

Laboratorio Práctico: Desarrollar un Formulario de Encuesta de Retroalimentación de Productos Usando Eventos



Tiempo estimado necesario: 30 minutos

Lo Que Aprenderás

En este laboratorio, aprenderás a crear un formulario de retroalimentación de usuarios para los productos de una marca de lujo utilizando HTML y JavaScript. Descubrirás cómo diseñar un formulario estructurado para recopilar diversa información del usuario, como nombre, edad, correo electrónico, trabajo y retroalimentación sobre el producto. Verás cómo utilizar las funciones de JavaScript para gestionar el envío del formulario, capturar los valores de entrada y mostrar dinámicamente la retroalimentación proporcionada por el usuario en la página web. También observarás los mecanismos de manejo de eventos.

Objetivos de Aprendizaje

Después de completar este laboratorio, podrás:

- **Manejo de formularios y captura de entradas de usuario:** Explorar el proceso de crear un formulario interactivo utilizando elementos HTML, como entradas de texto, opciones de selección y áreas de texto, para recopilar diversa información del usuario, incluyendo nombre, edad, correo electrónico, trabajo, retroalimentación sobre el producto, y más.
- **Manipulación del DOM con JavaScript:** Aprender cómo JavaScript interactúa con el Modelo de Objetos del Documento (DOM) para recuperar valores de entrada de los elementos del formulario y actualizar dinámicamente el contenido de la página web. Esto incluye modificar el estilo de visualización de un elemento HTML y poblar su contenido basado en la entrada del usuario.
- **Manejo de eventos e interacciones del usuario:** Explorar los mecanismos de manejo de eventos en JavaScript, específicamente la implementación de oyentes de eventos para clics de botones y pulsaciones de la tecla **Enter** para activar acciones, como enviar los datos del formulario y mostrar la retroalimentación de manera dinámica, proporcionando una experiencia de usuario fluida.
- **Diseño de interacción de la interfaz de usuario:** Analizar la importancia de presentar la información del usuario recopilada en un formato legible en la página web. Los estudiantes obtendrán información sobre cómo mostrar dinámicamente la retroalimentación proporcionada por el usuario de manera estructurada dentro de la página web, mejorando la interfaz y la experiencia general del usuario.

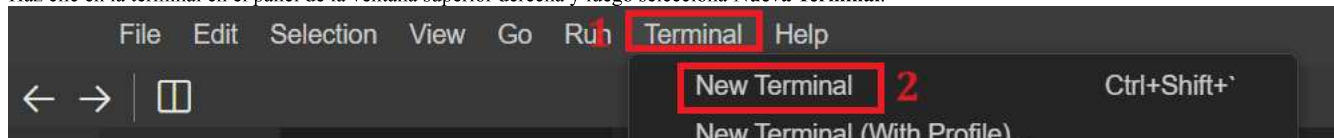
Requisitos Previos

- Conocimientos básicos de HTML.
- Comprensión básica de eventos y funciones en JavaScript.
- Navegador web con consola (Chrome DevTools, Firefox Console, etc.).

Paso 1: Configuración del entorno

1. Primero, necesitas clonar tu repositorio principal en el **Entorno de Skills Network** que has creado en el primer laboratorio y donde has subido todos tus archivos y carpetas de laboratorios anteriores. Sigue los pasos dados:

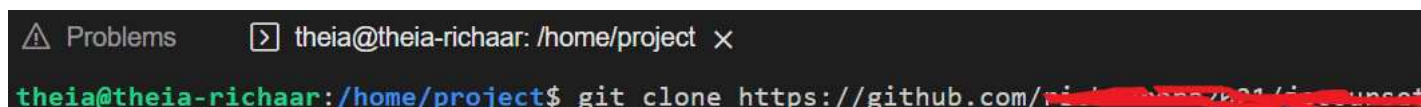
- Haz clic en la terminal en el panel de la ventana superior derecha y luego selecciona **Nueva Terminal**.



- Realiza el comando `git clone` escribiendo el comando dado en la terminal.

```
git clone <github-repository-url>
```

Nota: Coloca tu propio enlace de repositorio de GitHub en lugar de `<github-repository-url>`.



