ChatGPT

BandTrack – Cahier des charges version 2

1 – Objet

BandTrack est une application web destinée aux groupes de musique pour partager des suggestions de morceaux, suivre les répétitions et planifier des prestations. Après un premier prototype fonctionnant sans serveur, la version 2 centralise toutes les données sur un backend et intègre de nombreuses améliorations fonctionnelles et ergonomiques.
Ce document précise l'ensemble des fonctionnalités attendues, les contraintes techniques et les règles de gestion mises en œuvre dans cette nouvelle version.

2 – Architecture générale

- **Application monopage (SPA)** : l'interface est une application web dynamique, installable en tant que PWA, qui interagit avec le serveur via des appels HTTP.
- **Backend centralisé** : un serveur Python fournit une API REST (`/api/...`) et stocke toutes les informations (utilisateurs, suggestions, répétitions, prestations, paramètres) dans une base SQLite. Les sessions sont gérées via des cookies HTTP-Only et SameSite pour plus de sécurité [1] [2] .
- **Persistence des données** : les données sont partagées par tous les membres du groupe. Un volume Docker (`bandtrack-data`) conserve la base de données (`bandtrack.db`) et les fichiers audio des notes.
- **Front-end** : l'interface (HTML/CSS/JS) est servie par le backend. Elle utilise l'API REST pour toutes les opérations et ne recourt plus au `localStorage` .

3 – Authentification et gestion des utilisateurs

- **Création de compte et connexion** : chaque utilisateur doit s'inscrire avec un nom d'utilisateur et un mot de passe. Le premier compte créé devient administrateur par défaut. Les mots de passe sont hachés en base (PBKDF2-SHA256).
- **Sessions** : après la connexion, une session est créée côté serveur et un cookie signé est envoyé au navigateur. Les sessions expirent automatiquement au bout d'une semaine ou lors de la déconnexion.
- **Gestion des rôles** : un attribut `isAdmin` permet de distinguer les administrateurs. Les administrateurs peuvent modifier ou supprimer toutes les données et gérer les rôles des autres utilisateurs.
- **Paramètres du profil** : les utilisateurs peuvent changer le nom du groupe et basculer entre mode sombre et clair.
- **Gestion des comptes** (pour les administrateurs) : une rubrique dans les paramètres affiche la liste des utilisateurs avec une case à cocher « Administrateur ». Il est possible de promouvoir ou rétrograder un utilisateur, à l'exception de soi-même.

4 – Suggestions (section « J'aime »)

- **Ajout d'une suggestion** : tout utilisateur connecté peut proposer un morceau. La fiche comprend :

- *Titre* (obligatoire),
- *Auteur* (optionnel),
- *Lien YouTube* (optionnel).
- **Affichage** : les suggestions sont listées avec le titre, l'auteur et un lien cliquable si un URL YouTube a été fourni. Un texte indique qui a ajouté la suggestion.
- **Suppression** : seule la personne qui a créé la suggestion ou un administrateur peut la supprimer. Une confirmation est demandée avant la suppression.

5 – Répétitions (section « Morceaux en cours de travail »)

- **Ajout d'un morceau** : chaque utilisateur peut ajouter un morceau à travailler. Les champs disponibles sont :
- *Titre* (obligatoire),
- *Auteur* (optionnel),
- *Lien YouTube* (optionnel),
- *Lien Spotify* (optionnel).
- **Carte de répétition** : chaque morceau apparaît sous forme de carte affichant le titre, l'auteur et les liens. La carte permet :
- de régler son **niveau personnel** via un curseur 0 – 10 ; la valeur est enregistrée immédiatement et affichée sur la carte ; le design du curseur se colore en fonction du niveau ;
- de saisir des **notes textuelles** personnelles ;
- d'**ajouter une note audio** (fichier mp3/wav jusqu'à 5 Mo) qui est encodée et stockée en base ; si une note audio existe, un lecteur audio et un bouton de suppression apparaissent ;
- de consulter les **niveaux, notes et notes audio des autres membres** (affichées de manière anonyme par prénom/pseudo) ; les noms sont comparés de manière insensible à la casse pour éviter les doublons ;
- de **modifier** (titre, auteur, liens) ou **supprimer** un morceau si l'on en est le créateur ou si l'on est administrateur.

6 – Prestations

- **Ajout d'une prestation** : un formulaire permet de saisir le nom, la date et de sélectionner les morceaux (répétitions) qui seront joués.
- **Affichage** : la page est divisée en deux sections : « À venir » et « Passées », classées selon la date. Chaque carte de prestation est cliquable pour ouvrir une fenêtre détaillant la liste des morceaux (avec titre et auteur).
- **Modification et suppression** : le créateur de la prestation ou un administrateur peut modifier le nom, la date et la liste des morceaux, ou supprimer la prestation.
- **Accès aux répétitions depuis une prestation** : dans la vue détaillée d'une prestation, chaque morceau est cliquable et ouvre sa fiche détaillée (niveau, notes, note audio) pour consulter ou ajuster ses propres informations.

