Doc técnica Resultify

Pasos para correr la aplicación (**De manera local**).

**Front – end**

Abre una terminal, powershell o CMD (windows) dentro de la carpeta “**Resultify”**, dirigete a “**Project”** (que es la carpeta que contiene todo el front-end) una vez dentro de esa ruta, teclea el siguiente comando:

**$ npm start**

**A screenshot of a computer code

AI-generated content may be incorrect.**



Esto comenzará a correr el proyecto para React js en el puerto 3000 (yo lo configure en ese puerto, lo puedes cambiar en cualquier momento si así lo deseas). Dirígete a la url <http://localhost:3000/> y ahí estará la pagina de /Login, aunque normalmente al lanzar el comando y este se ejecuta exitosamente, te abre una nueva ventana del navegador con el componente inicial.

**Back – end**

1.- Verifica que la instancia de MYSQL este activa:

A screenshot of a computer

AI-generated content may be incorrect.

En la maquina de Windows, configure que se inicie la instancia al prender la maquina. Debe de estar corriendo para que el back-end realice sus consultas a la bd, si no, el archivo “**connection.js”** y “**myql.js”** marcarán un error.

Una vez que verifiques que la instancia de la bd local esta encendida:

Abre una terminal, powerShell o CMD (windows) dentro de la carpeta de Resultify, dirigete a la carpeta de “**server”** (**Que es la carpeta que contiene todo el back-end**), muévete a la carpeta “**src**” y escribe el siguiente comando:

**$ npx nodemon start**

Esto iniciara el servidor Express en el puerto 3001 (yo lo configure en ese puerto, lo puedes cambiar en cualquier momento si así lo deseas)

A screenshot of a computer

AI-generated content may be incorrect.

Una vez corriendo el servidor, podrás iniciar sesión sin ningún problema en la aplicación.

Ahora te explicare lo más detallado posible como esta estructurado internamente el proyecto, desde los paquetes instalados, lenguaje de programación, hasta las librerías usadas.

Comencemos con el front-end:

La aplicación esta desarrollada en **React js 16.2.0**, use esta librería por que me fue mas cómodo codificar html y javascript juntos, además que reutilice varios componentes como el side bar o barra lateral, resultados de los submits, archivos cargados y archivos procesados. Es fácil de usar y si no conoces esta herramienta es fácil de aprender. Además de que para el back-end use Node js, entonces es mucho mas fácil juntar estas dos tecnologías.

Para React js, instale las siguientes dependencias:

* "axios": "^1.8.2" : axios te permite hacer peticiones al back-end.
* "jwt-decode": "^4.0.0": json web token para codificar o decodificar datos, por ejemplo: contrseñas, datos sensibles, etc.
* "lucide-react": "^0.477.0": iconos e imágenes para mostrar en pantalla.
* "react": "^19.0.0",
* "react-dom": "^19.0.0",
* "react-router-dom": "^7.4.0",
* "react-scripts": "5.0.1",
* "web-vitals": "^2.1.4": Sirve para realizar métricas de rendimiento de la app, por ejemplo: tiempos de espera, cargas saturadas, trafico de usuarios, etc.

A screenshot of a computer

AI-generated content may be incorrect.

Dentro de la carpeta **src**, encontraras **Components**, carpeta que contiene todos los componentes separados por carpetas. Cada carpeta incluye su **component.jsx** y su **style.css**

A screenshot of a computer

AI-generated content may be incorrect.

Dentro de la carpeta **Hooks**, hay un solo archivo de momento, que es el **useAuthCheck.js,** este .js es un “**hook**”y sirve para estar validando el rol del usuario en cada momento de la navegación; si el usuario navega o logra acceder a una pagina donde no tiene autorización, este “**hook**“, lo redirigirá a la pagina de inicio. Si creas un componente nuevo o modificas uno existente, puedes agregar o quitar este hook desde el encabezado del .**jsx**, para dar o quitar acceso según el rol del usuario (alumno o docente).

A computer screen shot of a number

AI-generated content may be incorrect.