

# TP COMPLET : WHATSAPP CLONE EN VUE.JS - MEVN

## OBJECTIF GLOBAL

Créer une application de messagerie instantanée complète en stack MEVN (MongoDB, Express, Vue.js, Node.js) avec authentification, conversations en temps réel, gestion des contacts, partage de médias et monitoring avec Sentry.

---

## SECTION 1 : ARCHITECTURE ET PLANNING

### Étape 1.1 : Étudier l'architecture d'une application de messagerie

Vous devez comprendre :

- L'architecture client-serveur pour la messagerie
- Le rôle des WebSockets pour la communication en temps réel
- Les concepts de rooms et de sessions
- Comment gérer les connexions utilisateur
- Les défis de la synchronisation des données
- Les stratégies de cache local
- Les patterns de reconnexion
- Les considérations de scalabilité

### Étape 1.2 : Planifier la structure technique

Documentez :

- Les technologies choisies (Node.js, Vue.js, MongoDB, Socket.io)
- L'architecture des dossiers
- La séparation des responsabilités
- Les couches de l'application
- Les patterns de communication
- Les flux de données
- Les points de défaillance possibles
- Les stratégies de récupération

### Étape 1.3 : Définir les user stories

Créez une liste complète des fonctionnalités :

- Inscription et connexion
- Gestion des contacts
- Création de conversations

- Envoi et réception de messages
- Édition de messages
- État des messages (envoyé, livré, lu)
- Partage de médias
- Groupes de discussion
- Recherche de conversations
- Statuts utilisateur

#### Étape 1.4 : Planifier les données

Documentez les modèles de données :

- Utilisateurs
- Conversations
- Messages
- Contacts
- Groupes
- Médias
- Interactions utilisateur

#### Étape 1.5 : Planifier les tests

Listez les scénarios de test :

- Authentification
- Envoi de messages
- Réception en temps réel
- Gestion des connexions perdues
- Partage de fichiers
- Création de groupes

---

## SECTION 2 : BACKEND - AUTHENTIFICATION ET UTILISATEURS

### Étape 2.1 : Initialiser le projet backend

Vous devez :

- Créer la structure des dossiers (models, routes, controllers, middleware)
- Initialiser npm avec les dépendances
- Configurer les variables d'environnement
- Mettre en place le logging
- Créer les fichiers de configuration
- Initialiser la connexion à MongoDB
- Configurer Sentry pour le monitoring

## Étape 2.2 : Implémenter l'authentification

Créez les fonctionnalités suivantes :

- Enregistrement des utilisateurs
- Hachage des mots de passe
- Vérification des emails uniques
- Génération de tokens JWT
- Validation des tokens
- Middleware d'authentification

## Étape 2.3 : Implémenter la gestion des utilisateurs

Vous devez gérer :

- Profil utilisateur (nom, avatar, statut)
- Statut en ligne/hors ligne
- Dernière vue
- Informations de connexion
- Mise à jour du profil
- Suppression de compte
- Photo de profil avec upload

## Étape 2.4 : Implémenter les contacts

Créez les fonctionnalités :

- Ajout de contacts
- Suppression de contacts
- Liste des contacts
- Recherche de contacts
- Blocage de contacts
- Déblocage de contacts

## Étape 2.5 : Implémenter la gestion des sessions

Vous devez mettre en place :

- Enregistrement des sessions actives (IP)
  - Historique des connexions
  -
-

## SECTION 3 : BACKEND - MESSAGES ET CONVERSATIONS

### Étape 3.1 : Implémenter le modèle de conversation

Vous devez créer :

- Conversations one-to-one
- Groupes de conversation
- Informations de conversation (créateur, membres, titre)
- Paramètres de notification par conversation
- Archivage de conversations
- Suppression de conversations
- Recherche de conversations
- Tri et filtrage
- Aperçu du dernier message
- Compteur de messages non lus

### Étape 3.2 : Implémenter le modèle de message

Vous devez créer :

- Stockage des messages
- Types de messages (texte, image, vidéo, fichier)
- Timestamps
- Références des utilisateurs
- Conversations liées
- Statut des messages (envoyé, livré, lu)
- Édition des messages
- Suppression des messages

### Étape 3.3 : Implémenter les états des messages

Vous devez gérer :

- Message en attente
- Message envoyé
- Message livré
- Message lu
- Timestamp pour chaque état
- Notifications des changements d'état
- Synchronisation des états

### Étape 3.4 : Implémenter le partage de médias

Créez les fonctionnalités :

