

TheWave

Rapport du Projet "The Wave Music"

Présentation du Projet

The Wave Music est une plateforme musicale open-source qui permet aux utilisateurs de découvrir, écouter et gérer des morceaux de musique. Le site propose une interface intuitive pour naviguer à travers les albums, artistes, groupes et playlists. Les utilisateurs peuvent également obtenir des suggestions personnalisées basées sur leurs écoutes précédentes.

Manuel Utilisateur

1. Prérequis :

- postgresql@15 (assurer vous d'avoir la version 15)
- python3
- Depuis votre terminal, lancer cette commande dans le **root** du projet pour installer toutes les librairies nécessaires pour le bon fonctionnement du projet :
 - `python3 -m venv venv`
 - `source venv/bin/activate`
 - `pip3 install -r requirements.txt --force-reinstall`
 - `pip3 list` (vérifier la bonne installation des librairies)
 - assurer que votre IDE est configuré sur le correct Interpréteur Python :
 - Dans *VSCode* :
 - Ouvrez la Palette de Commandes (Ctrl+Shift+P).
 - Recherchez **Python: Select Interpreter**.
 - Choisissez l'interpréteur lié à votre projet ou environnement virtuel.
 - Dans *PyCharm* :
 - Allez dans **Paramètres > Interpréteur de projet**.
 - Configurez l'interpréteur correctement.

2. Setup de la base de données en local :

- Assurez-vous de bien démarrer le service postgresql@15 :
 - Pour windows :
 - `pg_ctl -D "C:\Program Files\PostgreSQL\15\data" start`
 - Pour linux :
 - `sudo systemctl start postgresql@15`
 - `sudo systemctl status postgresql@15`
 - Pour MacOS :
 - `brew services start postgresql@15`
 - `ps aux | grep postgresql@15`
- Se connecter à postgres en s'assurant d'utiliser en tant que l'utilisateur **superuser** existant
- Créer l'utilisateur DB pour notre projet et lui attribuer le privilège de manière suivante :
 - `CREATE USER postgres WITH PASSWORD 'postgres';`
 - `ALTER USER postgres CREATEDB;`
- Quitter la connexion actuelle et se connecter à la base de données par défaut en tant que ce nouveau utilisateur :
 - `\q` ou `exit`

- `psql -U postgres -d postgres`
- Créer la base de données et le schéma DB pour le projet :
 - `CREATE DATABASE thewave;`
 - `\c thewave`
 - `CREATE SCHEMA dataset;`
- Quitter de nouveau la connexion actuelle et lancer l'import du dump avec la commande psql :
 - `\q` ou `exit`
 - `psql -U postgres -d thewave -f /<path-to-dump-file>/thewave-2024_12_11_21_42_00-dump.sql`

3. Démarrer l'application :

- Depuis un terminal, lancer la commande suivante dans le **root** du projet :
 - `python3 -m app.main`
- Une fois l'application démarrée, elle sera accessible via un navigateur à l'adresse suivante :
 - <http://127.0.0.1:8080/>

4. Connexion :

- Pour accéder à toutes les fonctionnalités, les utilisateurs doivent se connecter.
 - Pour le moment, l'application ne propose pas la possibilité de s'inscrire aux utilisateurs.
 - Dans la base de données du projet, il existe 3 utilisateurs différents afin de tester l'application :
 - Utilisateur de test 1 :
 - Login : `rockfan123@example.com`
 - Mot de passe : `securePass123`
 - Utilisateur de test 2 :
 - Login : `musiclover@example.com`
 - Mot de passe : `password456`
 - Utilisateur de test 3 :
 - Login : `johnndoe@example.com`
 - Mot de passe : `johnsPassword`
- Une fois connectés, ils peuvent naviguer dans les différentes sections du site.

5. Navigation :

- **Accueil** : Affiche les morceaux les plus écoutés.
- **Albums** : Liste des albums disponibles.
- **Artistes** : Liste des artistes.
- **Groupes** : Liste des groupes.
- **Suggestions Personnalisées** : Recommandations basées sur les écoutes de l'utilisateur.
- **Recherche** : Permet de rechercher des morceaux, artistes, albums et playlists.

6. Écoute de Musique :

- Les utilisateurs peuvent écouter des morceaux en cliquant sur le bouton "Écouter" associé à chaque morceau.

7. Gestion des Playlists :

- Les utilisateurs peuvent créer et gérer leurs propres playlists.

Section Technique

Présentation des Étapes 1 et 2

1. Étape 1 : Configuration Initiale

- Mise en place de l'environnement de développement avec Flask et PostgreSQL.

- Création des modèles de base de données pour les utilisateurs, morceaux, albums, artistes, groupes, et playlists.

2. Étape 2 : Développement des Fonctionnalités

- Implémentation des routes pour la gestion des utilisateurs, morceaux, albums, artistes, groupes, et playlists.
- Intégration de la base de données avec SQLAlchemy.
- Développement de l'interface utilisateur avec Jinja2 et CSS.

Description Technique des Fonctionnalités [↗](#)

1. Gestion des Utilisateurs :

- Inscription, connexion et déconnexion des utilisateurs.
- Utilisation de Flask-Login pour la gestion des sessions.

2. Écoute de Morceaux :

- Route `/morceau/<int:morceau_id>/ecoute` pour incrémenter le nombre d'écoutes d'un morceau.
- Exemple de requête SQL :

```
1 INSERT INTO dataset.ecoute (numUsr, numMorc, date, nombreEcoute) VALUES (%s, %s, %s, %s)
```

3. Suggestions Personnalisées :

- Route `/suggestions` pour afficher des recommandations basées sur les écoutes de l'utilisateur.
- Exemple de requête SQL :

```
1 SELECT Morceau.numMorc, Morceau.titreMorceau, Morceau.duree,
2         array_agg(concat(Artiste.nom, ' ', Artiste.prenom)) AS artistes
3 FROM Morceau
4 JOIN ApparaîtDans ON ApparaîtDans.numMorc = Morceau.numMorc
5 JOIN Ecoute ON Ecoute.numMorc = Morceau.numMorc
6 WHERE Ecoute.numUsr = %s
7 GROUP BY Morceau.numMorc
8 ORDER BY sum(Ecoute.nombreEcoute) DESC
9 LIMIT 10
```

4. Recherche :

- Route `/search` pour rechercher des morceaux, artistes, albums et playlists.
- Exemple de requête SQL pour rechercher des morceaux :

```
1 SELECT * FROM Morceau WHERE titreMorceau ILIKE %s
```

Pistes d'Améliorations [↗](#)

1. Amélioration de la Sécurité :

- Implémenter des mesures de protection contre les attaques CSRF et XSS.

2. Optimisation des Performances :

- Utiliser des index sur les colonnes fréquemment utilisées dans les requêtes SQL.
- Mettre en place un système de cache pour les requêtes les plus lourdes.

3. Nouvelles Fonctionnalités :

- Pour le moment, la plateforme ne supporte pas d'inscription de nouveaux utilisateurs. Il serait indispensable d'investir sur cette nouvelle fonctionnalité.
- Ajouter la possibilité de suivre des artistes et des groupes.
- Implémenter un système de notifications pour informer les utilisateurs des nouvelles sorties.

