

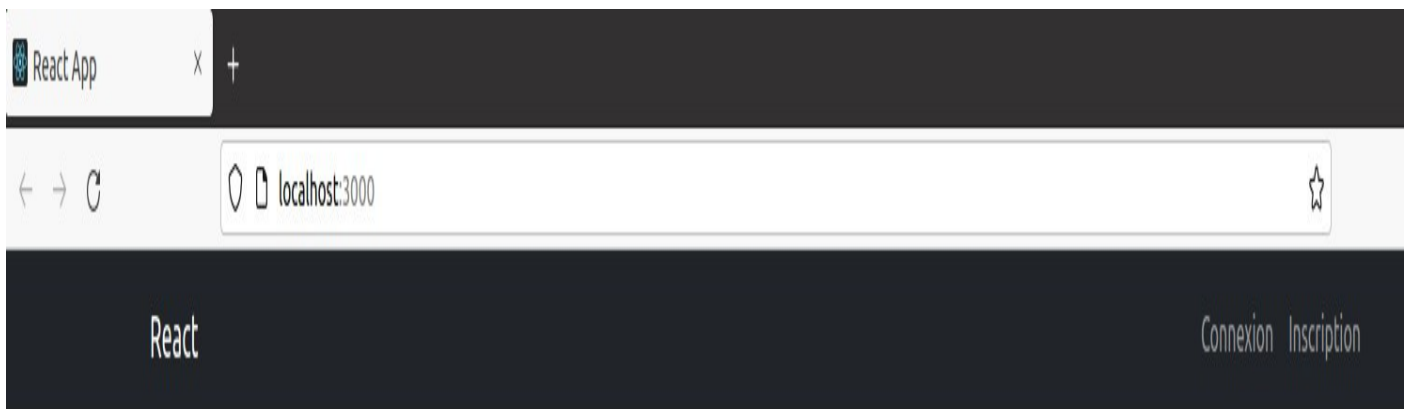
Compte rendu de mon projet Inscription et Authentification

J'ai réalisé un projet personnel (**Inscription et Authentification**) pour faire une inscription et une connexion qui permet d'enregistrer ou de récupérer les informations sur la base de données MySQL. Pour réaliser ce projet, j'ai développé en deux parties : Front-End (côté client) en utilisant axios qui va envoyer une requête HTTP au serveur, et Back-End (côté serveur) en utilisant express.json() qui va retourner la réponse par la requête HTTP client.

1) Front-End :

