Práctica de Front-end

Crear una aplicación Front-end

En esta práctica añadiremos el componente que le faltaba a las prácticas anteriores la interactividad que nos provee JavaScript para manipular el DOM. De esta forma vamos a crear una aplicación funcional del lado del lado del cliente (front-end). Además, ahora podremos realizar solicitudes de HTTP para realizar operaciones de altas, bajas, cambios y consultas.

Descripción de la práctica

Completaremos la parte del front-end de la aplicación de registro de usuarios. Lo cual implicará realizar lo siguiente.

1. Autenticar a un usuario (login)
2. Hacer una petición al back-end de los usuarios y ver el listado de usuarios
3. Registrar un nuevo usuario (solicitar a back-end)
4. Editar un usuario
5. Consultar usuarios por nombre
6. Borrar algún usuario

A todo esto se le añade que se deben activar/desactivar botones o inputs en algunos momentos, ocultar o mostrar elementos del HTML, solicitar confirmación para realizar una acción.

Para tratar de hacer más realista la práctica se ha compartido el código del back-end que contiene una REST API (que desarrollamos en la práctica anterior).

Antes de iniciar

Descarga de Teams el código de la práctica 4 que es el de inicio. O si lo desaeas reutilizar tu código de prácticas anteriores adelante. Podrás quitar los botones que no se necesitan y añadir nuevos en caso de necesitarse. Tienes libertad de modificar el html siempre que cumplas con los requerimientos. **Todo el código que realices debe estar en archivos externos de JavaScript**. **No olvides importarlos en el html. El archivo de home.html cámbialo por index.html**

Además encontrarás una función que agrega un listener al evento de DOMCOntentLoaded. En este evento podrás escribir inicializaciones o algún otra asignación de eventos.

Otra cosa, a pesar que el back-end es el mismo de la práctica anterior, se hizo una modificación en la respuesta de usuarios para facilitar la paginación:

En caso de utilizar el proyecto de teams. Recuerda correr npm install para que te instale las dependencias.

login

Al loguearse obtén el valor del correo y del password y mándalo al back-end para obtener un token del usuario. Puedes también guardarlo en una cookie. Si la respuesta es exitosa debes redirigir a la sección de alumnos **window.location.href = "url"** la redirección conviene que sea relativa ("/consultas.html")

Registrar usuario

Lo primero que realizaremos será trabajar con el registro de un usuario para ello debemos hacer varias cosas.

1. No permitas que el botón de guardar este activo si el formulario no es válido (si faltan campos, o si se pusieron valores incorrectos). Esto implica la creación de un evento al formulario para atrapar cuando hay cambios (observa este ejemplo http://jsfiddle.net/54dftpL6/) lo que puedes hacer es con un **querySelectorAll** atrapar los input que sean inválidos usando la pseudoclase **:invalid** si la respuesta es una lista de 1 o más elementos significa que hay elementos inválidos y por lo tanto el botón de guardar debe estar desactivado de otra forma solo revisa que los password sean iguales y permite que se active el botón. Con una regla simple de CSS marca los

inputs inválidos con un borde rojo. (asegúrate que el input del correo electrónico sea tipo "**email**" y no "mail"

2. Al darle clic en submit (con todos los inputs correctos) debes atrapar el evento **submit** del formulario, crea un handler que reciba el event, usa **event.preventDefault()** para que no te recargue la página cuando sometes el formulario. Este handler debe atrapar todos los valores del formulario en un objeto que cumpla con lo que pide el back-end (**en el atributo sexo debe ser H o M**). Manda la información al back-end (**mira los anexos al final de este documento**) y en caso de que todo sea correcto muestra un mensaje de que indique que el usuario ha sido registrado de lo contrario mostrar un mensaje de error. (puedes usar las alertas de Bootstrap)

Administración de usuarios

Al cargar está página debes mostrar todos los usuarios que están registrados, para ello debes solicitarlos al backend y mapearlos a html. Cuando los muestras en pantalla hay 3 botones para el botón de la lupa es el más sencillo ya que solo basta con crear un elemento de tipo a href.

Para el icono de lápiz (editar) este se convierte en algo más retador. Observa que este HTML ya contiene un data-user con el contenido del usuario. Con el fin de cargar el contenido en el formo usaremos el evento de **show.bs.modal** junto con jQuery (lo veremos en una clase más adelante. Este evento se dispara cuando se muestra el modal. Deberás agregar la lógica de carga de formulario en la función **que carga cuando el html está listo**, verás que ya está el evento de jquery.

