Visión del proyecto

### **Descripción General del Proyecto**

El proyecto consiste en desarrollar una **plataforma de reventa de entradas** que permita a los usuarios comprar y vender entradas para eventos de manera segura, eficiente y transparente. La plataforma está diseñada para abordar problemas comunes en el mercado secundario de entradas, como la falsificación y el fraude, brindando a los usuarios una experiencia confiable y sin complicaciones.

La aplicación se construirá utilizando la **stack MERN**, con una actualización significativa en el frontend utilizando **Next.js**. Next.js permitirá optimizar la velocidad de carga de las páginas a través del renderizado del lado del servidor (SSR) y mejorar la optimización para motores de búsqueda (SEO). Además, Next.js facilitará la creación de rutas dinámicas y la integración de funciones avanzadas como la autenticación y la pre-renderización.

**Objetivos del Proyecto:**

* Crear un entorno seguro y transparente para la reventa de entradas.
* Facilitar la experiencia del usuario con una interfaz intuitiva y rápida.
* Implementar medidas robustas de seguridad para proteger a los usuarios y sus transacciones.

La plataforma se compone de dos partes principales:

1. **Frontend:** Desarrollado en Next.js, proporcionando una interfaz de usuario dinámica y responsiva, accesible tanto en dispositivos móviles como de escritorio.
2. **Backend:** Implementado en Node.js con Express, manejando la lógica del negocio, la gestión de la base de datos y la autenticación de usuarios.

### **Funciones para los Usuarios**

1. **Registro y Autenticación:**
   * **Registro de Usuario:** Los usuarios pueden registrarse proporcionando su correo electrónico, contraseña, nombre completo y número de teléfono. Incluye la opción de verificar su identidad, como subir un documento oficial.
   * **Inicio de sesión:** Los usuarios pueden iniciar sesión utilizando su correo electrónico y contraseña, o mediante OAuth (Google, Facebook).
   * **Recuperación de contraseña:** Los usuarios pueden recuperar su contraseña a través de un enlace enviado a su correo electrónico.
2. **Gestión de Perfil:**
   * **Verificación de Identidad:** Permitir a los usuarios cargar documentos oficiales para la verificación de identidad.
   * **Actualización de Perfil:** Los usuarios pueden actualizar su información personal, como nombre, foto de perfil, correo electrónico y teléfono.
3. **Navegación y Búsqueda:**
   * **Búsqueda de Eventos:** Los usuarios pueden buscar eventos por nombre, fecha, ubicación o categoría.
   * **Filtrado de Entradas:** Opciones de filtrado como precio, ubicación del asiento y fecha del evento.
   * **Visualización de Eventos:** Mostrar detalles del evento, incluyendo descripción, fecha, ubicación y entradas disponibles.
4. **Gestión de Entradas:**
   * **Publicar Entradas en Venta:** Los usuarios (vendedores) pueden listar sus entradas para la venta(solo si estas son digitales y no físicas), proporcionando detalles como evento, precio y asientos, además deberán enviar una foto del detalle de la entrada para entregarla al cliente una vez la compre, esta será la que usará para entrar al evento
   * **Edición y Eliminación de Listados:** Los vendedores pueden editar o eliminar listados de entradas antes de que se vendan.
   * **Transferencia de Entradas:** Para entradas electrónicas, permitir la transferencia a otros usuarios a través de la aplicación.
5. **Compra de Entradas:**
   * **Carrito de Compras:** Los compradores pueden agregar entradas a un carrito y proceder al pago cuando estén listos.
   * **Pagos Seguros:** Implementar un sistema de pago seguro que soporte múltiples métodos de pago (tarjeta de crédito, PayPal, etc.).
   * **Confirmación de Compra:** Enviar un correo electrónico de confirmación con los detalles de la compra y acceso a la entrada.
   * **Funcion de retencion**: Al comprar una entrada el dinero se le retendrá al vendedor hasta que la fecha del evento haya pasado y el usuario confirme su uso, si el usuario no confirma su uso en tres días, este se confirmara solo y liberara el dinero al vendedor
6. **Sistema de Valoraciones y Reseñas:**
   * **Dejar Reseñas:** Los compradores pueden dejar reseñas y calificaciones para los vendedores después del evento.
   * **Visualización de Reseñas:** Mostrar las calificaciones y comentarios de los vendedores para que otros compradores puedan tomar decisiones informadas.
7. **Notificaciones:**
   * **Notificaciones Push y Email:** Enviar notificaciones sobre actualizaciones de la cuenta, nuevas entradas disponibles, confirmaciones de compra y alertas de seguridad.

### **Funciones para los Administradores**

1. **Gestión de Usuarios:**
   * **Moderación de Usuarios:** Administrar la verificación de identidad, bloquear o eliminar usuarios que infrinjan las reglas.
   * **Gestión de Soporte:** Responder a tickets de soporte y resolver disputas entre compradores y vendedores.
2. **Moderación de Contenidos:**
   * **Revisión de Listados:** Revisar y aprobar o rechazar listados de entradas sospechosas o fraudulentas.
   * **Gestión de Eventos:** Crear, editar o eliminar eventos en la base de datos, asegurando que la información sea precisa y actualizada.
3. **Gestión de Transacciones:**
   * **Monitoreo de Transacciones:** Visualizar y auditar transacciones realizadas en la plataforma, incluyendo pagos retenidos en escrow.
   * **Políticas de Reembolso:** Administrar reembolsos y resolver disputas sobre transacciones problemáticas.
4. **Generación de Reportes:**
   * **Reportes de Actividad:** Generar reportes sobre la actividad de usuarios, ventas de entradas y transacciones para análisis y toma de decisiones.
   * **Reportes Financieros:** Generar reportes sobre ingresos, pagos retenidos y comisiones de la plataforma.

