

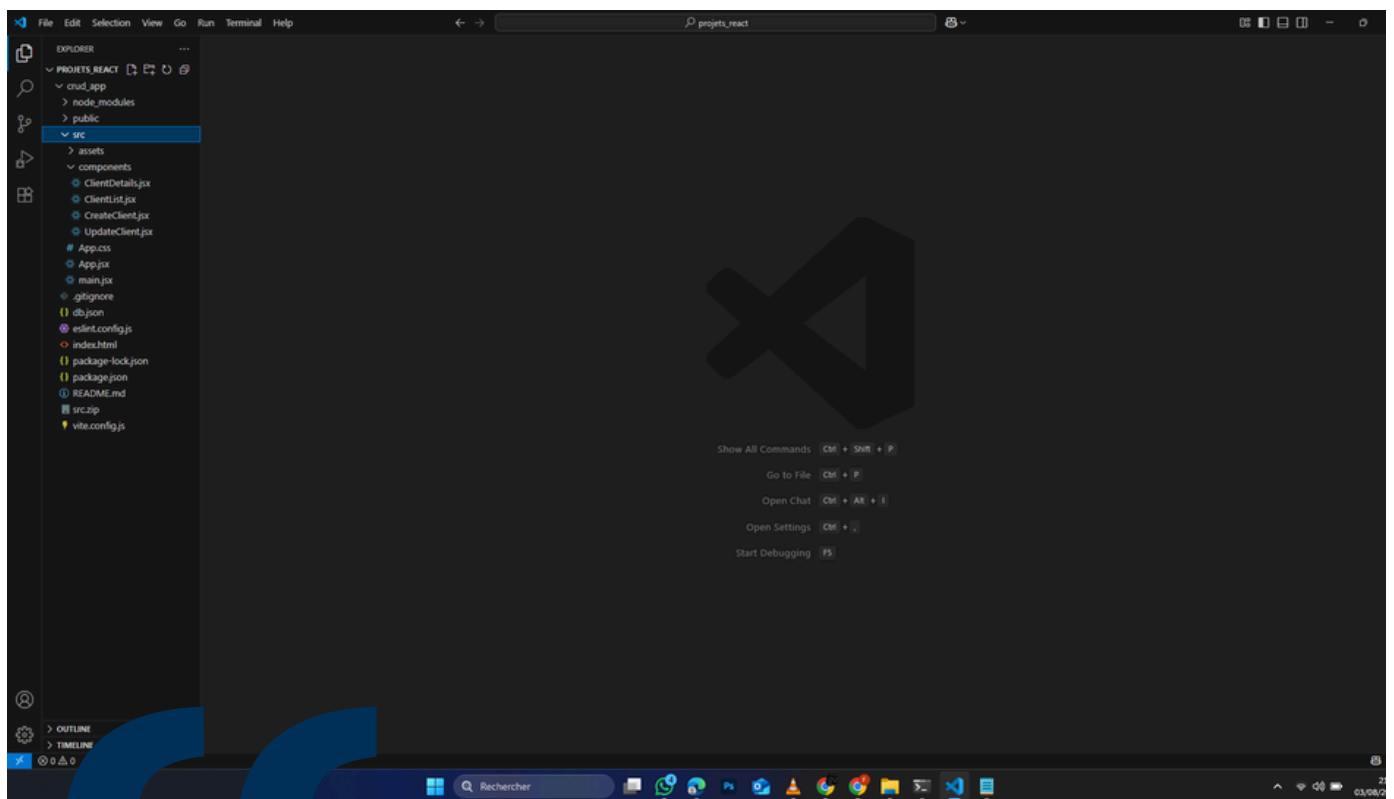
PDF DE RÉALISATION

CRÉATION D'UNE APPLICATION CRUD AVEC REACT JS

Fortuné ADJAGBA

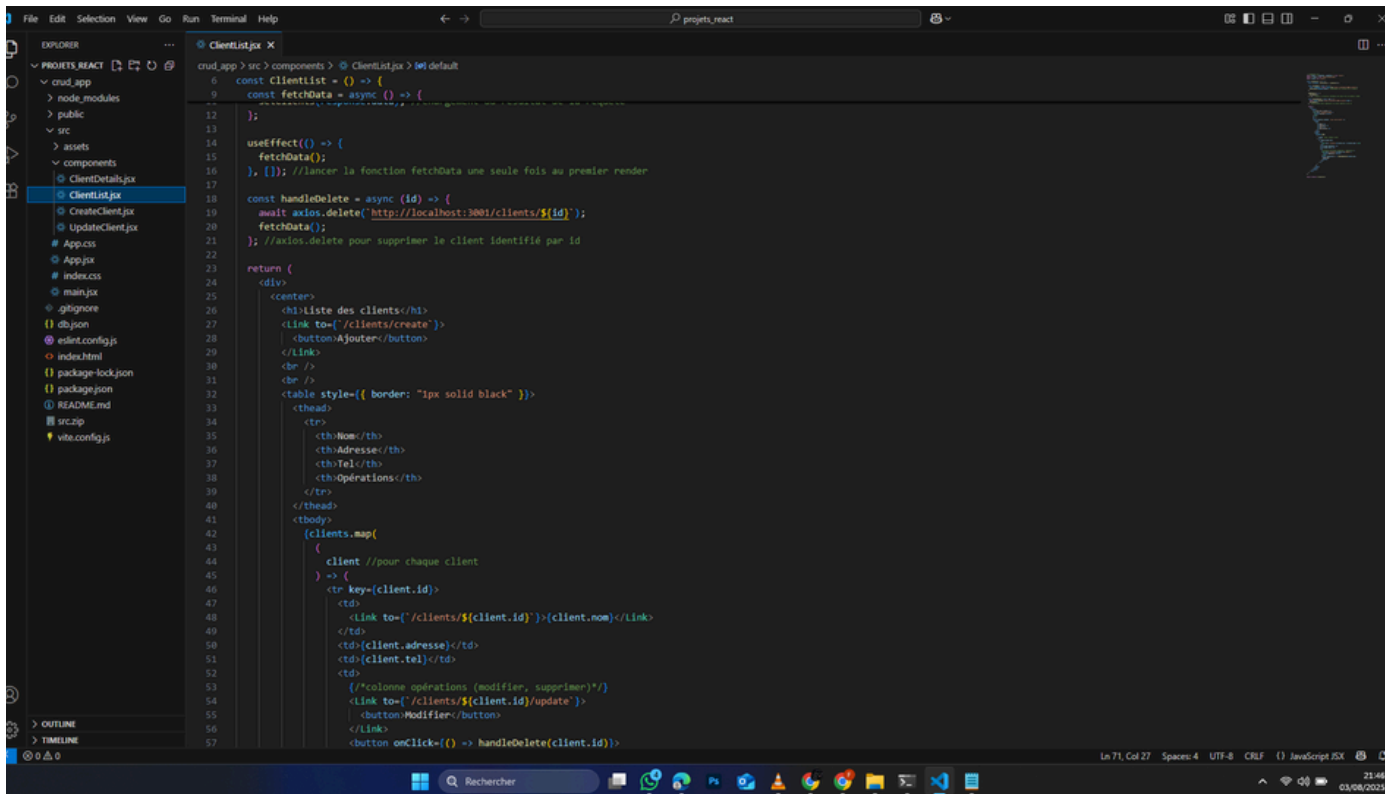
1. CONFIGURATION DU PROJET ET STRUCTURE DES FICHIERS

Capture d'écran montrant l'arborescence de mon projet après la création du dossier component et des fichiers .jsx nécessaires (**ClientList.jsx**, **ClientDetails.jsx**, **CreateClient.jsx** et **UpdateClient.jsx**).

