

TITRE DU PROJET

Plateforme interactive de gestion et partage de playlists vidéo (Mini-YouTube Personnel)

MOTS-CLÉS

Interface interactive, vidéos YouTube non répertoriées, tableau de vignettes, sélection par case à cocher, partage de playlists, API YouTube, responsive design, développement mobile et web, coaching, démonstrations pédagogiques

DESCRIPTION DU PROJET

L'objectif est de créer une application multi-plateforme qui permet aux professionnels (coach, formateurs, thérapeutes, etc.) et à leurs patients d'accéder, sélectionner et partager des playlists de vidéos. Pour simplifier la gestion des contenus, les vidéos seront hébergées sur YouTube en mode non répertorié, évitant ainsi la gestion d'un stockage dédié, souvent lourd et coûteux.

Les principales fonctionnalités de l'application sont les suivantes :

- **Affichage interactif** : Un tableau de vignettes affichera les vidéos hébergées sur YouTube. Chaque vignette est sélectionnable via une case à cocher, permettant aux utilisateurs de constituer une playlist personnalisée.
- **Création et partage de playlists** : L'utilisateur peut créer une liste de vidéos sélectionnées et la partager facilement avec un patient ou un autre utilisateur. Le partage se fera par lien sécurisé ou via une interface intégrée.
- **Lecture flexible** : Le destinataire de la playlist pourra lire les vidéos individuellement ou en mode séquence, avec la possibilité de modifier sa sélection directement via l'interface.
- **Cas d'usage élargi** : La solution s'adresse non seulement au domaine médical, mais également au coaching, aux démonstrations pédagogiques ou à toute situation nécessitant une gestion intuitive de contenus vidéo.

TRAVAUX À RÉALISER

1. Analyse des besoins et conception fonctionnelle

- **Recueil des besoins utilisateurs :** Identifier les exigences spécifiques des professionnels et des patients (accessibilité, confidentialité, ergonomie).
- **Spécifications fonctionnelles :** Rédiger un cahier des charges décrivant le comportement interactif du tableau, le mécanisme de sélection, la génération et le partage de playlists, ainsi que le mode de lecture personnalisé.
- **Prototypage :** Concevoir des wireframes et maquettes pour valider l'ergonomie de l'interface (tableau de vignettes, navigation intuitive, responsive design).

2. Architecture technique et choix technologiques

- **Frontend :**
 - Développement de l'interface web en HTML5, CSS3 et JavaScript/TypeScript, avec l'aide d'un framework (React, Angular ou Vue.js) pour une interactivité optimale.
 - Pour une version mobile, envisager un développement natif (Kotlin/Java pour Android, Swift pour iOS) ou une approche cross-platform (Flutter, React Native).
- **Backend :**
 - Mise en place d'une API REST pour gérer les playlists, les profils utilisateurs et la synchronisation des données. Technologies possibles : Node.js, Python/Django ou Java/Spring Boot.
 - Implémentation d'un système d'authentification et de gestion des droits d'accès, garantissant la sécurité des données partagées.
- **Intégration avec YouTube :**
 - Utilisation de l'API Data de YouTube pour récupérer automatiquement les miniatures, titres et métadonnées des vidéos hébergées en mode non répertorié.
 - Gestion des liens directs vers YouTube, permettant une lecture intégrée via un lecteur vidéo personnalisé ou une redirection optimisée.
- **Base de données :**
 - Choix d'un SGBD (MySQL, PostgreSQL, ou SQLite) pour stocker les métadonnées des vidéos, les playlists et les informations utilisateurs.
- **Sécurité et confidentialité :**

- Veiller à ce que l'hébergement en mode non répertorié réponde aux exigences de confidentialité, notamment dans le cadre d'un usage médical ou pédagogique.
- Gestion des quotas et surveillance de l'utilisation de l'API YouTube pour garantir une expérience utilisateur fluide.

3. Développement et intégration

- **Interface interactive** : Création du tableau de vignettes avec des cases à cocher pour sélectionner les vidéos. Chaque cellule affichera la miniature, le titre et éventuellement une courte description.
- **Module de partage** : Développement d'un système permettant de générer des liens de partage ou des codes d'accès aux playlists créées.
- **Lecture vidéo** : Intégration d'un lecteur vidéo capable de lire directement les vidéos YouTube (via l'iframe API ou le player intégré) en mode individuel ou séquence.
- **Optimisation mobile et accessibilité** : Adaptation de l'interface pour divers supports (desktop, mobile, tablette) et respect des normes d'accessibilité (contrastes, taille de police, navigation simplifiée).

4. Tests et validation

- **Tests unitaires et d'intégration** sur l'ensemble des modules (interface, API, intégration YouTube).
- **Tests utilisateurs** pour recueillir des retours sur l'ergonomie et l'efficacité de la solution.
- **Itérations de développement** basées sur les feedbacks pour affiner la solution avant déploiement final.

5. Déploiement et maintenance

- **Déploiement** : Publication de l'application sur un serveur cloud pour la version web et via les stores pour les applications mobiles (Google Play, App Store).
- **Suivi** : Mise en place d'un système de monitoring pour suivre l'usage et détecter d'éventuels dysfonctionnements.
- **Documentation** : Rédaction de guides techniques et utilisateurs pour faciliter l'adoption et la maintenance de la solution.

OUTILS MATÉRIELS / LOGICIELS POSSIBLES / SUPPORT

- **Développement Frontend :**
 - Langages : HTML5, CSS3, JavaScript/TypeScript
 - Frameworks : React, Angular ou Vue.js ; Flutter ou React Native pour mobile
- **Développement Backend :**
 - Langages : JavaScript (Node.js), Python, Java
 - Frameworks : Express.js, Django, Spring Boot
- **Base de données :** MySQL, PostgreSQL, ou SQLite
- **Intégration YouTube :** API Data de YouTube pour récupérer et gérer les métadonnées des vidéos en mode non répertorié
- **Environnements de développement :** Visual Studio Code, Android Studio, Xcode
- **Outils de versioning et collaboration :** Git, GitHub ou GitLab

• LIENS ET RESSOURCES

- Documentation API YouTube
- Guide sur l'intégration des vidéos non répertoriées YouTube
- [Tutoriels sur les API REST](#)
- [Normes d'accessibilité web \(W3C\)](#)
- [Exemples d'interfaces interactives](#)