

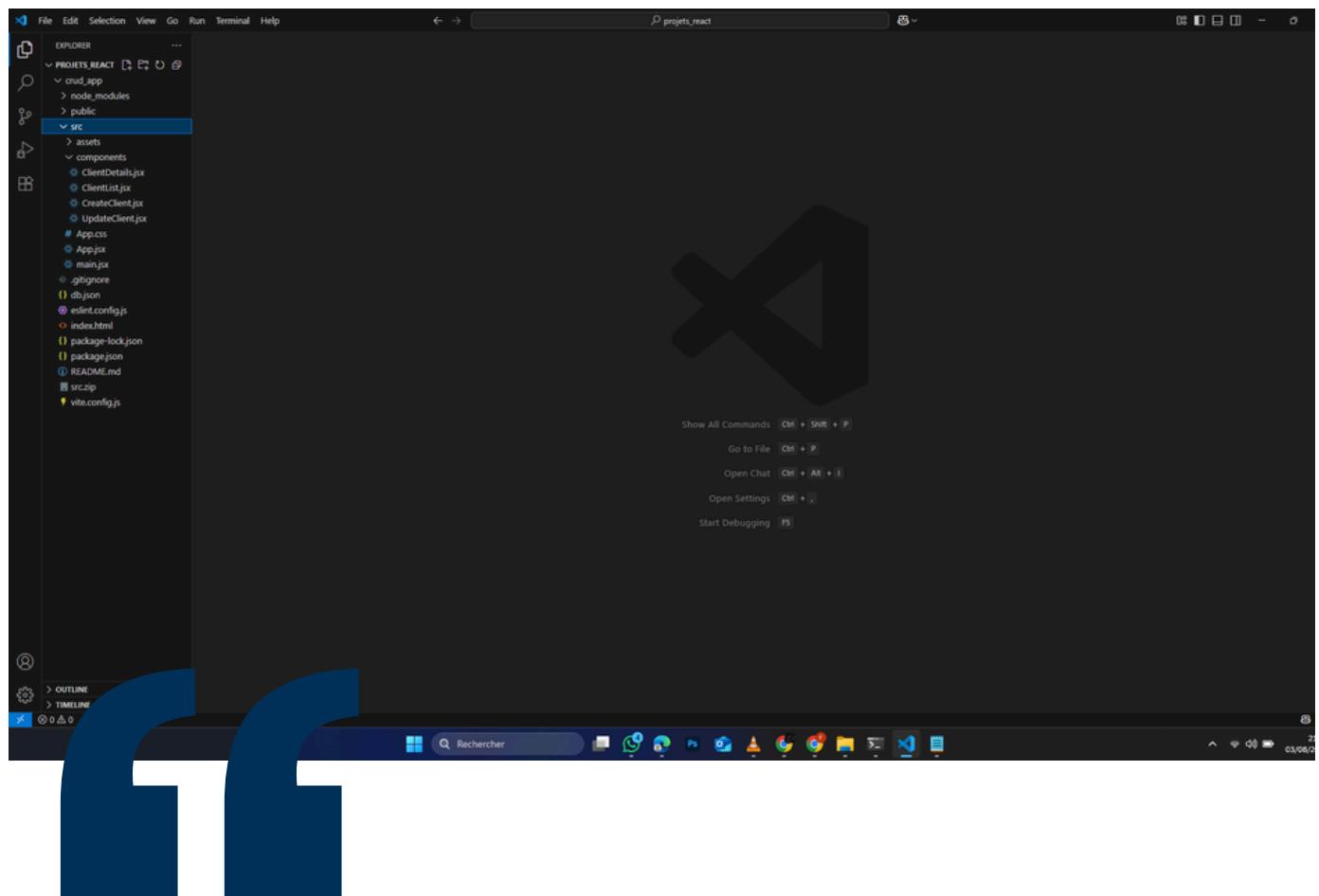
PDF DE RÉALISATION

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**).