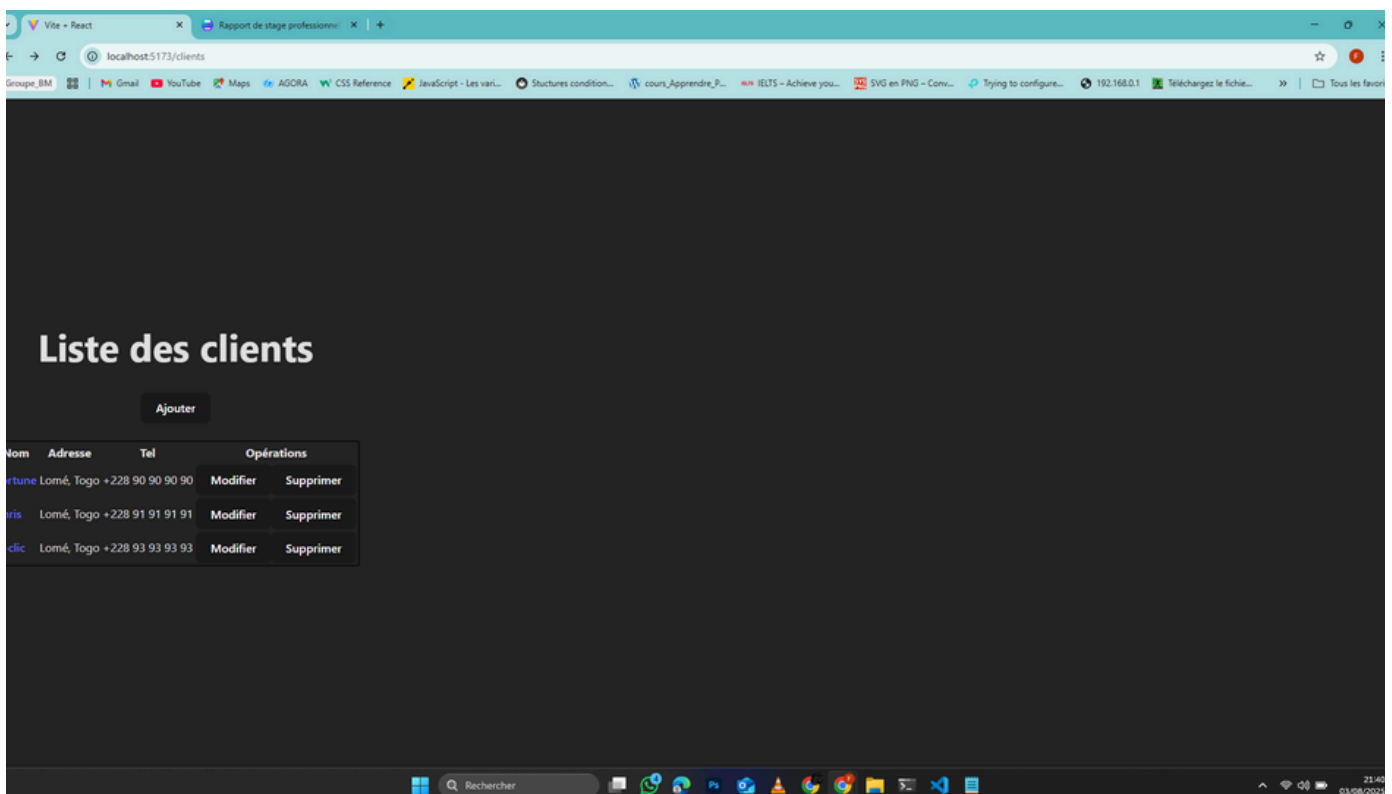


2. IMPLÉMENTATION DU COMPOSANT CLIENTLIST.JSX

- Capture d'écran du code ClientList.jsx initial (avant les styles Bootstrap).
- Capture d'écran montrant l'affichage de la liste des clients dans le navigateur, avec la table et les boutons **"Ajouter"**, **"Modifier"**, **"Supprimer"** fonctionnels.

