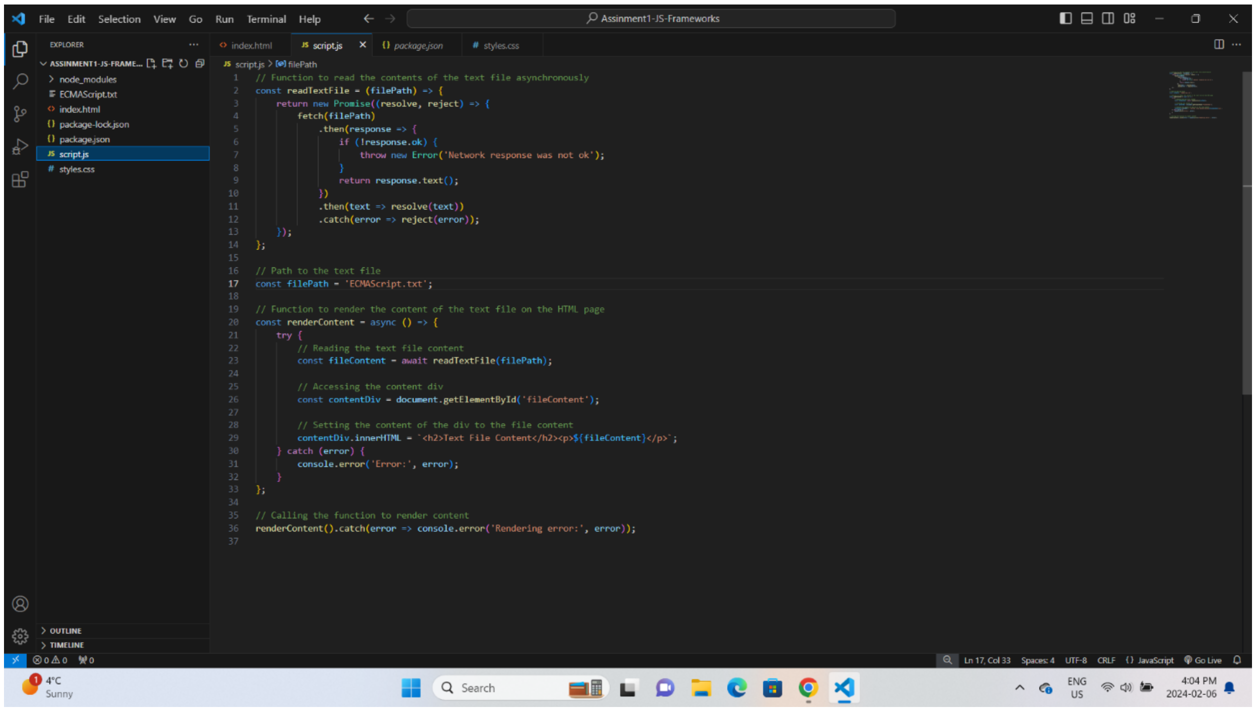
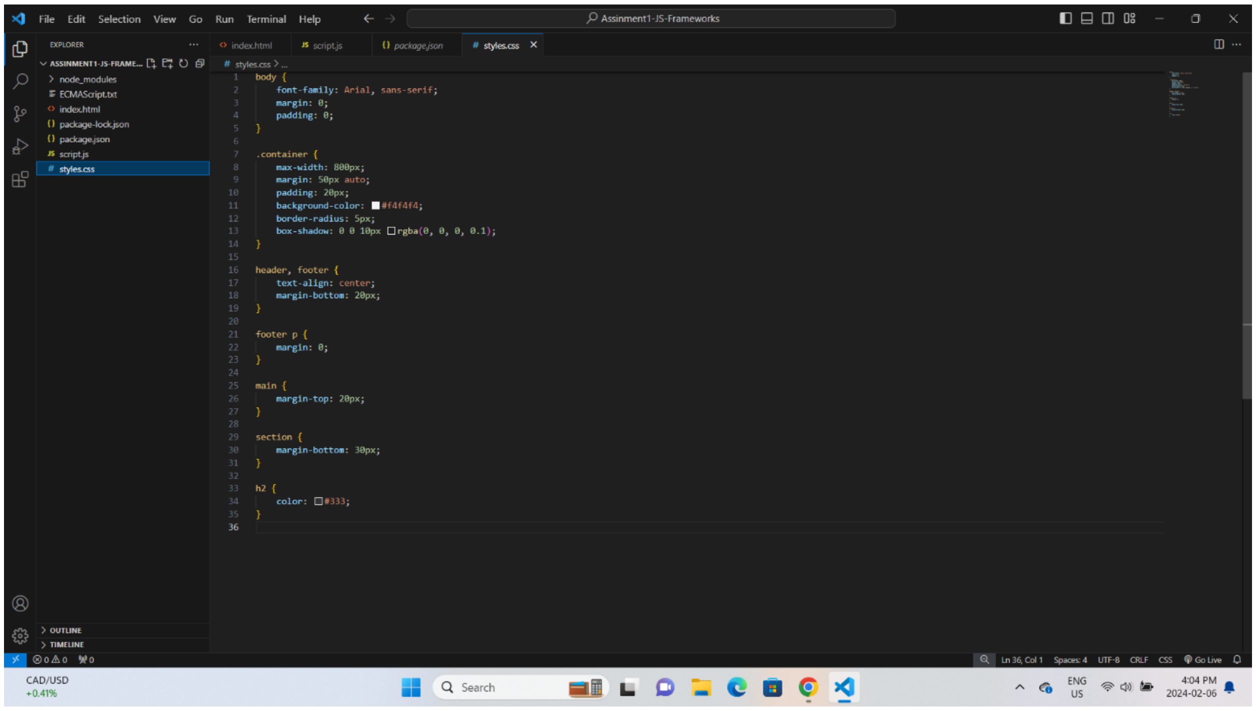


```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>JavaScript and Node.js Assignment</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <header>
            <h1>JavaScript and Node.js Assignment</h1>
            <p>Course: JavaScript Programming</p>
            <p>Assignment: Reading Text File with Node.js and Rendering with HTML</p>
        </header>
        <main>
            <div id="fileContent"></div>
            <section id="additionalInfo">
                <h2>Additional Information</h2>
                <p>ECMAScript 6 (ES6) is the latest version of the ECMAScript standard. It brings significant enhancements to the JavaScript language, including arrow functio
                <p>Node.js provides a runtime environment for executing JavaScript code outside of a web browser. It allows developers to build server-side applications using
                <p>This assignment demonstrates reading a text file asynchronously in Node.js and rendering its content dynamically on an HTML page using JavaScript and JSDOM
            </section>
        </main>
        <footer>
            <p>Programmed by Sukhpreet Singh and Deepesh Talwar</p>
        </footer>
    </div>
    <script src="script.js"></script>
</body>
</html>
```

```
PS C:\Users\sukhp\OneDrive\Desktop\Assinment1-JS-Frameworks>
```

