Para esse projeto eu pensei em fazer um serviço bem simples, que pode ser facilmente implementado e bem leve.

Utilizei um framework chamado flask, que é muito leve e mais enxuto em relação ao django especificamente.

O projeto de geração de senha tem 3 chamadas, sendo uma delas a entrada para colocar as especificações para a geração da senha, que são elas :

- settings_pass -> na rota /generate, que é a entrada para a tela onde serão especificados os argumentos para geração da senha, além disso esse método pega o log de quem passa por ele para análise futura.
- copping_pass -> na rota /generated, que recebe um POST, com os argumentos: description, limit_access, limit_time, have_strings, have_numbers, have_special_characters, total_characters, respectivamente. Todos estão presentes na class Generator que administra a senha. esse método é o principal que gera a senha, fazendo um log de quem gerou e retorna o link com o argumento code_access criptografado de acesso a senha que é passado para quem solicitou a geração da senha.
- passwords_getter -> na rota /, que recebe um argumento: code_access, esse método gera um log de quem o acessou, faz a verificação para certificar que a senha está ativa e se estiver tudo correto a retorna.

Após a especiação, a senha é gerada de forma (pseudo) randômica, utilizando a biblioteca random, o método choice onde é passado uma lista das strings de acordo com o que o usuário solicitou e a quantidade de caracteres solicitadas pelo usuário. Recomendo a utilização desse método somente para a utilização de baixo e médio risco, para alto risco é necessária a utilização de uma biblioteca verdadeiramente randômica.

Após a senha gerada, é retornada uma URL seguindo de uma Serialized Key utilizando uma biblioteca nativa do flask para dificultar a decodificação. Assim, com esse link é possível que qualquer pessoa visualize a senha dentro do tempo especificado e pela quantidade de vezes especificadas.

As senhas são todas salvas em memória para uma questão de segurança, pois dessa forma fica mais difícil de acessá-las e se o servidor sofrer um ataque as senhas seriam apagadas por segurança, os logs também mas podem ser futuramente migrados para serem salvos em um banco de dados para manter o histórico e para futuras análises.

Foi implementado um método que roda de 30 em 30 minutos para verificar se as senhas estão inspiradas.