

DOCUMENTATION :

1. Introduction :

Le projet Whatsapp Web a été conçu dans le but de faciliter l'échange d'informations en temps réel entre utilisateurs. Cette application s'adresse à toutes les personnes possédant un numéro de téléphone, offrant ainsi une solution de communication universelle et accessible.

Pour sa réalisation, le projet utilise Bootstrap, tawind, Html et Css pour le développement du Frontend, garantissant une interface fluide et réactive sur les appareils mobiles, et Laravel 12 pour le Backend, assurant une gestion sécurisée et performante des données et des communications.

Whatsapp Web vise à fournir une expérience de messagerie instantanée moderne, intuitive et fiable, combinant rapidité, simplicité et sécurité, afin de répondre aux besoins quotidiens de communication de ses utilisateurs.

2. Installation et Environnement :

2.1. Frontend :

Initier bootstrap et tawind en initiant les liens html et css :

```
-<link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.0/font/bootstrap-icons.css" rel="stylesheet">
```

```
-<script src="https://cdn.tailwindcss.com"></script>
```

2.2.Backend:

- *Créer des données fictives avec Seeder :*

```
php artisan db :seed
```

- *Lancement du server :*

```
php artisan serve
```

3. Architecture du projet :

Pour illustrer l'architecture de ce projet, nous nous appuierons sur l'illustration suivant :

