

# Présentation du projet

## 1 – Présentation

Notre projet LarbraChat est un chat web en temps réel, pour faciliter l'inscription et la connexion à notre application, nous avons mis en place la possibilité de s'inscrire et de se connecter avec Google et Twitter. Le système de temps réel, lui, fonctionne grâce à des web sockets.

Un système de groupe est mis en place pour pouvoir discuter avec plusieurs personnes en même temps et un système de conversion basique est, lui aussi, implémenté.

## 2 – Technologies utilisées

Pour mettre en place et faire fonctionner ce projet, nous avons choisi plusieurs technologies qui vont permettre de se compléter et donc d'avoir un back-end et un front-end complètement indépendant et fonctionnel.

Concernant la technologie back-end, nous avons choisi Laravel qui est un framework de PHP très utilisé dans le web. Dans Laravel, il était très facile d'implémenter des web sockets et de créer une API pour pouvoir envoyer des informations vers le front-end, ce qui nous a confortés dans notre choix. Nous avons logiquement utilisé Eloquent l'ORM de Laravel qui nous a simplifié la vie pour la gestion des données avec la Base de données.

Concernant la technologie front-end, nous avons décidé d'utiliser Vue.js car c'était le framework front le plus adapté pour une utilisation avec Laravel. Il est très simple d'utilisation et permet de faire des appels API très simplement avec Axios.

On a ensuite utilisé Echo pour communiquer avec les websockets et donc interagir en temps réel sans rechargement de page.

### 3 – Rôles de chacun

Durant ce projet, nous n'avons pas défini nos rôles au préalable, chacun était développeur et les décisions étaient prises ensemble. Chaque fonctionnalité a été validée en amont et une fois le développement fini, nous faisons un point ensemble pour ajuster si nécessaire et vérifier si cela correspondait bien aux attentes.

### 4 – Structure algorithmique

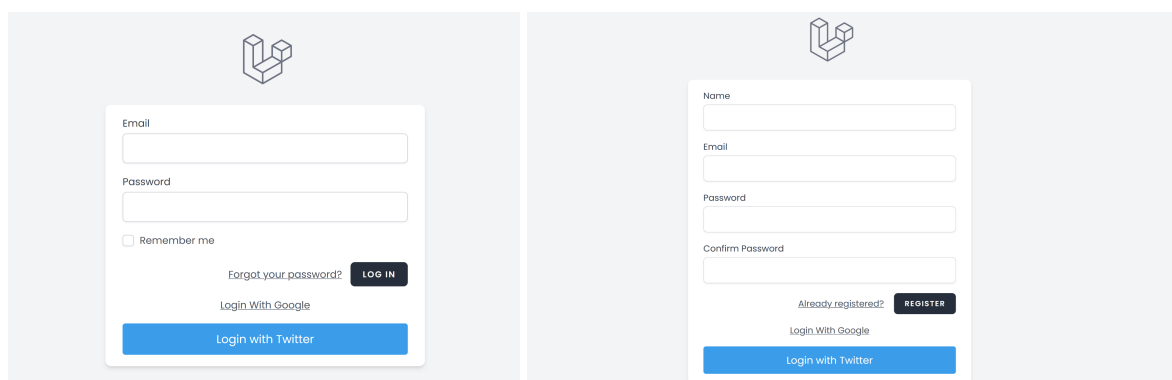
Pour notre structure algorithmique, nous avons choisi de mettre en place une architecture en MVC avec une API qui envoie les données au front-end.

### 5 – Liste et explications des fonctionnalités principales

Voici la liste des fonctionnalités principales et leurs explications :

- Système d'Auth

La première fonctionnalité principale était un système d'authentification simple pour pouvoir se connecter et avoir ses conversations sauvegardées. Nous avons choisi d'utiliser Breeze, une librairie de Laravel qui permet de mettre en place très rapidement l'authentification.



