

# Cahier des Charges : API de Traitement des Données

## 1. Contexte

Le projet vise à concevoir une API permettant d'automatiser le traitement des données, notamment le nettoyage et la normalisation des fichiers CSV. Cette API simplifiera la manipulation des données en prenant en charge la détection et le traitement des valeurs manquantes, aberrantes et dupliquées.

## 2. Objectifs

- Développer une API qui facilite le nettoyage des données brutes.
- Automatiser les tâches de traitement des fichiers CSV.
- Fournir une interface utilisateur permettant l'importation des fichiers et la récupération des données nettoyées.
- Garantir une architecture modulaire et extensible.

## 3. Périmètre Fonctionnel

L'API prendra en charge les fonctionnalités suivantes :

1. Chargement de fichiers CSV.
2. Traitement des données :
  - Détection et gestion des valeurs manquantes.
  - Détection et gestion des valeurs aberrantes.
  - Détection et suppression des doublons.
  - Normalisation des données.
3. Exportation des données nettoyées au format CSV.
4. Interface utilisateur pour l'importation des fichiers.
5. Documentation de l'API.

## 4. Technologies et Outils

- Backend : Flask (Python) pour la gestion des requêtes API.
- Frontend : HTML, CSS, JavaScript, Bootstrap.
- Base de données : (Si nécessaire) SQLite ou PostgreSQL.
- Bibliothèques Python : Pandas, NumPy, Scikit-learn.

## **5. Architecture du Système**

- Utilisateur : Dépose un fichier CSV via l'interface web.
- Frontend : Interface web qui permet l'interaction avec l'API.
- Backend : API Flask qui traite les fichiers et renvoie les données nettoyées.
- Stockage temporaire : Stockage des fichiers traités (si nécessaire).

## **6. Contraintes et Exigences**

- Performances : Optimisation du traitement des fichiers.
- Sécurité : Validation des fichiers uploadés.
- Ergonomie : Interface simple et intuitive.
- Compatibilité : Prise en charge des navigateurs modernes.

## **7. Délais et Livrables**

- Durée : 14 jours à compter de la réception du sujet.
- Livrables :
  - Code source de l'API et du frontend.
  - Documentation technique et utilisateur.
  - Rapport de projet.

## **8. Équipe et Organisation**

- Travail en binôme.
- Répartition des tâches : Backend, Frontend, Tests et Documentation.