Proyecto S3 s5

Para el proyecto de la semana los estudiantes deberán construir tanto el frontend como el backend del módulo de inicio de sesión.

- Durante las sesiones 4 y 5 de la semana, el formador hace la explicación de los diferentes componentes necesarios para construir un módulo de inicio de sesión
- Se deben construir los documentos explicados durante la semana 1, documento de requerimientos, documento del proyecto, casos de uso e historias de usuario; estos enfocados al proyecto de la semana.
- Durante la sesión 4 se hace la explicación del documento de pruebas, este nuevo documento deberá ser construido por los estudiantes para el proyecto de la semana.

La interfaz debe contar con un campo para el nombre de usuario, otro campo para contraseña y un botón realizar la acción de inicio de sesión, al ingresar deberá mostrar un escritorio para el usuario en donde se visualicen su nombre y correo. En caso de fallar, se debe indicar que hubo un error en el inicio de sesión.

Para la calificación del proyecto de la semana, los estudiantes deben conservar una estructura base, esta estructura garantizará que se puedan realizar pruebas automáticas. En caso de no cumplir con la estructura base alojada en github, la prueba arrojará un fallo que afectará la calificación.

la estructura base link github:

https://github.com/Tecnalia-Cilco-3/semana-3

requisitos obligatorios estructura backend:
ruta:
se debe contar un una ruta por medio de método post para el inicio de sesión de la siguiente manera:

```
'/api/auth/signin'
```

cuando esta ruta es consumida desde el frontend la api debe responder en tres casos diferentes :

1. Cuando el usuario se loguea exitosamente ,debe responder con un status 200 y propiedad accessToken de la siguiente manera :

   ```
   res.status(200).send({ accessToken: token });
   ```

2. El usuario no existe en la bases de dato, debe responder con un status 404 de la siguiente manera:

   ```
   res.status(404);
   ```

esta respuesta se puede complementar con un mensaje como por ejemplo:

```
res.status(404).send('User Not Found.');
```

3. El usuario ingresa una contraseña inválida, debe responder con un status 401 de la siguiente manera:

```
res.status(401);
```

esta respuesta se puede complementar con un mensaje y propiedades dependiendo de lo que espera recibir en el fontend como por ejemplo:

```
res.status(401).send({ auth: false, accessToken: null, reason:
"Invalid Password!" });
```

bases de datos :

en el directorio config/config.json encontrarán las credenciales para la conexión de las bases de datos de la siguiente manera:

```
{
    "development": {
        "dialect": "sqlite",
        "storage": "./database.sqlite3"
    },
    "test": {
        "dialect": "sqlite",
        "storage": "./database.sqlite3"
    },
    "production": {
        "dialect": "sqlite",
        "storage": "./database.sqlite3"
    }
}
```

queda de elección de cada grupo utilizar la bases de datos localmente que deseen ya sea la predeterminada en el archivo o utilizar mysql como se explicó en las sesiones anteriores, estas modificacion solo se deben realizar en el objeto "**development**", las otras por ningún motivo deben ser modificadas esto podría alterar el resultado de la prueba y por ende su calificación.

Modelo:
se debe crear un modelo con el nombre **user** como se explicó en la sesión ,por medio de sequelize cli con los atributos obligatorios : **name,email , password** de tipo string.

Finalmente, como todos los proyectos, deben ser subidos a un repositorio en Github