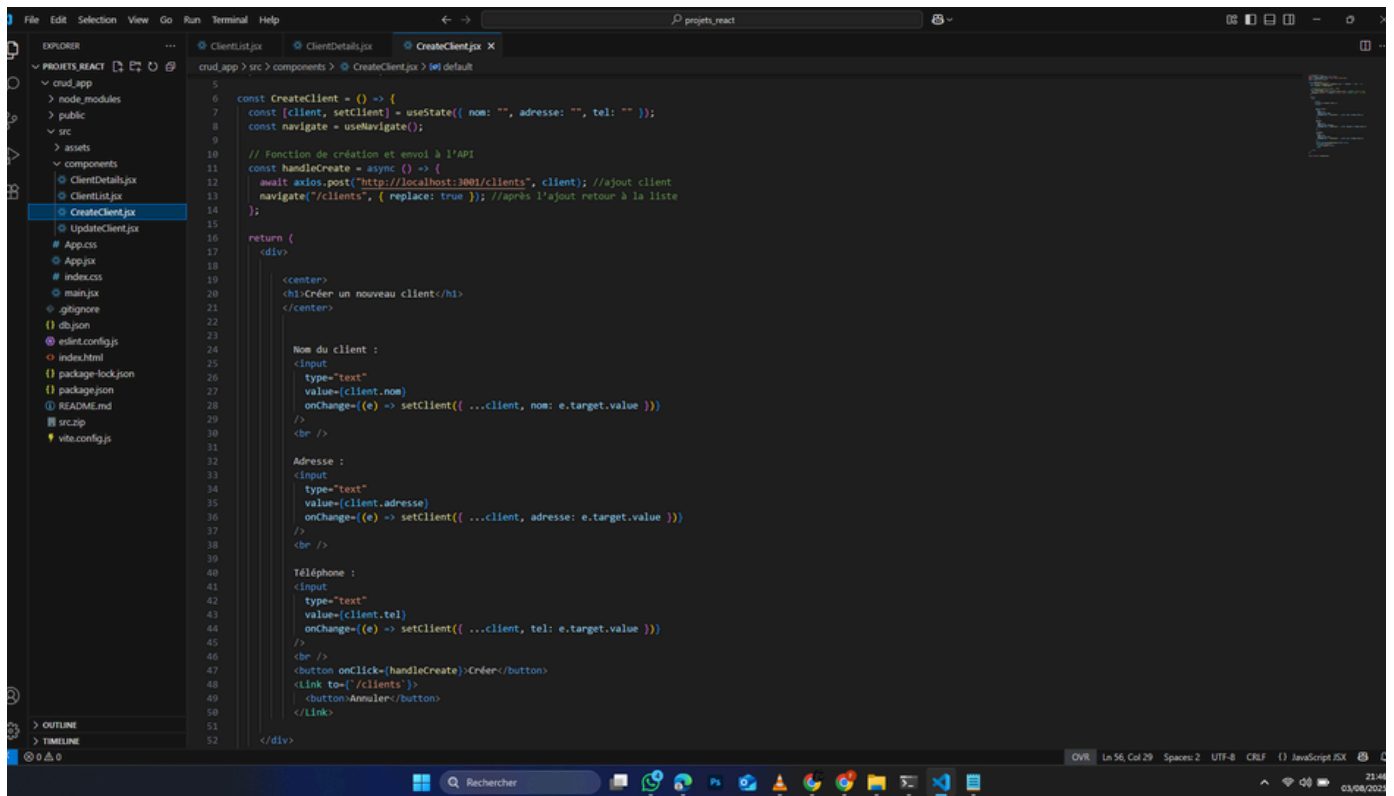


```
6 const ClientList = () => {
9   const fetchData = async () => {
12  };
13
14   useEffect(() => {
15     fetchData();
16   }, []); //lancer la fonction fetchData une seule fois au premier render
17
18   const handleDelete = async (id) => {
19     await axios.delete(`http://localhost:3001/clients/${id}`);
20     fetchData();
21   }; //axios.delete pour supprimer le client identifié par id
22
23   return (
24     <div>
25       <h1>
26         <h2>
27         <Link to="/clients/create">
28         <button>Ajouter</button>
29         </Link>
30         <br />
31         <br />
32         <table style={{ border: "1px solid black" }}>
33           <thead>
34             <tr>
35               <th>Nom</th>
36               <th>Adresse</th>
37               <th>Tel</th>
38               <th>Opérations</th>
39             </tr>
40           </thead>
41           <tbody>
42             <clients.map(
43               (
44                 client //pour chaque client
45               ) => (
46                 <tr key={client.id}>
47                   <td>
48                     <Link to="/clients/${client.id}">{client.nom}</Link>
49                   </td>
50                   <td>{client.adresse}</td>
51                   <td>{client.tel}</td>
52                   <td>
53                     {/*colonne opérations (modifier, supprimer)*/}
54                     <Link to="/clients/${client.id}/update">
55                       <button>Modifier</button>
56                     </Link>
57                     <button onClick={() => handleDelete(client.id)}>
```

