Spécification du projet Kafka

## **Objectifs**

L’objectif principal de ce projet est de développer une solution en Java permettant la mise en place d’un système de médiation de contenu basé sur le modèle **publish-subscribe**. Ce système sera conçu pour :

* **Indexer les contenus** partagés par les émetteurs (« publishers »).
* **Diffuser de manière sélective ces contenus** aux abonnés (« subscribers ») ayant préalablement enregistré leurs centres d'intérêt sous forme de mots-clés, à la manière du service Google Alerts.
* S’appuyer sur des technologies robustes comme **Apache Kafka** et **Apache Zookeeper** pour garantir la scalabilité, la tolérance aux pannes, et une gestion efficace des messages.

## **Fonctionnalités**

Le système à développer inclut les fonctionnalités suivantes, basées sur les interfaces fournies :

### **Fonctionnalités principales**

#### **1. Gestion des utilisateurs**

* **Connexion et inscription** :
  + Les utilisateurs peuvent créer un compte avec des informations telles que prénom, nom, nom d'utilisateur, adresse email et mot de passe.
  + Une interface intuitive permet de se connecter avec un email et un mot de passe.
  + Une option « Mot de passe oublié ? » est présente pour la réinitialisation.

#### **2. Publication de contenu**

* **Interface de publication** :
  + Les utilisateurs (publishers) peuvent sélectionner un topic existant ou créer un nouveau topic.
  + Ils peuvent publier un message ou un lien vers un site web (un des deux seulement).

#### **3. Consultation et gestion des alertes**

* **Personnalisation des abonnements** :
  + Les utilisateurs (subscribers) peuvent rechercher et s’abonner à des topics spécifiques.
  + Une liste des topics abonnés est affichée pour faciliter la gestion.
* **Planning de réception des alertes** :
  + Les utilisateurs peuvent choisir de recevoir des notifications à chaque nouveau message.
  + Ils peuvent configurer une heure de réception quotidienne et la fréquence (en jours).

### **Fonctionnalités supplémentaires**

#### Fonctionnalités implémentées en plus des exigences initiales :

1. **Interface utilisateur moderne** :

Design adaptatif avec une répartition claire entre les éléments de connexion/inscription et de gestion des alertes.

1. **Interface de consultation** :

Les utilisateurs peuvent visualiser les messages publiés dans chaque topic abonné via une interface claire.

1. **Filtrage par thèmes** :

Recherche rapide des thèmes pour trouver des topics précis.

1. **Notifications et planification** :

Planification avancée pour personnaliser la réception des alertes (heure et fréquence).

## **Contraintes**

### **Contraintes techniques**

* **Technologie** : Le système doit être implémenté en **Java**, en assurant la compatibilité avec la version la plus récente du langage.
* **Technologies support** : L'utilisation de **Apache Kafka** et **Apache Zookeeper** est obligatoire pour la gestion des messages et des clusters.

### **Contraintes organisationnelles**

* **Travail collaboratif** : Les étudiants doivent travailler en équipe de cinq pour favoriser la coopération.
* **Livraison et deadlines** :
  + Date limite pour la livraison : **31/01/2025**.
  + Tous les livrables doivent être hébergés sur un dépôt GitHub, incluant :
    - Les spécifications du projet.
    - L’architecture du système.

### **Contraintes de performance**

* Le système doit pouvoir gérer une forte charge de publications et de souscriptions simultanées sans dégradation notable des performances.
* La latence pour la diffusion des messages doit être minimale pour assurer une expérience utilisateur fluide.

* r