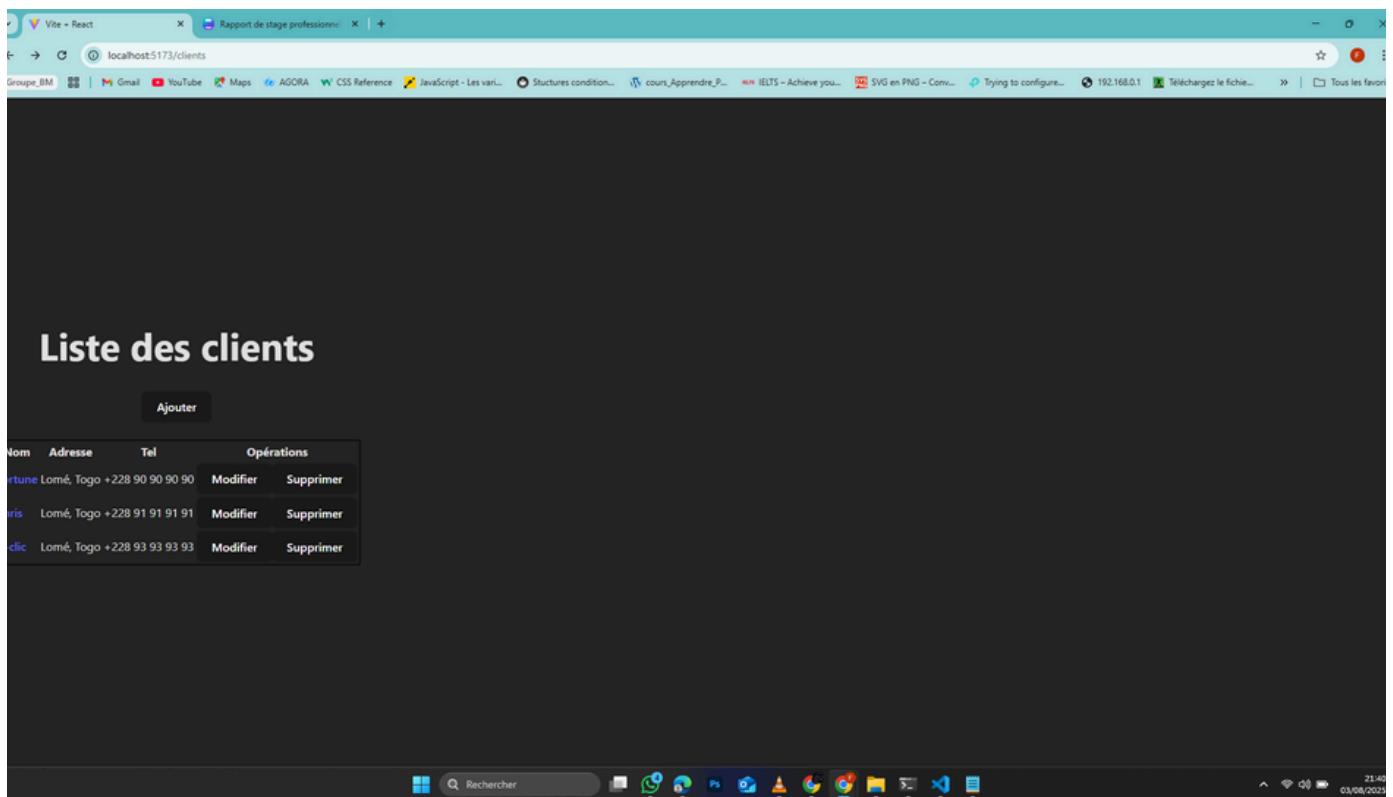


2. IMPLÉMENTATION DU COMPOSANT CLIENTLIST.JSX

- 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.

