**Ejercicio Práctico: Creación de una Aplicación de Lista de Tareas con Autenticación Social**

**Descripción:** Crear una aplicación web de lista de tareas utilizando React para el frontend y Django para el backend. La aplicación debe permitir a los usuarios crear, leer, actualizar y eliminar tareas. Además, se debe implementar la autenticación social utilizando Google y Apple, junto con un manejo adecuado de errores en el frontend y el backend.

**Pasos:**

1. **Backend (Django):**
   * Configura un proyecto Django.
   * Crea un modelo **Task** con los campos **title**, **description**, **completed**, y **created\_at**.
   * Implementa una API RESTful usando Django REST Framework para manejar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para las tareas.
   * Agrega manejo de errores para garantizar respuestas adecuadas en caso de solicitudes fallidas.
2. **Autenticación Social:**
   * Implementa la autenticación social utilizando Google y Apple.
   * Utiliza las bibliotecas o SDKs proporcionados por Google y Apple para integrar la autenticación en el backend de Django.
   * Almacena la información del usuario (por ejemplo, nombre, correo electrónico) obtenida de Google y Apple en la base de datos de la aplicación.
3. **Frontend (React):**
   * Configura un proyecto React.
   * Crea componentes para mostrar la lista de tareas, agregar una nueva tarea, marcar una tarea como completada y eliminar una tarea.
   * Implementa la autenticación social en el frontend utilizando las API proporcionadas por Google y/o Apple.
   * Maneja los errores de manera adecuada, mostrando mensajes de error al usuario cuando sea necesario.
4. **Integración:**
   * Integra el frontend de React con el backend de Django para que la aplicación pueda enviar y recibir datos de la API.
   * Asegúrate de que la autenticación social funcione correctamente y que los usuarios puedan realizar todas las funciones CRUD en la aplicación de lista de tareas.

**Requisitos Adicionales:**

* La lista de tareas debe actualizarse dinámicamente sin necesidad de recargar la página.
* Agrega estilos CSS para mejorar la apariencia y usabilidad de la aplicación.
* Implementa un manejo robusto de errores tanto en el frontend como en el backend, mostrando mensajes claros al usuario en caso de problemas.

**Entrega:**

* Sube el código fuente de la aplicación a un repositorio en GitHub y proporciona el enlace al evaluador mediante correo electrónico: transformacion@spiglobal.net
* Incluye instrucciones claras sobre cómo ejecutar la aplicación localmente, así como los pasos necesarios para probar la autenticación con Google y/o Apple.
* Tiempo aprox.= 1 semana (due date 23 Febrero 2024)