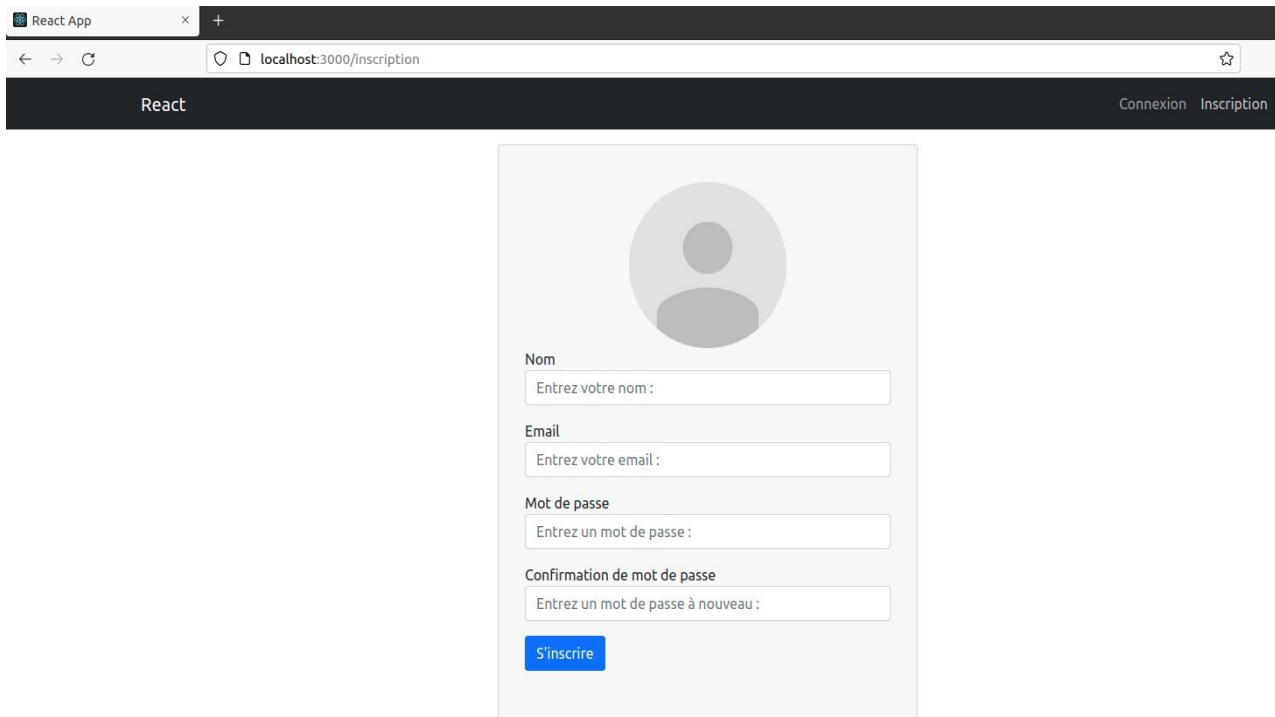
Dans la partie Front-End, j'ai utilisé le framework ReactJs (bibliothèque JavaScript) pour créer une interface utilisateur interactive, j'utilise aussi Axios (clients HTTP les plus populaires pour les navigateurs et les nœuds basés sur des promesses) pour envoyer une requête POST au serveur, puis enregistrer les données JSON reçues dans la base de données MySQL.

Page d'accueil : [accueil.js](#)



Dans la page d'accueil, j'ai créé deux liens via le bouton Connexion et Inscription qui une fois cliqué permettent de se rediriger vers le lien exact.

Page Inscription : inscription.js



React App

localhost:3000/inscription

React

Connexion Inscription

Nom

Entrez votre nom :

Email

Entrez votre email :

Mot de passe

Entrez un mot de passe :

Confirmation de mot de passe

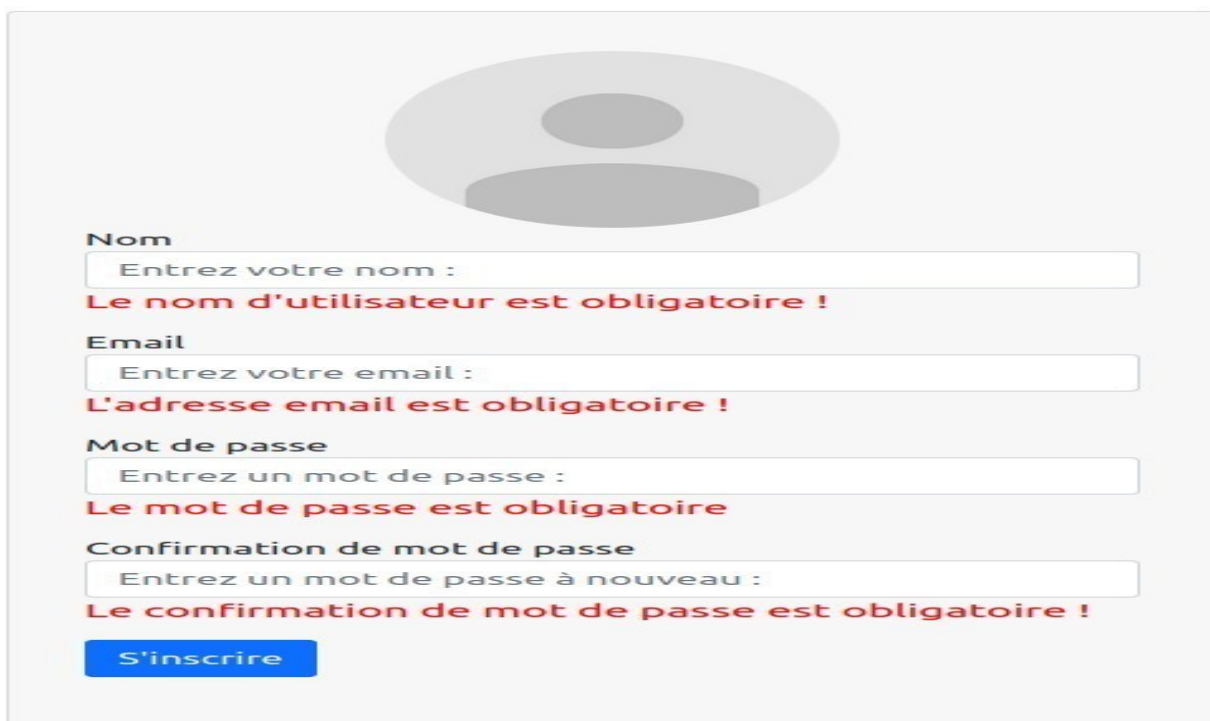
Entrez un mot de passe à nouveau :

