# **Justification de nos choix technologiques**

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

## Nous avons opté pour Jinja comme moteur de templates afin de gérer l’interface utilisateur, car il permet de combiner aisément du code Python avec des fichiers HTML et CSS. Cette intégration fluide est essentielle pour générer dynamiquement les pages web en fonction des données issues du backend. Jinja nous permet d’injecter directement les résultats produits par l’algorithme Python dans les pages HTML, ce qui facilite la gestion des préférences exprimées par les étudiants, l’affichage des groupes et le retour d’information aux utilisateurs. Candice, ayant déjà travaillé avec Jinja lors de son stage en première année, a pu guider efficacement l’équipe sur son utilisation et ses bonnes pratiques.

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

## Nous avons choisi Flask, un framework Python léger et modulaire, pour développer la logique métier de notre application ainsi que la gestion de l’authentification. Flask offre une grande simplicité d’utilisation tout en étant suffisamment puissant pour gérer l’ensemble des opérations serveur : traitement des formulaires, gestion des sessions, interaction avec la base de données, exécution de l’algorithme de regroupement, etc. Sa structure minimaliste facilite l’intégration avec Jinja pour le rendu dynamique des pages HTML, mais aussi avec MySQL pour gérer les comptes utilisateurs et récupérer les données nécessaires (liste des étudiants, rôles, etc.). Ce choix technologique nous a permis d’avoir une architecture claire, évolutive et bien adaptée au contexte pédagogique du projet.

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

Nous avons choisi d’utiliser le format JSON pour stocker les données temporaires, notamment les préférences des étudiants et les groupes générés. Ce format est particulièrement adapté car il permet une lecture et une écriture simples, tout en offrant une grande flexibilité dans la structure des données.  
Par exemple, comme l’enseignant peut définir librement la taille des groupes, il est essentiel de pouvoir gérer dynamiquement des listes de tailles variables, ce que JSON permet naturellement.  
De plus, les préférences exprimées par les étudiants sont très hétérogènes : certains peuvent répartir leurs poids sur de nombreux camarades, tandis que d’autres concentrent tout leur poids sur un seul. Grâce à sa structure souple, JSON permet de gérer facilement ces cas très différents sans rigidité ni limitation technique.

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

## Nous avons opté pour une base de données SQL afin de gérer les informations des utilisateurs. MySQL s’est imposé comme un choix pertinent car il facilite le traitement des inscriptions et des connexions : dès que l’utilisateur clique sur le bouton "S’inscrire", ses données sont directement enregistrées et stockées dans la base. Ce choix est également avantageux pour la récupération des étudiants lors de la saisie des préférences de vote. Une simple requête SQL suffit à extraire les noms depuis la table users, ce qui rend le système à la fois rapide et fiable.

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

Pour ce projet, nous avons fait le choix de travailler sur une seule branche GitHub. Bien que cette méthode ne soit pas optimale pour la collaboration en équipe, il est en effet recommandé de créer une branche par développeur, puis de fusionner régulièrement, elle s’est révélée adaptée à notre organisation.  
En effet, nous étions en communication constante à trois, ce qui nous permettait de nous coordonner efficacement sur les moments de *push* et *pull*. De plus, nous travaillions souvent sur les mêmes fichiers, ce qui aurait compliqué la gestion des conflits avec des branches séparées.  
Dans le cadre d’un projet à distance ou avec des horaires décalés, il aurait été préférable d’adopter une approche plus rigoureuse avec des branches individuelles et des *merge* réguliers accompagnés de réunions de synchronisation. Néanmoins, dans notre contexte, nous avons su gérer collectivement les conflits lorsqu’ils survenaient.

**6.Pourquoi avoir utilisé l’algorithme CP-SAT de Google OR-Tools**

L’algorithme CP-SAT (Constraint Programming - Satisfiability) proposé par Google OR-Tools a été retenu pour sa capacité à résoudre efficacement des problèmes d’optimisation sous contraintes. Dans notre cas, le problème à résoudre consiste à affecter un ensemble d’étudiants à des groupes fixes tout en maximisant leur satisfaction mutuelle (via des préférences pondérées).  
CP-SAT est particulièrement bien adapté à ce type de problème combinatoire, car il permet de définir des variables booléennes (x[i, g]) et des contraintes complexes (taille des groupes, affectation unique, maximisation de la satisfaction mutuelle) de manière déclarative. Il explore ensuite intelligemment l’espace des solutions pour trouver le meilleur possible.  
Nous avons préféré ce solveur à une méthode heuristique ou à une simple boucle Python, car il garantit une solution optimale ou quasi-optimale dans un temps raisonnable, même pour des dizaines d’élèves.

**7.Pourquoi avoir réalisé un diagramme de séquence**  
Nous avons produit un diagramme de séquence pour mieux comprendre les interactions entre les différents composants du système, en particulier lors du processus de vote et de génération des groupes.  
Ce diagramme nous a permis de clarifier les rôles de chaque entité (élève, professeur, back-end Flask, base de données, fichiers JSON) dans un scénario utilisateur concret. Il a été très utile pour structurer notre développement, identifier les points critiques (par exemple : moment où les préférences sont sauvegardées, ou où les groupes sont calculés), et anticiper les éventuels bugs liés à l’ordre des appels ou au partage des données.  
Enfin, il nous a aussi permis de mieux répartir les tâches dans l’équipe, chaque membre se concentrant sur une séquence bien définie (authentification, saisie des préférences, exécution de l’algorithme, affichage des résultats).