- Upload de fichiers
- Validation de types de fichiers
- Limitation de taille
- Miniatures pour images/vidéos
- Téléchargement de fichiers
- Suppression de fichiers
- Compression d'images

### Étape 3.5 : Implémenter la recherche de messages

Vous devez créer :

- Recherche simple par texte
- Recherche par expéditeur
- Pagination des résultats

### Étape 3.6 : Implémenter les messages spéciaux

Vous devez gérer :

- Messages réactifs (réactions avec emojis)
- Réponses à des messages (citations)
- Messages de notification système

### Étape 3.7 : Implémenter les groupes

Créez les fonctionnalités :

- Création de groupes
- Ajout/suppression de membres
- Rôles dans les groupes (admin, modérateur, membre)
- Permissions selon les rôles
- Description du groupe
- Photo de groupe

---

## SECTION 4 : BACKEND - WEBSOCKETS ET TEMPS RÉEL

### Étape 4.1 : Configurer Socket.io

Vous devez :

- Initialiser Socket.io
- Configurer CORS
- Implémenter l'authentification Socket

### Étape 4.2 : Implémenter les événements de messages

Vous devez créer les événements :

- Envoi de message
- Réception de message
- Confirmation d'envoi
- Notification de livraison
- Notification de lecture
- Suppression de message
- Édition de message
- Réaction au message

### Étape 4.3 : Implémenter la gestion de la présence

Créez les fonctionnalités :

- Statut en ligne/hors ligne
- Timestamp de présence
- Dernière vue pour chaque conversation
- Indicateur d'écriture
- Indicateur de saisie

### Étape 4.4 : Implémenter les notifications en temps réel

Vous devez gérer :

- Notifications de nouveau message
- Notifications d'ajout à un groupe
- Notifications de changement de groupe
- Notifications de suppression de message

#### Étape 4.5 : Gérer la reconnexion et les déconnexions

Vous devez implémenter :

- Détection automatique des déconnexions
  - Récupération des messages manqués
  - Notification de déconnexion aux autres utilisateurs
- 

### SECTION 5 : BACKEND - MONITORING ET ALERTES

#### Étape 5.1 : Implémenter l'intégration Sentry

Vous devez :

- Initialiser Sentry
- Capturer les erreurs
- Ajouter les breadcrumbs
- Implémenter le context utilisateur
- Tracer les transactions
- Profiler les performances
- Configurer les alertes
- Intégrer Slack

#### Étape 5.2 : Implémenter les alertes de sécurité

Créez les alertes pour :

- Nouvelles connexions
- Modifications de profil
- Modifications de paramètres
- Nouveau contact ajouté
- Blocages de contact

#### Étape 5.3 : Implémenter les logs applicatifs

Vous devez mettre en place :

- Logs des connexions
  - Logs des actions utilisateur
  - Logs des erreurs
  - Logs des WebSockets
  - Rotation des logs
  - Centralisation des logs
  - Alertes sur logs critiques
-

## SECTION 6 : FRONTEND - AUTHENTIFICATION ET SETUP

### Étape 6.1 : Initialiser le projet Vue.js

Vous devez :

- Créer le projet Vue.js avec Vite ou CLI
- Installer les dépendances
- Configurer les variables d'environnement
- Mettre en place la structure des fichiers
- Initialiser Sentry
- Configurer le logging

### Étape 6.2 : Implémenter les pages d'authentification

Créez les pages suivantes :

- Page de connexion avec email/mot de passe
- Page d'enregistrement
- Page de récupération de mot de passe
- Page de choix de pseudo
- Page de photo de profil
- Gestion des erreurs d'authentification
- Mémorisation de la connexion

### Étape 6.3 : Implémenter la gestion de session

Vous devez gérer :

- Stockage du token
- Déconnexion

### Étape 6.4 : Implémenter le setup initial

Créez les fonctionnalités :

- Upload de photo de profil
- Accès à la caméra
- Accès au stockage
- Configuration des paramètres par défaut



---

## SECTION 7 : FRONTEND - INTERFACE PRINCIPALE

### Étape 7.1 : Implémenter la sidebar

Vous devez créer :

- Liste des conversations
- Recherche de conversations
- Filtrage des conversations
- Statut des conversations
- Notifications non lues
- Menu d'options
- Archivage de conversation
- Suppression de conversation

### Étape 7.2 : Implémenter la zone de chat

Créez les fonctionnalités :

- Affichage des messages
- Scroll automatique vers les nouveaux messages
- Horodatage des messages
- Lecture des fichiers
- Téléchargement des fichiers

### Étape 7.3 : Implémenter la zone de saisie

Vous devez gérer :