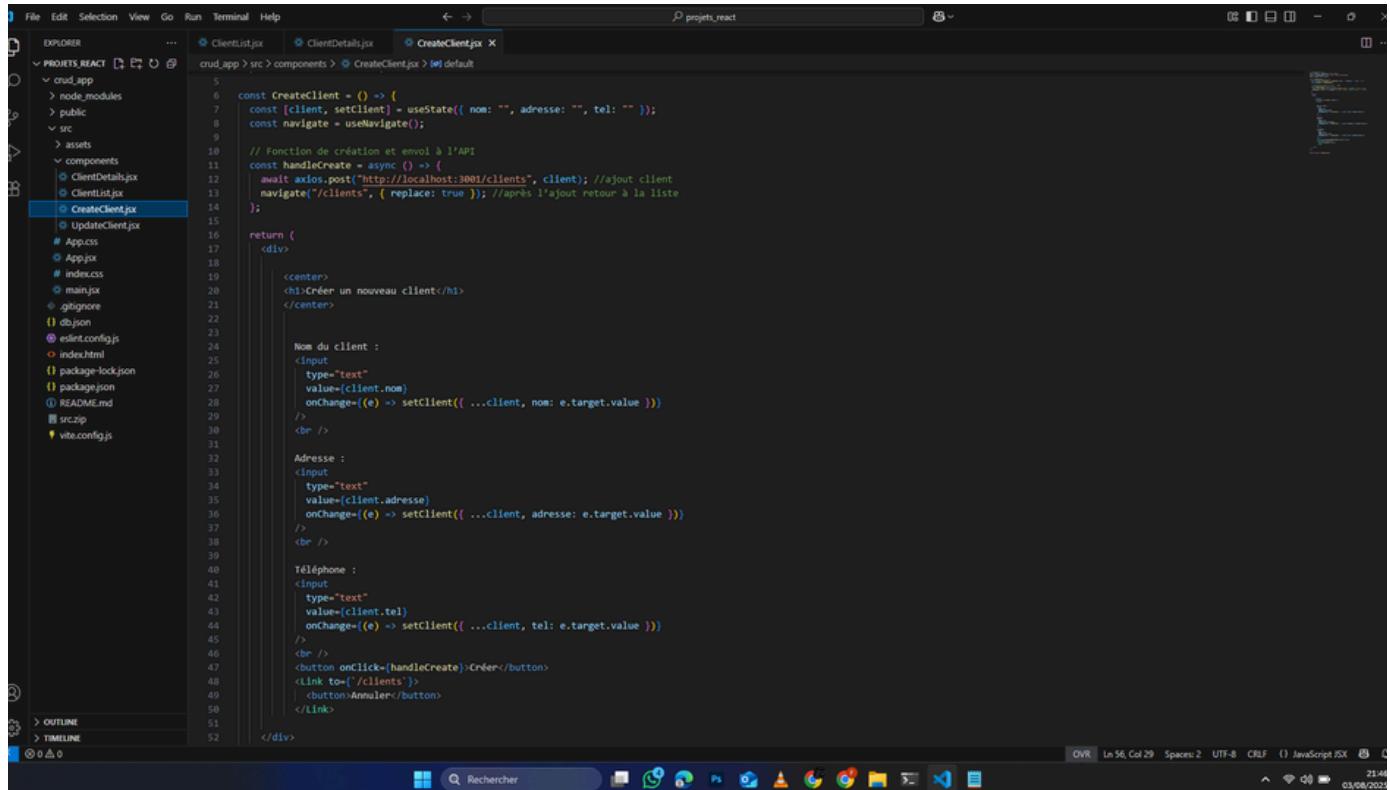
The screenshot shows a Microsoft Visual Studio Code (VS Code) window with the following details:

- File Explorer (Left):** Shows the project structure under "PROJECTS.REACT". The "src" folder contains "crud_app", "node_modules", "public", "assets", and "components". Inside "components", there are files: "ClientDetails.jsx", "ClientList.jsx" (which is currently selected), "CreateClient.jsx", and "UpdateClient.jsx". Other files in "src" include "App.css", "App.jsx", "index.css", "main.jsx", ".gitignore", "db.json", "eslint.config.js", "index.html", "package-lock.json", "package.json", "README.md", "src.zip", and "vite.config.js".
- Code Editor (Center):** Displays the content of "ClientList.jsx". The code uses functional components and hooks like useState and useEffect. It handles data fetching with axios and provides a delete operation via a button. The UI part includes a header with a "Ajouter" button and a table with columns for Nom, Adresse, Tel, and Opérations.
- Terminal (Top):** Shows the command "Run Terminal Help".
- Status Bar (Bottom):** Shows "Line 71, Col 27, Spaces: 4, UTF-8, CR/LF, JavaScript JSX".



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.