S'inscrire

Lorsque je clique le lien via le bouton Inscription, ça redirige vers le lien :

<http://localhost:3000/inscription>

Dans cette page, j'ai créer 4 zones de textes et un bouton pour S'inscrire à la base de données.



Nom

Entrez votre nom :

Le nom d'utilisateur est obligatoire !

Email

Entrez votre email :

L'adresse email est obligatoire !

Mot de passe

Entrez un mot de passe :

Le mot de passe est obligatoire

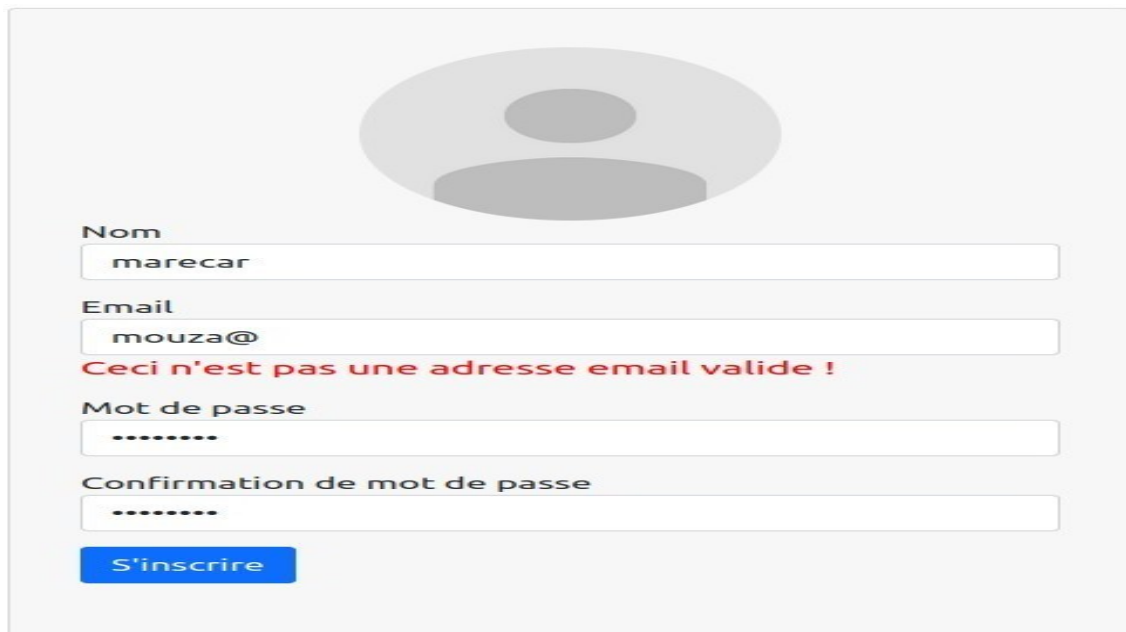
Confirmation de mot de passe

Entrez un mot de passe à nouveau :

Le confirmation de mot de passe est obligatoire !

