**Projet : Développement du site web de La sicRoom – Salle de concert SMAC**

**Introduction et Contexte**

Le projet consiste à développer une application web pour une salle de concert indépendante spécialisée dans les musiques actuelles : **La sicRoom**. Cette plateforme doit permettre de **présenter la programmation musicale**, **vendre des billets en ligne**, **proposer un espace d’accompagnement pour les artistes**, et **rendre accessible du contenu multimédia** (vidéos, live stream, ressources pédagogiques).

Le site s’adresse à plusieurs types d’utilisateurs : le grand public, les artistes locaux, et les administrateurs de la salle.

**Bloc 1 – Développement Front-End de sites web responsives et accessibles**

**Outils et Technologies**

* **Langages** : HTML5, CSS3, JavaScript (vanilla)
* **Librairies éventuelles** : Bootstrap ou équivalent pour la responsivité
* **Accessibilité & SEO** : Utilisation des balises sémantiques, rôles ARIA, titres hiérarchisés, attributs alt, description des liens
* **Outils de test** : Google Lighthouse, Wave, Responsive Design Mode

**Principales Fonctionnalités Attendues**

* Page d’accueil avec texte de présentation, lecteur vidéo (embed), section live direct (ex : flux intégré ou espace média).
* Agenda des concerts avec navigation par trimestre et liens vers les fiches des groupes.
* Page billetterie : formulaire de réservation simple, responsive et accessible.
* Formulaire d’accompagnement des artistes : champs dynamiques, envoi des données, validation front.
* Affichage optimisé sur mobile, tablette et desktop.
* Respect des normes WCAG (accessibilité) et bonnes pratiques SEO.

**Livrables Attendus**

* Code HTML/CSS/JS structuré et commenté
* Design responsive (mobile first ou équivalent)
* Maquettes (figma ou équivalent)
* Rapport décrivant les choix UX, responsive et les actions pour l’accessibilité

**Tests et Validation**

* Vérification de la responsivité sur différents devices
* Audit d’accessibilité avec Lighthouse ou Wave
* Vérification manuelle de la navigation clavier et lecteurs d’écran

**Bloc 2 – Développement Back-End d’applications web dynamiques**

**Outils et Technologies**

* **Langages** : Node.js (sans framework), JavaScript
* **Base de données** : MySQL
* **Modélisation** : MCD, MPD, scripts SQL de création de base
* **Architecture** : MVC codé from scratch, POO, routes manuelles
* **Authentification** : email/mot de passe

**Principales Fonctionnalités Attendues**

* Système de gestion d’utilisateurs avec deux rôles : **utilisateur** (public) et **administrateur**
* Billetterie en ligne :
  + Réservation de places (création de tickets)
  + Affichage du nombre de places restantes
  + E-mail de confirmation (simulation ou console)
* Agenda concert :
  + Ajout, modification et suppression d’événements par les admins
  + Affichage public trié par date
* Formulaire artistes :
  + Enregistrement de la demande dans la base
  + Interface d’administration pour lire et traiter les demandes
* (Optionnel) Gestion de vidéos de cours :
  + Upload de vidéos par les musiciens
  + Lecture vidéo côté utilisateur

**Livrables Attendus**

* Code source structuré, commenté, sans framework
* Modèle relationnel de base de données + script SQL complet
* Diagrammes UML (cas d’utilisation, classes, séquence)
* Documentation technique : endpoints, architecture MVC, sécurité basique

**Tests et Validation**

* Tests unitaires simples sur les fonctions critiques (CRUD, login)
* Tests manuels des rôles et permissions
* Revue de code par les pairs (si travail en groupe)

**Bloc 3 – Développement d’applications web avec un framework**

**Outils et Technologies**

* **Framework** : React.js (création via Vite ou Create React App)
* **Librairies associées** : React Router, Axios, éventuellement Zustand ou Redux
* **API** : récupération des données du back-end Node.js
* **Enrichissements UX** : Tailwind CSS (optionnel)

**Principales Fonctionnalités Attendues**

* Refonte du front avec React :
  + Affichage dynamique des événements (agenda)
  + Réservation d’un billet via un formulaire connecté
  + Formulaire d’accompagnement en React
* Gestion des rôles :
  + Connexion admin/éditeur et accès au back-office React
  + Interface de gestion des concerts et formulaires artistes
* Composants modulaires et réutilisables
* Navigation SPA fluide, sans rechargement complet

**Livrables Attendus**

* Application React fonctionnelle, proprement architecturée
* Documentation technique sur les composants, appels API et gestion d’état
* Déploiement en ligne (Netlify, Vercel ou autre)

**Tests et Validation**

* Tests unitaires des composants (ex : avec Jest)
* Simulation de navigation utilisateur (playwright ou tests manuels)
* Optimisation des performances et bonne gestion des erreurs API