

Roomies : L'Expérience Multijoueur en ligne



Objectifs et Problématique

Le projet Roomies est né du besoin de créer une plateforme de jeux en ligne complète et engageante, répondant à plusieurs défis clés du développement web moderne.

Simplifier l'Interaction

Fournir une plateforme intuitive pour les jeux multijoueurs avec communication instantanée.

Sécurité Robuste

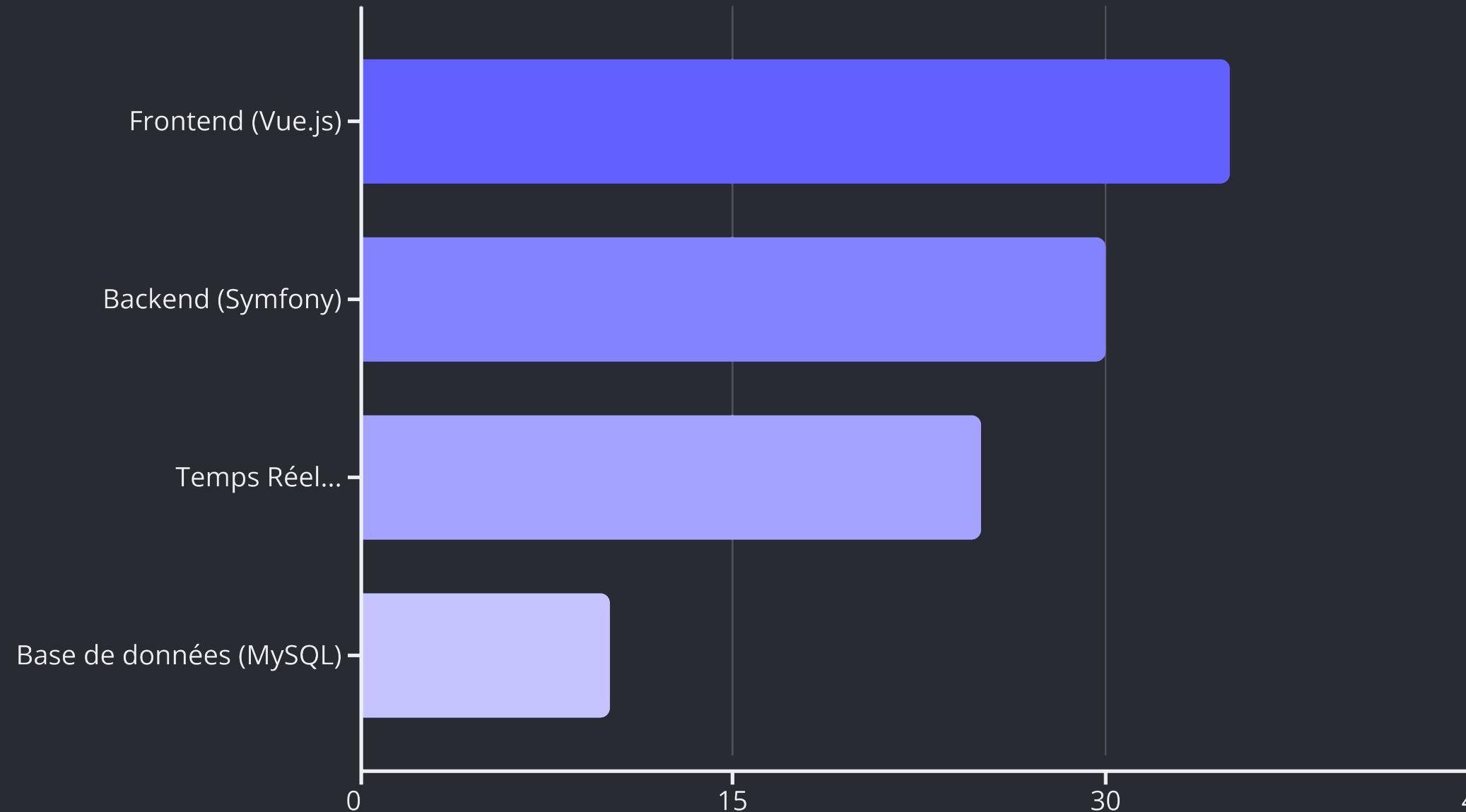
Garantir une authentification fiable et une protection des données utilisateur.