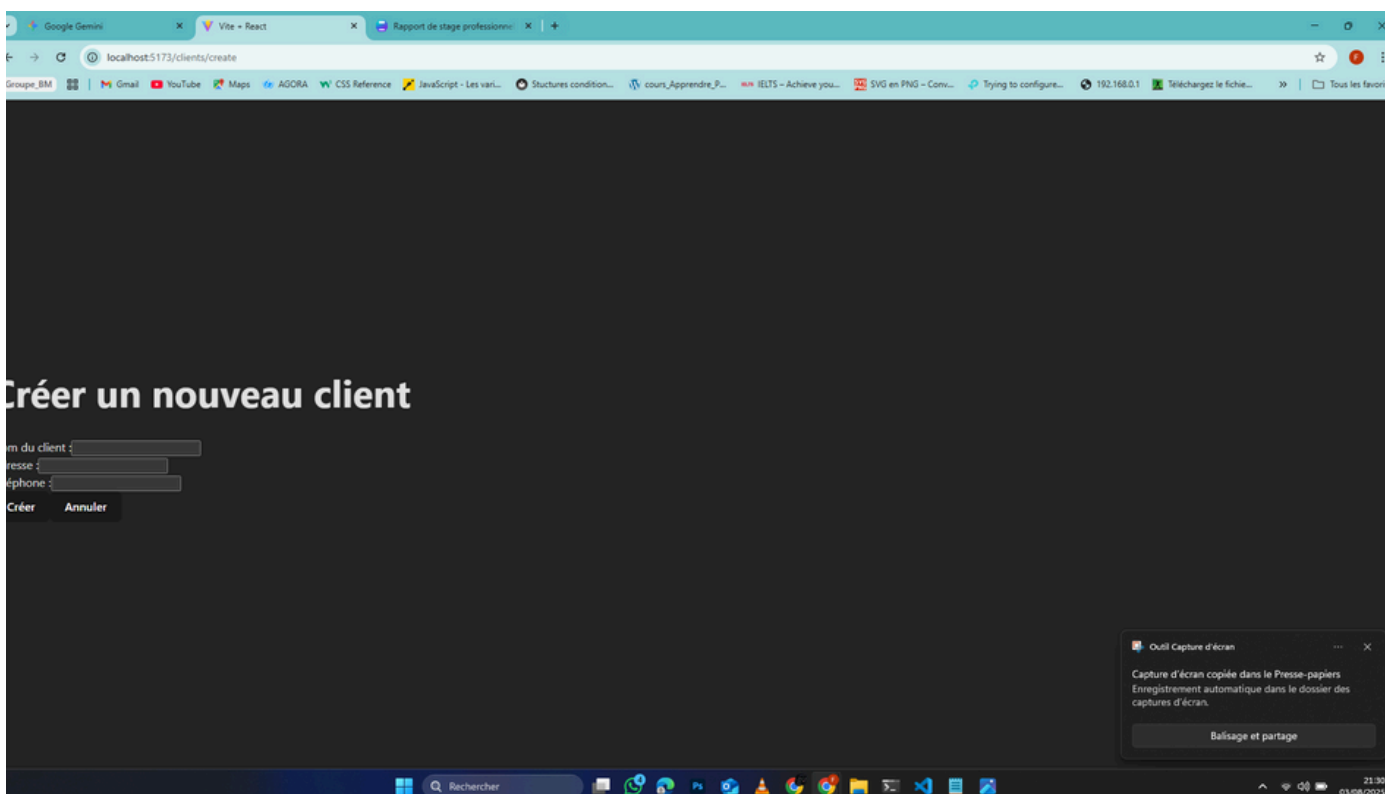


3. IMPLÉMENTATION DE LA CRÉATION DU CLIENT

- Capture d'écran du code CreateClient.jsx initial.
- Capture d'écran montrant l'affichage du formulaire de création d'un nouveau client.

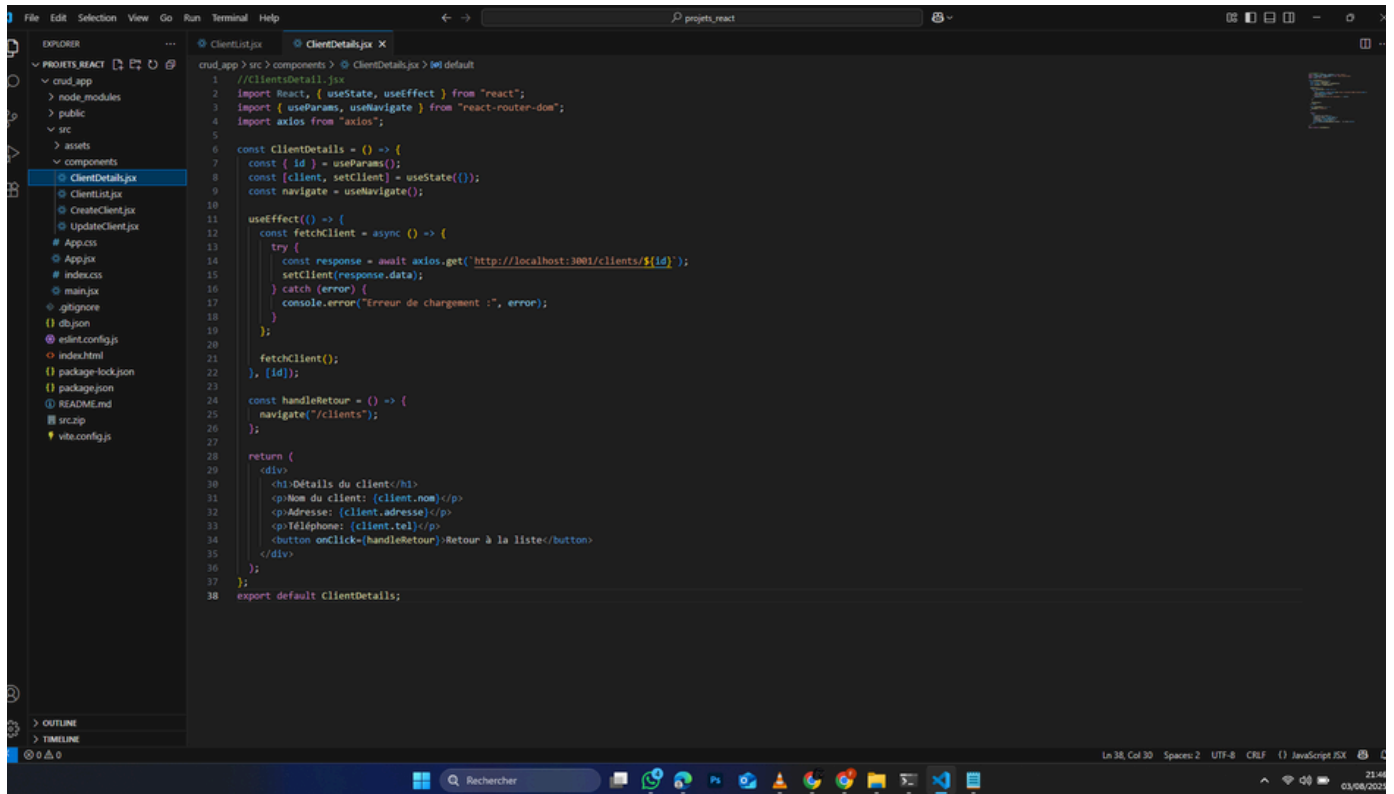


```
5
6 const CreateClient = () => {
7   const [client, setClient] = useState({ nom: "", adresse: "", tel: "" });
8   const navigate = useNavigate();
9
10  // Fonction de création et envoi à l'API
11  const handleCreate = async () => {
12    await axios.post("http://localhost:3001/clients", client); //ajout client
13    navigate("/clients", { replace: true }); //après l'ajout retour à la liste
14  };
15
16  return (
17    <div>
18
19      <center>
20        <h1>Créer un nouveau client</h1>
21      </center>
22
23      Nom du client :
24      <input
25        type="text"
26        value={client.nom}
27        onChange={e => setClient({ ...client, nom: e.target.value })}
28      />
29      <br />
30
31      Adresse :
32      <input
33        type="text"
34        value={client.adresse}
35        onChange={e => setClient({ ...client, adresse: e.target.value })}
36      />
37      <br />
38
39      Téléphone :
40      <input
41        type="text"
42        value={client.tel}
43        onChange={e => setClient({ ...client, tel: e.target.value })}
44      />
45      <br />
46      <button onClick={handleCreate}>Créer</button>
47      <Link to="/clients"><button>Annuler</button></Link>
48    </div>
49  );
50
51
52
```

