

Lesson 03 Demo 04

Implementing Form Validation

Objective: To demonstrate the implementation of JavaScript-based form validation for a login page, ensuring user authentication by verifying specified username and password criteria

Tools required: Visual Studio code and Node.js

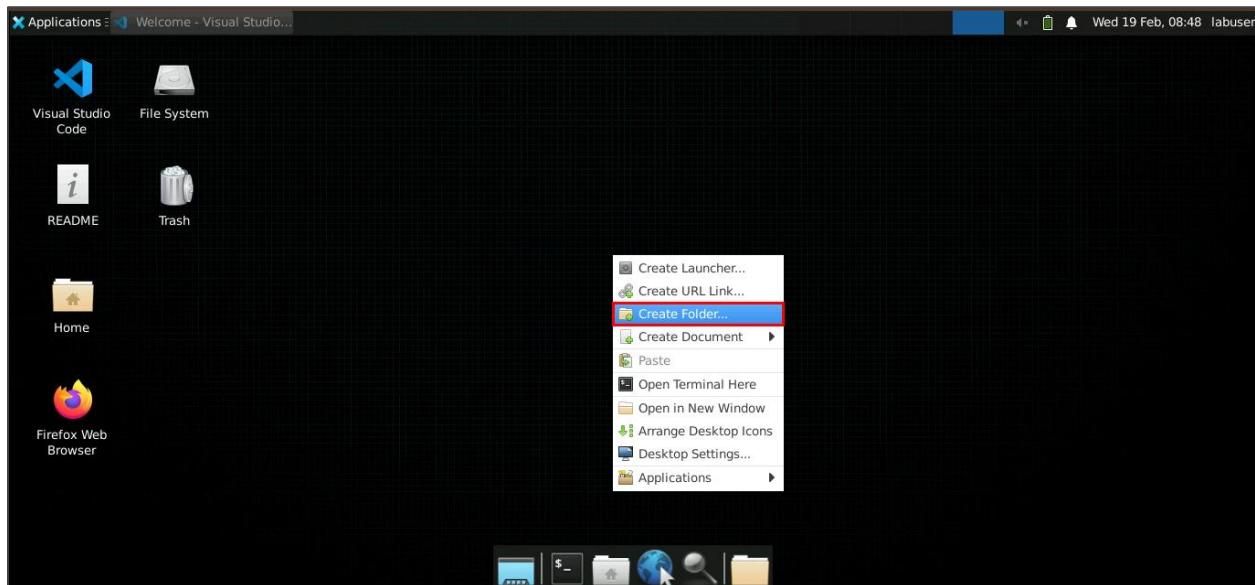
Prerequisites: Basic understanding of functions in JavaScript

Steps to be followed:

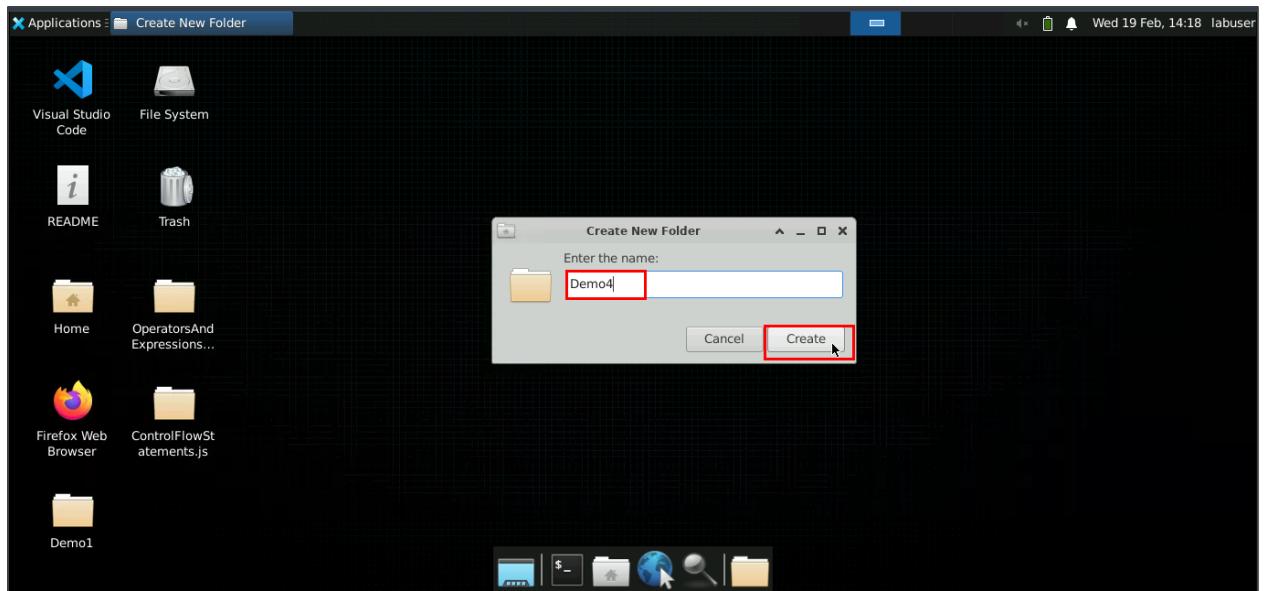
1. Create a folder
2. Execute the JavaScript file

