

ÉCOLE SUPERIEURE PRIVÉE DES TECHNOLOGIES DE  
L'INFORMATION ET DE MANAGEMENT DE NABEUL



Mini Projet React  
Mini Projet « Tavel Adventure »

**Sujet :**

**Conception et développement d'une Music App**

Elaboré par

Ahmed Badreddine Atbi

Validé par

MR.Mohamed anis mahjoubi

Année universitaire

2021– 2022

## I. Présentation du projet :

Notre projet consiste à développer une application web de lecture de musique intitulé « MUSIC GO ». Le but de notre projet est de faciliter la recherche et la lecture et d'écouter de la musique pour l'utilisateur afin d'avoir une expérience aisée et plaisante.

## II. Problématique :

L'utilisateur trouve des difficultés, parmi lesquelles :

- Difficulté de la recherche et de l'écoute de la musique sur internet
- Risque de perte des morceaux favoris.
- Difficulté de stockage.
- Limiter la consommation des données mobile.
- Perte de temps.

## III. Solution(s) proposée(s) :

La solution proposée consiste à développer une application web consistante et conviviale.

La réalisation de ce site se résume en deux phases de développement

- Front-end : Partie publique accessible pour tous les utilisateurs.
- Back-end : Partie privée : Un espace réservé à l'administration.

## IV. Analyse des besoins :

### 1. Les besoins fonctionnels

Au cours de cette étape, nous allons extraire les besoins fonctionnels que notre application doit satisfaire :

### Authentification :

L'identification de chaque utilisateur par un login et un mot de passe afin de garantir la sécurité.

### Gestion des titres (l'affichage et la manipulation des titres) :

- L'utilisateur peut consulter les morceaux.
- L'utilisateur peut écouter les morceaux.

### Gestion des listes titres des favoris (l'affichage et la maintenance la liste de favoris) :

- L'utilisateur peut consulter la liste des favoris.
- L'utilisateur peut ajouter, supprimer, chercher et mettre à jour la liste des titres favoris.

### Gestion des utilisateur :

- L'administrateur peut consulter la liste des utilisateurs.
- L'administrateur peut ajouter, supprimer, chercher un utilisateur et mettre à jour les informations des utilisateurs.

## 2. Identification des acteurs

Au niveau de cette phase, nous présentons les différents acteurs susceptibles d'interagir avec le système.

Dans le cas de notre projet nous présentons les deux acteurs principaux :

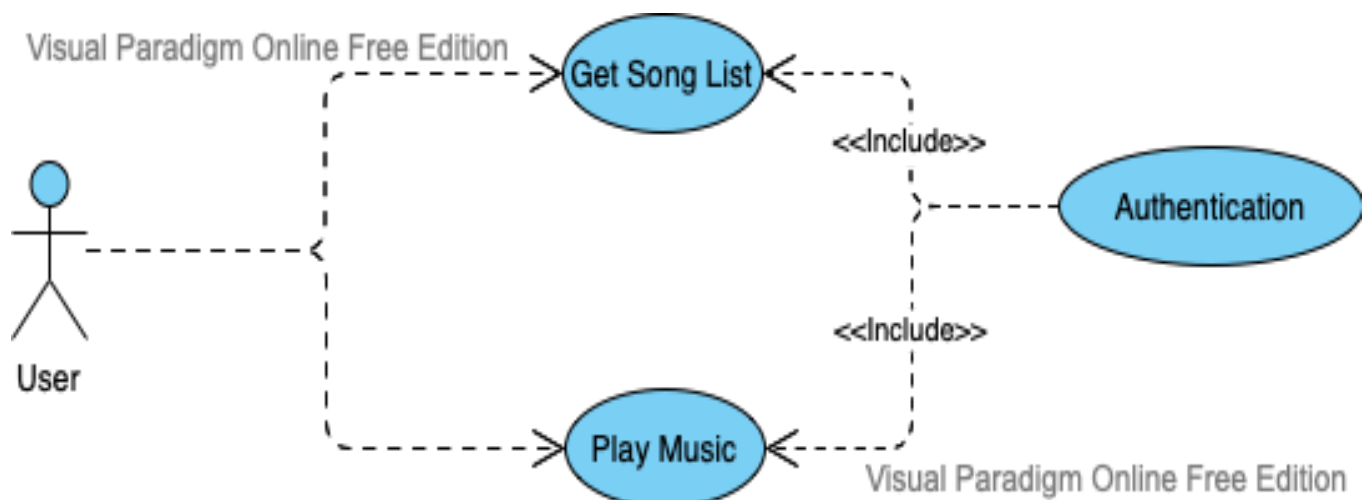
Acteur	Rôle
Utilisateur	<ul style="list-style-type: none"><li>● S'inscrire</li><li>● S'authentifier pour avoir l'accès</li><li>● Consulter les titres.</li><li>● Ecouter les titres.</li></ul>

