Documentation Technique - Time's Hub

Table des matières

1. Présentation Générale

Time’s Hub est une application mobile de jeu d’équipe inspirée de Time’s Up. Elle permet à plusieurs utilisateurs de participer à une partie composée de différentes manches pour faire deviner des mots. L'application fonctionne aussi bien en ligne qu'en mode invité hors-ligne.

2. Architecture de l'application

2.1. Frontend (Client)

- **Technologie principale** : React Native via Expo
- **Navigation** : React Navigation
- **Rendu conditionnel** : hooks (useState, useEffect) + composants personnalisés
- **Communication avec le backend** : Axios

2.2. Backend (API Node.js)

- **Framework** : Express.js
- **Base de données** : MongoDB Atlas (via Mongoose)
- **Endpoints principaux** :
    - POST /api/auth/register
    - POST /api/auth/login
    - GET /api/history (protégé par JWT)
    - POST /api/words/generate (utilise API d’IA)

2.3. Intelligence Artificielle

- **Service** : Gemini (API REST)
- **Fonction** : Générer dynamiquement 40 mots à partir d’un thème utilisateur ou via une liste par défaut

3. Fonctionnalités Clés

3.1. Authentification JWT

- Jeton généré à la connexion
- Stocké localement et ajouté à chaque requête API

3.2. Génération de mots

- Utilisation d’un appel API vers OpenAI/Gemini avec prompt personnalisé
- Réponse parsée côté backend avant envoi au client

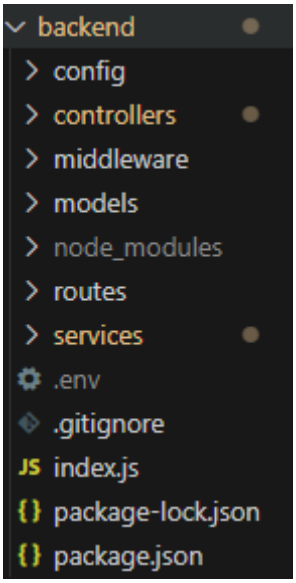
3.3. Gestion de parties

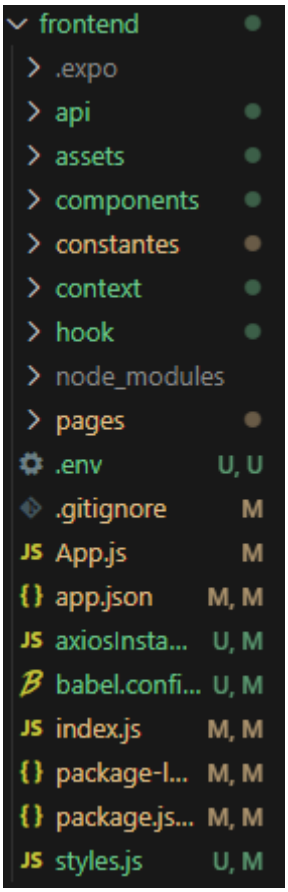
- 3 manches successives : Libre, 1 mot, Mime
- Chronomètre automatique de 30 secondes
- Passage automatique entre équipes
- Modal de changement d'équipe ou fin de manche

3.4. Historique utilisateur

- Sauvegarde côté backend si utilisateur connecté
- Enregistrement : thème, gagnant, date, nombre d'équipes
- Accessible via l'écran profil

4. Structure du Projet





5. Sécurité

- Données utilisateurs chiffrés via JWT
- Validation côté serveur des entrées
- Politique CORS appliquée

6. Déploiement & Test

- **Développement local** : Expo Go ou Android/iOS Simulator
- **Production** : génération d'un build via Expo Application Services (EAS)
- **Base de données** : MongoDB Atlas (cloud)

7. Dépendances Clés

"react-native": "0.74.x",

"expo": "^53.x.x",

"axios": "^1.x",

"@react-navigation/native": "^7.x",

"mongoose": "^7.x",

"express": "^4.x",

"jsonwebtoken": "^9.x"

8. Limitations Actuelles

- Pas de mode multijoueur temps réel (hors partage local)
- Génération IA dépendante de l'API tierce

9. Évolutions Possibles

- Ajout d'un mode multijoueur en ligne
- Traduction multilingue de l'interface
- Mode jour/nuit
- Statistiques avancées par joueur