▼ whatsapplaravel

➤ app

➤ bootstrap

➤ config

➤ database

➤ public

➤ resources

➤ routes

➤ storage

➤ tests

➤ vendor

⚙ .editorconfig

⚙ .env

💰 .env.example

📁 .gitattributes

📁 .gitignore

≡ artisan

{ } composer.json

{ } composer.lock

{ } package-lock.json

{ } package.json

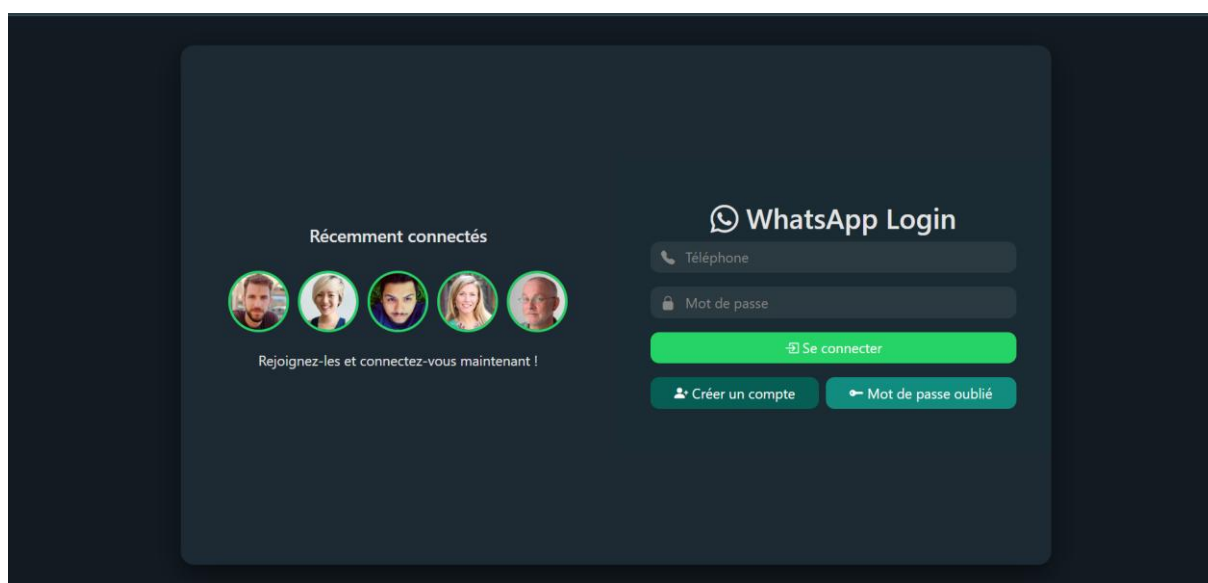
📡 phpunit.xml

ℹ README.md

⚡ vite.config.js

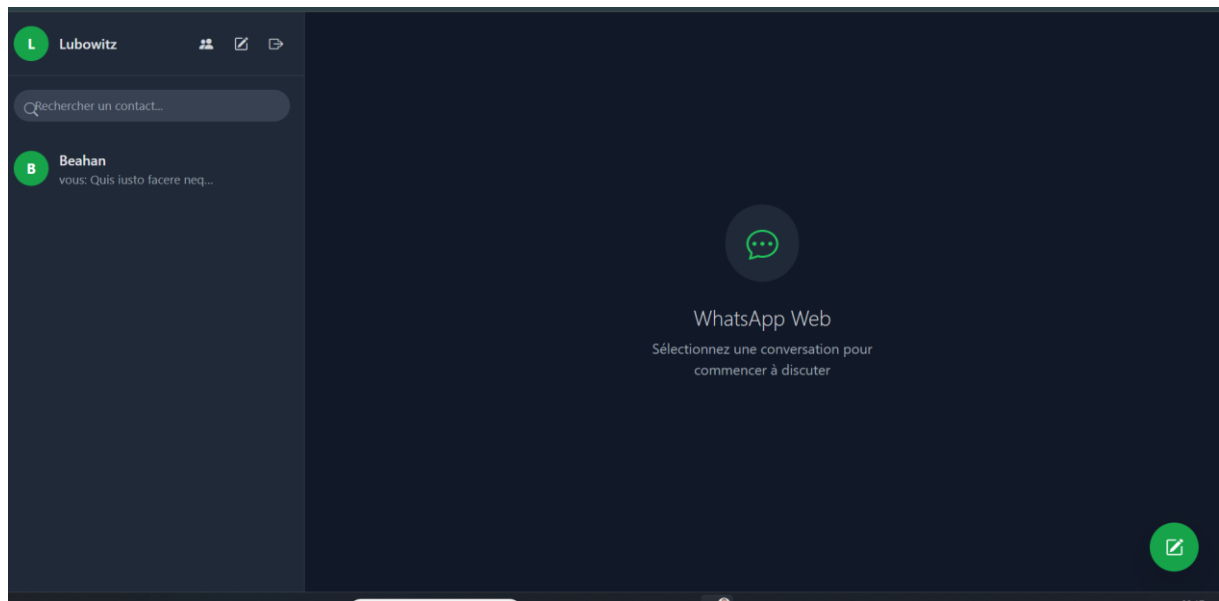
4. Guide d'utilisation mobile :

Lors du lancement du projet un écran d'accueil très intuitive apparait, cette écran redirige les utilisateurs vers l'écran d'authentification :

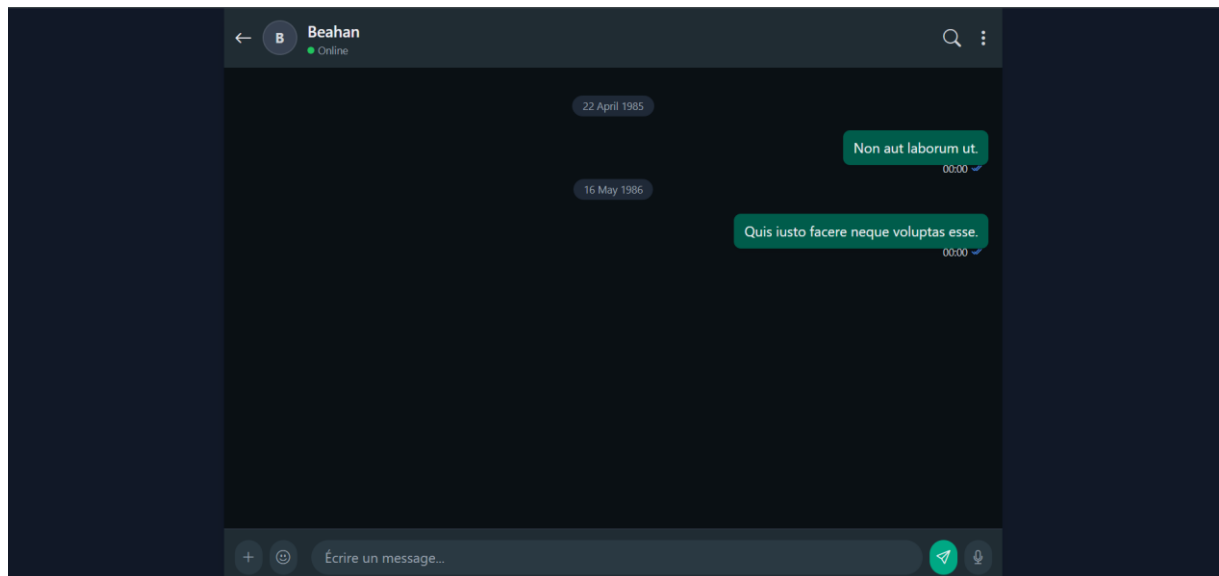


En second, une fois dans l'écran de l'authentification, l'utilisateur doit entrer son numéro de téléphone pour accéder à son liste de discussion. Si l'utilisateur n'est pas dans la base de donnée, il ne sera pas reconnu et une message « Itendifiant Incorrect » en rouge apparait. Ce système ne possède pas encore d'inscription. Il est important de lancer la **commande php artisan db:Seed** pour avoir des utilisateurs.