Step 1: Create a folder

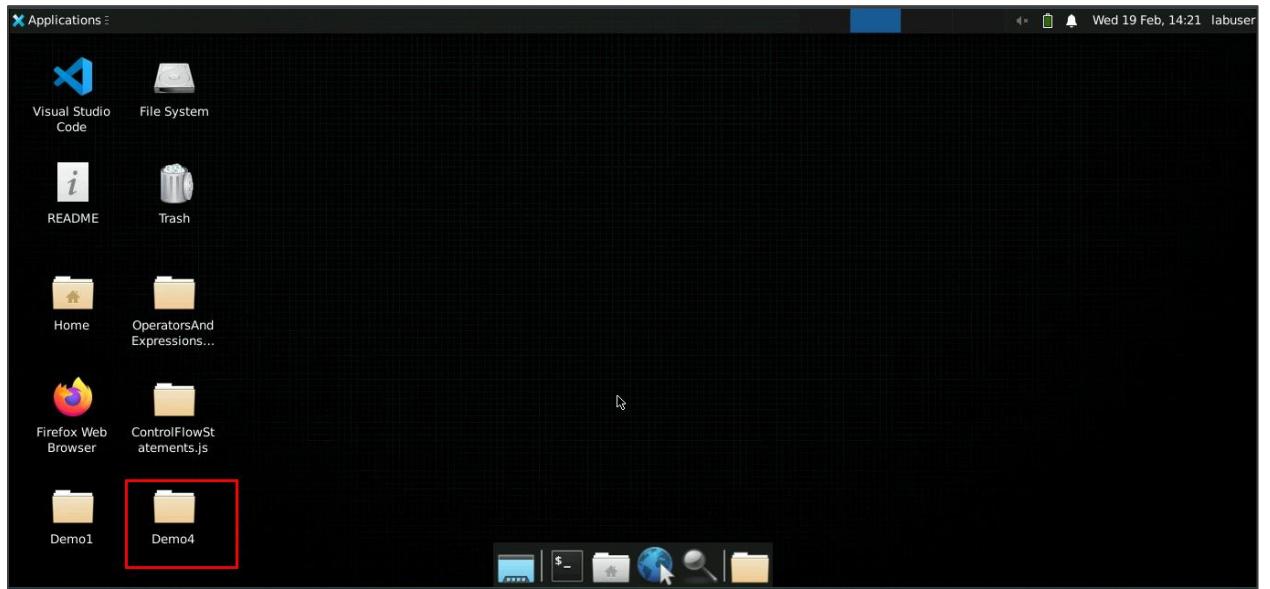
1.1 Right-click on the desktop and click on **Create Folder...**