- Champ de saisie simple
- Sélecteur d'emojis
- Indicateur d'écriture
- Émojis et reactions
- Envoi par bouton ou raccourci clavier

### Étape 7.4 : Implémenter le partage de médias

Créez les fonctionnalités :

- Sélection de fichiers
- Prévisualisation avant envoi
- Upload avec progression
- Annulation d'upload
- Drag and drop

### Étape 7.5 : Implémenter les actions sur les messages

Vous devez permettre :

- Suppression de message
- Édition de message
- Copie de message
- Réaction au message
- Réponse au message (citation)
- Téléchargement du fichier

### Étape 7.6 : Implémenter les paramètres de conversation

Créez les options :

- Information du contact/groupe
  - Recherche dans la conversation
  - Médias dans la conversation
  - Fichiers dans la conversation
  - Blocage du contact
  - Suppression
- 

## SECTION 8 : FRONTEND - GESTION DES CONTACTS ET GROUPES

### Étape 8.1 : Implémenter la gestion des contacts

Vous devez créer :

- Page de contacts
- Ajout de contact
- Suppression de contact
- Recherche de contact
- Tri des contacts
- Filtrage des contacts
- Blocage de contact
- Édition de contact
- Photo du contact

### Étape 8.2 : Implémenter la création de groupes

Créez les fonctionnalités :

- Page de création de groupe
- Sélection des membres
- Définition du titre
- Upload de photo de groupe

- Description du groupe
- Restrictions de groupe
- Validation avant création
- Messages de confirmation

### Étape 8.3 : Implémenter la gestion des groupes

Vous devez gérer :

- Page d'information du groupe
- Ajout/suppression de membres
- Promotion/demotion d'admin
- Modification du titre du groupe
- Modification de la description
- Modification de la photo
- Paramètres de groupe
- Quitter le groupe
- Supprimer le groupe

### Étape 8.4 : Implémenter les rôles dans les groupes

Créez les permissions pour :

- Créateur (tous les droits)
  - Admin (gestion des membres, modification d'info)
  - Modérateur (suppression de messages)
  - Membre (envoi de messages)
  - Interface adaptée selon le rôle
- 

## SECTION 9 : FRONTEND - PROFIL ET PARAMÈTRES

### Étape 9.1 : Implémenter la page de profil

Vous devez créer :

- Affichage du profil utilisateur
- Modification du pseudo
- Modification de la photo
- Modification du statut
- Statut en ligne/hors ligne
- Historique des connexions
- Appareils connectés
- Informations du compte

### Étape 9.2 : Implémenter les paramètres

Créez les sections suivantes :

- Paramètres de compte
- Paramètres de notification
- Paramètres d'apparence (Dark Mode)
- Paramètres de langue

### Étape 9.3 : Implémenter les paramètres de confidentialité

Vous devez gérer :

- Blocage de contacts

### Étape 9.4 : Implémenter les paramètres de sécurité

Créez les fonctionnalités :

- Changement de mot de passe
- Historique des connexions

---

## SECTION 10 : SÉCURITÉ DE BASE

### Étape 10.1 : Implémenter la sécurité de communication

Créez les fonctionnalités :

- HTTPS obligatoire
- WSS (WebSocket Secure) obligatoire
- CORS restrictif

### Étape 10.2 : Implémenter la gestion des permissions

Vous devez gérer :

- Permissions des contacts
- Permissions des groupes
- Permissions des fichiers

### Étape 10.3 : Implémenter le contrôle d'accès

Créez les fonctionnalités :

- Authentification requise
  - Rôles et permissions
  - Élévation de privilèges
  - Revocation d'accès
- 

## SECTION 11 : PERFORMANCE ET OPTIMISATION

### Étape 11.1 : Optimiser le backend

Vous devez :

- Implémenter la pagination
- Implémenter le cache (Redis)
- Optimiser les temps de réponse
- Réduire la charge serveur

### Étape 11.2 : Optimiser le frontend

Créez les optimisations :

- Lazy loading des composants Vue
- Optimisation des images
- Réduction du bundle size
- Optimisation des composants

### Étape 11.3 : Implémenter le mode hors ligne

Vous devez gérer :

- Stockage local des messages
  - Synchronisation lors du reconnexion
- 

## SECTION 12 : INTÉGRATIONS SIMPLES

### Étape 12.2 : Intégrer Sentry complet

Vous devez :

- Capturer toutes les erreurs
- Ajouter les breadcrumbs pertinents
- Implémenter les transactions

