Presentacion Tecnica - JocQuiz

Hecho por : Josthyn Loma y Kevin Hoyos

Problemas y soluciones

Problemas:

Hemos tenido varios problemas durante el proceso , los más destacados eran , o que desde vue no enviaba bien la información a la API o en symfony había problemas a la creación , normalmente salen errores al hacer fetch con el método POST. Salia este error porque no construía bien el json que tendría que recibir la API o desde backend pedía una json bastante complejo

Soluciones:

Encontraba una forma coherente en la que pueda enviar bien el json y utilizando la función JSON.stringify(array) para que se pueda enviar o en symfony(backend) para solucionarlo buscaba otra manera en la que pueda recibir el json que envía fetch correctamente

Problemas Actuales:

No está bien hecho el diseño, el administrador no funciona correctamente, y no hay control para introducir datos en crearQuiz y no comprueba si un campo de texto está vacío

Aspectos Técnicos - Frontend

- **Registro** : Está compuesta por dos archivos , una que está toda la información que saldrá por pantalla y que enviará los datos que introducimos con un POST y otro archivo en la que hará que que se muestre por pantalla
- **Login:** Con un fetch hará un POST que dará un mensaje si esta logueado o no, si esta logueado con pinia guardamos toda la información del usuario para crear una sesión
- **Admin**: Con un fetch POST correctamente dará un mensaje si eres admin o no, comprueba con un if si el mensaje llegado desde symfony es correcto o no, y guardará la información a pinia
- **Perfil**: Recoge toda la información de pinia para que se refleje en los campos de texto y con un método POST desde VUE y método PUT desde symfony se hará el cambio si el usuario desea cambiar su información.
- **Quiz Creados:** Cojera la información de los quiz que ha creado el usuario con un fetch GET y con un método DELETE podrá eliminar el quiz con la API que ha creado desde backend

Frontend

- **Quiz Jugados:** Fetch GET para recoger la información de los quiz que ha realizado el usuario
- **Jugar Quiz**: Fetch GET que recogerá las preguntas dependiendo del quiz y al terminar hara un POST a la api para enviar las respuestas
- **Crea Quiz** : Con un fetch POST conseguimos enviar todas las preguntas, respuesta y su correspondiente resultado. Para hacer bien este POST se ha tenido que construir la estructura del json y en el body escribir cómo se enviará sin necesidad de la variable datosEnvio
- **Página Inicia**: Compuesta por varios Gets que hara mostrar mejores jugadores,mejores cuestionarios y que tipos de cuestionarios hay
- **Mostrar Quiz:** Hará un get para recoger todos los cuestionarios de cierto tema

Aspectos Técnicos - Backend