The screenshot shows a code editor with the 'CreateClient.jsx' file selected in the left sidebar. The code is a functional component for creating a new client. It uses useState to manage state for name, address, and phone, and axios to post data to a local API. The UI consists of three input fields and two buttons: 'Créer' and 'Annuler'.

```
const CreateClient = () => {
  const [client, setClient] = useState({ nom: "", adresse: "", tel: "" });
  const navigate = useNavigate();

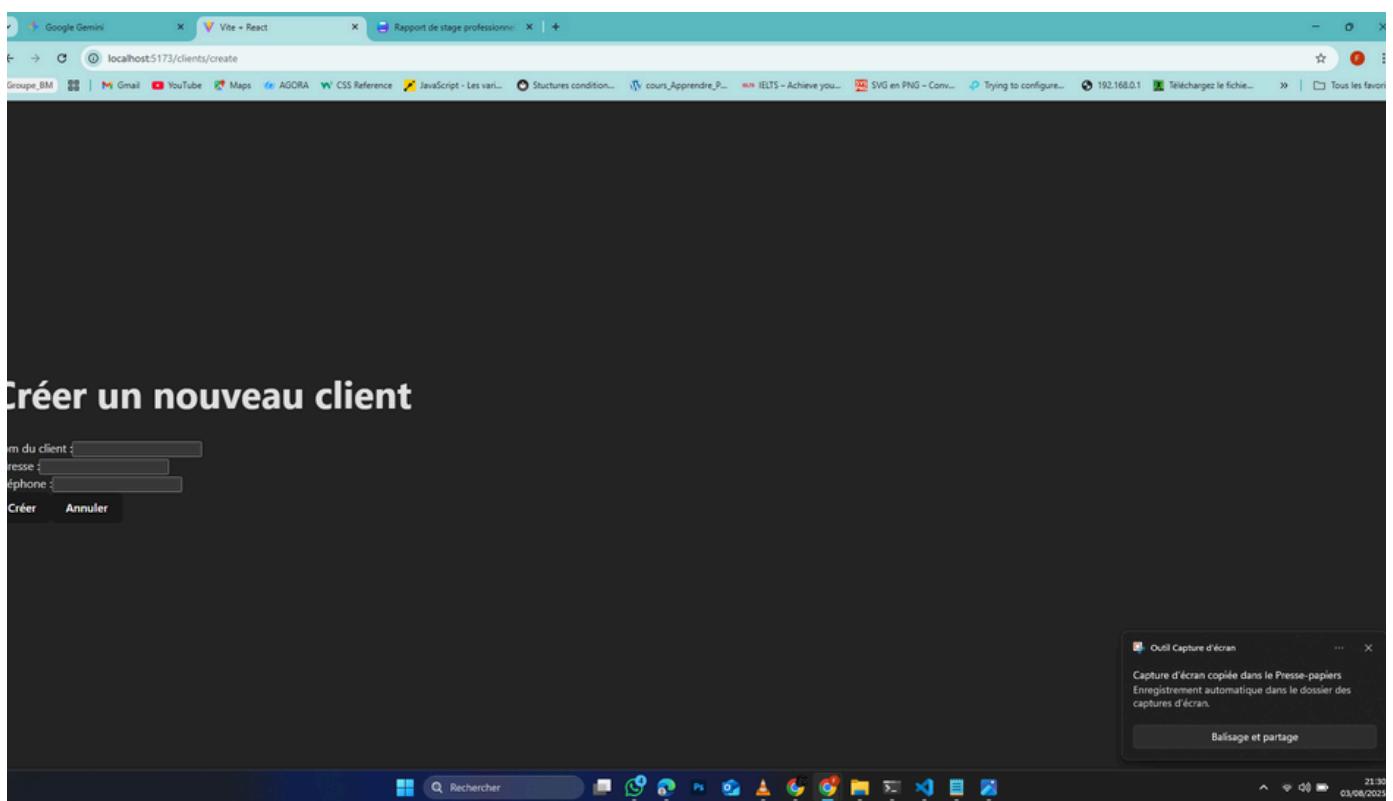
  // Fonction de création et envoi à l'API