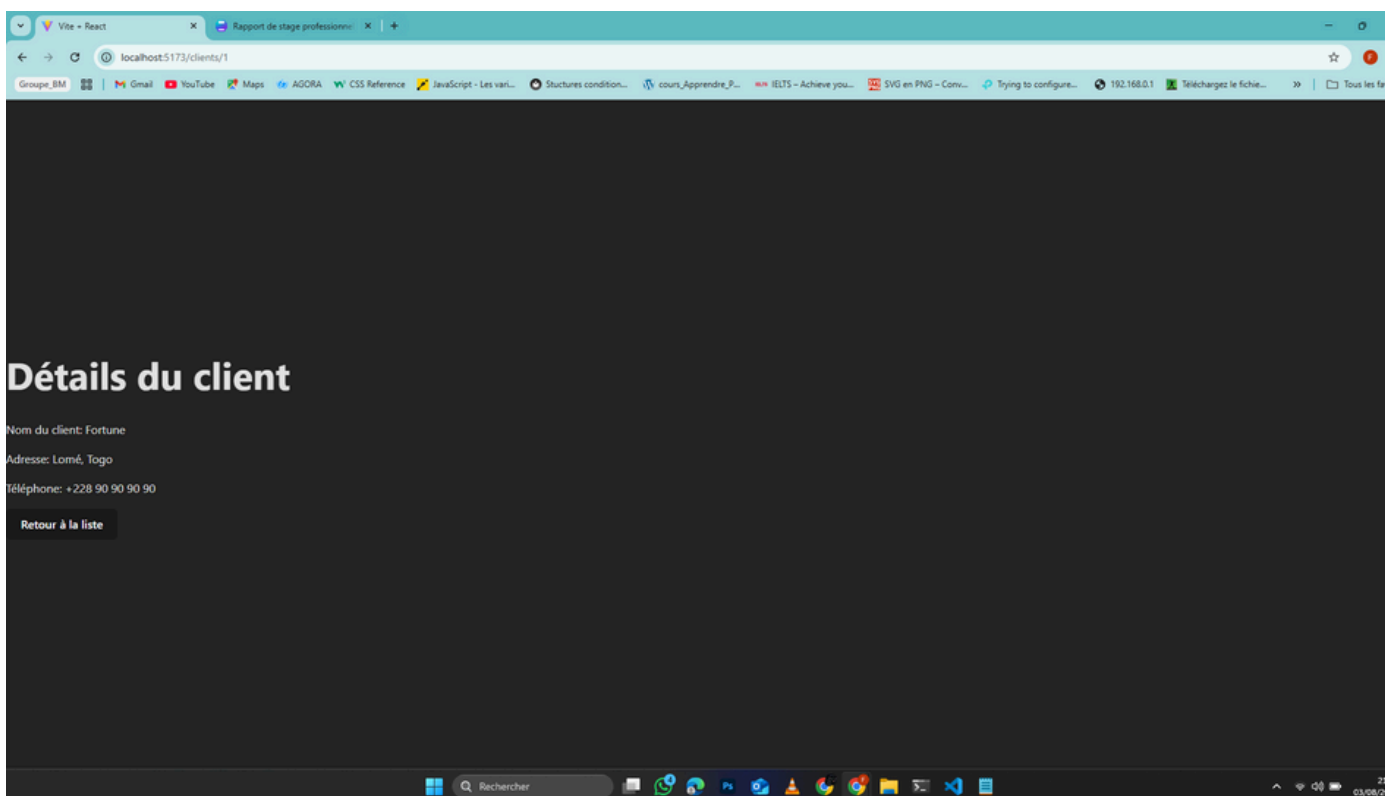


4. IMPLÉMENTATION DES DÉTAILS D'UN CLIENT

- Capture d'écran montrant les détails d'un client spécifique (ClientDetails.jsx) après avoir cliqué sur son nom dans la liste.