- El paso anterior clonará la carpeta de tu repositorio de GitHub en la carpeta del proyecto en el explorador. También tendrás múltiples carpetas dentro de la carpeta clonada.
- Ahora necesitas navegar dentro de la carpeta clonada. Para esto, escribe el comando dado en la terminal:

```
cd <repository-folder-name>
```



Nota: Escribe el nombre de tu carpeta clonada en lugar de <repository-folder-name>. Realiza git clone si has cerrado sesión en el **Entorno de Skills Network** y no puedes ver ningún archivo o carpeta después de iniciar sesión.

- Ahora selecciona **Nombre de la Carpeta Clonada**, haz clic derecho sobre ella y selecciona **Nueva Carpeta**. Ingresas el nombre de la carpeta como **productSurvey**. Esto creará la carpeta para ti. Luego selecciona la carpeta **productSurvey**, haz clic derecho y selecciona **Nuevo Archivo**. Ingresas el nombre del archivo como **product_survey.html** y haz clic en Aceptar. Esto creará tu archivo HTML.
- Ahora, selecciona nuevamente la carpeta **productSurvey**, haz clic derecho y selecciona **Nuevo Archivo**. Ingresas el nombre del archivo como **product_survey.js** y haz clic en Aceptar. Esto creará tu archivo JavaScript.
4. Crea una estructura básica de plantilla del archivo HTML agregando el contenido proporcionado a continuación.

```
<!DOCTYPE html>
<html>
<head>
  <title>Comentarios sobre Marca de Lujo</title>
</head>
<body>
  <h1>Bienvenido a los Comentarios sobre Productos de Marca de Lujo</h1>
  <h2>Proporciona Tu Comentario</h2>
  <label for="name">Nombre:</label><br>
  <input type="text" id="name" placeholder="Tu Nombre" required><br>
  <label for="age">Edad:</label><br>
  <input type="number" id="age" placeholder="Tu Edad" required><br>
  <label for="email">Correo Electrónico:</label><br>
  <input type="email" id="email" placeholder="Tu Correo Electrónico" required><br>
  <label for="job">Trabajo:</label><br>
  <input type="text" id="job" placeholder="Tu Trabajo" required><br>
  <label for="designation">Designación:</label><br>
  <input type="text" id="designation" placeholder="Tu Designación" required><br>
  <label for="productType">Tipo de Producto:</label><br>
  <select id="productType" required>
    <option value="perfume">Perfume</option>
    <option value="cream">Crema</option>
    <option value="oils">Aceites</option>
  </select><br>
  <label for="feedbackText">Comentarios:</label><br>
  <textarea id="feedbackText" rows="5" cols="30" placeholder="Ingresa tus comentarios aquí" required></textarea><br>
  <button id="submitBtn">Enviar Comentarios</button>
<hr>
<div id="userInfo" style="display: none;">
  <h2>Información del Usuario</h2>
  <p>Nombre: <span id="userName"></span></p>
  <p>Edad: <span id="userAge"></span></p>
  <p>Correo Electrónico: <span id="userEmail"></span></p>
  <p>Trabajo: <span id="userJob"></span></p>
  <p>Designación: <span id="userDesignation"></span></p>
  <p>Comentarios para <span id="userProductChoice"></span>: <span id="userFeedback"></span></p>
</div>
<script src="./product_survey.js"></script>
</body>
</html>
```

- **Formulario de comentarios del usuario:** La estructura HTML comprende campos de entrada para el nombre, edad, correo electrónico, trabajo, designación del usuario, un menú desplegable para seleccionar un tipo de producto (perfume, crema, aceites) y un área de texto para comentarios.

- **Botón de envío:** El botón “Enviar Comentarios” activa el procesamiento de los detalles ingresados por el usuario utilizando JavaScript al hacer clic.
- **Visualización dinámica:** Al enviar, JavaScript procesa la información ingresada y muestra dinámicamente los detalles proporcionados por el usuario, incluyendo nombre, edad, correo electrónico, trabajo, designación y comentarios sobre el tipo de producto elegido.
- **Visualización de sección oculta:** Inicialmente oculta, una sección debajo del formulario se vuelve visible después del envío, mostrando la información y comentarios del usuario y proporcionando una vista simplificada de los detalles ingresados.
- Para incluir el archivo js en **product_survey.html**, se utiliza la etiqueta de script en el archivo HTML por encima de la etiqueta `</body>` para incluir el archivo de script.