S'inscrire

Lorsque je clique sur le bouton S'inscrire et que un ou plusieurs zones de textes sont vides, j'affiche les messages erreurs pour ces zones de texte.



A user registration form with a light gray background. At the top center is a large, light gray circular placeholder for a profile picture. Below it are four input fields, each with a label to its left: 'Nom' (containing 'marecar'), 'Email' (containing 'mouza@'), 'Mot de passe' (containing seven dots), and 'Confirmation de mot de passe' (containing seven dots). A red error message, 'Ceci n'est pas une adresse email valide !', is positioned below the email field. At the bottom left is a blue button with the text 'S'inscrire'.

Nom
marecar

Email
mouza@

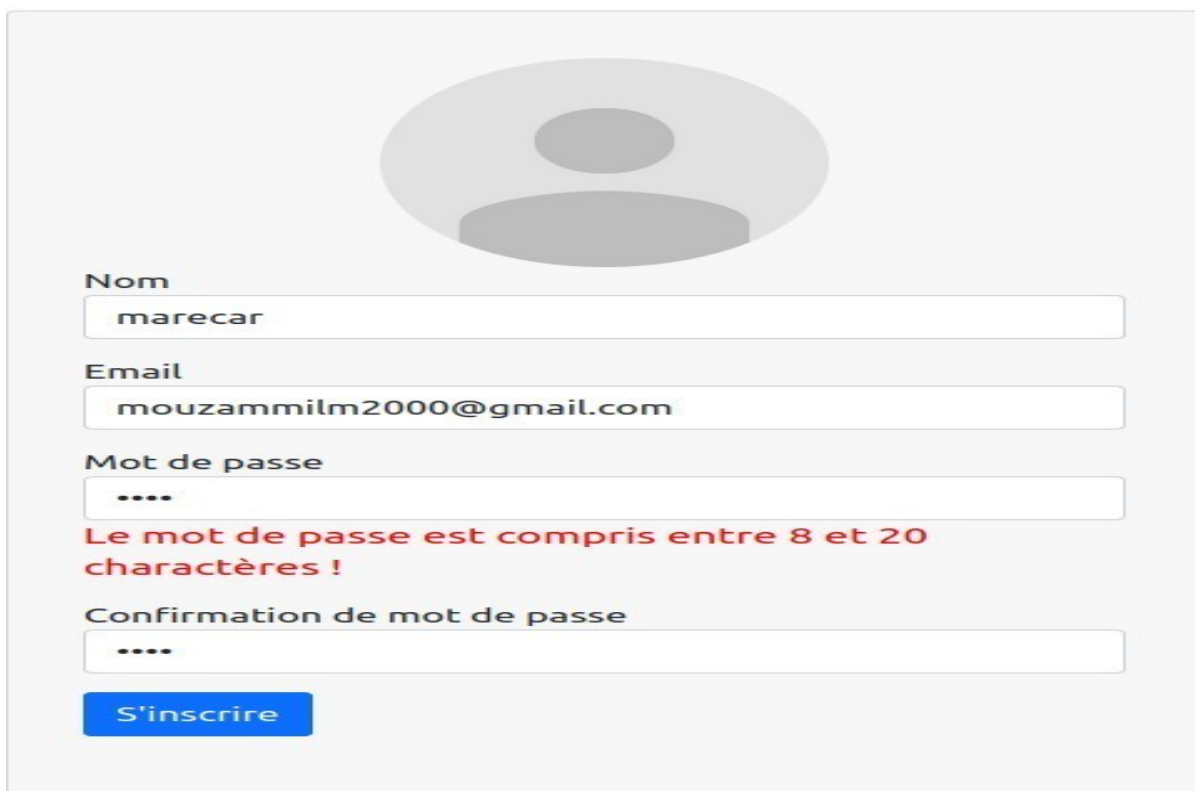
Ceci n'est pas une adresse email valide !

Mot de passe
.....

Confirmation de mot de passe
.....

S'inscrire

Si l'adresse email est en format incorrect, alors j'affiche une erreur de message en bas de la zone de texte email et ces données ne seront pas enregistrés dans la base de données.