1.2 Enter the name as **Demo4** and click on **Create**

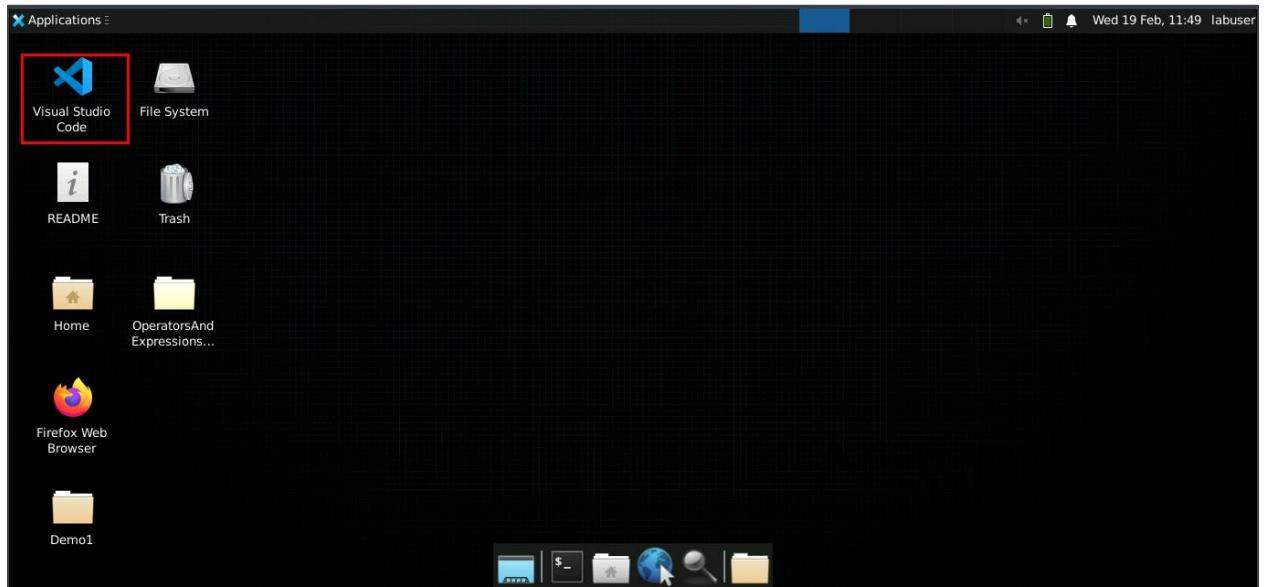


The **Demo4** folder gets created as shown below:

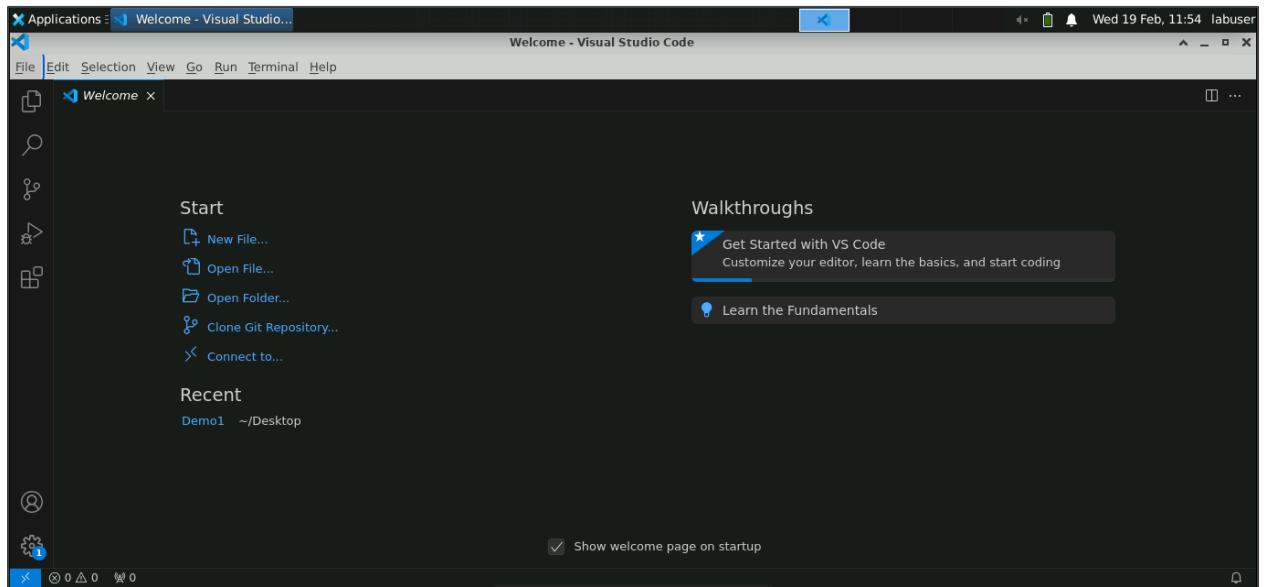


Step 2: Execute the JavaScript file

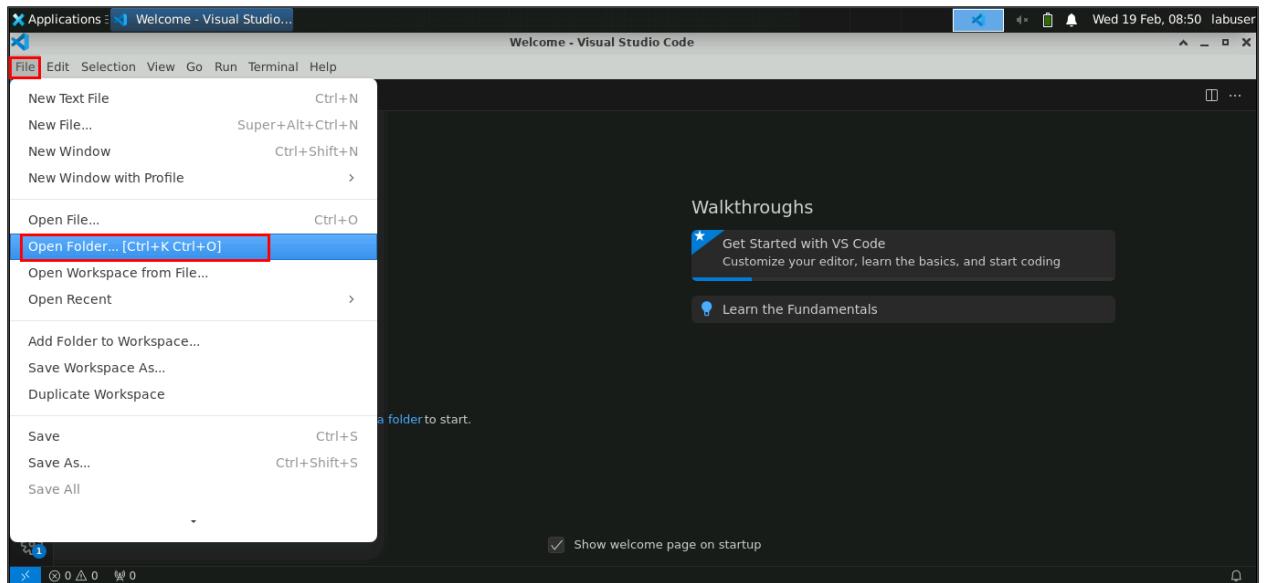
2.1 Double-click on the **Visual Studio Code** icon to open it



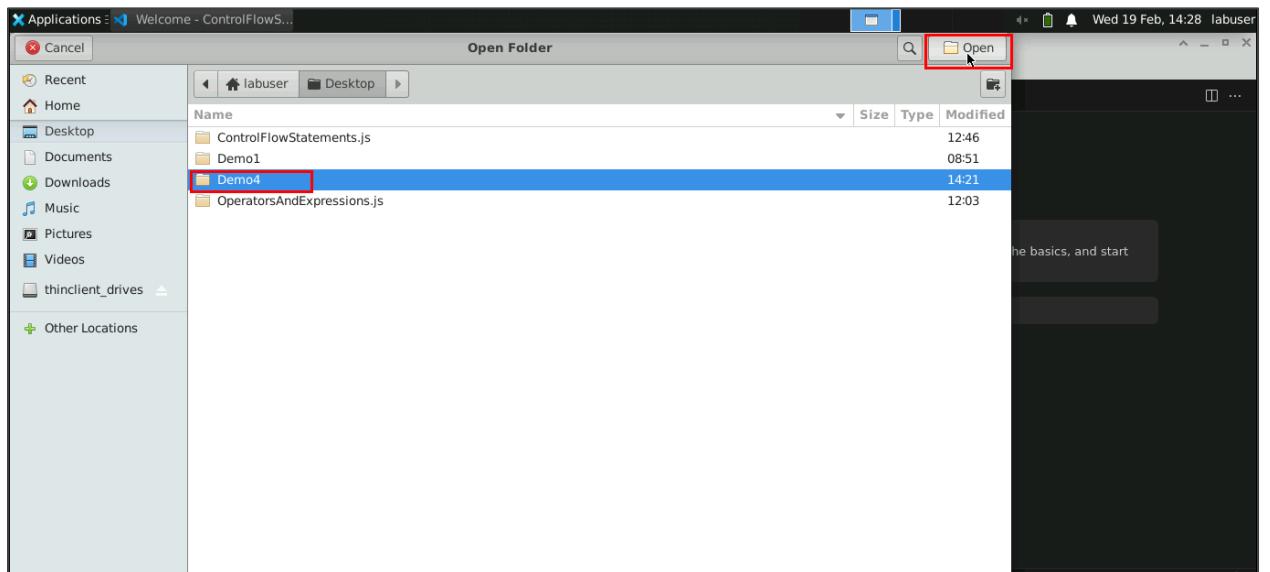
The **Visual Studio Code** opens as shown below:



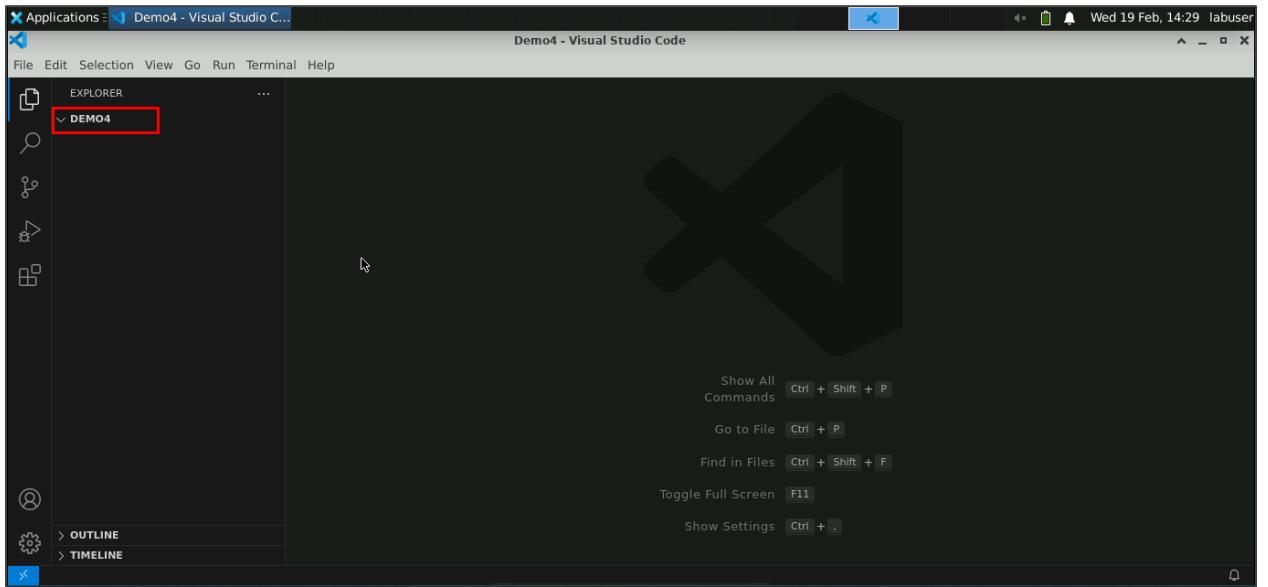
2.2 Click on **File**, then click on **Open Folder...**



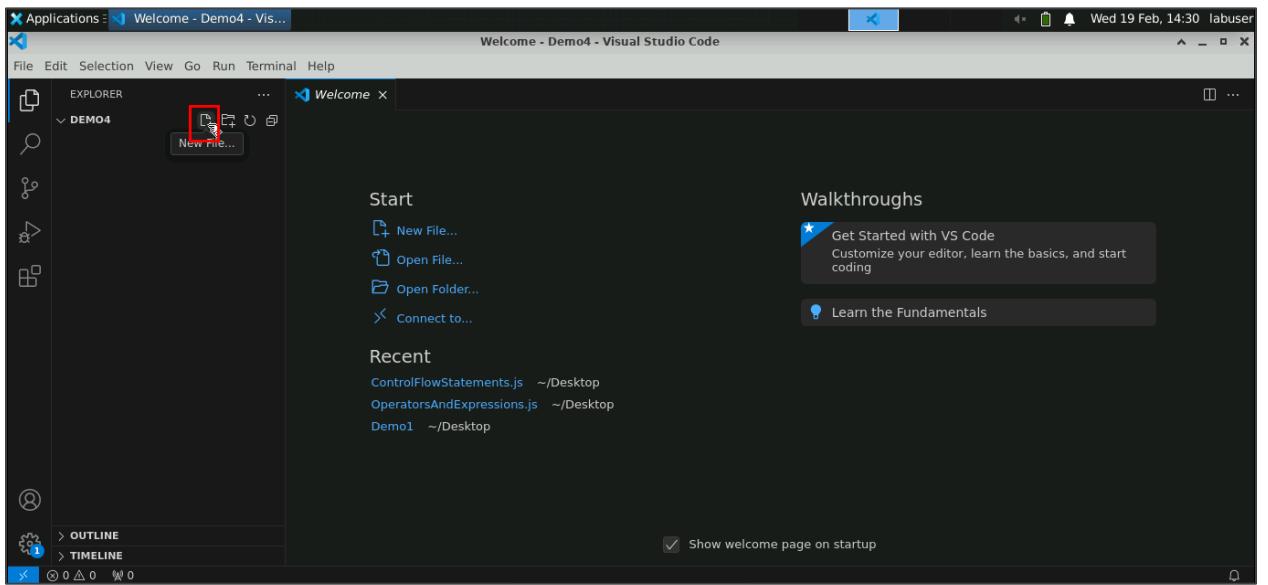
2.3 Select the **Demo4** folder and click on the **Open** icon to open the folder in Visual Studio Code