## Administrateur

- S'authentifier pour avoir l'accès
- Gérer les utilisateurs
- Gérer les titres

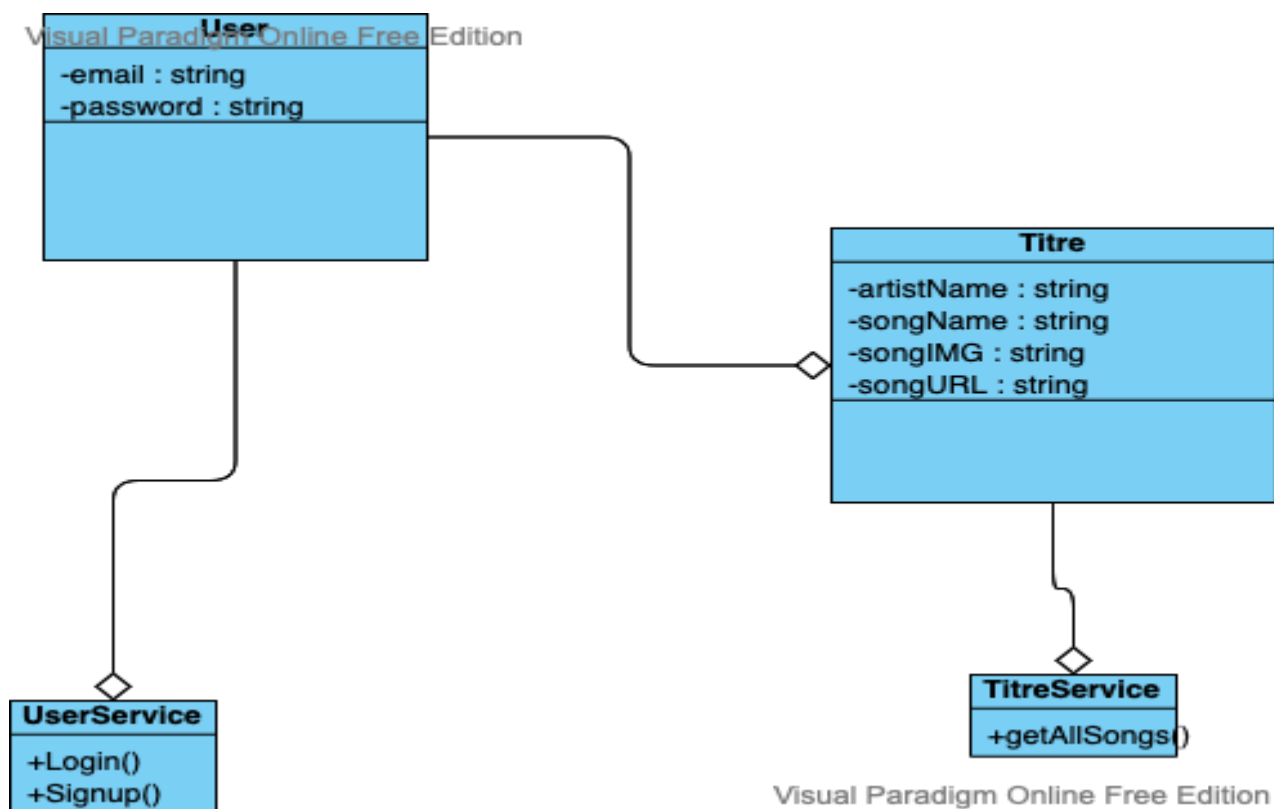
## V. Diagramme de cas d'utilisation général

Ci-dessous, le schéma représentatif du diagramme de cas d'utilisation de notre projet :



## VI. Diagramme des classes :

Ci-dessous, le schéma représentatif du diagramme des classes général de notre projet :



