1. **Introduction :** Objectif du projet
2. **Technologies utilisées :** Python, Flask, JS, Chart.js
3. **Structure du site :** Accueil, SQL, HTTP, Historique, Graphiques
4. **Fonctionnement du modèle :** SVC / MultinomialNB
5. **Détection d’attaques :** Types, exemple de détection
6. **Interactivité :** Historique dynamique, analyse en temps réel
7. **Améliorations possibles**
8. **Conclusion**

**1. 🔍 Page de recherche ou filtre de l'historique**

* Permet de filtrer les requêtes par type (danger/safe) ou mot-clé.

**2. 🧠 Ajout d’une explication du résultat**

* Exemple : pour chaque attaque détectée, afficher la raison (ex : "Caractère suspect ' OR 1=1 détecté").

**3. 📂 Upload de fichiers logs**

* Permettre à l'utilisateur d’envoyer un fichier .txt ou .log contenant des requêtes.
* Le serveur les analyse ligne par ligne et affiche les résultats.

**4. 📈 Page d’administration avec statistiques plus poussées**

* Nombre total de requêtes analysées % de requêtes dangereuses
* Graphique évolution horaire ou journalière (fictif ou réel)

**5. 🧪 Ajout de tests automatiques**

* Petits tests dans un fichier Python (test\_predict.py) pour vérifier si les prédictions sont correctes.

**6. 🔐 Ajout d’un système de connexion (authentification simple)**

* Pour que seuls certains utilisateurs puissent accéder aux pages sensibles comme /analystic.
* Flask avec session suffit.