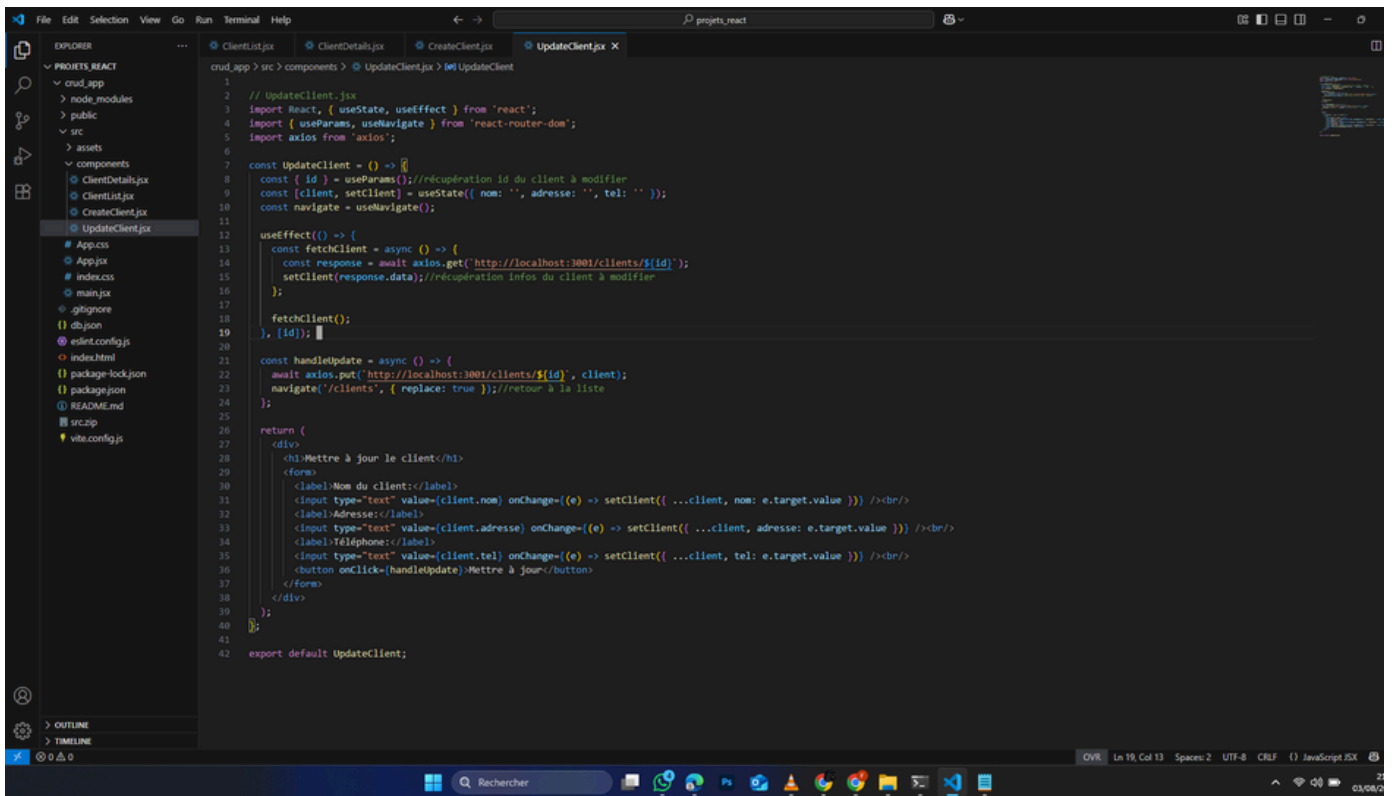


```
1 //ClientDetails.jsx
2 import React, { useState, useEffect } from "react";
3 import { useParams, useNavigate } from "react-router-dom";
4 import axios from "axios";
5
6 const ClientDetails = () => {
7   const { id } = useParams();
8   const [client, setClient] = useState({});
9   const navigate = useNavigate();
10
11   useEffect(() => {
12     const fetchClient = async () => {
13       try {
14         const response = await axios.get("http://localhost:3001/clients/${id}");
15         setClient(response.data);
16       } catch (error) {
17         console.error("Erreur de chargement :", error);
18       }
19     };
20     fetchClient();
21   }, [id]);
22
23   const handleRetour = () => {
24     navigate("/clients");
25   };
26
27   return (
28     <div>
29       <h3>Détails du client</h3>
30       <p>Nom du client: {client.nom}</p>
31       <p>Adresse: {client.adresse}</p>
32       <p>Téléphone: {client.tel}</p>
33       <button onClick={handleRetour}>Retour à la liste</button>
34     </div>
35   );
36 };
37
38 export default ClientDetails;
```