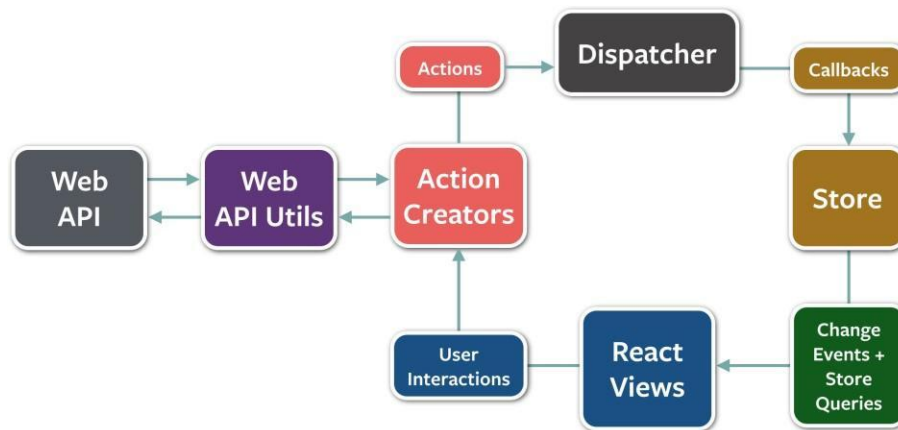
## *Chapitre 3 : Réalisation*

## Introduction :

Ce chapitre représente le dernier volet de ce rapport, il sera consacré à l'implémentation de notre système. Nous commençons par la présentation des ressources matérielles et logicielle utilisées. Nous passons ensuite à présenter des captures d'écran dans le but de mettre en évidence l'aspect ergonomique et fonctionnel des interfaces développées.

### 1. Environnement de développement :

#### 1.1 Architecture Technique :



Contrairement à d'autres bibliothèques et frameworks d'interface utilisateur, Reactjs n'applique pas de modèle d'architecture. C'est juste une vue qui s'adresse à l'interface utilisateur.



Juste en dessous de l'interface utilisateur se trouve une arborescence de plusieurs composants React. En termes généraux, un composant de réaction n'est rien d'autre que l'unité structurelle centrale - quelque chose de minimal, comme un bouton, une étiquette ou une étiquette de saisie de texte ou peut-être quelque chose de plus complexe, comme un formulaire d'inscription, un profil utilisateur, etc.

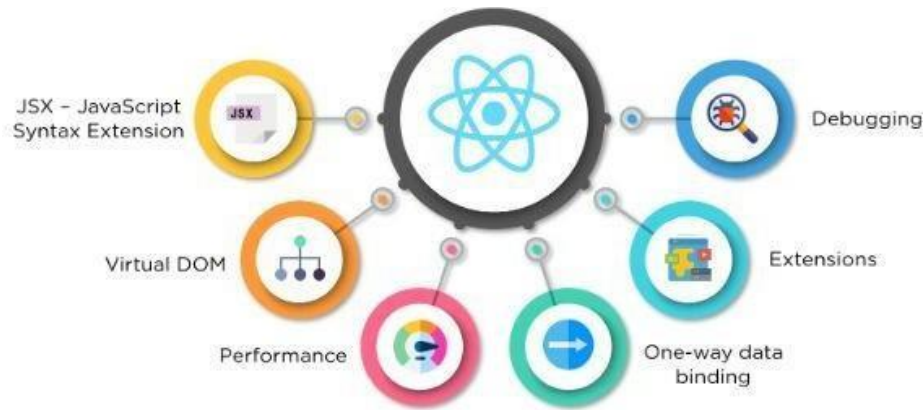
Maintenant, voici la partie passionnante : tous les composants de React peuvent contenir un état ; cependant, ce n'est pas obligatoire. Par état, nous entendons les données nécessaires à suivre pour faire fonctionner l'application. De plus, l'état de votre application ne cesse de changer de temps en temps, en fonction de l'action entreprise par votre utilisateur. C'est seulement l'état des données qui détermine ce qui s'affiche sur l'interface utilisateur d'une application.

React propose déjà un mécanisme prêt à l'emploi appelé l'état local. Cependant, le mécanisme n'est pas suffisant à lui seul pour être utilisé dans des applications du monde réel. C'est là que les bibliothèques de gestion d'état comme Redux entrent en jeu.

## **1.2 Environnement logiciel :**

Nous avons énuméré au cours de cette partie les différents outils utilisés tout au long de ce projet pour l'étude et la mise en place de notre application.

