CAHIER DES CHARGES

**1. Contexte du projet**

L'Université Sorbonne Paris Nord possède une collection de près de 17 000 jeux de société, certains datant du XIXe siècle. La gestion actuelle de cette collection repose sur un fichier Excel unique, contenant le legs initial, qui est insuffisant pour répondre aux besoins de curation de l'équipe.

**2. Objectif du projet**

L’objectif est de développer une application web avec une base de données relationnelle et des scripts de nettoyage et d'importation de données pour :

- Faciliter la gestion des informations des jeu.

- Gérer l’administration et une recherche avancée au sein de la collection.

- Gérer des extensions pour des fonctionnalités futures (statistiques, API, etc.).

**2. Périmètre du projet**

* Le projet comprend trois phases :
  + Nettoyage et importation des données depuis le fichier Excel.
  + Développement de l'application web pour gérer la collection.
  + Documentation pour les utilisateurs et l'administration.

**3. Fonctionnalités du projet**

***Nettoyage des données et scripts d’importation***

* Scripts de nettoyage des données :
  + Identification et correction des erreurs : scripts pour détecter les erreurs comme les doublons, les champs mal remplis et les données manquantes
  + Validation des données : détection des champs vides, dates incorrectes et valeurs anormales.
  + Normalisation des données : formatage uniforme (ex. : noms d’auteurs, dates, éditeurs).

***Importation et structuration des données :***

- Importation du fichier Excel : Script pour importer les données nettoyées dans la base de données de l’application.

- Modèle de base de données : création de tables pour les entités principales (jeux, auteurs, éditeurs, catégories, années de publication).

- Intégrité des données : mise en place de contraintes (relations correctes entre les auteurs, éditeurs et jeux) pour assurer la cohérence.

***Interface utilisateur pour la gestion de l’inventaire***

* Tableau de bord d’administration :
  + Vue globale de l’inventaire avec recherche et filtres (année, auteur, catégorie, etc.).
  + Vue détaillée des fiches de chaque jeu (titre, date de publication, éditeur, description, etc.).
* Recherche avancée avec filtres :
  + Recherche par titre, auteur, éditeur, année de publication, nombre de joueurs, type de jeu (jeu de plateau, cartes, etc.).
  + Filtres dynamiques permettant de raffiner les résultats.
* Gestion des collections :
  + Ajout, modification et suppression des jeux dans l’inventaire.
  + Gestion des auteurs et éditeurs, avec possibilité de lier un jeu à plusieurs auteurs ou éditeurs.
* - Suivi des prêts et de la localisation physique des jeux :
  + Fonctionnalité pour suivre le prêt des jeux et enregistrer leur localisation dans les archives.

***Fonctionnalités d’administration et de gestion***

* Gestion des utilisateurs et rôles :
  + Système de droits d'accès avec rôles (utilisateur simple, gestionnaire, administrateur).
* Suivi des modifications :
  + Historique des actions réalisées par les utilisateurs (ajouts, modifications, suppressions).
* Gestion des collections supplémentaires :
  + Possibilité d’ajouter de nouvelles collections et catégories (par exemple, ajouter de nouvelles périodes ou types de jeux)
* Extensions et intégrations :
  + Prévoir un cadre extensible pour ajouter des fonctionnalités futures, telles que l'intégration d'une API externe (ex. : base de données publique de jeux).

***Évolutivité et extensions potentielles***

* Numérisation :
  + Ajout d'images et de scans (ex : photos de boîtes, règles de jeu) associés aux jeux.
* Statistiques et rapports :
  + Génération de statistiques (ex : nombre de jeux par période, répartition par type de jeu).
  + Exportation des données : //JSP
  + Fonction pour exporter des jeux ou des listes filtrées au format Excel ou CSV.

**4. Documentation et support**

* Manuel utilisateur :
  + Guide pour les utilisateurs avec instructions pour la gestion des jeux, l’ajout de nouveaux jeux et la modification des collections.
* Guide d'administration technique :
  + Documentation technique pour l’installation, maintenance, et mise à jour de l’application, ainsi que les scripts d’import/export (incluant le schéma des tables de la base de données).
* Gestion des versions et sauvegardes :
  + Mise en place de sauvegardes régulières de la base de données et d'un système de restauration.

**5. Technologies et contraintes**

* Langages et Frameworks : Backend en Python (Flask) ou PHP ; Frontend en HTML, CSS, JavaScript.
* Base de données : **PostGreSQL (pgAdmin4)**

**6. Délais**

***À REMPLIR***

**7. Livrables**

1. Scripts de nettoyage et importation des données.
2. Base de données relationnelle avec tables et relations.
3. Application web pour la gestion de la collection.
4. Documentation utilisateur et guide d’administration.
5. Rapport final et possibilités d'évolutions.