



**GUARDIA**  
CYBERSECURITY SCHOOL

## **CAHIER DES CHARGES**

**Fondamentaux de l'algorithmie &  
Programmation Python**



## INTRODUCTION

Nom du module : Fondamentaux de l'algorithmie & Programmation Python

Nom de l'UE : DevSecOps

Nombre de crédits ECTS : 5

Date : du 09 décembre au 09 janvier

Classe : GC1

### MODALITÉS

Ce projet se fait individuellement

### CONTEXTE ET DESCRIPTION DU PROJET

Ce projet a pour but de mettre en pratique les notions de programmation en Python vues en première partie de ce module. Il se fera en trois temps.

Premièrement, vous allez développer une application qui permet à un commerçant de gérer une liste de ses produits. L'application permettra à l'utilisateur (le commerçant en question) de lire un fichier texte contenant une liste de produits et d'effectuer des opérations comme l'ajout, la recherche et le tri de ces produits. Vous devrez structurer votre code pour qu'il soit clair, efficace et modulaire.

Deuxièmement, vous devrez étendre la première partie du projet à une plateforme de gestion de commerçants. Pour ce faire, il vous faudra une base de données locale sous forme d'un fichier CSV contenant des utilisateurs et leur mot de passe. Chacun aura sa liste de produits à gérer. Il vous faudra alors renforcer la sécurité de la base de données. Le long de ce processus, vous allez détecter les mots de passe compromis et proposer des suggestions pour les renforcer. Vous généraliserez la détection des mots de passe ou des adresses e-mail compromis

en utilisant une API. Votre gestionnaire devra alors envoyer des alertes aux utilisateurs concernés.

Et enfin, munir votre gestionnaire d'une interface graphique simple en intégrant un outil comme Tkinter ou PyQt. Vous ajouterez la possibilité de recevoir des commandes. La base de données doit être adaptée en conséquence. Par la suite vous devrez permettre d'afficher des statistiques des données sous forme de graphiques avec des bibliothèques comme Matplotlib ou Seaborn.

En plus de Git pour gérer les versions de votre code, une dockerisation de votre application est aussi demandée.

En bonus, vous pouvez créer votre propre API pour exposer les données produits et recevoir les commandes.



## OBJECTIFS PÉDAGOGIQUES ET PROFESSIONNELS DU PROJET

A l'issue de ce projet, les étudiants seront capables de :

### Savoirs

- Comprendre les structures de données en Python (listes, dictionnaires).
- Apprendre à manipuler des fichiers texte et CSV en Python.
- Apprendre à manipuler des bases de données locales en Python..
- Comprendre les concepts avancés de hachage (salage, force brute).
- Comprendre l'importance de la gestion des mots de passe et des compromis dans la sécurité.
- Comprendre les API REST, apprendre à interagir avec des API tierces..
- Apprendre à envoyer des emails de notification en Python.

### Savoir-être

- Respecter les délais.
- Évaluer sa propre contribution.
- Structurer le code de manière claire et modulaire pour faciliter la maintenance et l'évolution.
- Travailler de manière organisée pour gérer plusieurs fonctionnalités.
- Assurer une gestion responsable et sécurisée des mots de passe utilisateurs.

## Savoir-faire

- Développer une application Python modulaire et fonctionnelle.
- Utiliser des structures de contrôle pour gérer des choix utilisateur via un menu.
- Implémenter des processus de salage et de hachage, puis comparer les hachages avec une base de données compromise.
- Développer une application Python qui interagit avec une API externe et stocke les résultats dans une base de données.
- Mettre en place un système de surveillance pour informer les utilisateurs des nouveaux compromis.

## RESSOURCES

### 1. A savoir/à apprendre :

- Git/Github pour la gestion des versions du projet.

### 2. A lire/à consulter :

- **Livre** : "Automate the Boring Stuff with Python" par Al Sweigart (Chapitre sur les fichiers et les structures de contrôle).
- **Tutoriel** : [How To Hash Passwords In Python](#).
- **Documentation** : [Have I Been Pwned API Documentation](#)

### 3. A installer :

- Python 3.
- Le module **requests** pour manipuler des requêtes HTTP en Python.
- Le module **hashlib** pour la génération de hachages.
- Les modules **PyQT**; **Matplotlib** pour l'aspect visualisation.



## ROADMAP

### DESCRIPTION

Veillez trouver ci-dessous une description des livrables attendus ainsi que les dates d'échéance associées. Il est essentiel de respecter les échéances suivantes pour assurer une progression harmonieuse et structurée du projet. Chaque livrable représente une étape importante dans le processus de réalisation et permet d'évaluer l'avancement du travail. Les dates limites fixées doivent être rigoureusement respectées afin de garantir une évaluation équitable et de permettre un feedback constructif en temps opportun.

