# **Justification de nos choix technologiques**

## **1. Choix de Jinja pour l’interface utilisateur**

Nous avons choisi **Jinja** comme moteur de templates pour la partie interface, car il permet de combiner facilement du code Python avec des fichiers HTML/CSS. Cette intégration fluide est essentielle pour créer des pages web dynamiques qui s’adaptent aux besoins des utilisateurs, notamment les élèves et les enseignants.

Grâce à Jinja, nous pouvons injecter les données calculées par l’algorithme en Python directement dans les templates HTML, ce qui facilite la gestion des votes, des affinités et des résultats des groupes.

Candice ayant déjà utilisé cette techno au sein de son stage en L1 elle nous a donc expliqué comment il fonctionnait.

## **2. Utilisation de Flask pour la logique métier et l’authentification**

Nous avons retenu **Flask**, un framework Python léger et flexible, pour développer la logique métier et la gestion de l’authentification. Flask permet d’implémenter facilement les fonctionnalités serveur nécessaires au traitement des données, à l’exécution de l’algorithme de regroupement des étudiants, et à la sécurisation de l’accès via un système de comptes utilisateurs.

Sa simplicité et sa modularité facilitent l’intégration avec Jinja pour le rendu des pages, ainsi qu’avec la base de données SQL pour la gestion des utilisateurs et des groupes.

## **3. Utilisation de JSON pour la gestion des données temporaires**

Nous avons choisi de gérer les groupes à l’aide de fichier JSON, car cette techno permet de gérer facilement les taillis de groupes. Comme le prof peut choisir le nombre d'élèves par groupe, il faut pouvoir gérer la taille des groupes dans les fichiers et JSON le permet bien.

De plus, un élève peut mettre un poids de 1 par personne, soit une liste de 100 élève, alors qu’un autre peut mettre un poids de 100 sur un seul élève. JSON est utile car il pourra les gérer en même temps.

## **4. Choix de la base de données SQL et organisation simplifiée**

Nous avons choisi d’utiliser une base de données sql pour gérer notre table avec les utilisateurs. MySQL nous permet de mieux gérer les pages d’inscription et de connexion car dès le boutent enregistrer les éléments sont envoyé et stocké dans la base de données. C’est aussi très pratique pour les appels d'étudiants dans les votes car il suffit d’une simple requête pour récupérer les données.

## **5. Utilisation de GitHub pour la gestion du projet**

Pour l’utilisation de Github, nous avons choisi de gérer sur une seule branche. On sait que c’est vraiment pas optimisé ni pratique pour des projet de groupes, il aurait fallu que chacun crée sa branche afin de travailler de son côté et faire un merge régulier afin de tous travailler sur des projet similaire.

Nous avons fait ce choix car pour ce projet on était constamment à trois a se parler et communiquer sur quand push et pull et également nous avons travaillé sur les même fichier en même temps ce qui est parfois compliqué avec des branché séparés. Si nous avions ce travail en distanciel, il aurait été logique de partir sur des branches personnelles avec des réunions régulières pour merge.