Proyecto FRONT-END

Realizado por:

- ☐ Iker García López
- ☐ Pablo Alcolea Aguilar
- ☐ Oier Barrutiabengoa Errandonea
- ☐ Mikel Elola Echechipia

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Descripción:

En esta primera parte del proyecto estamos diseñando una aplicación web, en el ámbito de front-end, que trata sobre la grabación de audios de voz. Para ello, vamos a utilizar el entorno “IntelliJ”, donde gracias al uso de node.js y express (plantillas EJS), podremos conectar a nuestro cliente con un servidor. Antes de meternos en el front-end, nuestro objetivo en la resolución de esta primera parte del proyecto es ofrecer cuatro diferentes funcionalidades:

- Grabar un audio
- Escuchar el audio
- Guardar el audio
- Conseguir enlace de un audio

A continuación, vamos a desarrollar cada una de estas.

1. Grabar un audio:

Nosotros, como cliente que accede a la aplicación web, vamos a poder grabar nuestras propias notas de voz. Para ello, contamos con el API HTML5Media. El formato de esta nota de voz estará dentro de un container ogg (audio vorbis).

Su duración máxima serán 300 segundos, que a medida que el audio esté siendo grabado, esta cantidad será actualizada a los segundos restantes.
Para poder iniciar la grabación de este, deberemos aceptar los permisos a nuestro micrófono mediante el comando getUserMedia,este como parámetro nos devolverá un stream.

A continuación, llamará a las funciones initAudio() e initRecord().
La primera función se encargará de inicializar el estado del audio a “cargado” o a “reproduciendo”. Esto será útil más adelante para, dependiendo del estado, poder controlar diferentes eventos.

La segunda función, a la que le hemos pasado el parámetro stream, tiene dos métodos diferentes. El primero de ellos hace que mientras se esté grabando el audio, toda la información se irá guardando en un array llamado “audioChunks”. El segundo método indica que una vez parada la grabación del audio, se genera un objeto de tipo blob, que a su vez llama a una función llamada loadBlob() que se encarga de conseguir la URL de este objeto.

Diferentes funciones han sido implementadas, como record() y stopRecording(), que inician la grabación y la detienen, además de contar con un temporizador. Este, cuando se está grabando, se activa y empieza a contar y cuando se detiene el audio, para. Para tratar sobre el temporizador del audio , también hemos definido la función convertirSegundosAMinutosYSegundos() que como su nombre indica, dados “x” segundos, los convierte a minutos y segundos.

2. Escuchar el audio:

Una vez guardado el audio, podremos reproducirlo. El botón de reproducir, al ser pulsado, llamará a la función playAudio() que reproducirá el audio grabado previamente. Si el audio está en reproducción, al pulsar el botón de nuevo, se llamará a la función stopAudio() que parará la reproducción.

3. Guardar el audio:

El audio se puede subir y almacenar. Para ello utilizaremos el botón de upload. Cuando se pulsa el botón, se llama a la función upload() que hace que se almacene el audio mediante una petición POST.

4. Conseguir el enlace del audio:

Al guardar el audio, el botón de copiar genera y copia en el portapapeles un enlace para acceder al audio guardado. (Posteriormente el audio sera implementado)

5.Avisar al usuario de que el link ha sido correctamente copiado:

Mediante el uso de las librerías de snackbar hemos hecho que la aplicación avise al usuario de que el link se ha copiado correctamente

Problemas

Al realizar el proyecto nos hemos dado cuenta de que en el código que nos proporcionaron para realizar las grabaciones de audio hay un fallo, al parar la grabación y al volver a realizarla posteriormente nos hemos dado cuenta de que solo se reproduce el último trozo de audio y no todo completo, en nuestra aplicación recomendamos realizar solo 1 grabación seguida y no realizar ninguna más.

Enlace de github:
https://github.com/RosherB/PracticaFrontEnd