La réalisation de l'application a été développée avec les outils suivants :



React : React est une bibliothèque qui ne gère que l'interface de l'application, considéré comme la vue dans le modèle MVC. Elle peut ainsi être utilisée avec une autre bibliothèque ou un framework MVC comme AngularJS. La bibliothèque se démarque de ses concurrents par sa flexibilité et ses performances, en travaillant avec un DOM Virtuel et en ne mettant à jour le rendu dans le navigateur qu'en cas de nécessité [3]

## { JSON }

JSON Server fournit une fausse API REST complète qui ne nécessite aucune configuration ni aucun codage pour la configuration. Avec son aide, nous pouvons configurer rapidement et facilement un backend pour le prototypage et la moquerie, sans avoir à nous soucier de la logique sous-jacente. De cette façon, nous pouvons nous concentrer davantage sur le développement du frontend et moins sur la configuration du backend.



**React Router** est une bibliothèque standard pour le routage dans React. Il permet la

navigation entre les vues de divers composants dans une application React, permet de

modifier l'URL du navigateur et maintient l'interface utilisateur synchronisée avec l'URL.



**Axios** est une bibliothèque qui sert à créer des requêtes HTTP qui sont présentes en externe. Cela ressort clairement du fait que nous pouvons parfois, dans les applications React, avoir besoin d'obtenir des données de la source externe. Il est assez difficile de récupérer ces données afin qu'elles puissent être normalement affichées sur le site Web.



**HTML5** : est la dernière révision majeure du HTML (format de données conçu pour représenter les pages web). Cette version a été finalisée le 28 octobre 2014. HTML5 spécifie deux syntaxes d'un modèle abstrait défini en termes de DOM.



**CSS** : Les feuilles de styles (en anglais "Cascading Style Sheets", abrégé CSS) sont un langage qui permet de gérer la présentation d'une page Web. Le langage CSS est une recommandation du World Wide Web Consortium (W3C), au même titre que HTML ou XML.



**Bootstrap** : est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc ) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option.



**JavaScript :** JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web. À chaque fois qu'une page web fait plus que simplement afficher du contenu statique — afficher du contenu mis à jour à des temps déterminés, des cartes interactives, des animations 2D/3D, des menus vidéo défilants. JavaScript a de bonnes chances d'être impliqué. C'est la troisième couche des technologies standards du web. [5]



**Figure 10 W3layouts**

W3layouts est une boutique en ligne pour télécharger des modèles de sites Web pour tous les types d'entreprises. Son réutilise des éléments de page Web statiques tout en définissant des éléments dynamiques en fonction des paramètres de demande Web. Les modèles Web prennent en charge le contenu statique, fournissant une structure et une apparence de base. Les développeurs peuvent implémenter des modèles à partir de systèmes de gestion de contenu, de cadres d'applications Web et d'éditeurs HTML.

## **2. Présentation des interfaces de l'application**

Cette section comporte des captures d'écran de quelques interfaces de l'application réalisée accompagnée par une brève description.

## **2.1. Front-end**

Le front-end c'est la partie du code qui est reçue par le client. Je rappelle que le client c'est notre navigateur Web. Il s'agit finalement des éléments du site web que l'on aperçoit à l'écran et avec lesquels on pourra interagir.

### **2.1.1 Interface Home**

Cette interface représente la page d'accueil qui s'affiche dès l'accès à notre site web, elle est constituée d'un menu qui contient un ensemble des sections qui permettent à accéder aux autres interfaces du site.

