Reporte Técnico

FinancIA

1. Web Principal

Características Técnicas:

Framework Frontend: Streamlit
Lenguaje de Programación: Python
Librerías adicionales: pandas, numpy.

Funcionalidad principal: Proporciona información sobre "Banorte FinancIA" y tiene un botón que redirige a una subweb.

Personalización de UI: Se ha utilizado HTML y CSS incrustados para personalizar el diseño, incluyendo el color de fondo y la inserción de imágenes de fondo.

Diseño: Se compone de un título, subtítulo, descripciones, y secciones que describen la visión y el objetivo de la plataforma. Además, muestra categorías de servicios en un formato 2x2.

Enlace externo: Existe un botón que redirige a "http://10.22.234.131:8501" mediante el uso de una función open_page que utiliza JavaScript.

2. Subweb

Características Técnicas:

Framework Frontend: Streamlit
Lenguaje de Programación: Python
Librerías adicionales: openai, pandas.
API utilizada: OpenAI GPT-3.5 Turbo para chat.

Funcionalidad principal: La subweb actúa como un asistente financiero especializado en Banorte, proporcionando información relacionada con Banorte y ofreciendo distintas opciones financieras para el usuario.

Personalización de UI: Utiliza CSS para cambiar el color de fondo del cuerpo de la página a rojo (#ff0000).

Interactividad: El usuario puede seleccionar una opción de una lista desplegable y escribir mensajes en un área de texto. Al hacer clic en "Enviar", la aplicación se comunica con la API de OpenAI y muestra la respuesta.

Mensajes del sistema: Incluye mensajes predefinidos del sistema para orientar al usuario sobre el uso adecuado del asistente financiero y la posición de Banorte frente a otros bancos.

Cambios:
En esta versión actualizada del código, se han eliminado las líneas que cargaban los archivos ‘clientes.txt’ y ‘fondosdeinversion.txt’ mediante pandas, por lo que estos datos no se utilizan en esta versión de la subweb.

Consideraciones de Seguridad:
Es importante asegurarse de que las claves API y cualquier información sensible no sean expuestas. Considerar usar variables de entorno u otros métodos seguros para manejar claves API.

Es recomendable validar y sanear todas las entradas del usuario para evitar vulnerabilidades como Cross-Site Scripting (XSS).