

Consideraciones para evaluar el proyecto

PARTE INTERFACES

- Buenas prácticas CSS: Mantiene coherencia en nombrar las clases y ordena el código CSS
- Estructura coherente de las páginas
 - o Encabezado con logo
 - o Menú bien posicionado, con hover en sus elementos, etc.
 - o Contenido principal de cada página (indicado más abajo)
 - o Pie
- Combinación de colores adecuada.
- Las unidades de medida son adecuadas (no mezcla pixeles, con %, con em...)
- Tiene la interfaz diseñada de las páginas:
 - o Login
 - o Registro
 - o Gestión de usuarios en zona admin (puede estar todo el crud en la misma página)
 - o Gestión de categorías (puede estar todo el crud en la misma página)
 - o Gestión de preguntas (solo crear y listar)
 - o Gestión de exámenes (solo crear y listar)
 - o Gestión de intentos:
 - Ver lista de intentos pendientes y hechos
 - Realización de intento de examen
 - Visualización de los resultados del intento
- CSS avanzado:
 - o Uso de flex o grid en lugar de float para posicionar elementos
 - o Uso de iconos o imágenes
 - o Uso de sombras
 - o Uso de fuentes
 - o Uso de animaciones, transiciones o transformaciones
 - o Uso de filtros o mezclas
 - o Otros (no obligatorio, pero se valorará positivamente su uso): gradientes, variables, funciones matemáticas, gestión de modo oscuro y claro, tooltip
- No se califica que sea responsive, aunque si hacemos un buen diseño con flex, grid y unidades de medida, seguramente se adapte automáticamente.

PARTE CLIENTE

- Login
 - Se debe comprobar que exista el usuario en la lista y que esté validado para iniciar sesión, y según su rol se dirige a una página u otra, o se ocultan las opciones que tiene.
 - Si es correcto, se guarda en localStorage el usuario logueado a modo cookie, para cuando entre sin cerrar sesión, se mantenga la sesión iniciada.
 - Cuando hace logout, este dato desaparece.
- Registro
 - Se deberá tener una lista de usuarios guardada en localStorage donde se van almacenando los usuarios cuando se registren.
- Gestión de usuarios en zona admin
 - Se muestra una lista de usuarios registrados.
 - Debe de poder modificarse cualquier dato del usuario, menos el nombre (contraseña, rol, y validado)
 - No es necesario crear usuarios desde esta gestión, siempre que se haya hecho el Registro
- Gestión de categorías
 - Listar, crear, y borrar utilizando el localStorage
- Gestión de preguntas
 - Crear, guardando en localStorage la lista de preguntas.
 - Para crear necesitamos la lista de categorías, debemos obtenerlas del LS y pintar el select con tantas opciones como categorías usando elementos del DOM.
 - Listar
- Gestión de exámenes
 - Crear, guardando en localStorage la lista de exámenes.
 - Para crear necesitamos la lista de preguntas, debemos obtenerlas del LS y pintar la lista usando elementos del DOM.
 - Los filtros de categoría y dificultad son opcionales.
 - Cada vez que se cree un examen se generan tantos intentos de ese examen como alumnos haya guardados, para que los alumnos cuando accedan vean su lista de exámenes pendientes de hacer (intentos).
 - Listar
- Gestión de intentos:
 - Ver lista de intentos pendientes (con fecha posterior a la actual) y hechos
 - Realización de intento de examen
 - Podemos visualizarlo a modo carrousel controlando la visibilidad de la pregunta actual añadiendo clases mediante js.
 - Si no da tiempo, se muestra la lista completa del tirón.
 - Visualización de los resultados del intento

Para ir pintando los elementos de cada lista en las páginas correspondientes, se usarán los elementos DOM creados desde javascript y se irán añadiendo al html con los métodos de gestión del DOM.

Cuando se crean o eliminan elementos, además de mostrarlo gráficamente, recordad que también tienen que persistir esas modificaciones en los datos que haya en localStorage.