

*  **Documentation du Projet React**

1. Introduction

L'application **ViteAmine** propose une navigation dynamique et plusieurs fonctionnalités interactives : **Quiz, gestion de tâches (Todo), recherche de films, quiz géographique et playlists musicales**. Grâce à **React Router**, l'utilisateur peut naviguer aisément entre les différentes pages.

2. Installation

Prérequis :

- **Node.js** installé
- **npm** ou **yarn** installé

Étapes d'installation :

1. **Cloner le projet :**

Git clone [URL-du-repository] cd nom-du-dossier

2. **Installer les dépendances :**

Npm install

3. **Lancer l'application :**

Npm start

3. Structure du Projet

Dossier/Fichier	Rôle
src/App.jsx	Contient la configuration des routes avec React Router
src/Layout.jsx	Gère la barre de navigation commune
src/pages/Quiz.jsx	Affichage d'un quiz interactif
src/pages/Todo.jsx	Gestion d'une liste de tâches avec stockage local
src/pages/Movies.jsx	Recherche de films via l'API OMDB
src/pages/GeoQuiz.jsx	Quiz géographique utilisant une API de pays
src/pages/Music.jsx	Application musicale intégrant iTunes API

Dossier/Fichier	Rôle
src/components/PlaylistDrawer.jsx	Composant permettant de gérer une playlist musicale



4. Navigation & Fonctionnement

L'application utilise **React Router** pour organiser la navigation entre plusieurs pages.

Routes définies dans App.jsx :

```
<Routes>
  <Route path="/" element= {<Layout />} />
  <Route index element= {<Home />} />
  <Route path="quiz" element= {<Quiz />} />
  <Route path="todo" element={<Todo />} />
  <Route path="movies" element= {<Movies />} />
  <Route path="geoQuiz" element= {<GeoQuiz />} />
  <Route path="mysic" element= {<Mysic />} />
  <Route path="mysic/album/ :id" element= {<Album />} />
  <Route path="*" element= {<Nopage />} />
</Route>
</Routes>
```

Composant Layout.jsx : Barre de navigation

-  Contient les liens vers **Home, Quiz, Movies, GeoQuiz, Todo, Mysic**.
-  Intègre **PlaylistDrawer**, un gestionnaire de musique.

5. Présentation des Fonctionnalités

Quiz (Quiz.jsx)

- **Objectif** : Tester les connaissances de l'utilisateur avec un QCM.
- **Logique** :
 - Stocke la **question en cours** et le **score** avec useState.
 - Vérifie si la réponse sélectionnée est correcte.
 - Affiche le score final avec option de redémarrage.
 -

Todo List (Todo.jsx)

- **Objectif** : Permet à l'utilisateur d'ajouter et gérer des tâches.
- **Stockage** :
 - Utilise **localStorage** pour sauvegarder les tâches.
 - Vérifie la longueur des tâches pour éviter les erreurs.

Recherche de Films (Movies.jsx)

- **Objectif** : Permet de rechercher des films grâce à **OMDB API**.
- **Fonctionnalités** :
 - **Recherche dynamique** via l'API.
 - **Ajout/suppression des favoris**.
 - **Affichage des résultats** sous forme de cartes.

GeoQuiz (GeoQuiz.jsx)

- **Objectif** : Quiz interactif sur les pays du monde.
- **Logique** :
 - Affiche un **drapeau aléatoire**.
 - Propose **4 réponses** avec un pays correct.
 - Ajoute un **score final** et une sauvegarde de score via API locale.

Musique (Mysic.jsx)

- **Objectif** : Recherche d'albums via **iTunes API**.
- **Fonctionnalités** :
 - Affichage dynamique des albums.
 - Ajout à la playlist.
 - **Lecture d'extraits audio**.

Playlist Drawer (PlaylistDrawer.jsx)

- **Objectif** : Gestion de la playlist musicale.

- **Fonctionnalités :**

- Ajout/suppression de chansons.
- Lecture des extraits audio.
- Interface avec **MUI components**.

6. Dépannage et Problèmes Courants

Problème	Solution
L'application ne démarre pas	Vérifier les dépendances (npm install)
Erreur 404 Not Found	Vérifier que React Router est bien configuré
Aucun film ne s'affiche	Vérifier la clé API OMDB et les permissions
La playlist ne fonctionne pas	Vérifier que localStorage est bien activé
Erreur API sur le quiz géographique	Vérifier l'API restcountries.com (requêtes trop fréquentes)

7. Améliorations possibles

- **Ajout d'une authentification** pour gérer les utilisateurs.
- **Stockage des favoris** avec **base de données**.
- **Amélioration du design avec animations** (Framer Motion).
- **Optimisation des performances avec Lazy Loading** (React Suspense).
- **Ajout d'une pagination** pour les films et albums.

8. Conclusion

L'application React présentée offre une **navigation fluide** et des **fonctionnalités interactives** : quiz, musique, tâches, films et géographie. Grâce à **React Router** et différentes API, elle garantit une **expérience dynamique** et **évolutive**.

Ce projet peut être amélioré en intégrant une **authentification**, un **stockage plus avancé** et une **optimisation des performances**. Il constitue une base solide pour des projets interactifs et éducatifs.

 **Merci d'avoir parcouru cette documentation !**