MANUAL TÉCNICO

Introducción

Este manual proporciona una guía técnica para la instalación, configuración y mantenimiento de la aplicación móvil para la gestión de citas médicas en un asilo.

Requisitos del Sistema

- Frontend: React Native
- . Backend: Node.js con Express
- Base de Datos: Firebase Firestore
- Autenticación: Firebase Authentication para correo electrónico y Google OAuth
- · Manejo de Estado: Redux
- · Notificaciones: Firebase Cloud Messaging (FCM)

Instalación

Clonar el Repositorio

- git clone https://github.com/tu-usuario/tu-epositorio.git
- cd tu-repositorio

Instalar Dependencias

• npm install

Configurar Firebase

- 1. Crear un Proyecto en Firebase
 - Visite Firebase Console.
 - Cree un nuevo proyecto y configúrelo según las instrucciones.
- 2. Configurar Firebase Authentication
 - En la consola de Firebase, vaya a "Authentication" y habilite los métodos de autenticación por correo electrónico y Google OAuth.
- 3. Configurar Firebase Firestore
 - En la consola de Firebase, vaya a "Firestore Database" y cree una nueva base de datos en modo de producción o prueba.

4. Configurar Firebase Cloud Messaging (FCM)

En la consola de Firebase, vaya a "Cloud Messaging" y siga las instrucciones para configurar las notificaciones push.

5. Agregar Archivos de Configuración

- Para Android: Descargue el archivo google-services.json desde la consola de Firebase y colóquelo en el directorio android/app.
- Para iOS: Descargue el archivo GoogleService-Info.plist desde la consola de Firebase y colóquelo en el directorio iOS/.

Configurar Variables de Entorno

Cree un archivo .env en la raíz del proyecto con las siguientes variables:

- FIREBASE_API_KEY=tu-api-key
- FIREBASE_AUTH_DOMAIN=tu-auth-domain
- FIREBASE_PROJECT_ID=tu-project-id
- FIREBASE_STORAGE_BUCKET=tu-storage-bucket
- FIREBASE_MESSAGING_SENDER_ID≔tu-messaging-sender-id
- FIREBASE_APP_ID=tu-app-id

Ejecución de la Aplicación

• npm start

Estructura del Proyecto

Directorio src

- · components: Contiene los componentes de React Native.
 - SplashScreen.js: Pantalla de inicio.
 - 。LoginScreen.js: Pantalla de inicio de sesión.
 - HomeScreen.js: Pantalla principal.
 - Profiles: Componentes para los perfiles de usuarios.
 - DoctorProfile.js
 - PatientProfile.js
 - Appointments: Componentes para la gestión de citas.
 - AppointmentsCalendar.js
 - · AppointmentForm.js

- MedicalHistory.js: Componente para el historial médico.
- Chat: Componentes para el chat.
 - . ChatScreen.js
- Notifications: Componentes para las notificaciones.
 - Notifications.js
 - · ManualNotificationScreen.js
- Settings: Componentes para configuración y soporte.
 - Settings.js
 - Support.js
- Admin: Componentes administrativos.
 - UserManagementScreen.js
 - · ReportGenerationScreen.js

- redux: Contiene la configuración de Redux.
 - 。 actions.js: Acciones de Redux.
 - 。 reducers.js: Reductores de Redux.
 - store.js: Configuración del store de Redux.
- utils: Contiene utilidades y funciones auxiliares.
 - 。 api.js: Funciones para la comunicación con el backend.
 - o notifications.js: Configuración y manejo de notificaciones.

Seguridad y Permisos

- Autenticación: Utiliza Firebase Authentication para gestionar los accesos.
- Roles de Usuario: Control de acceso basado en roles implementado en el backend.

Mantenimiento

- Actualizaciones: Mantenga las dependencias actualizadas con npm update.
- Backup: Realice backups regulares de la base de datos Firebase
 Firestore.

Enlaces Útiles

- · Manual de usuario: Enlace al Manual de Usuario
- · Manual técnico: Enlace al Manual Técnico
- . Trello o Notion: Enlace a Trello o Notion
- Diseños Mock Ups: Enlace a los Diseños Mock Ups

Licencia

Este proyecto está licenciado bajo la Licencia Creative Commons

• CC BY-NC 4.0.