**Sistema de Gestión de Gastos y Seguimiento (Full-Stack Python/Flask)**

Este proyecto es una aplicación web full-stack diseñada para la gestión y el seguimiento de gastos del personal (titulares y familiares), con un sistema de autenticación basado en roles para garantizar la seguridad y el aislamiento de los datos.

**⚙️ Arquitectura del Proyecto**

|  |  |  |
| --- | --- | --- |
| **Componente** | **Tecnología** | **Función** |
| **Frontend (Cliente)** | HTML5, Tailwind CSS, JavaScript, Chart.js | Interfaz de usuario (UI), lógica de navegación, gráficos y comunicación con la API. |
| **Backend (Servidor)** | Python (Flask) | Manejo de peticiones HTTP, autenticación (JWT), lógica de negocio, y acceso a la base de datos. |
| **Base de Datos** | SQLite3 | Persistencia de datos de titulares, familiares, gastos, y usuarios. |
| **Seguridad** | JWT (JSON Web Tokens), passlib (Hash de Contraseñas) | Autenticación y control de acceso basado en roles (admin y titular\_view). |

**🚀 Guía de Instalación y Ejecución**

Para poner en marcha esta aplicación, necesitas tener **Python 3** y **XAMPP** instalados.

**1. Preparación del Entorno Python**

1. **Instalar Dependencias:** Abre la terminal de Visual Studio Code dentro de la carpeta del proyecto (htdocs/app\_gastos/).
2. Ejecuta el siguiente comando para instalar las bibliotecas necesarias:
3. pip install Flask Flask-Cors passlib pyjwt

**2. Configuración de Archivos**

Asegúrate de que los dos archivos principales (index.html y server.py) estén en la misma carpeta (htdocs/app\_gastos/).

**3. Ejecución del Servidor Backend (Flask)**

1. En la terminal del proyecto, ejecuta el servidor de Python:
2. python server.py

*(Verás un mensaje que indica que el servidor está corriendo en http://127.0.0.1:5000 y que las tablas de la base de datos se han creado/actualizado.)*

**4. Acceso a la Aplicación (Frontend)**

1. Inicia el servidor **Apache** en el Panel de Control de XAMPP.
2. Abre tu navegador y navega a la URL local de XAMPP:
3. http://localhost/app\_gastos/index.html

**🔒 Flujo de Autenticación y Roles**

La aplicación utiliza un estricto control de acceso para proteger los datos.

**1. Rol ADMINISTRADOR (admin)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Acción** | **Usuario** | **Contraseña** | **Permisos** |
| **Login** | La primera cédula registrada. | La primera contraseña registrada. | Acceso total a todos los módulos. |
| **Funciones** | **Solo el Admin** | Puede agregar Titulares, Familiares, Gastos, modificar estados y mensajes, y ver reportes completos. |  |

**2. Rol TITULAR (titular\_view)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Acción** | **Usuario** | **Contraseña** | **Permisos** |
| **Login** | Cédula del titular. | La Cédula del titular (como contraseña inicial). | Acceso limitado a **Seguimiento de Gastos** y **Reportar Error**. |
| **Funciones** | **Solo el Titular** | Puede ver el estado de sus propios gastos, los mensajes del administrador y sus gráficos de consumo mensual. **No puede modificar ni registrar nuevos datos.** |  |

**🔑 Creación de Usuarios**

1. **Crear Admin:** La **primera vez** que ejecutas /register en la pantalla de Login, se crea el usuario admin.
2. **Crear Titulares/Usuarios de Acceso:** El **Administrador** debe usar el módulo **Titulares** para registrar un nuevo Titular. Automáticamente, se crea una clave de acceso en el backend donde:
   * **Usuario = Cédula del Titular**
   * **Contraseña = Cédula del Titular**

**📄 Estructura de Reportes Exportados (CSV)**

El archivo CSV exportado del módulo **Reporte de Gastos** tiene la siguiente estructura de columnas:

|  |  |
| --- | --- |
| **Columna** | **Descripción** |
| **Nombre y Apellido** | Nombre completo del Titular. |
| **Vicerrectorado** | Vicerrectorado asociado al Titular. |
| **Año** | Año del gasto. |
| **Mes** | Mes del gasto. |
| **Condicion** | Condición laboral del Titular. |
| **Tipo de Gasto** | Categoría del gasto (Medicina Variable, Lentes, etc.). |
| **Monto** | Monto del gasto en Bolívares (Formato numérico). |