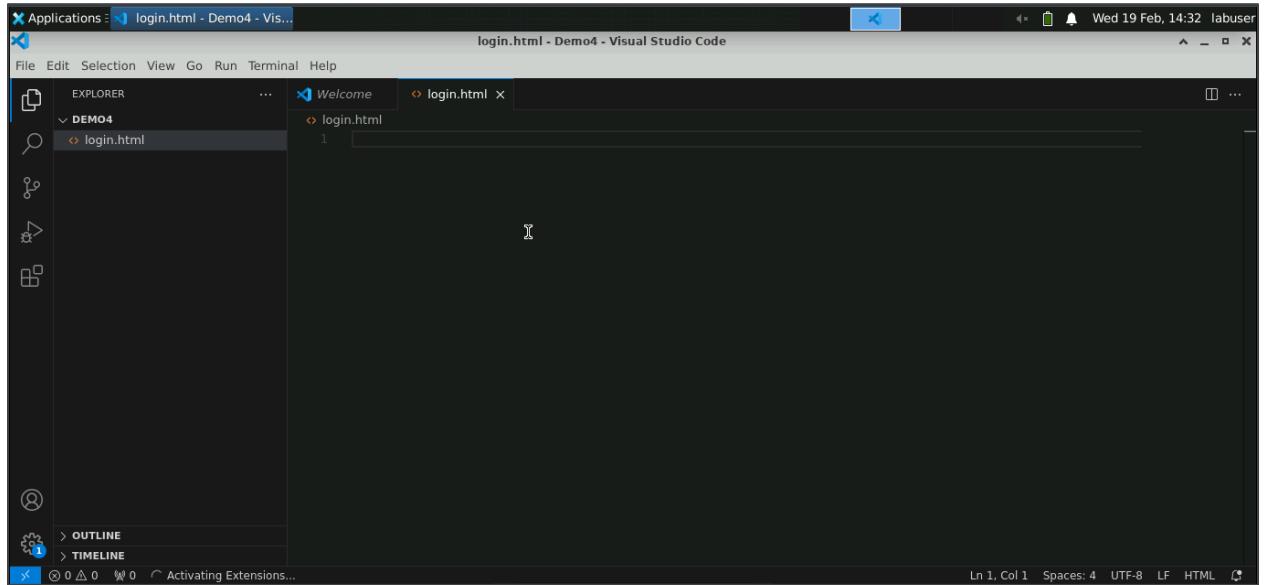
The folder opens in **Visual Studio Code** as shown below:



2.4 Click on the **File** icon to create a new file named **login.html**



The file gets created as shown below:



2.5 Enter the code below and save the file to create a form with validation:

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Form Validation</title>
</head>
<body>
    <h2>Login Form</h2>
    <form method="post" onsubmit="return validation()">
        <input type="text" id="user" placeholder="Username"><br>
        <input type="password" id="pwd" placeholder="Password"><br>
        <input type="submit" value="Submit">
    </form>

    <script>
        function validation() {
            let un = document.getElementById('user').value.trim();
            let pwd = document.getElementById('pwd').value.trim();

            if (!un || !pwd) {
                alert("Please enter both username and password");
                return false;
            }
        }
    </script>

```

