**PROYECTO-ISOFT**

**Descripción del Proyecto**

Desarrollo de un proyecto llamado "CELERIUM, herramienta gestora de proyectos informáticos". Este sistema tiene como objetivo ofrecer una plataforma completa para la gestión eficiente de proyectos tecnológicos. La herramienta incluye módulos para tareas, reportes, análisis de datos, y administración general tanto para desarrolladores como para lideres de proyecto.

**Requisitos del Proyecto**

* **Backend:** El backend del proyecto está desarrollado en Node.js, utilizando Express como framework principal. A continuación, se listan las dependencias necesarias:
  + *Dependencias*
    - cors: Manejo de cabeceras y seguridad entre dominios.
    - express: Framework para la creación de servidores.
    - joi: Validación de datos.
    - jsonwebtoken: Manejo de autenticación mediante tokens.
    - mongoose: Conexión y manejo de bases de datos MongoDB.
    - newrelic: Monitorización de rendimiento.
    - nodemailer: Envio de correos electrónicos.
    - request: Realizar solicitudes HTTP.
    - uuid: Generación de identificadores únicos.
  + **Instalación de Dependencias**
    - Ejecuta el siguiente comando en la carpeta del backend: **npm install**
* **Frontend:** El frontend está desarrollado con React y utiliza Vite como herramienta de desarrollo. Incluye librerías avanzadas para la creación de interfaces interactivas y componentes responsivos.
  + *Dependencias*
    - @mui/material: Componentes de interfaz basados en Material Design.
    - @mui/icons-material: Íconos listos para usar.
    - axios: Manejo de solicitudes HTTP.
    - react-router-dom: Manejo de rutas en React.
    - react-hook-form: Manejo de formularios en React.
    - yup: Validación de formularios.
    - apexcharts y react-apexcharts: Gráficas avanzadas.
    - simplebar-react: Scrollbars personalizadas.
    - swiper: Carruseles y sliders interactivos.
  + **DevDependencies (Dependencias de Desarrollo)**
    - vite: Herramienta de desarrollo rápida.
    - typescript: Tipado estático para JavaScript.
    - eslint y prettier: Análisis estático y formateo de código.
  + **Instalación de Dependencias**
    - Ejecuta el siguiente comando en la carpeta del frontend: **yarn install (o si usas npm npm install)**

**Estructura del Proyecto**

* **Backend:**
  + src/controllers: Controladores para manejar la lógica de negocio.
  + src/models: Modelos de datos (MongoDB).
  + src/routes: Definición de rutas.
  + src/helpers: Funciones reutilizables.
  + src/middlewares: Middleware de validación y seguridad.
* **Frontend:**
  + src/components: Componentes reutilizables.
  + src/pages: Páginas principales.
  + src/routes: Configuración de rutas.
  + src/hooks: Hooks personalizados.
  + src/theme: Estilos y configuración de tema.
  + src/utils: Utilidades generales.

**Ejecución del Proyecto**

* **Backend:**
  + Asegúrate de que MongoDB esté corriendo en tu sistema.
  + Configura las variables de entorno en el archivo .env ubicado en la carpeta del backend.
  + Inicia el servidor ejecutando el siguiente comando en la terminal: **node src/app.js**
* **Frontend**:
  + Configura las variables de entorno en el archivo .env ubicado en la carpeta del frontend.
  + Inicia el servidor de desarrollo ejecutando el siguiente comando: **npm run dev (o si usas yarn, yarn dev)**