7 – Paramètres et personnalisation

- **Nom du groupe** : modifiable par les utilisateurs via la page « Paramètres ». Le nom est affiché dans le titre de la page et utilisé pour personnaliser l'application.
- **Mode sombre/clair** : un interrupteur permet d'activer le mode sombre ; l'état est enregistré dans les paramètres globaux du groupe.
- **Logo** : un logo est affiché dans un bandeau fixe en haut de toutes les pages afin d'identifier l'application.
- **Déconnexion** : permet de terminer la session en cours.

8 – Gestion des rôles et administration

- **Règles d'édition et de suppression** :
- *Suggestions* : seul le créateur ou un administrateur peut supprimer.
- *Répétitions* : seul le créateur ou un administrateur peut modifier le titre, l'auteur et les liens, ou supprimer le morceau. Tout utilisateur peut toutefois modifier son propre niveau, ses notes textuelles et sa note audio.
- *Prestations* : seul le créateur ou un administrateur peut modifier ou supprimer.
- **Gestion des utilisateurs** : les administrateurs peuvent accéder à une liste de tous les comptes et cocher/décocher le statut administrateur (à l'exception du leur).

9 – Interface et ergonomie

- **Navigation** : une barre fixe en bas de l'écran contient les onglets « J'aime », « Répétitions », « Prestations » et « Paramètres ». Les onglets sont accessibles uniquement après connexion.
- **Modales** : l'ajout et la modification des éléments (morceaux, prestations, suggestions) se font dans des fenêtres modales centrées avec validation et annulation.
- **Confirmation** : toute action destructive (suppression) est précédée d'une question de confirmation.
- **Responsive design** : l'interface est optimisée pour un usage sur mobile et bureau ; le mode sombre adapte les couleurs.
- **Accessibilité** : les champs de formulaires sont étiquetés et les liens externes s'ouvrent dans un nouvel onglet (`target="_blank"`).

10 – Sécurité et conformité

- **Stockage sécurisé des mots de passe** : utilisation de PBKDF2 avec SHA-256 et génération d'un sel aléatoire pour chaque utilisateur.
- **Protection contre le vol de session** : les jetons de session sont stockés en base et transmis via des cookies `HttpOnly` et `SameSite=Lax` [2]. Le front-end n'a pas accès au contenu du cookie.
- **Validation des entrées** : le backend valide systématiquement les données reçues (longueur des titres, format des dates, type d'URL) et rejette les requêtes non authentifiées.
- **Limitation de la taille des fichiers** : les notes audio sont limitées à 5 Mo et sont stockées sous forme d'URL de données complète pour éviter les problèmes de type MIME.

11 – Contraintes techniques

- **Écosystème** : le backend est un service Python exécuté dans un conteneur Docker (`python:3.11-slim`). La base de données SQLite est sauvegardée dans un volume externe.
- **API REST** : toutes les opérations sont exposées via des routes `/api/…` (enregistrement/login, suggestions, répétitions, prestations, paramètres, utilisateurs). Les réponses utilisent un format JSON homogène.
- **Déploiement** : l'application est fournie avec un `Dockerfile` et un `docker-compose.yml` pour faciliter l'installation sur un NAS ou un serveur. Le volume `bandtrack-data` doit être conservé pour maintenir les données et les enregistrements audio.

12 – Évolutions possibles

Les améliorations suivantes pourront être envisagées dans des versions ultérieures :

- Mise en place d'un système de notifications (p. ex. lorsqu'un nouveau morceau est ajouté ou lorsqu'une prestation est modifiée).
- Export du répertoire des morceaux et des notes au format PDF ou CSV.
- Intégration d'un lecteur audio universel pour pré-écouter les morceaux via des API musicales (Spotify, YouTube, etc.).
- Synchronisation en temps réel via WebSocket pour voir instantanément les mises à jour des autres membres.

Cette version 2 du cahier des charges reflète les retours des utilisateurs et les fonctionnalités développées tout au long du projet. Elle sert de base pour la maintenance de BandTrack et l'intégration d'évolutions futures.

---

1 Please Stop Using Local Storage - DEV Community
https://dev.to/rdegges/please-stop-using-local-storage-1i04

2 Managing user sessions: localStorage vs sessionStorage vs cookies
https://stytch.com/blog/localstorage-vs-sessionstorage-vs-cookies/