## **1. Présentation des interfaces de l'application**

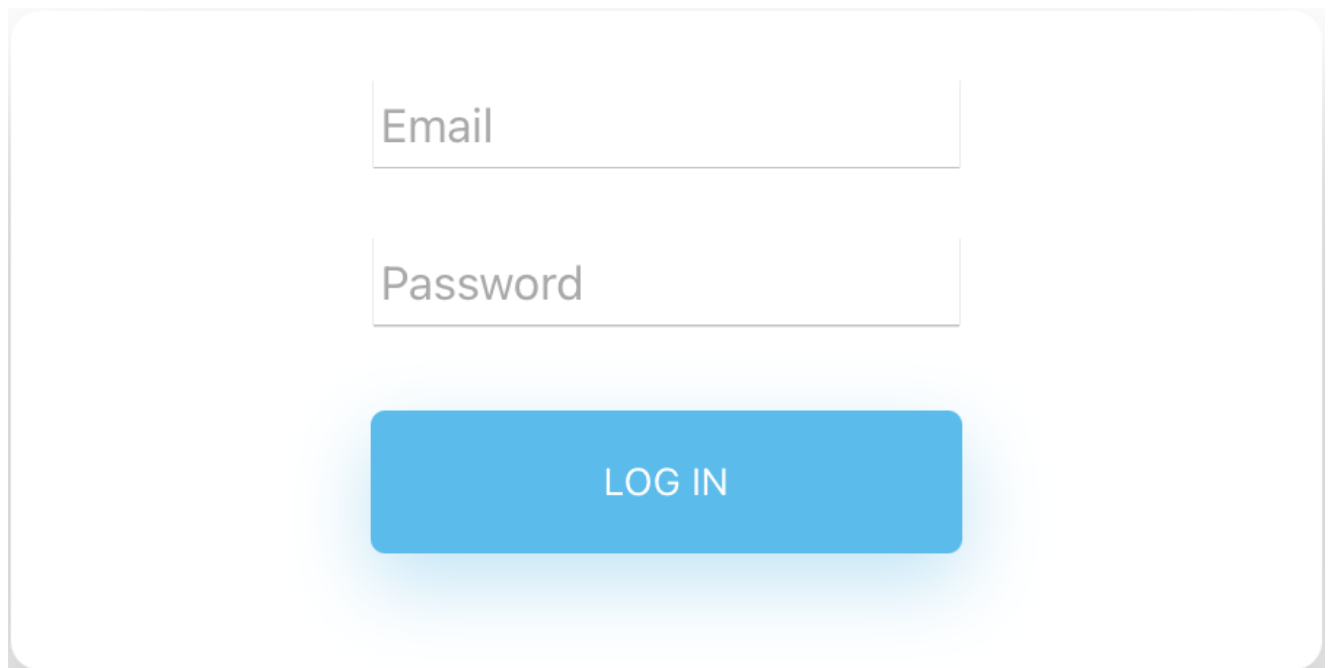
Cette section comporte des captures d'écran de quelques interfaces de l'application réalisée accompagnée par une brève description.

## 1.1. Front-end

Le front-end c'est la partie du code qui est reçue par le client. Je rappelle que le client c'est notre navigateur Web. Il s'agit finalement des éléments du site web que l'on aperçoit à l'écran et avec lesquels on pourra interagir.

### 1.1.1 Interface Login

Cette interface représente la page d'accueil qui s'affiche dès l'accès à notre site web, elle est constituée d'un menu qui contient un ensemble des sections qui permettent à accéder aux autres interfaces du site.

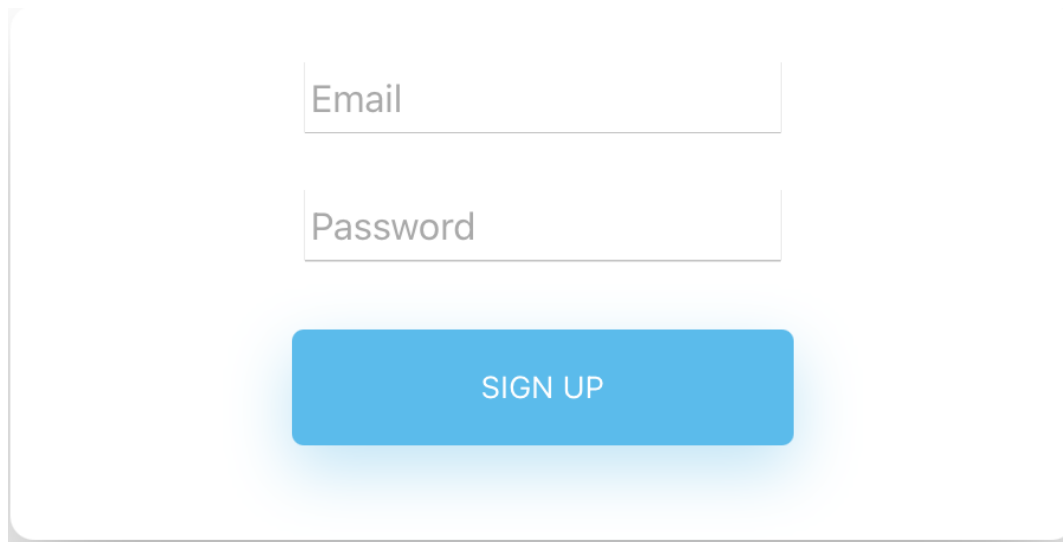
A mockup of a login interface. It features two input fields: the top one is labeled 'Email' and the bottom one is labeled 'Password'. Both fields have a light gray border and a subtle shadow. Below these fields is a prominent blue button with rounded corners and the text 'LOG IN' in white, uppercase letters. The entire interface is set against a light gray background with a soft gradient and a subtle shadow effect.