Expérience Temps Réel

Implémenter le chat et la synchronisation des jeux sans latence visible.

Aperçu Technique Global

Roomies s'appuie sur une architecture microservices conteneurisée avec Docker, assurant scalabilité et maintenance facilitée.



Le Frontend : Vue.js 3

Le frontend de Roomies repose sur **Vue.js 3**, avec **Vite** pour le bundling, **Tailwind CSS** pour un design moderne et responsive, et **PrimeVue** pour des composants UI performants et accessibles.

L'application gère :

- Une **authentification sécurisée par JWT** avec conservation des tokens côté client (LocalStorage → Pinia).
- Un **système de navigation protégé par des guards** (authGuard) empêchant l'accès aux routes non autorisées.
- Une **interface intuitive** pour créer, rejoindre et gérer des rooms multijoueur.
- Un **chat global et par room** en temps réel via WebSockets.
- Une **gestion des profils** avec édition, suppression de compte.
- Un **store centralisé** avec Pinia, permettant la synchronisation des données utilisateurs et des jeux.

Le code est organisé par pages, composants et hooks personnalisés, facilitant la **scalabilité** et la **maintenabilité**. L'ensemble est entièrement **responsive** et testé avec **Vitest** et **Cypress** (end-to-end).

The screenshot displays the Roomies application interface. At the top, there is a dark header bar with several navigation items: 'Accueil' (Home), 'Rooms 4' (with a dropdown arrow), 'Administration', 'Contact', 'Recherche' (Search), and a user icon. Below the header, the main content area features a large, stylized logo for 'ROOMIES GAMEROOMS' featuring a glowing blue video game controller icon. To the right of the logo, the tagline 'Play. Chat. Connect.' is displayed in large, bold, blue letters. Below this, a call-to-action button labeled 'Commencer' (Start) is visible. The main body of the page is divided into three main sections: 1) A user profile section for 'Salut admin,' showing a placeholder profile picture and the message 'Bienvenue dans ton espace rétro ✨'. 2) An 'Mes Informations' (My Information) section containing fields for 'Nom d'utilisateur' (User Name) set to 'admin' and 'Email' set to 'admin@admin.com', with a 'Modifier' (Edit) button below. 3) A 'Mes Jeux Favoris' (My Favorite Games) section which currently displays the message 'Tu n'as pas encore de jeux favoris.' (You don't have any favorite games yet.) and a button labeled 'Ajouter un jeu' (Add a game) with a plus sign icon. The overall design is modern and minimalist, utilizing a dark theme with blue and white text.

Le Backend : Symfony 7

Le **backend** de Roomies est construit sur **Symfony 7**, offrant une architecture RESTful claire et sécurisée. Il intègre un système d'authentification par **JWT** et utilise **Doctrine ORM** pour la gestion de la base de données **MySQL**.

Fonctionnalités clés du backend :

- **API REST** exposée pour les entités fondamentales : **User**, **Room**, **Game**, **Friendship**, etc.
- **Sécurisation robuste des endpoints** via authentification et autorisations basées sur les rôles.
- **WebSockets distincts** du service HTTP principal, assurant une modularité et une évolutivité accrues.
- Gestion centralisée des erreurs et des réponses JSON standardisées pour une communication cohérente.

Les contrôleurs sont rigoureusement testés avec **PHPUnit**, et les migrations de base de données sont versionnées via **Doctrine Migrations**.

L'API respecte les **bonnes pratiques REST**, garantissant une architecture claire, modulaire et bien documentée.