Nota: Cuando hayas pegado el código, guarda tu archivo.

Paso 2: Definición de variables y funciones

1. Crea una función llamada **submitFeedback** dentro de **product_survey.js** para tomar la retroalimentación del formulario de encuesta de usuarios creado en el archivo **product_survey.html**.

```
function submitFeedback() {
}
```

2. Variables como username, age, email, job, designation, productType y feedback se declaran y se les asignan valores de los respectivos elementos de entrada HTML utilizando `getElementById`. Por ejemplo:

```
const username = document.getElementById('name').value;
const age = document.getElementById('age').value;
const email = document.getElementById('email').value;
const job = document.getElementById('job').value;
const designation = document.getElementById('designation').value;
const productType = document.getElementById('productType').value;
const feedback = document.getElementById('feedbackText').value;
```

El código anterior recupera los valores utilizando elementos de entrada HTML con los IDs especificados y los almacena en variables.

3. A continuación, declara una variable llamada **submitButton** y inicialízala con `document.getElementById('submitBtn')`; Esta línea de código obtiene el elemento HTML con el ID 'submitBtn' utilizando su identificador único (`getElementById`) y lo asigna a la variable 'submitButton' para una interacción o manipulación posterior en JavaScript.

```
const submitButton=document.getElementById('submitBtn');
```

4. Ahora, asigna la función **submitFeedback** para que se ejecute cuando ocurra el evento **onclick** en el elemento HTML referenciado por la variable **submitButton**. Incluye el código dado fuera de la función en el archivo **product_survey.js**.

```
submitButton.onclick = submitFeedback;
```

5. Incluye el código dado en la función **submitFeedback** después de la declaración de la variable feedback para mostrar un mensaje de agradecimiento cuando el usuario envíe el formulario.

```
alert('Thank you for your valuable feedback')
```

Paso 3: Mostrar comentarios de usuarios en la página web

1. Como la función **submitFeedback** obtendrá la información de los comentarios de los usuarios en el segundo punto del paso 2, los detalles también deben mostrarse en la página web para informar al usuario sobre la información ingresada.
2. Para lograr esto, incluye el siguiente código dentro de la función **submitFeedback** para mostrar la información.

```
document.getElementById('userName').innerHTML = username;
document.getElementById('userAge').innerHTML = age;
document.getElementById('userEmail').innerHTML = email;
document.getElementById('userJob').innerHTML = job;
document.getElementById('userDesignation').innerHTML = designation;
document.getElementById('userProductChoice').innerHTML = productType;
document.getElementById('userFeedback').innerHTML = feedback;
```

3. Por ejemplo, la línea de código `document.getElementById('userName').innerHTML = username;` encuentra el elemento HTML con el ID **userName** y reemplaza su contenido HTML interno con el valor almacenado en la variable **username**. Actualiza dinámicamente el contenido mostrado dentro de ese elemento HTML específico.
4. En el código HTML en la línea 29, todo el elemento `div`, que se utilizará para mostrar información, está estilizado con la propiedad `'display: none'` para ser visible solo después de que el usuario haga clic en el botón de enviar. Para hacer visible esta parte después del clic en el botón, incluye el siguiente código dentro de la función **submitFeedback**.

```
document.getElementById('userInfo').style.display = 'block';
```

5. La línea de código anterior utiliza JavaScript para acceder a un elemento HTML identificado por el ID `'userInfo'`. Luego modifica la propiedad de estilo CSS `'display'` de ese elemento, estableciéndola en `'block'`, lo que altera su visibilidad en la página web para que se muestre como un elemento de bloque.

