

Compte-rendu du Système de Suivi de Cohorte d'Étudiants

Yaël Frégier

20 février 2025

1 Problématique Générale

- **Objectif** : Mettre en place un système pour suivre une cohorte d'étudiants de licence de mathématiques.
- **Besoins fonctionnels** :
 - Importer une liste d'étudiants (noms et adresses email) à partir d'un fichier Excel.
 - Générer pour chaque étudiant un lien personnalisé contenant un token unique.
 - Envoyer un email à chaque étudiant avec ce lien.
 - Proposer un formulaire web pré-rempli (nom et email) où l'étudiant peut renseigner son parcours post-licence (formation, établissement, parcours professionnel).
 - Stocker les informations en base de données.
- **Contraintes techniques** :
 - Utilisation de frameworks compatibles avec Python (Flask, SQLAlchemy).
 - Hébergement sur un serveur local.
 - Gestion des envois d'emails via un serveur SMTP (initialement avec Proton, puis adapté à Gmail).

2 Solution Proposée

2.1 Création de la Base de Données et des Modèles

- Utilisation de **Flask-SQLAlchemy** pour définir deux modèles principaux :
 - **Student** : stocke le nom, l'email, le token et la date d'envoi.
 - **Reponse** : enregistre la réponse via le formulaire (formation, établissement, parcours professionnel et date de réponse).
- Initialisation de la base de données via un script dédié (`init_db.py`) qui appelle `db.create_all()`.

2.2 Importation des Données et Envoi d'Emails

- Un script (`send_emails.py`) lit le fichier Excel (`liste_etudiants.xlsx`) grâce à **pandas** et **openpyxl**.
- Pour chaque étudiant, un token unique est généré (via **uuid**).

- L'étudiant est enregistré dans la base de données si son email n'existe pas déjà.
- Un email personnalisé est envoyé via **smtplib**, contenant un lien pointant vers un formulaire web accessible à l'URL `http://localhost:8000/form?token={token}`.

2.3 Application Web pour le Formulaire

- Développement d'une application web **Flask** qui héberge le formulaire accessible via le lien personnalisé.
- Le formulaire affiche les informations pré-remplies (nom et email) et permet aux étudiants de compléter leur parcours.
- À la soumission, les informations sont enregistrées en base (dans la table **Reponse**).

2.4 Configuration SMTP et Envoi d'Emails via Gmail

- Initialement, un compte Proton était utilisé, mais la solution a été adaptée pour Gmail.
- Paramètres utilisés :
 - Serveur SMTP : `smtp.gmail.com`
 - Port : 587 (avec l'utilisation de `starttls()`).
- Gestion des problèmes d'authentification via l'utilisation d'un mot de passe d'application ou l'activation de l'accès aux applications moins sécurisées.

2.5 Déploiement et Configuration Locale

- Le projet est destiné à être hébergé sur un serveur local.
- Utilisation d'un environnement virtuel et d'un fichier `requirements.txt` pour gérer les dépendances.
- Gestion du chemin de la base de données via une URI adaptée (par exemple, `sqlite:///instance/students.db`) avec vérification de l'existence du dossier.

3 Conclusion

La solution proposée intègre plusieurs fonctionnalités clés :

1. L'importation de données depuis un fichier Excel.
2. L'envoi d'emails personnalisés contenant un lien unique vers un formulaire web.
3. L'hébergement d'une application Flask qui sert le formulaire permettant aux étudiants de compléter leur parcours post-licence.
4. La gestion des données via une base de données SQLite, initialisée et manipulée par SQLAlchemy.

Ce système offre ainsi un suivi complet de la cohorte d'étudiants, en combinant l'automatisation de l'import et de l'envoi des emails avec une interface web interactive pour la collecte des réponses. Les principaux défis résidaient dans la configuration correcte de l'accès à la base de données et la gestion de l'authentification SMTP, qui ont été résolus par des ajustements de configuration et l'adaptation aux contraintes locales.