Voici de manière claire et détaillée les **fonctionnalités complètes** prévues pour le site web **5252** ainsi que le **script SQL** nécessaire à l'implémentation complète de la base de données associée.

**📌 Liste complète des fonctionnalités**

**🎯 Fonctionnalités Générales**

* **Consultation d'événements :**
  + Affichage des événements culturels (festivals, expositions, concerts, etc.)
  + Recherche et filtrage par catégorie, date, lieu
* **Calendrier interactif :**
  + Affichage dynamique mensuel des événements
  + Vue détaillée par jour et événement

**🧑‍💻 Fonctionnalités Utilisateurs**

**Utilisateur simple**

* Inscription, connexion, déconnexion
* Gestion de profil utilisateur (nom, email, mot de passe)
* Possibilité de commenter les événements
* Notifications internes (nouveaux événements, validations, etc.)

**Utilisateur professionnel (Entreprise)**

* Inscription nécessitant une validation administrative
  + Fourniture de documents légaux (RCCM, ID Nat)
* Gestion de profil entreprise après validation
* Soumission d’événements culturels pour validation
* Consultation des événements soumis et leur statut (validé, refusé, en attente)
* Prise de rendez-vous avec l'ONT

**Administrateur ONT**

* Validation manuelle des entreprises
* Validation manuelle des événements soumis
* Gestion complète des utilisateurs et des événements
* Communication directe avec les utilisateurs professionnels
* Gestion des notifications globales

**🌐 Multilingue**

* Français et Anglais pour tout le site (contenu et interface)

**📱 Fonctionnalités avancées Mobile/PWA**

* Responsive design (smartphones/tablettes)
* Installation possible comme PWA (Progressive Web App)
* Notifications push pour les utilisateurs connectés
* Fonctionnalité hors-ligne limitée (cache d'événements récents)

**🔐 Sécurité**

* Authentification sécurisée (hash des mots de passe)
* Validation des inputs utilisateurs
* Protection anti-CSRF, anti-XSS, anti-SQL Injection
* Gestion précise des rôles et permissions (utilisateur, entreprise, admin)
* Sécurisation de l'upload des fichiers (documents, images)

**📌 Script SQL pour la Base de Données**

Voici le **script SQL complet** à exécuter pour créer la base de données du projet 5252 :

sql

CopierModifier

CREATE DATABASE ont\_5252;

USE ont\_5252;

-- Table utilisateurs

CREATE TABLE users (

id INT PRIMARY KEY AUTO\_INCREMENT,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

role ENUM('user', 'company', 'admin') DEFAULT 'user',

is\_verified BOOLEAN DEFAULT FALSE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Table entreprises (utilisateurs pro)

CREATE TABLE companies (

id INT PRIMARY KEY AUTO\_INCREMENT,

user\_id INT NOT NULL,

company\_name VARCHAR(150) NOT NULL,

rccm VARCHAR(100) NOT NULL,

id\_nat VARCHAR(100) NOT NULL,

address TEXT NOT NULL,

status ENUM('pending', 'approved', 'rejected') DEFAULT 'pending',

justification TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE

);

-- Table événements

CREATE TABLE events (

id INT PRIMARY KEY AUTO\_INCREMENT,

company\_id INT,

title VARCHAR(150) NOT NULL,

description TEXT NOT NULL,

event\_date DATE NOT NULL,

end\_date DATE,

location VARCHAR(255),

image\_url VARCHAR(255),

status ENUM('pending', 'approved', 'rejected') DEFAULT 'pending',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (company\_id) REFERENCES companies(id) ON DELETE CASCADE

);

-- Table commentaires

CREATE TABLE comments (

id INT PRIMARY KEY AUTO\_INCREMENT,

event\_id INT NOT NULL,

user\_id INT NOT NULL,

content TEXT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (event\_id) REFERENCES events(id) ON DELETE CASCADE,

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE

);

-- Table notifications internes

CREATE TABLE notifications (

id INT PRIMARY KEY AUTO\_INCREMENT,

user\_id INT NOT NULL,

message TEXT NOT NULL,

is\_read BOOLEAN DEFAULT FALSE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(id) ON DELETE CASCADE

);

-- Table rendez-vous entre ONT et entreprises

CREATE TABLE meetings (

id INT PRIMARY KEY AUTO\_INCREMENT,

company\_id INT NOT NULL,

admin\_id INT NOT NULL,

proposed\_date DATETIME NOT NULL,

status ENUM('pending', 'confirmed', 'declined') DEFAULT 'pending',

notes TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (company\_id) REFERENCES companies(id) ON DELETE CASCADE,

FOREIGN KEY (admin\_id) REFERENCES users(id) ON DELETE CASCADE

);

-- Table fichiers (documents RCCM et images des événements)

CREATE TABLE files (

id INT PRIMARY KEY AUTO\_INCREMENT,

company\_id INT,

event\_id INT,

file\_type ENUM('rccm', 'id\_nat', 'event\_image') NOT NULL,

file\_url VARCHAR(255) NOT NULL,

uploaded\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (company\_id) REFERENCES companies(id) ON DELETE CASCADE,

FOREIGN KEY (event\_id) REFERENCES events(id) ON DELETE CASCADE

);

-- Index optimisés

CREATE INDEX idx\_user\_role ON users(role);

CREATE INDEX idx\_event\_status ON events(status);

CREATE INDEX idx\_company\_status ON companies(status);

CREATE INDEX idx\_notification\_read ON notifications(is\_read);