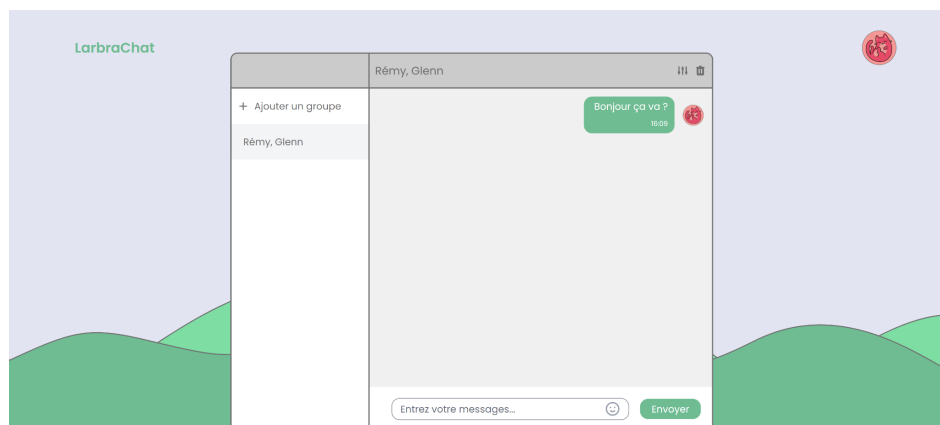
The image displays two side-by-side screenshots of a web application's authentication interface. Both screens feature a logo at the top center, consisting of three stacked cubes. The left screen is the login page, containing input fields for 'Email' and 'Password', a 'Remember me' checkbox, a 'Forgot your password?' link, a 'LOG IN' button, and social login options for 'Login With Google' and 'Login with Twitter'. The right screen is the registration page, containing input fields for 'Name', 'Email', 'Password', and 'Confirm Password', an 'Already registered?' link, a 'REGISTER' button, and social login options for 'Login With Google' and 'Login with Twitter'.

- Connexion / Inscription par Google, Twitter

Nous avons ensuite décidé de rajouter la connexion et l'inscription par Google et Twitter pour se connecter rapidement à notre application. Grâce à Breeze une implémentation de leur API était plutôt rapide.

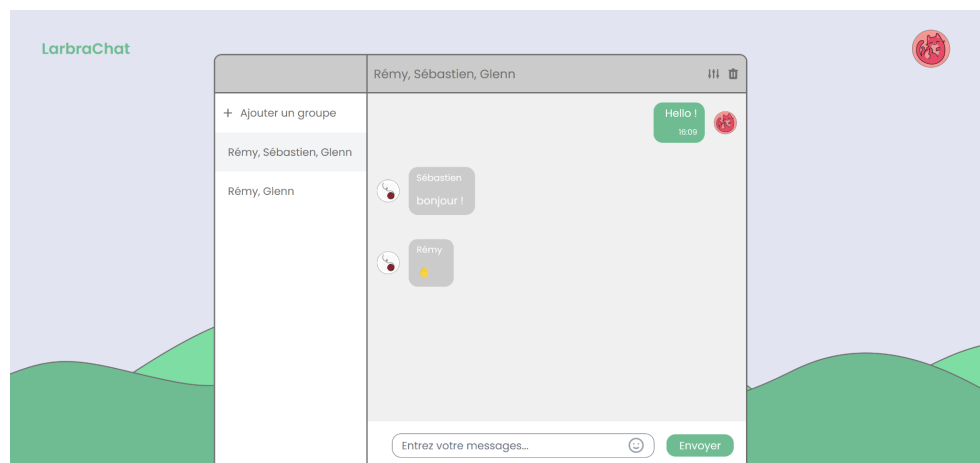
- Conversation privée

La conversation privée est une fonctionnalité très importante de notre chat, pour le fonctionnement cela fonctionne de la même manière qu'un groupe, chaque personne est reliée au groupe et donc chaque personne à accès à celui-ci et peut envoyer et recevoir un message.



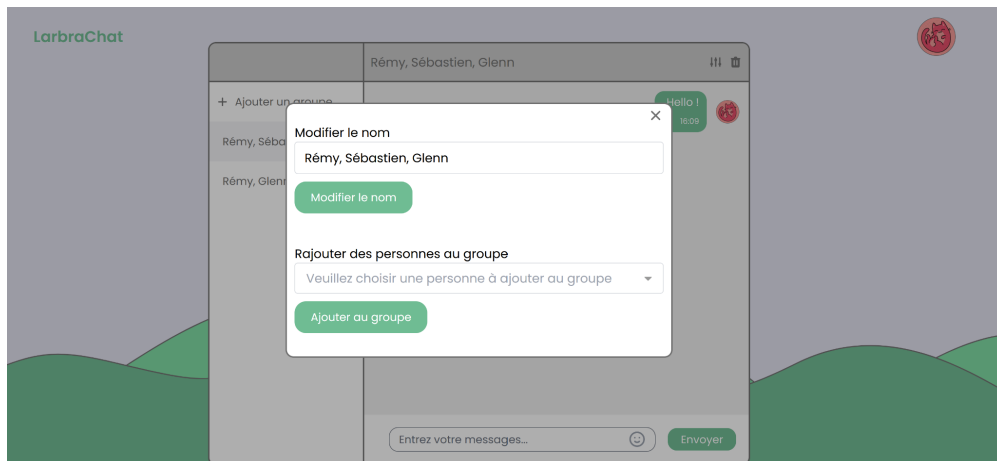
- Conversation par groupe

Pour la conversation par groupe, le fonctionnement est le même que la conversation privée sauf que 3 personnes minimum sont reliées au groupe.

