**Cahier des charges : Projet de site de streaming musical**

**1. Introduction**

Ce document décrit les objectifs, les fonctionnalités, et les spécifications techniques pour le développement d’un site de streaming musical. Le site vise à offrir une expérience utilisateur intuitive pour l’écoute de musique, tout en permettant une gestion efficace des contenus et des utilisateurs via un panneau d'administration.

**2. Objectifs**

* Permettre aux utilisateurs de streamer de la musique en ligne.
* Proposer une interface utilisateur simple et ergonomique.
* Offrir un système d'inscription et de connexion sécurisé.
* Fournir à l’administrateur les outils nécessaires pour gérer le contenu musical et les utilisateurs.

**3. Fonctionnalités**

**3.1. Interface utilisateur (Frontend)**

**3.1.1. Inscription et connexion**

* Création de compte utilisateur avec e-mail, mot de passe.poù !
* Connexion sécurisée avec validation des identifiants.
* Récupération de mot de passe via e-mail.

**3.1.2. Navigation et écoute de musique**

* Recherche de morceaux par titre, artiste ou genre.
* Lecture audio via un lecteur intégré avec les contrôles suivants : lecture/pause, volume, navigation dans le morceau.
* Possibilité de créer des playlists personnalisées.

**3.1.3. Profil utilisateur**

* Accès à un tableau de bord personnel.
* Modification des informations personnelles.
* Historique des morceaux écoutés.

**3.2. Interface administrateur (Backend)**

**3.2.1. Gestion des utilisateurs**

* Consultation de la liste des utilisateurs.
* Suppression de comptes utilisateurs.

**3.2.2. Gestion des contenus musicaux**

* Ajout de nouveaux morceaux avec les métadonnées suivantes : titre, artiste, album, genre, fichier audio.
* Suppression de morceaux existants.
* Modification des métadonnées d’un morceau.

**4. Contraintes techniques**

**4.1. Frontend**

* Langages : HTML, CSS, JavaScript.
* Frameworks recommandés : React.js ou Vue.js pour une interface dynamique.
* Responsive design compatible avec les appareils mobiles et tablettes.

**4.2. Backend**

* Langages : Node.js, Python (Django/Flask), ou PHP (Laravel).
* Base de données : MySQL ou MongoDB.
* Authentification : Utilisation de JSON Web Tokens (JWT) ou OAuth 2.0.

**4.3. Hébergement**

* Serveur cloud (AWS, Google Cloud, ou Azure).
* CDN pour une diffusion rapide des fichiers audio.

**4.4. Sécurité**

* Chiffrement des mots de passe avec bcrypt ou une solution équivalente.
* Protection contre les attaques par injection SQL et les failles XSS.

**5. Livrables**

* Interface utilisateur fonctionnelle avec toutes les fonctionnalités décrites.
* Panneau d'administration complet pour la gestion des utilisateurs et des contenus.
* Documentation technique et utilisateur.

**6. Planning prévisionnel**

1. **Semaine 1-2** : Analyse des besoins et conception des maquettes (UI/UX).
2. **Semaine 3-5** : Développement de l’interface utilisateur.
3. **Semaine 6-8** : Implémentation du backend et de la base de données.
4. **Semaine 9-10** : Intégration frontend-backend et tests.
5. **Semaine 11** : Lancement et feedback utilisateur.

**7. Équipe projet**

* Chef de projet : Supervise le développement et garantit le respect des délais.
* Développeur frontend : Responsable de l’interface utilisateur.
* Développeur backend : Gère la logique serveur et la base de données.
* Designer UX/UI : Conçoit les maquettes et améliore l’expérience utilisateur.

**8. Budget estimatif**

* Hébergement : 200 €/an.
* Outils et logiciels : 100 €.
* Temps de développement : Selon le tarif horaire des développeurs.