Jalon	Livrables attendus	Date limite	Moyens / formats
1	Menu de choix et fonctionnalités de base	semaine 1	- code source

2	Adapter les fonctionnalités à un fichier CSV		
3	Sécurisation de la base de données		
4	Détection des compromis en utilisant une API		
5	Système de notifications en cas de données compromises	semaine 2	- code source
6	Interface graphique simple		
7	Gestion des commandes		
8	Affichage de statistiques		
9	Création de sa propre api	semaine 3	- code source - vidéo de démo globale

Les fonctionnalités à implémenter pour chaque étape sont les suivantes :

### 1. Menu de choix et fonctionnalités de base

- Un menu interactif pour naviguer entre les différentes options.
- Lecture et écriture dans un fichier texte pour sauvegarder et charger les données.
- Utilisation de la programmation modulaire pour structurer le code.
- permettre d'ajouter de nouveaux produits ou de supprimer des produits de la liste.

- permettre de rechercher un produit par son nom. En plus de la recherche séquentielle, il vous est demandé de développer la recherche binaire.
- permettre d'afficher la liste des produits, triée par prix ou par quantité. Vous devrez développer les algorithmes de tri par vous même. Il vous est demandé de développer au moins deux algorithmes (par exemple : tri à bulles et tri rapide).

## 2. Adapter les fonctionnalités à un fichier CSV

- Implémentez une base de données pour stocker les utilisateurs et leurs mots de passe sous forme de fichier CSV.
- Permettez d'ajouter de nouveaux utilisateurs et de mettre à jour ou supprimer leurs informations dans la base de données.
- Générer une base de données de hachages compromis.
- Faire transition d'un fichier texte à une base de données pour les produits. Vous pouvez désormais utiliser un module comme pandas pour la recherche et le tri.

## 3. Sécurisation de la base de données

- Générer un hachage approprié des mots de passe des utilisateurs avec [hashlib](#).
- Comparer ces hachages avec la base de données de hachages compromis.
- Indiquer quels mots de passe sont compromis et proposer des suggestions pour renforcer les mots de passe.
- Implémenter un système de génération de salage des mots de passe avant de les hacher.
- Générer une mise à jour de la base de données avec un hachage adéquat.

## 4. Détection des compromis en utilisant une API

- Implémenter la vérification des mots de passe et des adresses e-mail via l'API « Have I Been Pwned » pour chaque élément.
- Générer une base de données locale avec l'historique des requêtes et des compromis.

## 5. Système de notifications en cas de données compromises

- Ajouter un système de notification ou d'alerte par envoi de mail en cas de nouveaux compromis détectés pour un utilisateur.

## **6. Interface graphique simple**

- Remplacer le menu console par une simple interface graphique.
- Permettre de voir ses produits et faire une recherche.

## **7. Gestion des commandes**

- Pour simplifier cette partie, vous mettrez les commandes sous forme d'objet json dans un fichier. Vous déclenchez manuellement le traitement de ces commandes pour vos tests.
- Vous adapterez votre base de données pour permettre aux commerçants de voir les commandes qu'elles soient traitées ou rejetées.

## **8. Affichage de statistiques**

- Améliorer l'affichage des commandes et de l'approvisionnement par des graphiques sur l'interface du gestionnaire.

## **9. Création de sa propre api**

- Mettre en place un serveur TCP local en écoute de requête GET.
- Décider d'un format de données pour les requêtes afin de les utiliser pour l'envoi de commande et pour l'exposition de produits pour les clients distants.
- Mettre en place une base de données pour stocker les utilisateurs avec la clé qui leur a été attribuée.
- Créer un log des requêtes reçues.





## EVALUATION

L'évaluation est conçue pour être holistique, prenant en compte non seulement le produit final, mais aussi le processus, les compétences acquises et les attitudes démontrées tout au long du projet. Les compétences sont classées en trois catégories principales : "savoirs" (connaissances théoriques), "savoir-faire" (compétences pratiques), et "savoir-être" (compétences interpersonnelles et attitudes).

### SYSTÈME DE NOTATION

#### Savoirs (Connaissances)

- **Compréhension théorique** : Évaluée soit en amont du projet pendant la semaine théorie soit lors de la soutenance et restitution du projet. Cela permet de mesurer la compréhension des concepts fondamentaux et des connaissances liées au projet des étudiants.

#### Savoir-faire (Compétences)

- **Compétences techniques et application** : Évaluées à travers la soumission finale du projet. Cela inclut la qualité, la fonctionnalité, et la précision technique du travail produit.
- **Gestion de projet** : Évaluée en fonction de l'organisation, du respect des échéances, et de l'utilisation efficace des ressources. Cela peut être évalué à travers la documentation du projet et les journaux de processus.

#### Savoir-être (Attitudes/Compétences interpersonnelles)

- **Travail d'équipe et collaboration** : Évalués à travers des évaluations par les pairs et les membres du groupe. Les critères incluent la communication, la coopération, et la contribution aux tâches du groupe.
- **Autonomie et initiative** : Évaluées en fonction des contributions individuelles, de la capacité à travailler de manière autonome, et de la résolution proactive des problèmes.