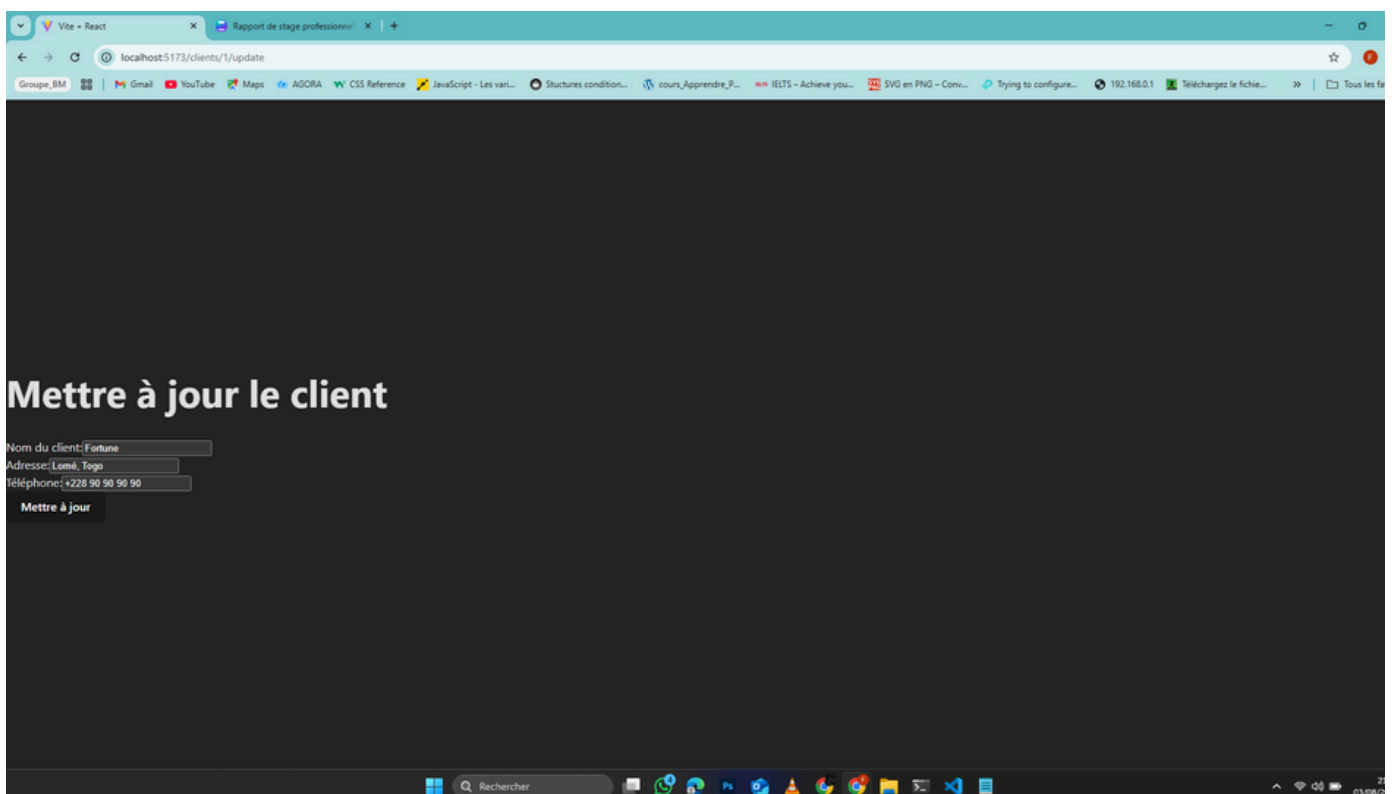


3. IMPLÉMENTATION DE LA MISE À JOUR ET DE LA SUPPRESSION D'UN CLIENT

- Capture d'écran du code UpdateClient.jsx initial.
- Capture d'écran montrant le formulaire de mise à jour d'un client (pré-rempli avec ses informations).

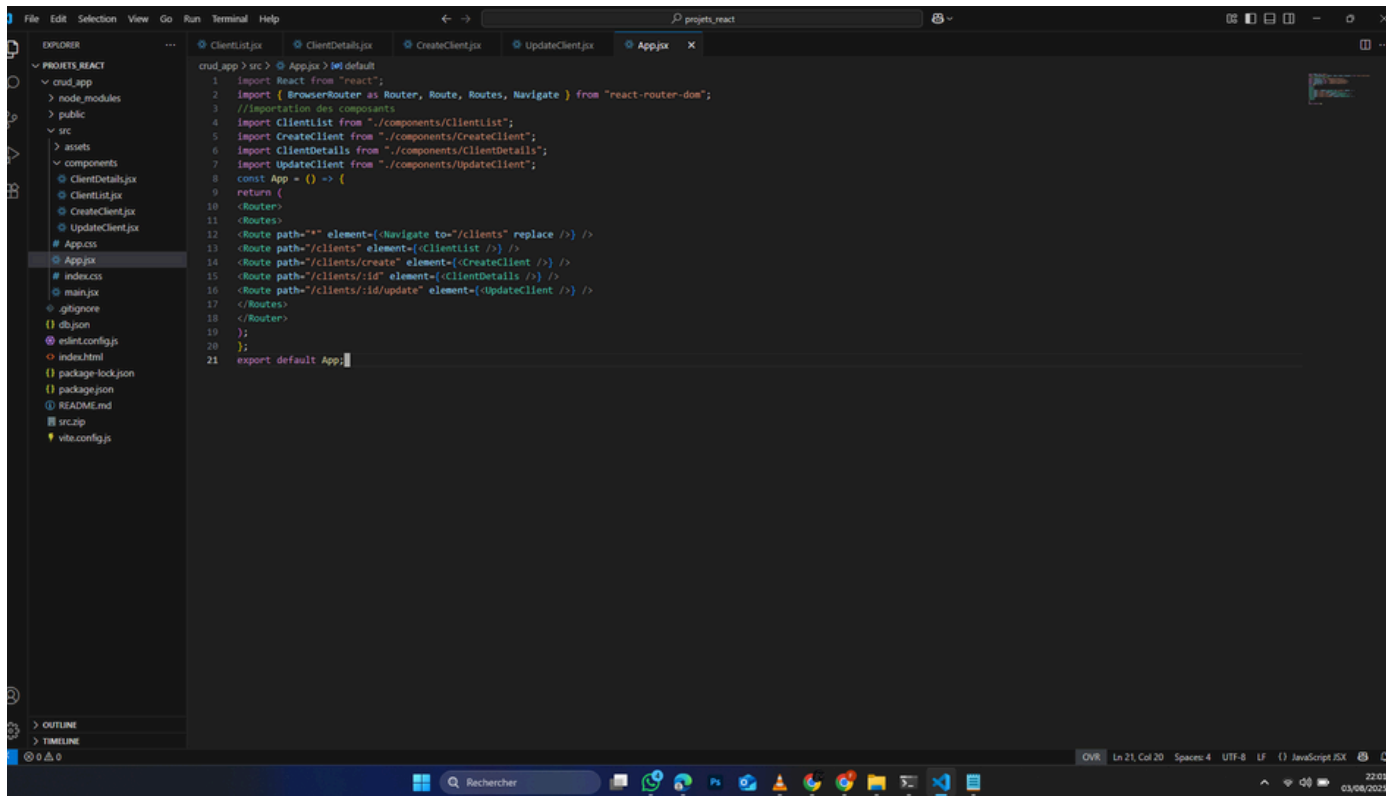


```
1 // UpdateClient.jsx
2 import React, { useState, useEffect } from 'react';
3 import { useParams, useNavigate } from 'react-router-dom';
4 import axios from 'axios';
5
6 const UpdateClient = () => {
7   const { id } = useParams(); // récupération id du client à modifier
8   const [client, setClient] = useState({ nom: '', adresse: '', tel: '' });
9   const navigate = useNavigate();
10
11   useEffect(() => {
12     const fetchClient = async () => {
13       const response = await axios.get('http://localhost:3001/clients/${id}');
14       setClient(response.data); // récupération infos du client à modifier
15     };
16     fetchClient();
17   }, [id]);
18
19   const handleUpdate = async () => {
20     await axios.put('http://localhost:3001/clients/${id}', client);
21     navigate('/clients', { replace: true }); // retour à la liste
22   };
23
24   return (
25     <div>
26       <h1>Mettre à jour le client</h1>
27       <form>
28         <label>Nom du client</label>
29         <input type="text" value={client.nom} onChange={e => setClient({ ...client, nom: e.target.value })} />
30         <label>Adresse</label>
31         <input type="text" value={client.adresse} onChange={e => setClient({ ...client, adresse: e.target.value })} />
32         <label>Téléphone</label>
33         <input type="text" value={client.tel} onChange={e => setClient({ ...client, tel: e.target.value })} />
34         <button onClick={handleUpdate}>Mettre à jour</button>
35       </form>
36     </div>
37   );
38 }
39
40 export default UpdateClient;
```