Lo mismo tendrás que repeitr para el modal de eliminar.

Editar usuario

Puedes copiar la ventana modal de registro de usuarios y traerla a la parte de administración de usuarios. Lo que importa aquí es que debes precargar los datos en los inputs y que no diga registrar usuario sino editar usuario. Desactiva el correo y el sexo para que solo se pueda cambiar todo lo demás. El botón que diga actualizar (no registrar). Si obtienes respuesta exitosa indicar con un mensaje "usuario actualizado". Cuando regreses a la ventana de la lista de usuarios ya debe aparecer la información actualizada.

Borrar usuario

Muestra una ventana modal con información del usuario a borrar y solicita una confirmación. Puedes tener una variable global que guarde el usuario seleccionado para que al darle clic en aceptar mandes el correo del usuario a eliminar. O seguir la lógica de consumir el email a través de data-email usando el event.target. Recuerda en asignar el evento al botón de borrar Al regresar al html ya debe estar borrado. Muestra mensaje de borrado o no encontrado según el caso.

Detalle de usuarios

Crea un archivo de detalle.html, que se abrirá cuando le den clic en la lupa, aquí se mostrará el detalle del usuario incluye la información adicional que provee el backend donde se incluye también la fecha y sexo. Puedes poner un botón que te permita regresar a la ventana anterior (de gestión de usuarios).

Puedes utilizar una cookie para especificar el usuario o puedes pasarlo como un query param en la url. Para esto cuando crees el icono de lupa en la vista de usuarios tendrás que especificar en el link el id. Por ejemplo: detalle.html?email=john@gmail.com. Dentro de detalle.html puedes usar el siguiente código para obtener el correo:

```
let urlParams = new URLSearchParams(window.location.search);
    let email = urlParams.get('email');
```

Filtro de usuarios

Muestra los usuarios de 3 en 3 usando los botones de paginación. Desactiva los botones de paginación que no deban estar activos y activa o añade los que falten.

Realiza filtros por nombre (solo primer nombre) pide al back-end que te de listado de usuarios siguiendo el esquema de paginación.

Hostear tu aplación

Sube todos tus archivos de front end (html, css, js) y back-end a la nube utilizando IBM Cloud – Cloud Foundry.

Recuerda modificar tu archivo manifest.yml los campos de name y host por el siguiente formado: iniciales-practica4.

Ejemplo: si te llamas Sergio perez entonces el valor que deberás poner es: sp-practica4.

Evaluación

En la entrega debes mandar todos tus archivos (recuerda no incluir node_modules) y un documento PDF que contenga tus conclusiones, la url de tu practica que apunta a IBM Cloud y una tabla donde indicas cuales requerimientos has cumplido.

Requerimiento	Valor	Realizado (Si/No)
1. Validación del formulario (el botón se activa cuando todo está correcto y se desactiva si no). Se marcan en rojo los que no están correctos.	10	
2. Es posible realizar el registro del usuario al back-end. Y muestra un mensaje de usuario registrado.	10	
3. Es posible loguearse (se guarda el token del usuario logueado en una cookie). Cuando todo es correcto me muestra un mensaje de bienvenida. En caso de error me muestra un mensaje de que ocurrió un error.	10	
4. Al entrar a la ventana de administración de usuarios es posible visualizar los usuarios que regresa el back-end (no los que estaban de ejemplo)	10	
5. Al dar clic en el botón de editar se abre el modal de	10	

edición con los datos precargados.		
6. Es posible realizar la actualización de la información correctamente y al regresar a la ventana es posible ver el dato actualizado.	10	
7. Es posible borrar un usuario, pero primero pregunta por confirmación de borrado.	10	
8. Al dar clic en la lupa es posible ver el detalle del usuario (redirecciona a otro archivo html)	10	
9. Es posible realizar el paginado (correctamente, mostrando el número de páginas correcto y activando desactivando botones prev y next) y poder realizar búsquedas por nombre.	20	
10. El proyecto se subió a la plataforma de IBM Cloud público de modo que es accesible y funcional desde internet.	10	

Si obtienes una calificación mayor a 100 te quedas con 100.

En el documento que entregues tu tabla también pon una reflexión sobre la práctica.