A user registration form with a light gray background. At the top center is a large, light gray circular placeholder for a profile picture. Below it are four input fields, each with a label to its left: 'Nom' (containing 'marecar'), 'Email' (containing 'mouzammilm2000@gmail.com'), 'Mot de passe' (containing four dots), and 'Confirmation de mot de passe' (containing four dots). A red error message, 'Le mot de passe est compris entre 8 et 20 caractères !', is positioned below the password field. At the bottom left is a blue button with the text 'S'inscrire'.

Nom
marecar

Email
mouzammilm2000@gmail.com


Mot de passe
....

Le mot de passe est compris entre 8 et 20 caractères !

Confirmation de mot de passe
....

S'inscrire

Si le mot de passe est inférieure à 8 ou supérieur à 20 caractères, j'affiche cet erreur et l'insertion ne sera donc pas effectuée.



Nom

Email

Mot de passe

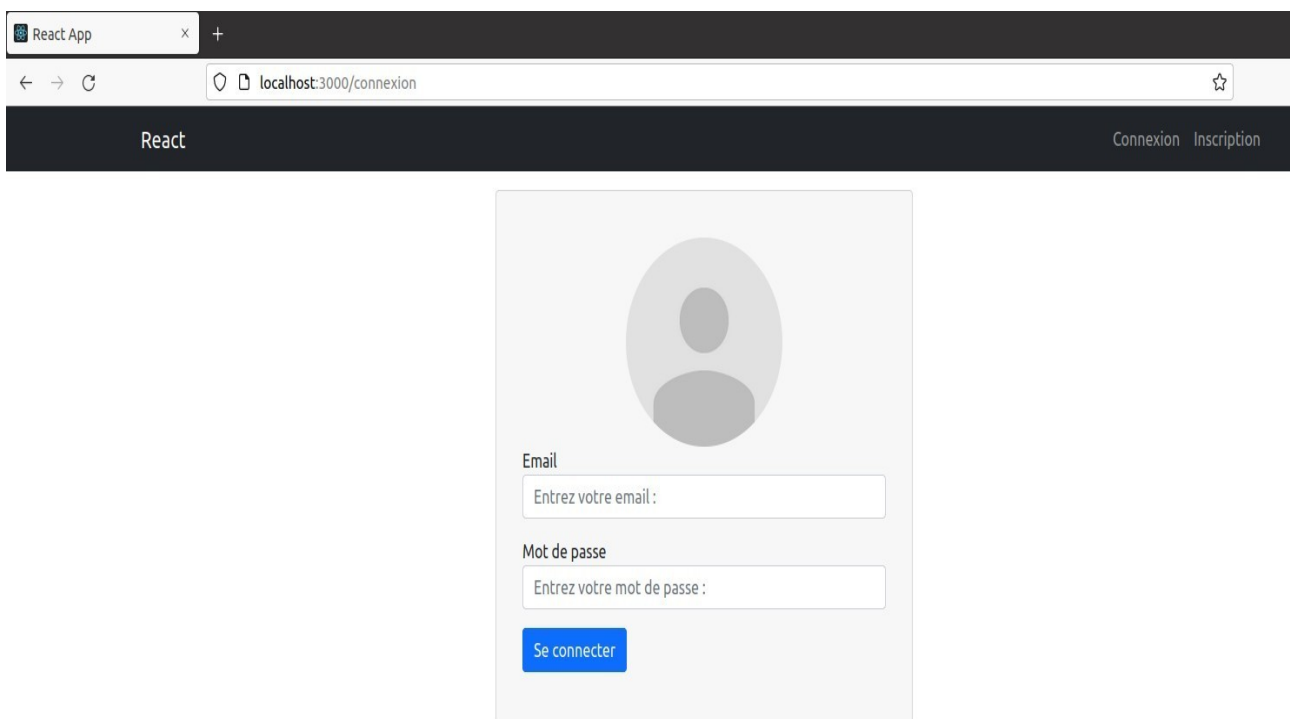
Confirmation de mot de passe

Le mot de passe ne correspond pas avec le nouveau mot de passe !