```

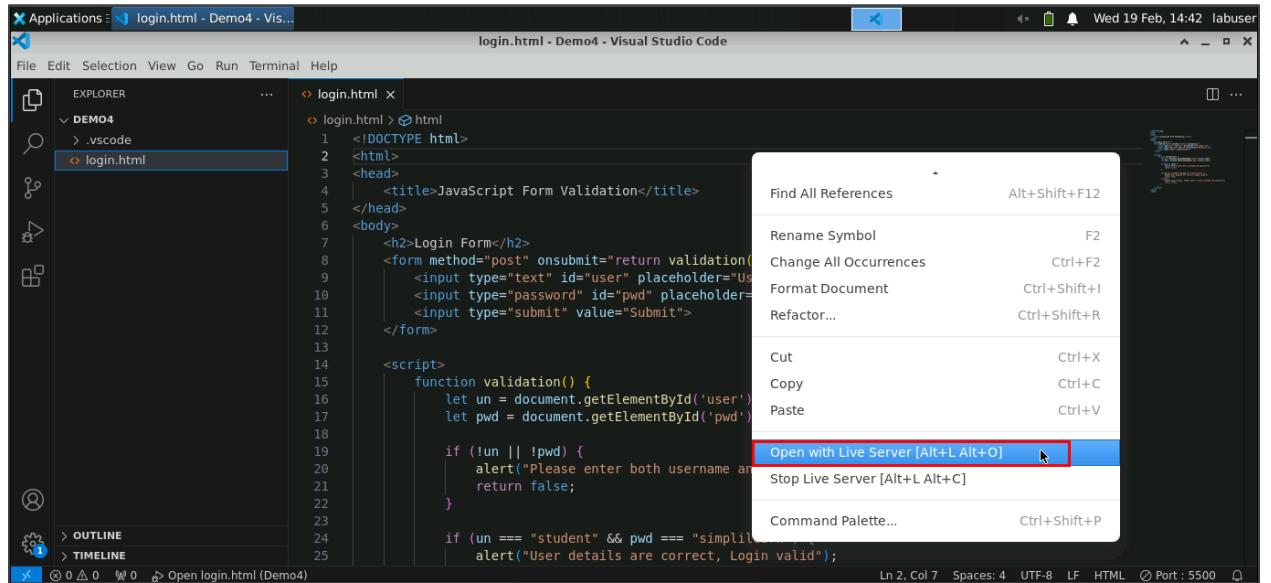
if (un === "student" && pwd === "simplilearn") {
    alert("User details are correct, Login valid");
    return true;
} else {
    alert("Login Failed, Please enter a valid username and password");
    return false;
}
}