### **Funciones Técnicas y Seguridad**

1. **Seguridad de la Información:**
   * **Cifrado de Datos:** Implementar cifrado para datos sensibles, incluyendo contraseñas y datos de pago.
   * **Autenticación Multifactor:** Ofrecer autenticación de dos factores (2FA) para aumentar la seguridad de las cuentas.
2. **Integraciones de API:**
   * **Integración con Proveedores de Ticketing:** Integrar con APIs de plataformas de ticketing para la validación de entradas y sincronización de datos de eventos.
   * **Pasarela de Pago:** Integrar con pasarelas de pago para gestionar transacciones de manera segura.
3. **Escalabilidad y Rendimiento:**
   * **Optimización de la Base de Datos:** Asegurar que la base de datos esté optimizada para manejar un gran volumen de transacciones y consultas.
   * **Manejo de Cargas:** Implementar soluciones de escalabilidad para manejar picos en la demanda durante eventos populares.
4. **Soporte al Cliente:**
   * **Sistema de Soporte:** Implementar un sistema de tickets para que los usuarios puedan solicitar ayuda o reportar problemas.
   * **Chat en Vivo:** Opcionalmente, ofrecer un chat en vivo para soporte en tiempo real.

### **Requisitos No Funcionales**

1. **Seguridad:**
   * **Cifrado de Datos:** Todo dato sensible, como contraseñas y detalles de pago, debe estar cifrado tanto en tránsito como en reposo.
   * **Autenticación Multifactor (2FA):** Implementar autenticación de dos factores para todos los usuarios para aumentar la seguridad de las cuentas.
   * **Protección contra Fraudes:** Integrar sistemas de detección y prevención de fraudes para monitorear transacciones sospechosas.
   * **Cumplimiento Normativo:** Asegurarse de que la plataforma cumpla con las normativas de protección de datos y privacidad, como el GDPR.
2. **Rendimiento:**
   * **Tiempo de Respuesta:** Las páginas deben cargar en menos de 3 segundos bajo condiciones normales.
   * **Escalabilidad:** La plataforma debe poder escalar para manejar picos de tráfico, especialmente durante eventos populares, sin degradación del rendimiento.
   * **Optimización de Consultas:** La base de datos y las consultas deben estar optimizadas para manejar un alto volumen de transacciones concurrentes.
3. **Disponibilidad:**
   * **Alta Disponibilidad:** La plataforma debe tener una disponibilidad del 99.9% para garantizar un tiempo de inactividad mínimo.
   * **Recuperación ante desastres:** Implementar un plan de recuperación ante desastres con copias de seguridad regulares y redundancia de datos.
   * **Balanceo de Carga:** Utilizar balanceo de carga para distribuir el tráfico de manera equitativa entre los servidores.
4. **Mantenibilidad:**
   * **Código Modular y Documentado:** El código debe estar bien documentado y estructurado en módulos para facilitar el mantenimiento y futuras expansiones.
   * **Control de versiones:** Uso de sistemas de control de versiones (como Git) para gestionar cambios en el código y colaborar eficientemente.
   * **Facilidad de Despliegue:** Los despliegues deben ser automatizados y tener un mínimo impacto en el servicio activo.
5. **Usabilidad:**
   * **Interfaz de Usuario Intuitiva:** La interfaz debe ser fácil de usar, con una experiencia de usuario coherente y fluida tanto en dispositivos móviles como en escritorio.
   * **Accesibilidad:** La plataforma debe cumplir con los estándares de accesibilidad (WCAG 2.1) para usuarios con discapacidades.
   * **Localización:** Soporte para múltiples idiomas y formatos regionales de fecha y moneda.
6. **Compatibilidad:**
   * **Soporte Multi-Navegador:** La plataforma debe ser compatible con los principales navegadores (Chrome, Firefox, Safari, Edge) y sus versiones recientes.
   * **Compatibilidad de Dispositivos:** La aplicación debe ser completamente funcional en diferentes tamaños de pantalla, incluyendo dispositivos móviles, tablet.
7. **Integrabilidad:**
   * **APIs Externas:** Las integraciones con APIs externas (proveedores de ticketing, pasarelas de pago, etc.) deben ser seguras, confiables y cumplir con los estándares de la industria.
   * **Modularidad:** La arquitectura del sistema debe permitir la fácil integración de nuevas funcionalidades o servicios externos sin afectar el rendimiento o la estabilidad del sistema.
8. **Monitoreo y Logística:**
   * **Monitoreo en Tiempo Real:** Implementar herramientas para monitorear el rendimiento del sistema en tiempo real y detectar posibles problemas de inmediato.
   * **Registros y Auditoría:** Mantener registros detallados de transacciones, accesos y errores para auditorías y análisis posteriores.