En troisième, la liste de discussion, cette liste présente les personnes avec qui l'utilisateur a réagit récemment, en appuyant sur une des personnes présentes sur la liste permet de discuter avec celle-ci.

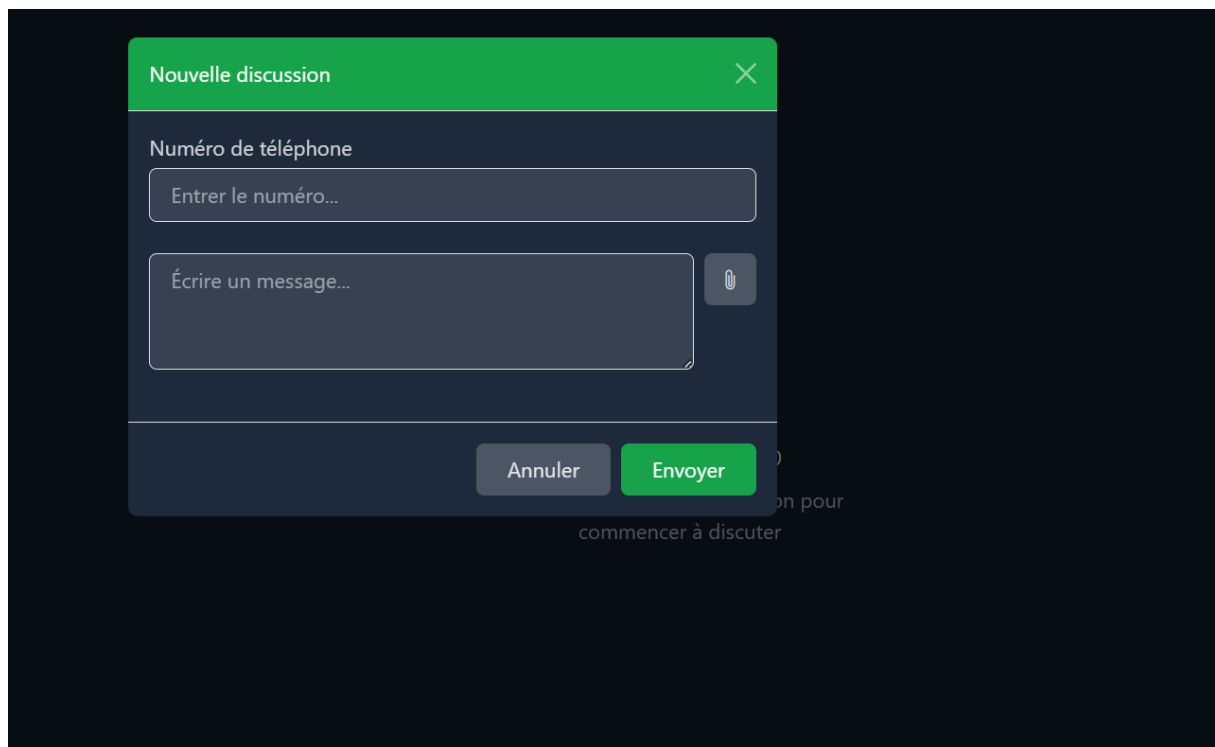


Ensuite, l'interface de discussion présenter ci-dessous présentera la fonctionnalité principale du projet. Elle permet de discuter entre un utilisateur et une autre.



Le flèche permet de retourner vers la liste de discussion et l'écriture 'Modifier' mène vers une interface de modification.

Après, voici l'interface d'ajout d'une discussion dans la liste de discussion.



Remplissez les champs et appuyer sur le bouton « Enregistrer », l'écran de discussion apparaîtra et vous pouvez discuter ensuite.

5. Contribution :

Voici le lien git pour cloner le projet :

git clone <https://github.com/freddy810/ProjetWhatsapp/tree/feature/Reactjs>

Voici quelque suggestion d'amélioration :

- annuler les retours vers la page précédente en utilisant les « middleware »
- utiliser les sanctums et le fonction de hashage pour hacher les mots de passes
- ajouter une interface d'administrateur pour le CRUD

6. Planning :

Le tableau ci-dessous décrit le planning du projet :

Date de début : 05 Août 2025

Date limite : 17 Août 2025

Tâches	Nature	Travaux Antérieur	Durée (jours)
--------	--------	-------------------	---------------

A	Création de la base de données	-	2
B	Création des interface dynamique	A	3
C	Configuration de l'authentification sur l'interface d'authentification	B	3
D	Affichage des listes de discussion pour chaque utilisateurs connecté	C	1
E	Configuration du discussion instantane	D	1
F	Ajout de discussion à la liste de discussion	E	1