

Proyecto gestión

Este proyecto es para llevar la gestión de proyectos y esta desarrollado con React y next js y framework TailwindCSS e iconos de lucide.

Y para el manejo de la información ocupa firebase

Funcionalidades

1. Permite a los usuarios autenticarse con Firebase Authentication
2. Verifica si el sistema está en mantenimiento a través de Firestore
3. Redirige al usuario a la página correspondiente según su rol
4. Creación de usuarios de Usuarios
5. Edición de usuarios
6. desactivación de usuarios
7. Listar usuarios
8. Generador de Password
9. Poner sitio en mantenimiento

Validaciones

En modulo de usuarios

1. validación de Campos
2. validación de Email Existente

En módulo de Proyectos

1. validación de campos
2. validación de fechas

Estructura del proyecto.

El proyecto tiene toda su funcionamiento en carpeta SRC

```
src/
├── app/                # Carpeta base de Next.js con el App Router
│   ├── layout.tsx      # Layout principal del sistema
│   ├── page.tsx        # Página principal
│   ├── login/          # Carpeta para la página de inicio de sesión
│   ├── admin/          # Carpeta para las funcionalidades de admin
│   ├── gerente/        # Carpeta para las funcionalidades de gerente
│   ├── miembro/        # Carpeta para las funcionalidades de miembro
│   ├── components/     # Componentes reutilizables
│   ├── hooks/          # Hooks personalizados
│   ├── lib/            # Funciones auxiliares y configuración
│   ├── services/       # Archivo para la services generales
│   ├── styles/         # Archivos de estilos globales
│   └── types/          # Tipos y modelos de datos
```

LoginPage.tsx

Importaciones:

- **"use client"** → Indica que es un componente cliente en Next.js.
- **useEffect, useState** → Hooks de React para manejar estado y efectos secundarios.
- **useRouter** → Para redirigir al usuario después del login.
- **login** (de authService) → Función para autenticar usuarios en Firebase.
- **getUserRole** (de userService) → Obtiene el rol del usuario desde Firestore.
- **Firestore (doc, getDoc, db)** → Para verificar si el sistema está en mantenimiento.

Funciones:

checkMaintenance: Verifica si el sitio este mantenimiento esto para Evitar que los usuarios ingresen cuando se está haciendo un mantenimiento al sistema.

Login: Encardo de realizar el login tomando las credenciales ingresadas y verificándola con Firebase Authentication.

getUserRole: Encardo de sacar el rol del usuario desde firebase.

Funcionalidad	Implementación
Verifica mantenimiento	Consulta Firestore (config/system).
autenticación	login(email, password).
Obtener Rol	getUserRole(user.uid)
Redirigir según rol	router.push("/")

Admin/page.tsx

Importaciones:

- **"use client"** → Indica que es un componente cliente en Next.js.
- **useEffect, useState** → Hooks de React para manejar estado y efectos secundarios.
- **useRouter** → Para redirigir al usuario después del login.
- **onAuthStateChanged**: Valida si el usuario tiene sesión activa desde firebase
- **logout**: (de authService) → Función para cerrar sesión de los usuarios

Funcionalidad	Implementación
Validar Sesión activa	onAuthStateChanged
Cerrar sesión	handleLogout

Admin/user/page.tsx

Importaciones:

- **"use client"** → Indica que es un componente cliente en Next.js.
- **useEffect, useState** → Hooks de React para manejar estado y efectos secundarios.
- **Firebase Firestore**:
 - **collection, getDocs** → Obtener documentos de Firestore.
 - **updateDoc, setDoc, doc** → Modificar y guardar datos en Firestore.
- **Firebase Authentication**
 - **createUserWithEmailAndPassword** → Crear nuevos usuarios en Firebase Auth.
 - **getAuth** → Obtener la instancia de autenticación.

Funcionalidad	Implementación
Listar usuarios	fetchUsers
Validar Email Valido	isValidEmail
Generador de Contraseña segura	generatePassword
Copiar a portapapeles	copyPassword
Editar usuarios	handleEdit
Crear usuarios	handleCreateUser
Registrar en Firebase Authentication	createUserWithEmailAndPassword
Mensaje de Confirmacion o Error.	setMessage

Admin/panel/page.tsx

Importaciones:

- **"use client"** → Indica que es un componente cliente en Next.js.
- **useEffect, useState** → Hooks de React para manejar estado y efectos secundarios.
- **Firebase Firestore:**
 - **collection, getDocs** → Obtener documentos de Firestore.
 - **updateDoc, setDoc, doc** → Modificar y guardar datos en Firestore.

Funcionalidad	Implementación
Mantenimiento del Sitio	handleToggle
Confirmacion de Cambio	handleConfirm

Gerente/page.tsx

Importaciones:

- **"use client"** → Indica que es un componente cliente en Next.js.
- **useEffect, useState** → Hooks de React para manejar estado y efectos secundarios.
- **useRouter** → Para redirigir al usuario después del login.
- **onAuthStateChanged:** Valida si el usuario tiene sesión activa desde firebase
- **logout:** (de authService) → Función para cerrar sesión de los usuarios

Funcionalidad	Implementación
Validar Sesión activa	onAuthStateChanged
Cerrar sesión	handleLogout

Gerente/panel/page.tsx

Importaciones:

- **"use client"** → Indica que es un componente cliente en Next.js.

- **useEffect, useState** → Hooks de React para manejar estado y efectos secundarios.
- **Firebase Firestore:**
 - **collection, getDocs** → Obtener documentos de Firestore.
 - **updateDoc, setDoc, doc** → Modificar y guardar datos en Firestore.
 - **deleteDoc** → Eliminar Documentos en Firestore

Funcionalidad	Implementación
Listar proyectos	fetchProjects
Validar Fechas	validateDates
Editar Proyectos	handleEdit
Crear proyectos	handleCreateProject
Eliminar Proyectos	handleDelete
Asignar tareas a los proyectos	handleAddTask

Lista de usuarios

email	password	rol
alexander.siguenza@udb.edu.sv	123456#	Administrador
ca140174@udb.edu.sv	123456	Gerente
ca140174_2@udb.edu.sv	123456	Miembro
ca140174_3@udb.edu.sv	l6(9LI7AKw6%	Gerente
ca140174_5@udb.edu.sv	SoloTd39wqK8	Gerente