*Figure 3 interfaces « Login »*

### 1.1.2 Interfaces « Sign up»

La figure ci-dessous présente une petite description de l'interface Sign up pour enregistrer un utilisateur avec son adresse email et son mot de passe



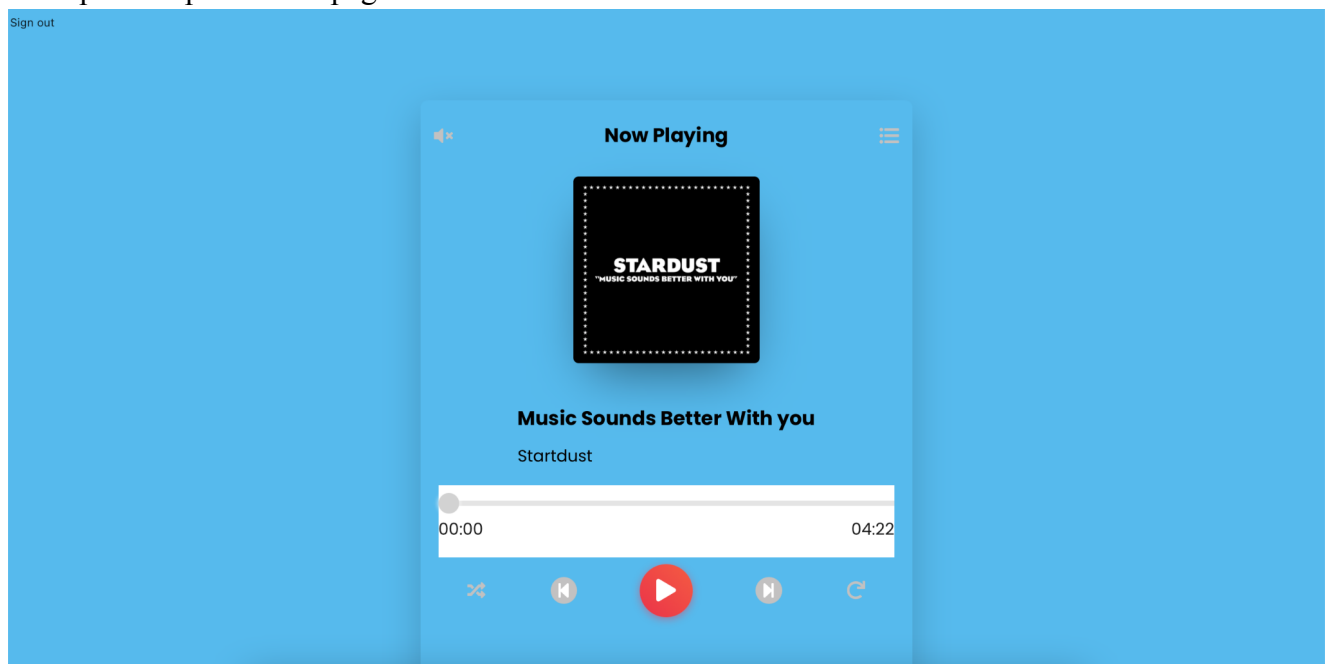


A sign-up form with two input fields: "Email" and "Password". Below the fields is a blue button labeled "SIGN UP". The form is set against a light gray background with rounded corners.