</script>
</body>
</html>

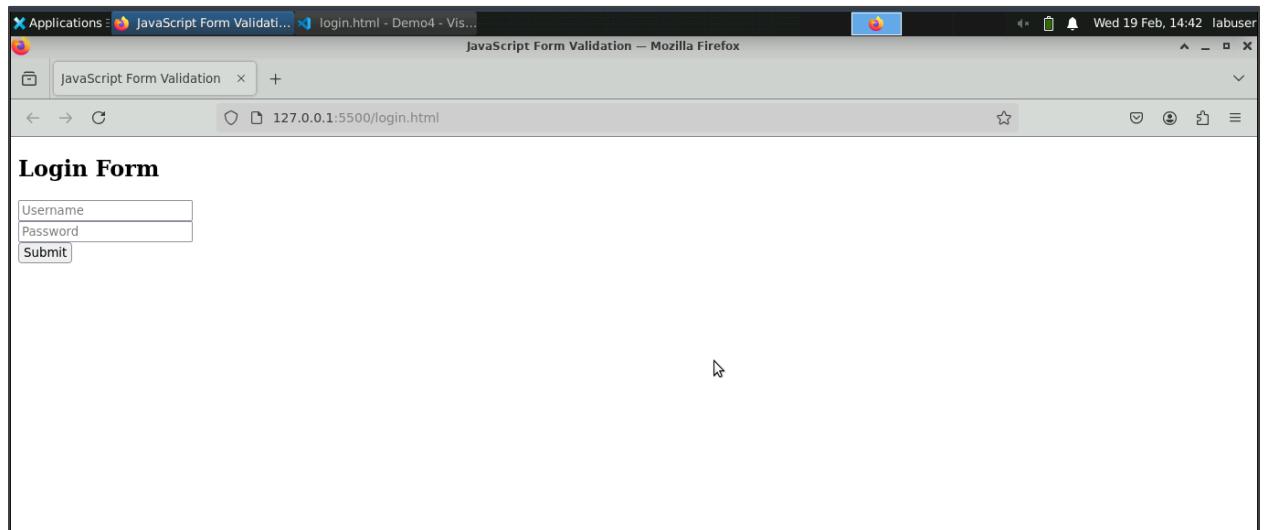
```

A screenshot of the Visual Studio Code interface. The title bar shows "Applications: login.html - Demo4 - Vis..." and "login.html - Demo4 - Visual Studio Code". The status bar at the bottom right indicates "Wed 19 Feb, 14:33 labuser", "Ln 35, Col 1", "Spaces: 4", "UTF-8", "LF", "HTML", and a file icon. The main area shows an HTML file with a red box highlighting the entire

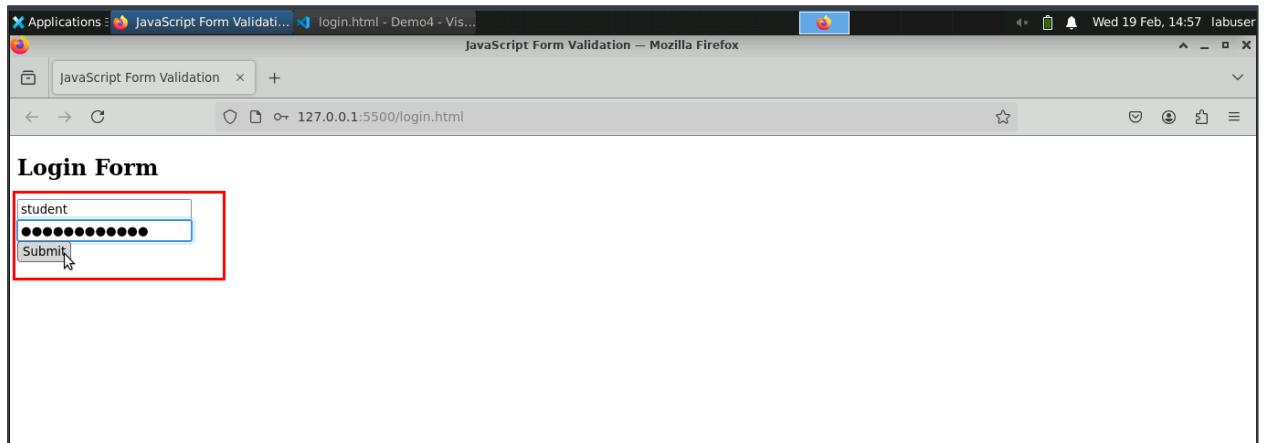
2.6 Right-click on the **login.html** file in the project and select the **Open with Live Server** option



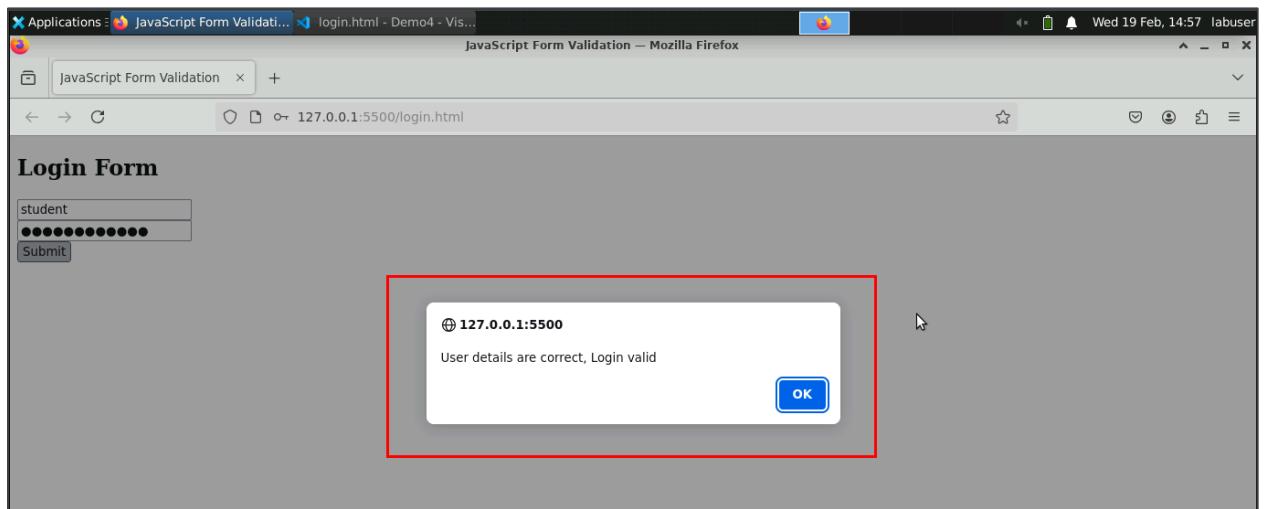
The server starts running as shown below:



2.7 Enter **student** in the username field, enter **simplilearn** in the password field, and click the **Submit** button



The form will validate the data and prompt the user if the details are correct, as shown below:



Note: The login page will be executed only once.

The above code demonstrates a basic HTML document with a JavaScript function for form validation. It validates a login form by checking if the entered username is **student** and the password is **simplilearn**, displaying an alert for a successful login or a failure message for incorrect credentials. Finally, it ensures proper validation logic to enhance user authentication.

By following the above steps, you have successfully implemented JavaScript-based form validation for a login page, ensuring authentication based on specified username and password criteria.