Documentação do Projeto: Cofre Digital de Senhas

1. Visão Geral

O Cofre Digital de Senhas é uma aplicação web segura, desenvolvida com Flask, projetada para armazenar, gerir e proteger as credenciais dos usuários. O projeto foi construído com um forte foco em segurança, usabilidade e funcionalidades avançadas como importação e exportação de senhas, medidor de segurança de senha e gerador de senha segura.

O objetivo principal foi criar uma ferramenta confiável onde os usuários possam centralizar as suas senhas com a confiança de que os seus dados estão protegidos por múltiplas camadas de segurança.

2. Funcionalidades Implementadas

2.1. Autenticação

- **Login e Sessão:** O controle de sessão é gerido pelo Flask-Login, garantindo que apenas usuários autenticados possam acessar às suas informações. As sessões expiram automaticamente após um período de inatividade para maior segurança.
- **Recuperação de Senha:** Implementado um fluxo de redefinição de senha onde o usuário valida a sua identidade respondendo corretamente à sua pergunta de segurança antes de poder definir uma nova senha.

2.2. Gestão de Senhas

- **Adicionar, Editar e Remover Senhas:** O usuário pode facilmente adicionar novas credenciais (serviço, nome de usuário e senha), atualizá-las quando necessário e removê-las de forma segura.
- **Visualização Segura:** As senhas nunca são exibidas diretamente no painel. O usuário precisa de clicar num botão "Visualizar" para revelá-las, minimizando a exposição acidental.

2.3. Ferramentas de Segurança e Usabilidade

- **Gerador de Senhas Fortes:** Nos formulários de adição e edição, existe um gerador de senhas integrado. O usuário pode definir o tamanho desejado e gerar uma senha complexa e aleatória com um único clique.
- **Medidor de Força da Senha:** Ao digitar uma nova senha, um slider e um texto analisam a sua força em tempo real (utilizando a biblioteca **zxcvbn**), incentivando o usuário a criar senhas mais seguras.
- **Copiar para a Área de Transferência:** No painel principal, ícones discretos permitem que o usuário copie o nome de usuário ou a senha revelada com um clique, melhorando a experiência e evitando erros de digitação.

2.4. Importação e Exportação de Dados

- **Exportação Criptografada:** O usuário pode exportar todas as suas senhas para um arquivo de backup (.json.enc). Este arquivo é totalmente criptografado com a chave de criptografia pessoal do usuário, garantindo que os dados permaneçam criptografados fora da sua conta.
- **Importação Segura:** É possível importar senhas a partir de um arquivo de backup previamente exportado. A aplicação desencripta o arquivo e adiciona as credenciais de volta ao cofre do usuário.

3. Arquitetura de Segurança

A segurança foi o pilar central deste projeto. As seguintes medidas foram implementadas:

- **Encriptação de Ponta a Ponta (AES):** Todas as senhas armazenadas no banco de dados são encriptadas utilizando o algoritmo AES através da biblioteca **cryptography.** Cada usuário possui uma chave de criptografia única, gerada no momento do registro, garantindo que os dados de um usuário não possam ser acessados com a chave de outro.
- **Hashing de Credenciais (Bcrypt):**
    - A senha de login do usuário é armazenada como um hash seguro usando Bcrypt.
    - A resposta à pergunta de segurança também é armazenada como um hash, garantindo que nem mesmo ela fique exposta em texto simples.
- **Proteção contra Vulnerabilidades Web:**
    - **SQL Injection:** Prevenida através do uso do ORM SQLAlchemy, que parametriza todas as consultas ao banco de dados.
    - **Cross-Site Scripting (XSS):** Mitigada pelo motor de templates Jinja2, que escapa automaticamente todas as variáveis renderizadas no HTML.
    - **Cross-Site Request Forgery (CSRF):** Protegida pela biblioteca Flask-WTF, que gera e valida tokens CSRF em todos os formulários.