Desafío Técnico | Full Stack Developer

Aplicación de Reserva de Espacios para Eventos

Descripción General:

El desafío consiste en desarrollar una aplicación web para la reserva de espacios (como salas de reuniones, auditorios, etc.) para eventos. La aplicación permitirá a los usuarios explorar los espacios disponibles, hacer reservas, gestionar sus propias reservas y visualizar la disponibilidad de los espacios.

Se debe construir una **SPA** en **Angular** para el frontend y una **API REST** en **Laravel** para el backend, utilizando autenticación de usuarios y asegurando la correcta interacción entre ambos. (Pueden ser dos repositorios separados)

Requerimientos Técnicos:

1. **Frontend (Angular)**:
    - **Vista principal**:
        - Página que muestra un listado de espacios disponibles, con filtros por tipo de espacio, capacidad y disponibilidad por fechas.
        - Al seleccionar un espacio, se debe mostrar información detallada (nombre, descripción, capacidad, fotos, horarios disponibles, etc.).
    - **ABM de espacios:**
        - Solo para usuarios administradores.
        - Requerimiento obligatorio utilizar MC-Table (de MC Kit) en el listado.
    - **Sistema de reservas**:
        - Formulario para que los usuarios reserven un espacio, indicando el nombre del evento, la fecha y la hora de inicio/fin.
        - Validación para evitar reservas en horarios ya ocupados.
    - **Gestión de reservas**:
        - Vista donde el usuario puede ver todas sus reservas actuales, modificar una reserva o cancelarla.
    - **Interfaz**:
        - Utilizar Angular Material, PrimeNg o Tailwind. (PrimeNg deseable)
        - Debe haber un sistema de notificaciones (toasts) para informar sobre el éxito o fallo de las acciones del usuario (por ejemplo, reserva exitosa o error al cancelar).
    - **Extras (Opcional):**
        - Testing de algún componente, mencionarlo en el Readme si se realiza.
        - Utilizar MC Kit, librería de formularios, filtros, tablas, etc (https://github.com/matiascamiletti/mc-kit)
2. **Backend (Laravel)**:

- **API RESTful**:
  - Implementar una API para gestionar:
    - Autenticación de usuarios mediante JWT (registro e inicio de sesión).
    - CRUD para la gestión de los espacios (crear, ver, editar y eliminar espacios).
    - CRUD para la gestión de las reservas (crear, ver, editar y eliminar reservas). Solo para el usuario actual.
  - Implementar lógica de validación para evitar la superposición de reservas en los mismos horarios.
  - Implementar alguna validación o funcionalidad no mencionada anteriormente que creas necesaria para el proyecto y documentarla en el readme.
- **Protección de rutas**:
  - Asegurar que solo los usuarios autenticados puedan gestionar sus reservas y que no puedan modificar o ver las reservas de otros.
- **Base de datos**:
  - Crear una base de datos para almacenar los espacios y reservas, usando **Migraciones y Seeders**
  - Relacionar los usuarios con sus respectivas reservas.
- **Tests**
  - Se debe crear una suite de testing validé el funcionamiento de la API.

3. **Requisitos adicionales**:
   - **Documentación**:
     - Proveer documentación básica para la API (por ejemplo, usando Swagger) que explique los endpoints disponibles y cómo interactuar con ellos.
     - Incluir un readme.md con instrucciones sobre cómo ejecutar el proyecto (instalación de dependencias, configuración de entorno, etc.). Y cualquier cosa a la que le hayas puesto amor para que pongamos más atención en ello.
4. **Extras (Opcional)**:
   - **Calendario**: Implementar un calendario que permita a los usuarios visualizar los horarios reservados y libres de un espacio en un formato gráfico.

Criterios de Evaluación:

- **Código limpio y organizado** tanto en front como en backend.
- **Uso adecuado de los frameworks Angular y Laravel**
- **Validez y seguridad** en la autenticación y autorización de usuarios.
- **Validación de formularios y manejo de errores** tanto en frontend como en backend.
- **Interfaz de usuario,** acá evaluamos tu creatividad, déjate llevar, se evalúa las habilidades del maquetado (html y css)
- **Calidad de la Documentación** en el / los archivos readme.md

- **Cumplimiento de los requisitos funcionales**.

Entrega:

- Enviar los link/s a el/los repositorios.
    - Se valorará el proyecto desplegado.
- Un breve texto contando tu experiencia al realizar el test y especificando dónde crees que está lo mejor de vos en el test.