Paso 4: Implementar la tecla de presión para enviar comentarios

1. Incluye el siguiente código al final del archivo **product_survey.js** para asegurar que si el usuario presiona la tecla 'Enter', el formulario de comentarios se envíe, similar a hacer clic en el botón de enviar:


```
document.addEventListener('keydown', function(event) {
  if (event.key === 'Enter') {
    submitFeedback();
  }
});
```

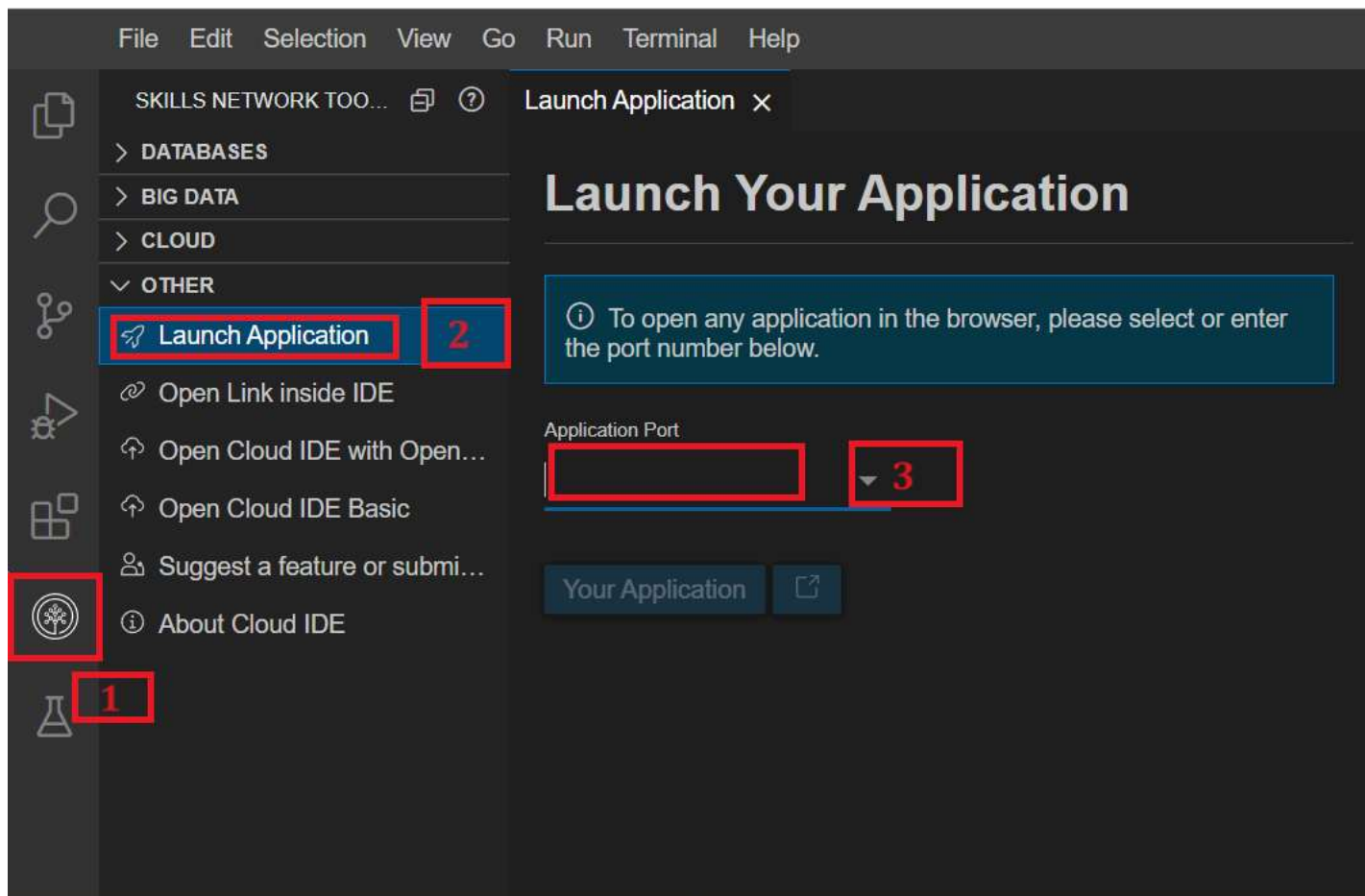
2. This code snippet adds an event listener to the entire document, listening for any keydown event. When the event occurs, it checks if the pressed key is 'Enter'. If it is, it calls the function `submitFeedback()`, executing the feedback submission functionality.