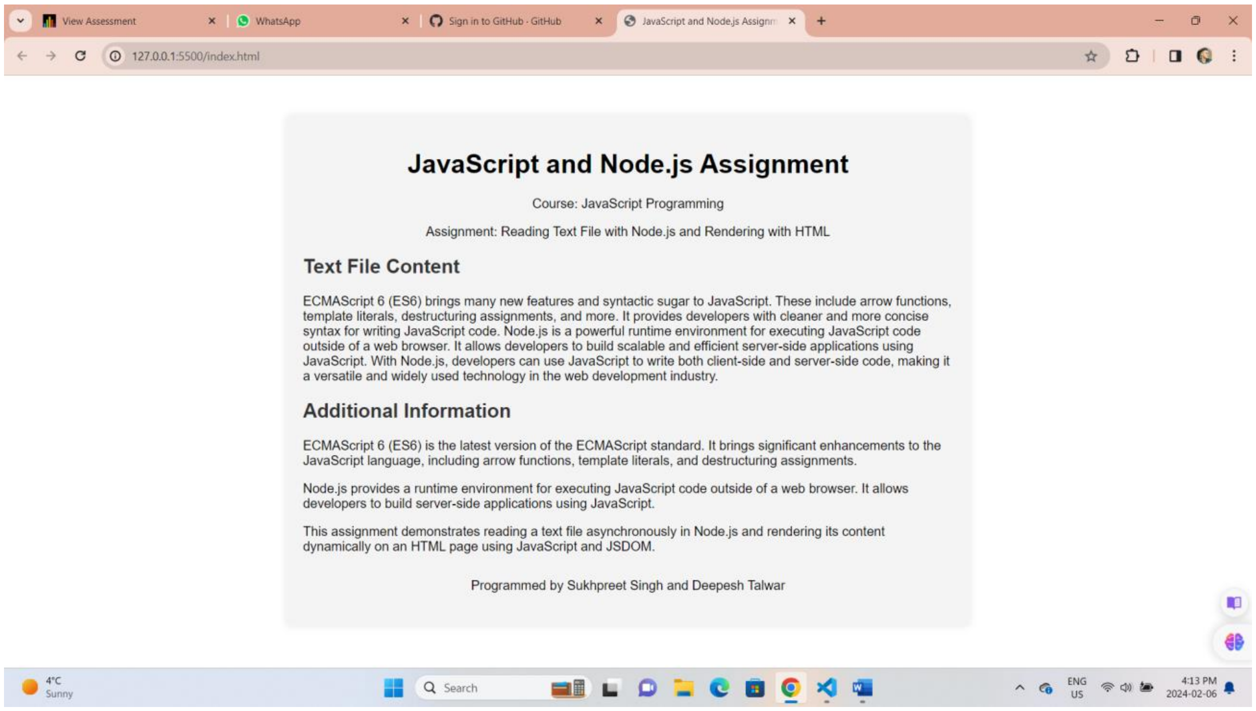


```javascript
// Function to read the contents of the text file asynchronously
const readTextFile = (filePath) => {
    return new Promise((resolve, reject) => {
        fetch(filePath)
            .then(response => {
                if (!response.ok) {
                    throw new Error('Network response was not ok');
                }
                return response.text();
            })
            .then(text => resolve(text))
            .catch(error => reject(error));
    });
};

// Path to the text file
const filePath = 'ECMAScript.txt';

// Function to render the content of the text file on the HTML page
const renderContent = async () => {
    try {
        // Reading the text file content
        const fileContent = await readTextFile(filePath);

        // Accessing the content div
        const contentDiv = document.getElementById('fileContent');

        // Setting the content of the div to the file content
        contentDiv.innerHTML = `<h2>Text File Content</h2><p>${fileContent}</p>`;
    } catch (error) {
        console.error('Error:', error);
    }
};

// Calling the function to render content
renderContent().catch(error => console.error('Rendering error:', error));
```



```css
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

.container {
    max-width: 800px;
    margin: 50px auto;
    padding: 20px;
    background-color: #f4f4f4;
    border-radius: 5px;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}

header, footer {
    text-align: center;
    margin-bottom: 20px;
}

footer p {
    margin: 0;
}

main {
    margin-top: 20px;
}

section {
    margin-bottom: 30px;
}

h2 {
    color: #333;
}

```

OUTPUT:



127.0.0.1:5500/index.html

JavaScript and Node.js Assignment

Course: JavaScript Programming

Assignment: Reading Text File with Node.js and Rendering with HTML

Text File Content

ECMAScript 6 (ES6) brings many new features and syntactic sugar to JavaScript. These include arrow functions, template literals, destructuring assignments, and more. It provides developers with cleaner and more concise syntax for writing JavaScript code. Node.js is a powerful runtime environment for executing JavaScript code outside of a web browser. It allows developers to build scalable and efficient server-side applications using JavaScript. With Node.js, developers can use JavaScript to write both client-side and server-side code, making it a versatile and widely used technology in the web development industry.

Additional Information

ECMAScript 6 (ES6) is the latest version of the ECMAScript standard. It brings significant enhancements to the JavaScript language, including arrow functions, template literals, and destructuring assignments.

Node.js provides a runtime environment for executing JavaScript code outside of a web browser. It allows developers to build server-side applications using JavaScript.

This assignment demonstrates reading a text file asynchronously in Node.js and rendering its content dynamically on an HTML page using JavaScript and JSDOM.

Programmed by Sukhpreet Singh and Deepesh Talwar