  const handleCreate = async () => {
    await axios.post("http://localhost:3001/clients", client); //ajout client
    navigate("/clients", { replace: true }); //après l'ajout retour à la liste
  };

  return (
    <div>
      <center>
        <h1>Créer un nouveau client</h1>
      </center>

      Nom du client :
      <input
        type="text"
        value={client.nom}
        onChange={(e) => setClient({ ...client, nom: e.target.value })}>
      />
      <br />

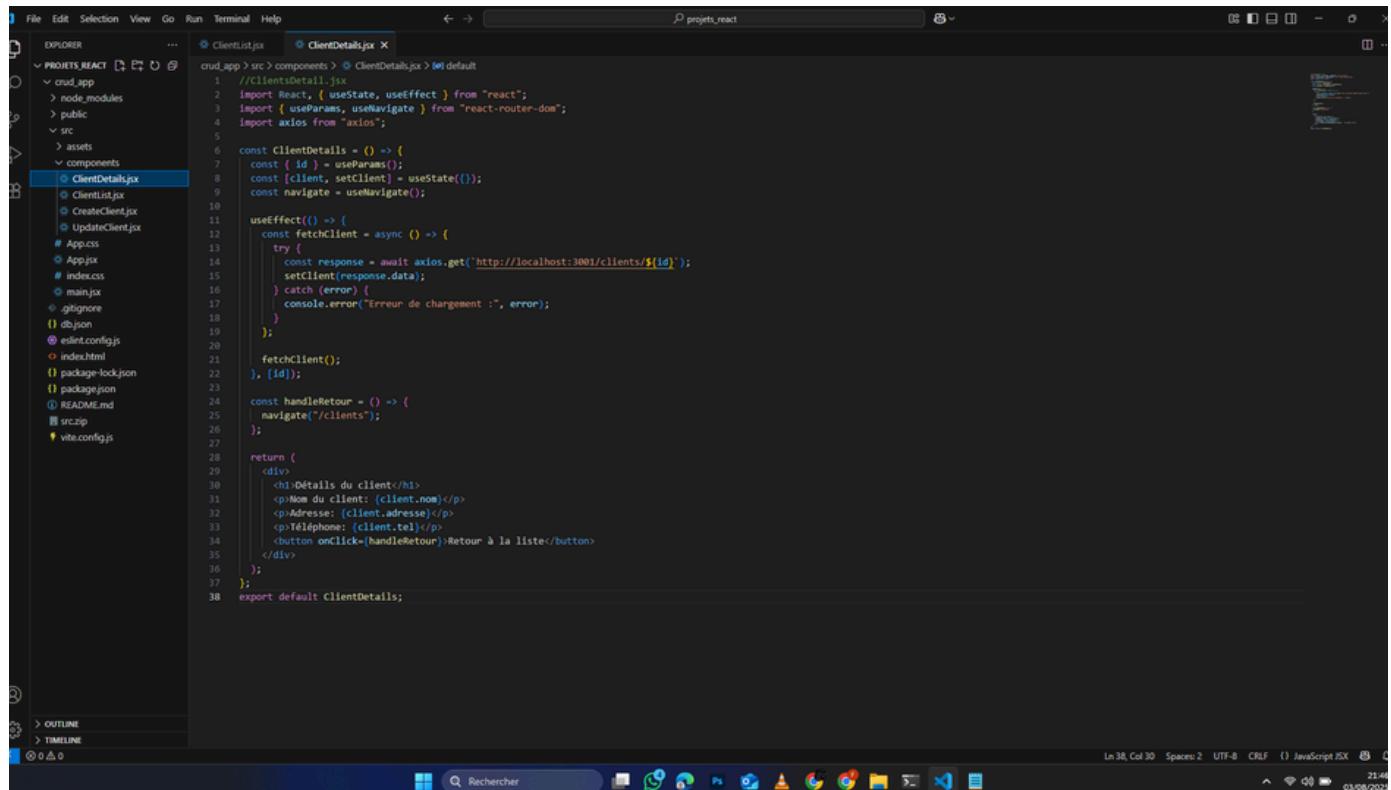
      Adresse :
      <input
        type="text"
        value={client.adresse}
