React, Node et MySql Login (Partie 2)

Étape 3 : Créer un premier formulaire côté React (email, pincode)
Vous pouvez récupérer le formulaire de bootstrap tout fait et l'adapter
https://getbootstrap.com/docs/5.2/forms/overview/#overview

Composant LoginScreen (à importer et intégrer dans App.js)

```
1  function LoginScreen(props) {
2    return (
3      <>
4        <form>
5          <div className="mb-3">
6            <label htmlFor="email" className="form-label">
7              Email address
8            </label>
9            <input
10             id="email"
11             name="email"
12             type="email"
13             className="form-control"
14           />
15         </div>
16         <div className="mb-3">
17           <label htmlFor="pincode" className="form-label">
18             Pincode
19           </label>
20           <input
21             id="pincode"
22             name="pincode"
23             type="text"
24             className="form-control"
25           />
26         </div>
27         <button type="submit" className="btn btn-primary">
28           Login
29         </button>
30       </form>
31     </>
32   );
33 }
34 export default LoginScreen;
```

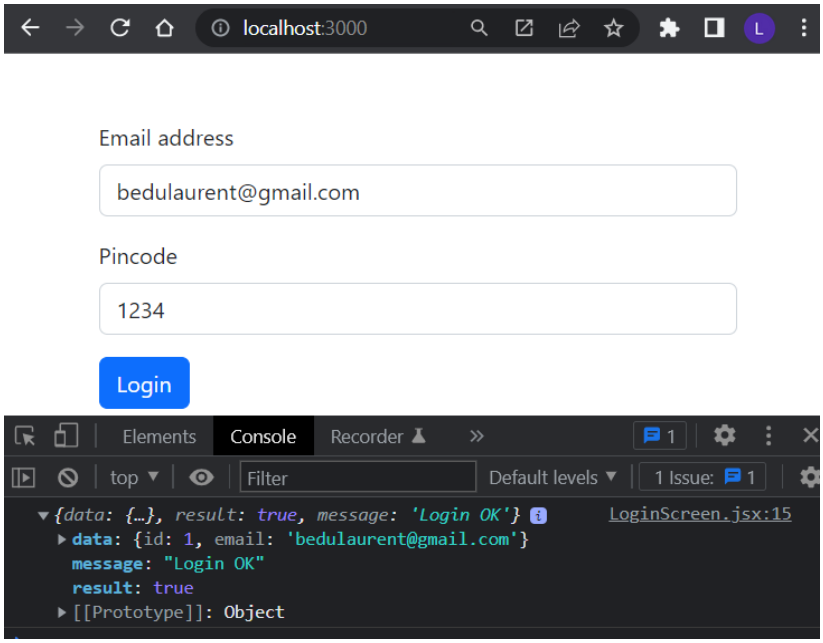
Ln 11 et 22 : Ne pas oublier d'ajouter un name à chaque input

Étape 4 : Ajouter un événement onSubmit sur la balise form qui exécute une méthode handleSubmit pour récupérer les données du formulaire et les envoyer vers l'API