[S'inscrire](#)

Si le mot de passe n'est pas égale avec ce nouveau mot de passe, j'affiche cet erreur, alors l'insertion de la base de données ne sera pas effectuée.

[Page Connexion : connexion.js](#)



React App x +

localhost:3000/connexion

React Connexion Inscription

Email

Mot de passe

[Se connecter](#)

Lorsque je clique le lien via le bouton Connexion, ça redirige vers le lien : <http://localhost:3000/connexion>

React App x +

localhost:3000/connexion

React Connexion Inscription

Email

Entrez votre email :

Veuillez entrez votre adresse email !

Mot de passe

Entrez votre mot de passe :

Se connecter

Si je clique sur le bouton Se connecter et que zone de texte email est vide, j'affiche cet erreur.

Email

mouzammilm2000@gmail.com

Mot de passe

Entrez votre mot de passe :

Veuillez entrez votre mot de passe !

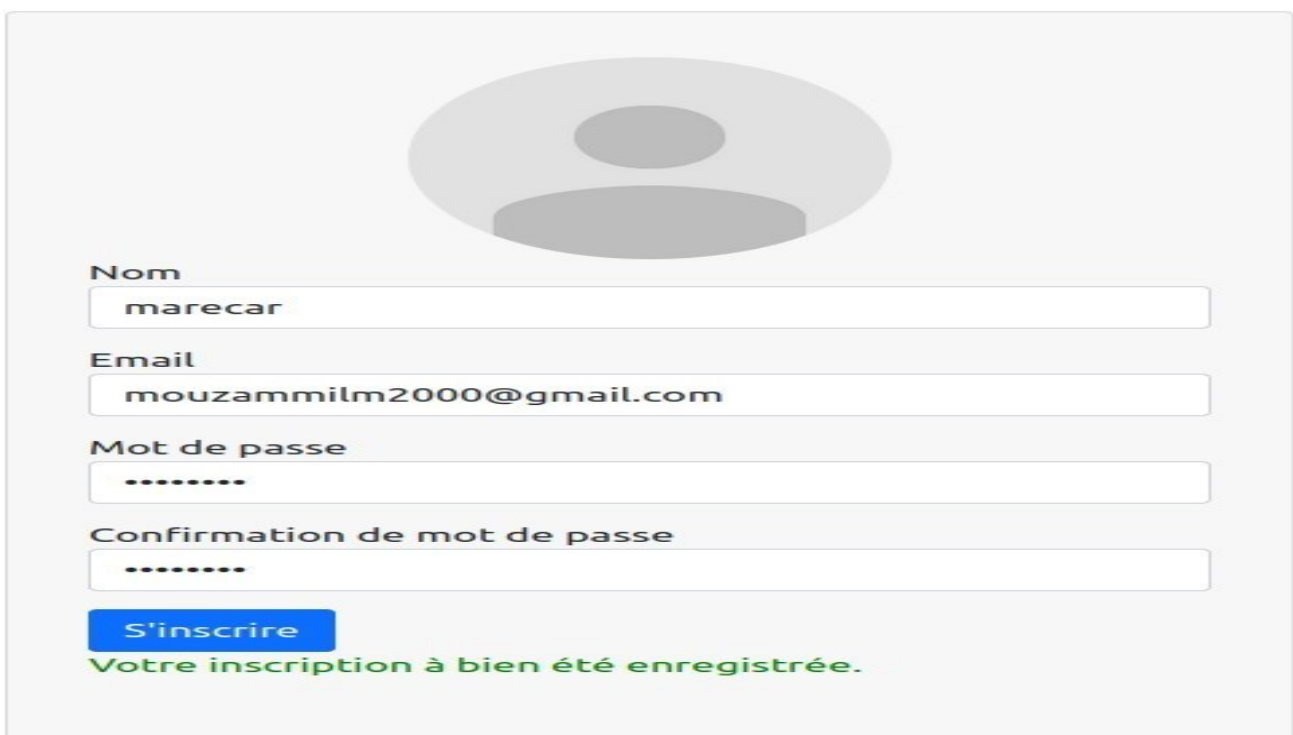
Se connecter

Si je clique sur le bouton Se connecter et que zone de texte mot de passe est vide, j'affiche cet erreur.