Step 5: Check the output

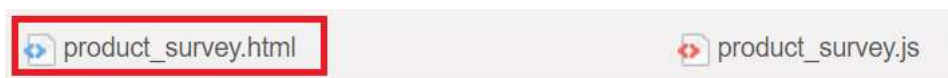
1. To view your HTML page, right-click on the **product_survey.html** file after selecting this file, then select "Open with Live Server".
 - The server should start on port 5500, indicated by a notification on the bottom right side.



2. Click on the Skills Network button on the left (refer to number 1), it will open the "Skills Network Toolbox". Click on Launch Application (refer to number 2).
From there, you enter the port no. as 5500 at number 3 and click on this button 



3. It will open your default browser where you will see **cloned-folder-name** folder name. Click on that **cloned-folder-name** folder name. After clicking you will see multiple folders name, among those folders name click on **productSurvey** folder. You will see files related to this folder where again you will click on **product_survey.html** file as shown below.



4. It will open the HTML page and you will see the feedback form as below.

Welcome to Luxury Brand Products Feedback

Provide Your Feedback

Name:

Age:

Email:

Job:

Designation:

Product Type:

Feedback:

5. Fill the feedback form and then click on **Submit Feedback** button.

6. As soon as you submit the feedback form, you will see an alert pop up box displaying the thank you message.

Thank you for your valuable feedback

7. After clicking the OK button in the alert popup box, the details of the feedback form will be displayed below the feedback form that the user filled before submitting. The screenshot below provides a demonstration of how the details will be displayed.

User Information:

Name: Peter

Age: 23

Email: peter@gmail.com

Job: Quality Assurance

Designation: Manager

Feedback for perfume: It has mild fragrance.

Note: After pasting the code, save your file. You can use any output method for this. If you edit your code, simply refresh your browser running through port number 5500. This way, there is no need to launch the application again and again.

Step 6: Perform Git commands

1. Perform `git add` to add the latest files and folder in the git environment.

```
git add --a
```

- Make sure the terminal has the path as follows:



2. Then perform `git commit` in the terminal. While performing `git commit`, terminal can show message to set up your `git config --global` for user.name and user.email. if yes, then you need to perform `git config` command as well for user.name and user.email as given.

```
git config --global user.email "you@example.com"
```

```
git config --global user.name "Your Name"
```

Then, perform commit command as given:

```
git commit -m "message"
```

3. Then perform `git push` just by writing given command in terminal.

```
git push origin
```

- After push command username and password will be asked to enter. You need to enter username for your GitHub account and password that you have created in first lab. After entering password and username, all of your latest folders and files will be pushed in your GitHub repository.

Practice task

1. In this task you need to create one more text area to ask users to share their experience with the product as well and show the result along with other details after clicking on **Submit Feedback** button.
2. To complete this task, you can use `<textarea>` tag in html file and give `id='userExperience'` as attribute for this tag.

3. Then access the details entered by user in javaScript file by accessing the input of this textarea using `document.getElementById('userExperience')` and store its value in one variable.
4. Then show the output after button is clicked with other user's entered details.

Summary

1. **Feedback form elements:** The HTML structure includes input fields for name, age, email, job, designation, a dropdown for product type selection, and a textarea for feedback submission. This comprehensive design ensures the collection of user-provided information.
2. **Dynamic user information display:** JavaScript function "submitFeedback()" processes user input upon button click or Enter key press, displaying entered details dynamically below the form, including name, age, email, job, designation, selected product type, and the provided feedback.
3. **Visibility toggle:** Initially hidden, a section below the form remains concealed until feedback submission. Upon submission, JavaScript alters its display to "block," showcasing the user's information and feedback for the selected product type.
4. **Event listeners implementation:** Event listeners are employed to trigger the feedback submission function, enabling user interaction via both the "Submit Feedback" button click and the Enter key press for a seamless experience.

© IBM Corporation. All rights reserved.