Attribut	Description
id	Identifiant unique de l'utilisateur
username	Nom d'utilisateur unique
email	Adresse email de l'utilisateur
password	Mot de passe haché
roles	Rôles attribués à l'utilisateur
created_at	Date de création du compte
last_active	Date de la dernière activité
avatar	Lien vers l'avatar de l'utilisateur

Détail de l'entité User

The screenshot shows the Symfony 7 Administration interface for the User entity. At the top, there are four summary cards: 'Total Utilisateurs' (2), 'Total Rooms' (3), 'Utilisateurs Actifs' (1), and 'Rooms Actives' (0). Below this is a navigation bar with links for 'Utilisateurs' (2) and 'Rooms' (3). The main area displays a table of users with columns: ID, Utilisateur, Rôles, Inscription, Dernière activité, and Actions. Two users are listed: 'test' (User, created 19 juil. 2025 23:52, last active 8 min ago) and 'admin' (User Admin, created 19 juil. 2025 23:35, last active now). The 'Actions' column for each user includes icons for view, edit, and delete.

Communication en Temps Réel : Node.js + Socket.IO

Le serveur Node.js avec Socket.IO gère toutes les interactions en temps réel, du chat aux jeux.



Chat Global et par Room

Échange instantané de messages entre tous les utilisateurs ou au sein d'une salle spécifique.



Synchronisation des Jeux

Mouvements de jeu synchronisés pour Morpion, 2048 et Échecs, garantissant une expérience fluide.



Architecture WebSocket

Connexions persistantes pour une faible latence et une mise à jour instantanée des états.

L'état des jeux est maintenu dans des objets Map, permettant une gestion efficace et rapide des interactions.

Fonctionnalités Clés Implémentées

Roomies offre un ensemble complet de fonctionnalités pour une expérience utilisateur riche et sécurisée.

Fonctionnalité	Description	Statut
Authentification	Inscription / Connexion sécurisée par JWT	Implémenté
Chat Global	Messagerie instantanée pour tous les utilisateurs	Implémenté
Gestion des Rooms	Création, jonction et gestion de salles de jeu	Implémenté
Profils Utilisateur	Affichage et modification des informations de profil	Implémenté
Ajout aux Favoris	Marquer des utilisateurs comme favoris	Implémenté
Sécurité JWT	Protection des API et des données	Implémenté
Utilisateur	Édition profil, suppression de compte	Implémenté
Chat par Room	Chat par room (WebSocket)	Implémenté
Jeux intégrés	Morpion, 2048, Échecs, Quizz	Implémenté (partiellement)
WebSocket	Gestion du temps réel (rooms, jeux, chat)	Implémenté
Tests	Tests unitaires (PHPUnit), E2E (Cypress), composants	Implémenté
Docker	Conteneurisation complète (frontend, backend, DB)	Implémenté
Fonctionnalités futures	Classement, WebRTC, historique parties	Doit être Planifié
Admin / Logs	Console backend, logs d'erreurs, debug	Doit être Planifié

Conclusion : Compétences et Vision

Le projet Roomies est une démonstration concrète de compétences fullstack, d'autonomie et d'une vision produit claire.



Vision Produit

Capacité à concevoir une application complète et centrée sur l'utilisateur.



Autonomie Technique

Maîtrise de l'ensemble du cycle de développement, de la conception au déploiement.



Compétences Fullstack

Expertise en frontend (Vue.js), backend (Symfony) et temps réel (Node.js/Socket.IO).

Roomies est bien plus qu'un simple projet fullstack : c'est un **écosystème complet**, combinant **authentification, gestion de profils, chat en temps réel et jeux synchronisés** avec une stack moderne et conteneurisée.

Ce projet met en avant :

- Une **maîtrise du développement web** (frontend + backend)
- Une **intégration temps réel avancée** via WebSocket
- Une **architecture Docker scalable** avec séparation des responsabilités
- Une **culture de tests (Vitest, Cypress, PHPUnit)** et de bonnes pratiques de code
- Une **autonomie technique** dans la conception et la réalisation d'un projet complexe

Roomies reflète une forte capacité à livrer un produit technique complet, collaboratif, maintenable, et prêt à évoluer.