

Projet de streaming de conférence

1. Base de données

- Base de Données **SQL** pour les vidéos:

SQL pour stocker les informations des vidéos, cela permet de structurer les données de manière relationnelle. Les informations telles que les métadonnées de la vidéo (titre, description, durée, catégories, etc.) se prêtent bien à une structure SQL.

Pour la gestion des vidéos elles-mêmes (fichiers), il vaut mieux stocker les fichiers vidéo sur un service de stockage comme **Google Cloud Storage**. Ensuite, dans la base de données, on stocke simplement les liens vers les fichiers.

- **MongoDB** pour les utilisateurs et l'historique:

MongoDB pour stocker les données utilisateur, surtout si elles sont semi-structurées (profils avec des préférences de personnalisation, historique, liste de conférences sauvegardées, etc.). MongoDB nous permettra également d'évoluer plus facilement, avec la flexibilité de modifier la structure des documents au fil du temps.

Cas d'utilisation de MongoDB :

1. Stockage de l'historique de visualisation et des préférences utilisateur (données facilement mises à jour).
2. Gestion des recommandations utilisateur, notamment pour stocker des données temporaires ou des informations sur les interactions utilisateur.

2. Technologie Backend

Pour le backend, il nous faut un framework robuste, rapide.

- **Node.js avec Express ou NestJS** : Node.js est parfait pour gérer des applications qui demandent beaucoup d'E/S comme les plateformes de streaming. Express est léger et facile à configurer pour des API REST, et NestJS ajoute une structure orientée objet qui facilite la maintenance et le développement.

3. Technologie Frontend

- **React & NextJS:** React est parfait pour des interfaces dynamiques et est bien supporté pour des applications frontend volumineuses et riches.
- **Tailwind CSS & Material UI :** Pour un design moderne, Tailwind CSS offre une grande flexibilité, tandis que Material UI (React) est bien adapté pour des composants plus sophistiqués.

4. Autres Points Clés

- **Authentification et Sécurité**

Pour l'authentification, on peut utiliser **JWT (JSON Web Tokens)** pour la gestion des sessions côté client, couplé avec **OAuth2** si on souhaite permettre la connexion via des services externes (Google, LinkedIn, etc.).

- **Infrastructure et Déploiement**

Docker : Pour isoler les services backend, frontend, base de données, et gestion des vidéos.

CI/CD : Met en place un pipeline d'intégration et de déploiement continu pour gérer facilement les mises à jour.

- **Optimisation de la Gestion des Vidéos**

Transcodage Dynamique : **Google Cloud Transcoder** peut être utile pour convertir les vidéos en plusieurs résolutions. Cela nous permettra de proposer différents formats de qualité (480p, 720p, etc.) en fonction de la bande passante de l'utilisateur.