5. DÉFINITION DES ROUTES ET RÉOLUTION DES ERREURS

- Capture d'écran du code **App.jsx** montrant la configuration des routes.
- Erreur de fichier introuvable (Remplacement du nom du fichier "ClientsList.jsx" en ClientList.jsx" pour la résolution.



The screenshot shows a code editor with the following content:

```
1 import React from "react";
2 import { BrowserRouter as Router, Route, Routes, Navigate } from "react-router-dom";
3 //Importation des composants
4 import ClientList from "../components/ClientList";
5 import CreateClient from "../components/CreateClient";
6 import ClientDetails from "../components/ClientDetails";
7 import UpdateClient from "../components/UpdateClient";
8 const App = () => {
9   return (
10     <Router>
11       <Routes>
12         <Route path="/" element=<Navigate to="/clients" replace /> />
13         <Route path="/clients" element=<ClientList /> />
14         <Route path="/clients/create" element=<CreateClient /> />
15         <Route path="/clients/:id" element=<ClientDetails /> />
16         <Route path="/clients/:id/update" element=<UpdateClient /> />
17       </Routes>
18     </Router>
19   );
20 };
21 export default App;
```

**[plugin:vite:import-analysis] Failed to resolve import
"./components/ClientList" from "src/App.jsx". Does the file exist?**

C:/Users/FORTUNE/Desktop/projets_react/crud_app/src/App.jsx:5:23

```
17 | import React from "react";
18 | import { BrowserRouter as Router, Route, Routes, Navigate } from "react-i
19 | import ClientList from "../components/ClientList";
    |                                     ^
20 | import CreateClient from "../components/CreateClient";
21 | import ClientDetails from "../components/ClientDetails";
```

```
at TransformPluginContext._formatLog (file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/node_modules/vite/dist/node/chunks/dep-0e3a0749.js:112:13)
at TransformPluginContext.error (file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/node_modules/vite/dist/node/chunks/dep-0e3a0749.js:112:13)
at normalizeUrl (file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/node_modules/vite/dist/node/chunks/dep-0e3a0749.js:112:13)
at process.processTicksAndRejections (node:internal/process/task_queues:105:5)
at async file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/node_modules/vite/dist/node/chunks/dep-0e3a0749.js:112:13
at async Promise.all (index 5)
at async TransformPluginContext.transform (file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/node_modules/vite/dist/node/chunks/dep-0e3a0749.js:112:13)
at async EnvironmentPluginContainer.transform (file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/node_modules/vite/dist/node/chunks/dep-0e3a0749.js:112:13)
at async loadAndTransform (file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/node_modules/vite/dist/node/chunks/dep-0e3a0749.js:112:13)
```

6. AMÉLIORATION DE L'INTERFACE AVEC BOOTSTRAP

- Capture d'écran du fichier **main.jsx** avec l'importation de Bootstrap.
- Capture d'écran du rendu final de la liste des clients (ClientList.jsx) avec les styles Bootstrap appliqués (les boutons colorés, le tableau avec des bordures, le conteneur centré).
- Capture d'écran du rendu final de la page de création ou de mise à jour des clients, montrant le formulaire stylisé avec les classes Bootstrap (card, form-control, ...).
- Capture d'écran du rendu final de la page des détails d'un client.