- Profiler les performances
  - Configurer les alertes
  - Intégrer Slack pour les alertes
  - Créer les dashboards
  - Mettre en place les rapports
- 

## SECTION 13 : TESTS

### Étape 13.1 : Implémenter les tests backend

Vous devez créer :

- Tests unitaires (Jest/Mocha)
- Tests d'intégration
- Tests des API REST
- Tests WebSocket
- Tests de base de données
- Couverture minimale 70%

### Étape 13.2 : Implémenter les tests frontend

Créez les tests :

- Tests de composants Vue (Vitest)
- Tests de performance
- Couverture minimale 70%

### Étape 13.3 : Implémenter les tests end-to-end

Vous devez tester :

- Flux complet d'enregistrement
- Envoi et réception de messages
- Partage de fichiers
- Création de groupe

### Étape 13.4 : Tester les aspects de sécurité basique

Créez les tests :

- Injection basique
  - XSS
  - Authentification
-

## SECTION 14 : DEPLOYMENT ET DEVOPS

### Étape 14.1 : Préparer le déploiement

Vous devez :

- Créer les Dockerfiles
- Configurer Docker Compose
- Mettre en place le CI/CD simple (ESLINT Back & Front et test ok )
- Configurer les environnements
- Écrire les scripts de déploiement
- Créer les configurations de production
- Préparer les bases de données

### Étape 14.1bis : Mettre en place le CI/CD simple

Vous devez :

- Configurer ESLint pour le backend (Node.js)
- Configurer ESLint pour le frontend (Vue.js)
- Exécuter les tests automatiques
- Vérifier que les tests passent (couverture 70%+)
- Mettre en place les pipelines GitHub Actions
- Lancer les checks avant déploiement
- Générer les rapports de couverture

### Étape 14.2 : Déployer sur le serveur

Créez le processus :

- Installation des dépendances
- Configuration des variables
- Migration des bases de données
- Déploiement du backend
- Déploiement du frontend

### Étape 14.3 : Configurer le monitoring

Vous devez mettre en place :

- Logs centralisés
- Dashboards simples
- Health checks

---

## SECTION 15 : DOCUMENTATION

### Étape 15.1 : Créer la documentation utilisateur

Vous devez documenter :

- Guide d'installation
- Guide d'utilisation
- FAQ

### Étape 15.2 : Créer la documentation développeur

Créez la doc :

- Architecture du système
- Guide de setup développement

### Étape 15.3 : Créer la documentation API

Documentez :

- Endpoints REST
- Événements WebSocket
- Formats de réponse
- Codes d'erreur
- Exemples de requêtes

---

## LIVRABLES ATTENDUS

À la fin de ce TP, vous devez avoir :

### Backend

- API REST complète fonctionnelle
- WebSockets en temps réel configurés
- Base de données avec tous les modèles
- Authentification JWT
- Monitoring Sentry complet
- Tests avec couverture 70%+

### Frontend

- Application Vue.js responsive
- Interface complète avec toutes les fonctionnalités



- Gestion des états avec Pinia
- Offline mode de base
- Notifications en temps réel
- Performance optimisée
- Tests avec couverture 70%+
- Accessible aux utilisateurs

#### Infrastructure

- Docker et Docker Compose
- CI/CD pipeline simple
- Scripts de déploiement
- Monitoring et alertes
- Documentation complète

#### Documentation

- Architecture documentée
- API documentée
- Guides utilisateur
- Guides développeur

---

### CRITÈRES D'ÉVALUATION - NOTATION SUR 20

1. **Fonctionnalité** (2 points) : Toutes les features implémentées et fonctionnelles
2. **Performance** (2 points) : Temps de réponse acceptable
3. **Sécurité** (2 points) : Authentification, autorisation, validation correctes
4. **Qualité du code** (2 points) : Lisible, maintenable, respecte les patterns Vue.js
5. **Tests** (2 points) : Couverture >70%, tests significatifs
6. **Documentation** (2 points) : Complète et à jour
7. **UX** (2 points) : Interface intuitive et fluide
8. **Monitoring** (1 point) : Erreurs capturées et tracées
9. **Scalabilité** (1 point) : Peut gérer plusieurs utilisateurs
10. **Production-ready** (2 points) : Peut être déployé en production

**Total : 20 points**

---

### RESSOURCES RECOMMANDÉES

- Socket.io documentation
- Vue.js 3 documentation
- Pinia documentation
- Node.js best practices
- MongoDB documentation

- [WebSocket security](#)
- [JWT best practices](#)
- [Testing frameworks Vue](#)
- [Docker documentation](#)