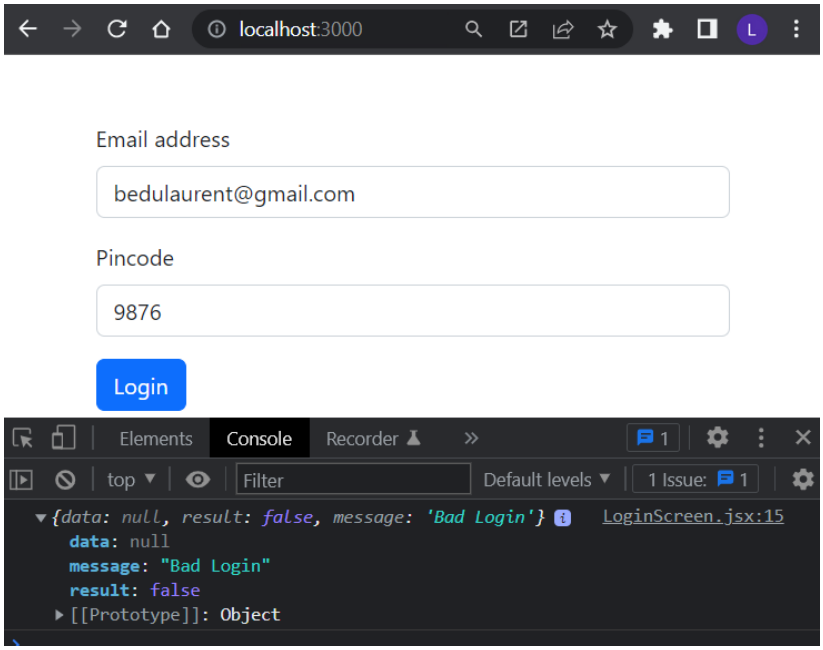
```
1   function LoginScreen(props) {
2
3     const handleSubmit = (e) => {
4       e.preventDefault();
5       const form = e.currentTarget;
6       const formData = new FormData(form);
7       const jsonData = Object.fromEntries(formData.entries());
8       const body = JSON.stringify(jsonData);
9       fetch("http://localhost:5000/login", {
10        method: "post",
11        headers: {
12          "content-type": "application/json",
13        },
14        body,
15      })
16      .then(resp => resp.json())
17      .then(json => console.log(json))
18      .catch(err => console.log(err));
19    };
20
21    return (
22      <>
23        <form onSubmit={handleSubmit}>
24          <div className="mb-3">
25            <label htmlFor="email" className="form-label">
26              Email address
```

Ln 4 : Arrêt du comportement par défaut de l'événement submit
Ln 5 à 8 : Construction du body de la requête HTTP à partir des inputs du formulaire
Ln 9 : Envoie de la requête HTTP vers l'API (Route POST/login) avec le body
Ln 16 à 18 : Traitement de la réponse (lecture du json et affichage dans la console)

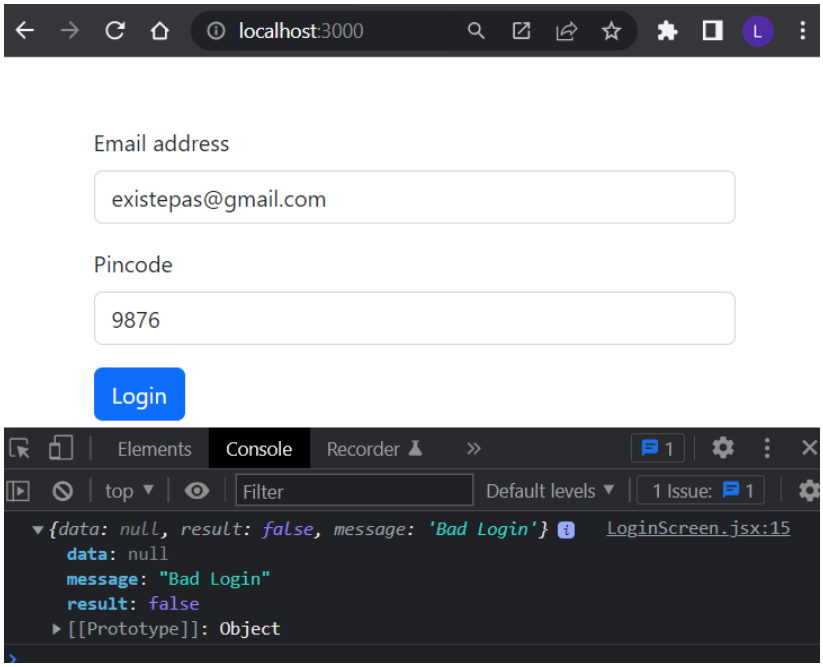
Tester tous les cas possible (idem étape 2) depuis le formulaire
Cas 1 :



Cas 2 :


localhost:3000
Email address
bedulaurent@gmail.com
Pincode
9876
Login
Elements
Console
Recorder
top
Filter
Default levels
1 Issue: 1
{data: null, result: false, message: 'Bad Login'}
LoginScreen.jsx:15
data: null
message: "Bad Login"
result: false
[[Prototype]]: Object

Cas 3 :


localhost:3000
Email address
existepas@gmail.com
Pincode
9876
Login
Elements
Console
Recorder
top
Filter
Default levels
1 Issue: 1
{data: null, result: false, message: 'Bad Login'}
LoginScreen.jsx:15
data: null
message: "Bad Login"
result: false
[[Prototype]]: Object