

# Projet : Kouak

## Livrables

- MCD : Un diagramme représentant le modèle conceptuel de données.
- Code Source : Le code source complet de l'application, incluant le backend et le frontend. (Github)

## Objectif

Le but de ce projet est de créer une application de chat en temps réel permettant à un groupe de personnes de communiquer.

L'application doit inclure des fonctionnalités d'inscription, de connexion, d'envoi de messages et de suivi de la conversation.

Vous devez également mettre en place des mesures de sécurité pour protéger les communications.

## Technologies Utilisées

- **Frontend** : React
- **Backend** : Node.js avec Express
- **Base de Données** : Sequelize ou Prisma
- **WebSocket** : Socket.io
- **Sécurité** : JWT (JSON Web Token)

## Fonctionnalités Requises

### 1. Inscription et Connexion

- Les utilisateurs doivent pouvoir s'inscrire avec un nom d'utilisateur, un email et un mot de passe.
- Les utilisateurs doivent pouvoir se connecter avec leur email et mot de passe.

### 2. Chat en Temps Réel

- Les utilisateurs doivent pouvoir envoyer des messages en temps réel.
- Les messages doivent être affichés dans l'ordre chronologique.
- Les utilisateurs doivent pouvoir voir les messages envoyés par d'autres utilisateurs en temps réel.

### 3. Interface Utilisateur

- L'application doit avoir une interface utilisateur intuitive et esthétique.
- Les utilisateurs doivent pouvoir voir une liste des utilisateurs connectés.
- Les messages doivent être affichés avec le nom de l'utilisateur qui les a envoyés.

#### **4. Sécurité**

- Mettre en place un middleware pour sécuriser les routes avec JWT.
- Les utilisateurs doivent être authentifiés pour accéder aux fonctionnalités de chat.

## **Étapes du Projet**

### **1. Conception de la Base de Données**

- Créer un MCD (Modèle Conceptuel de Données) pour représenter les entités et les relations nécessaires (utilisateurs, messages, etc.).
- Mettre en place la base de données avec Sequelize ou Prisma.

### **2. Backend**

- Configurer un serveur Node.js avec Express.
- Créer les routes nécessaires pour l'inscription, la connexion, l'envoi de messages et la récupération des messages.
- Mettre en place Socket.io pour la communication en temps réel.
- Implémenter un middleware pour sécuriser les routes avec JWT.

### **3. Frontend**

- Créer une application React pour l'interface utilisateur.
- Implémenter les composants nécessaires pour l'inscription, la connexion, l'envoi de messages et l'affichage des messages.
- Utiliser Socket.io pour la communication en temps réel.

### **4. Sécurité**

- Implémenter l'authentification JWT pour sécuriser les routes.
- Assurer que seuls les utilisateurs authentifiés peuvent accéder aux fonctionnalités de chat.