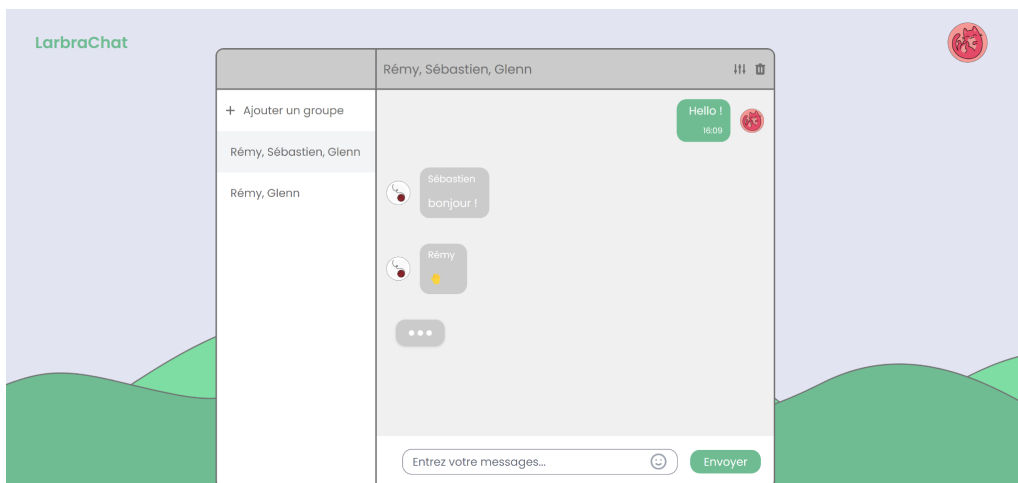


- Modification et personnalisation des conversations

Nous avons aussi ajouté un système de personnalisation des conversations. Chaque utilisateur relié à une conversation peut modifier le nom et rajouter des personnes au sein de la conversation, un utilisateur peut aussi quitter le groupe s'il le décide.



- Animation lorsque quelqu'un écrit

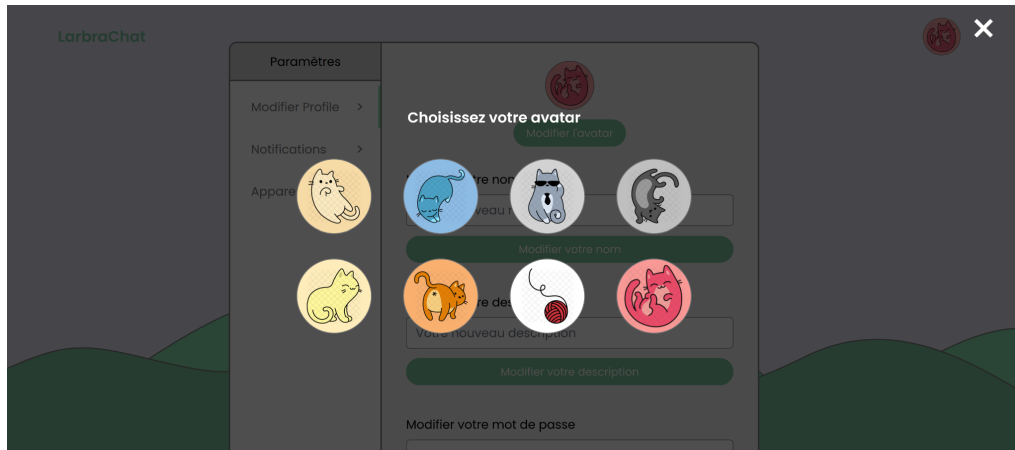


- Prise en charge des émojis

Pour la prise en charge des émojis comme on peut le voir plus haut, cela est fonctionnel grâce à la bonne configuration de la base de données.

- Système d'avatar

Pour le système d'avatar, à l'inscription d'une personne, un avatar par défaut est prédéfini à l'utilisateur, il peut ensuite choisir de le modifier dans les paramètres.



- Modification des informations utilisateur

Pour finir, nous avons ajouté la possibilité pour l'utilisateur de modifier complètement ses informations personnelles (mots de passe, nom et description).