*Figure 4 Interfaces « Sign Up »*

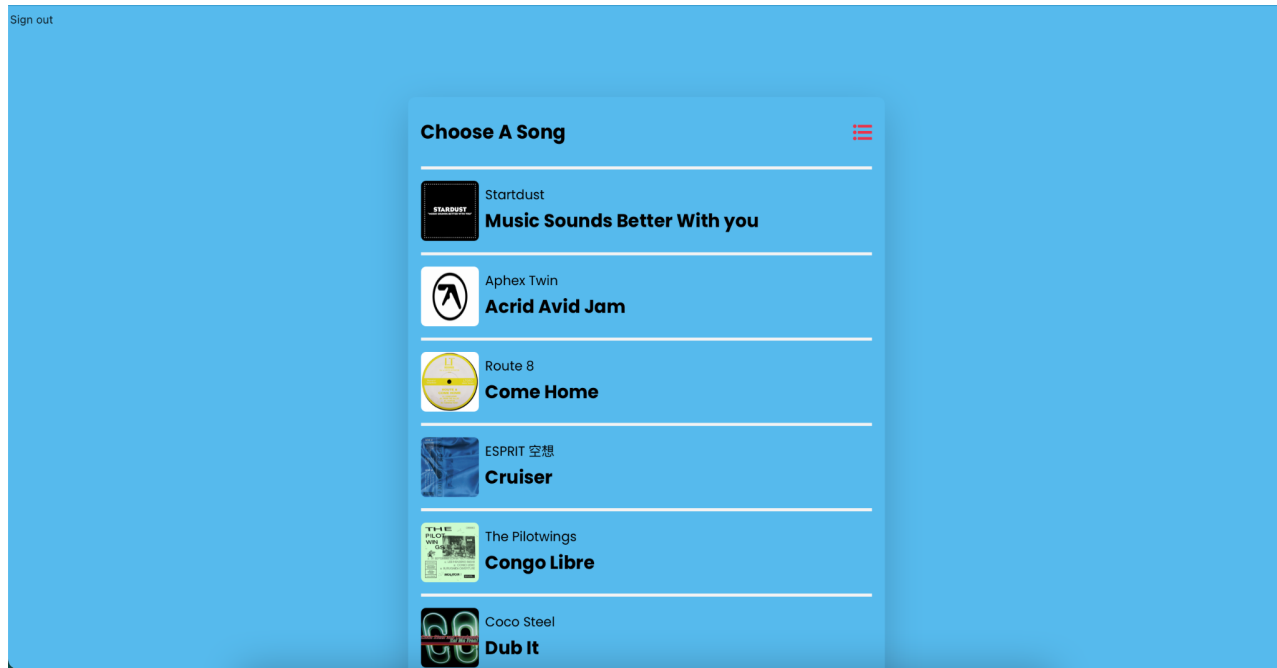
### 1.1.3 Interfaces Home

Cette partie représente la page du Home



*Figure 5 Interfaces home*

### 1.1.4 Interfaces Playlist



*Figure 6 Interfaces playlist*

## Conclusion général

Enfin notre mini projet abouti de créer un site de vente d'idées de décorations dans le domaine du e-commerce. Cette application permet de prendre en charge la majorité des tâches effectuées tel que les thèmes, idée etc. Ainsi ce site web représente une vitrine dans laquelle les clients de visiter le site et ajouter une nouvelle inspiration, Ainsi cette application permet aux architectes d'ajouter une nouvelle idée de décoration

Pour atteindre ce résultat final, on a essayé de retracer les différentes étapes de notre travail en partant par la mise du projet dans son cadre. Ensuite on s'est penché profondément sur le noyau de notre travail où on a présenté analyse et étude conceptuelle.

Dans le dernier chapitre, on a donné une vue sur les matériels et les logiciels utilisées dans notre projet et le développement de notre application.

Nous tenons à dire, au terme de ce travail, que la réalisation de ce mini projet était pour nous une occasion opportune pour développer nos savoirs et nos savoirs faire, pour acquérir de nouvelles connaissances, pour nous enrichir et pour exercer nos capacités d'observations, d'analyse, de créativité, de rédaction et de programmation. Cette expérience nous a permis ainsi de confronter l'acquis théorique à l'environnement pratique.

Tout en sachant que rien n'est définitif en matière de recherche et de progrès notamment, qu'il n'y a pas de bonnes réponses et qu'il n'y a que de bonnes questions, nous espérons que le présent rapport a été suffisamment clair, explicite et structuré pour que le lecteur ait une idée précise sur les différentes tâches auxquelles nous nous sommes livrées en réalisant notre projet sur les avantages qu'il présente et sur les nouveaux horizons qu'il ouvre.

Cependant, l'application peut être étendue par l'ajout de nouvelles fonctionnalités, à savoir, ajouter le paiement par carte bancaire après la réservation, développer et ajouter d'autres modules. Une application mobile est aussi envisageable.