Back-End :

Dans la partie Back-End, j'utilise une plateforme logicielle **NodeJS** pour exécuter du JavaScript côté serveur. J'utilise **ExpressJS** (une librairie qui permet de créer une application Web plus simplement qu'avec l'objet HTTP directement) pour fournir un ensemble de méthodes qui permet de traiter les requêtes HTTP. Dans ExpressJS, j'ai créer une route qui correspondra à un chemin d'une requête potentiel et j'ai utilisé `express.json()` pour envoyer des données au client. La route attend une requête POST (page d'inscription ou connexion) et une fois cette demande reçu, le serveur enverra les messages d'erreur ou résultat de la base de données MySQL en réponse (`res.send()`) au navigateur. J'utilise aussi **CORS** (mécanisme qui consiste à ajouter des en-têtes HTTP afin de permettre à un agent utilisateur d'accéder à des ressources d'un serveur situé sur une autre origine que le site courant.) pour faire une requête côté serveur de <http://localhost:3001/inscription> ou [connexion](#) sur un domaine différent côté client à <http://localhost:3000/inscription> ou [connexion](#). Et pour finir ,j'utilise **MySQL** pour enregistrer les données de la page d'inscription, et pour récupérer ses informations de la base de données dans la page connexion.

Page d'Inscription :



```
mysql> select * from client;
+----+-----+-----+-----+-----+
| id | name   | email                               | password | confirm_password |
+----+-----+-----+-----+-----+
| 91 | marecar | mouzammilm2000@gmail.com | azertyui | azertyui         |
+----+-----+-----+-----+-----+
1 row in set (0,02 sec)
```

Si toutes les zones de textes sont remplies et que le format de l'email est correct, et que le mot de passe est égale à ce nouveau, alors l'insertion sera effectuée. On peut voir ci-dessus que les données sont enregistrés dans la base de données.



Nom

Email

L'email existe déjà !

Mot de passe


Confirmation de mot de passe

[S'inscrire](#)

Si l'email existe déjà dans la base de données, alors j'affiche cette message d'erreur.

[Page Connexion :](#)

```
mysql> select * from client;
+-----+-----+-----+-----+-----+
| id | name      | email                               | password | confirm_password |
+-----+-----+-----+-----+-----+
| 91 | marecar   | mouzammilm2000@gmail.com | azertyui | azertyui         |
+-----+-----+-----+-----+-----+
1 row in set (0,02 sec)
```



Email

Mot de passe


[Se connecter](#)

91
marecar
mouzammilm2000@gmail.com

```
mysql> select * from client;
```

| id | name | email | password | confirm_password |
|----|---------|--------------------------|----------|------------------|
| 91 | marecar | mouzammilm2000@gmail.com | azertyui | azertyui |
| 93 | Luc | luc@luc.com | qsdghjk | qsdghjk |

2 rows in set (0,00 sec)




Email

Mot de passe

[Se connecter](#)

93
Luc
luc@luc.com

Si l'email existe dans la base de données et que le mot de passe est correct alors j'affiche le résultat de la ligne correspondante.



Email

Mot de passe

[Se connecter](#)

Email ou mot de passe incorrect !

Si l'email existe dans la base de données et que le mot de passe est incorrect, alors j'affiche le message d'erreur.

Conclusion :

Ce projet est ma première expérience en développement Full-Stack.

Dans ce projet, j'ai développé en deux parties Full-Stack : Front-End et Back-End. Dans la partie Front-End, le client demande une requête au serveur par Axios via la méthode POST, et dans Back-End, le serveur retourne le message d'erreur ou le résultat par requêtes HTTP client via la méthode POST et envoie des données par `express.json()` en réponse au navigateur.

Dans ce projet, j'ai réussi à se connecter du Front-End au Back-End en envoyant des requêtes HTTP client au serveur, et du Back-End au Front-End en retournant la réponse, j'ai donc réussi à insérer les données de la page d'inscription dans la base de données MySQL, et j'ai réussi à récupérer les informations de la base de données dans la page connexion.