        onChange={(e) => setClient({ ...client, adresse: e.target.value })}>
      />
      <br />

      Téléphone :
      <input
        type="text"
        value={client.tel}
        onChange={(e) => setClient({ ...client, tel: e.target.value })}>
      />
      <br />
      <button onClick={handleCreate}>Créer</button>
      <Link to="/clients"></Link>
      <button>Annuler</button>
    </div>
  );
}
```



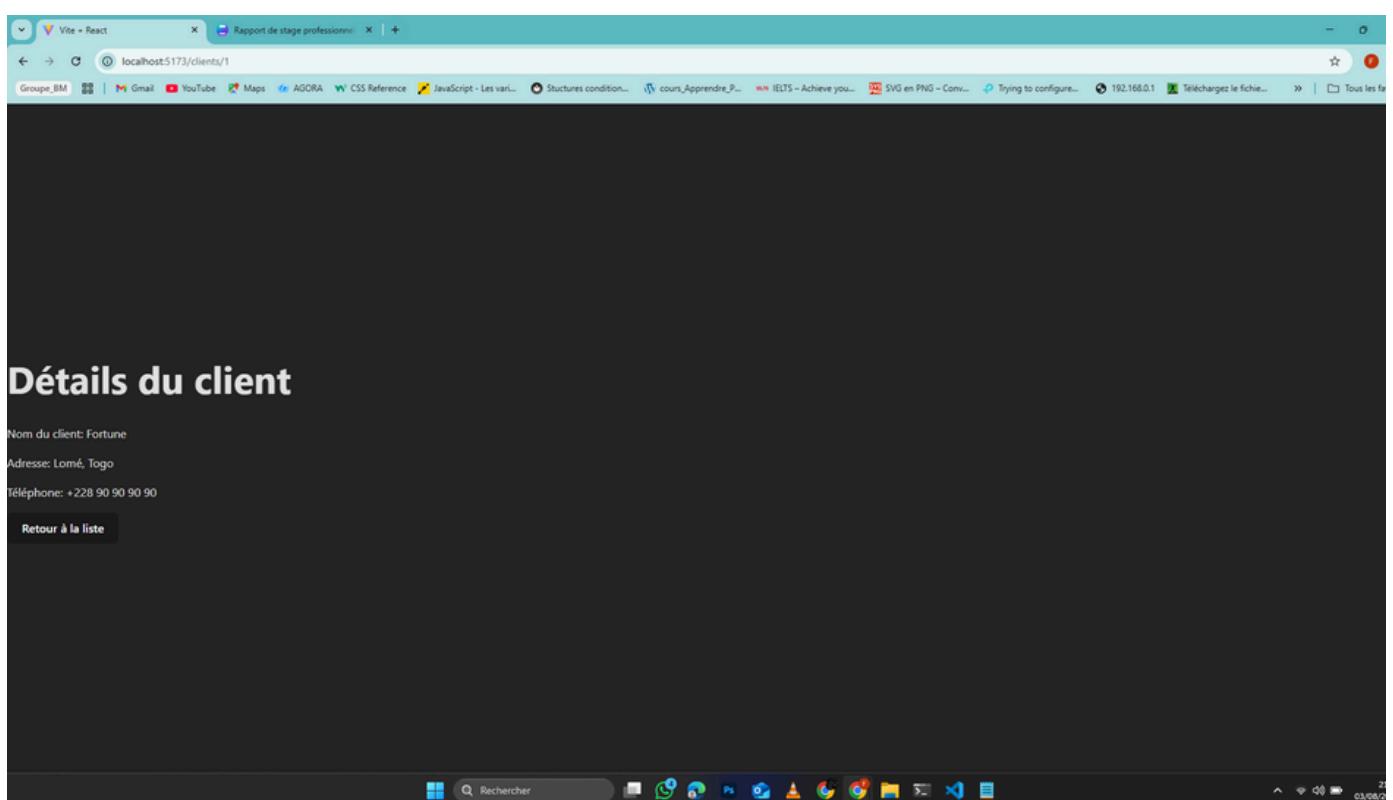
4. IMPLÉMENTATION DES DÉTAILS D'UN CLIENT

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



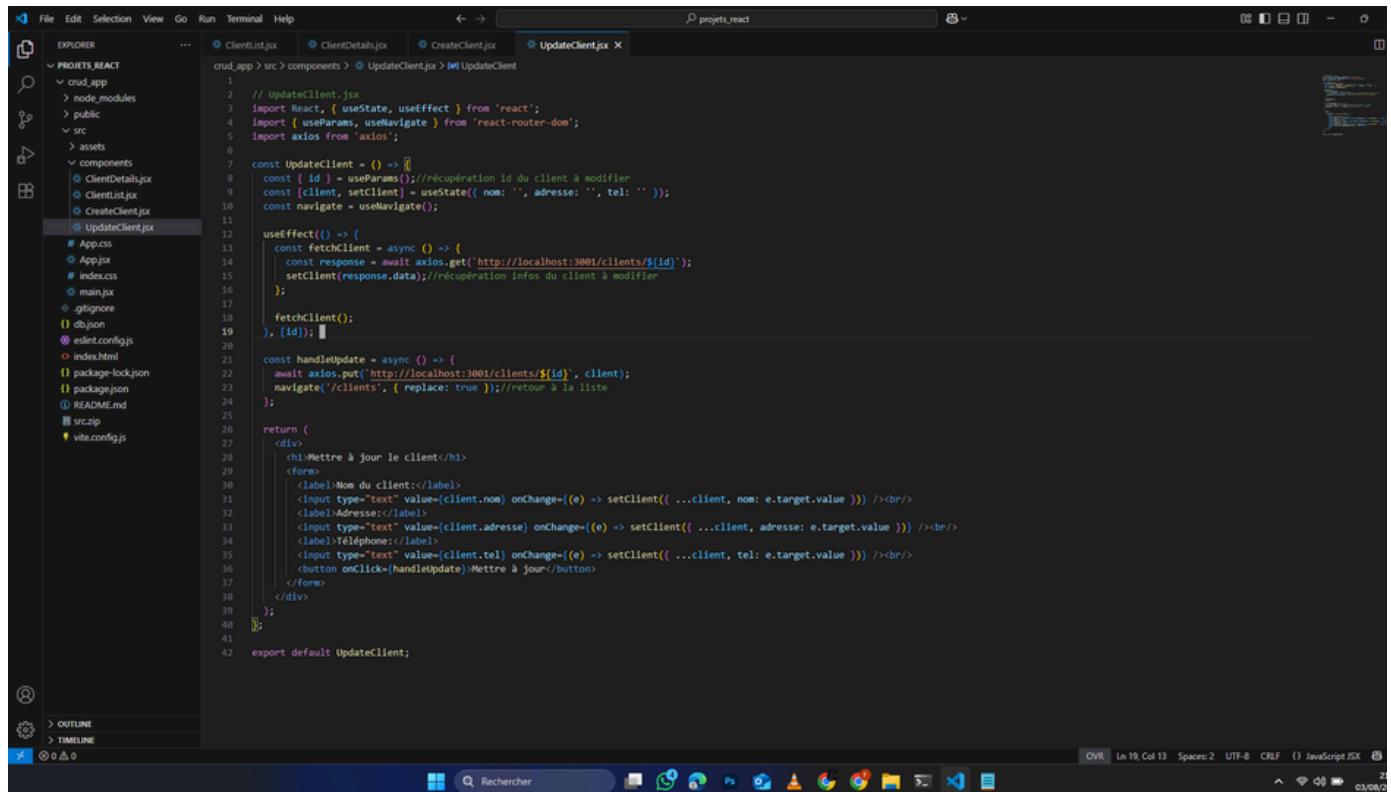
```
File Edit Selection View Go Run Terminal Help
PROJETS_REACT ...
crud_app
  > node_modules
  > public
  > assets
    > components
      ClientList.jsx
      ClientDetails.jsx
      CreateClient.jsx
      UpdateClient.jsx
    # App.css
    # App.jsx
    # index.css
    # main.jsx
    .gitignore
    db.json
    eslint.config.js
    index.html
    package-lock.json
    README.md
    src.zip
    vite.config.js

ClientList.jsx
1 //ClientDetail.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       <h1>Détails du client</h1>
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;
```



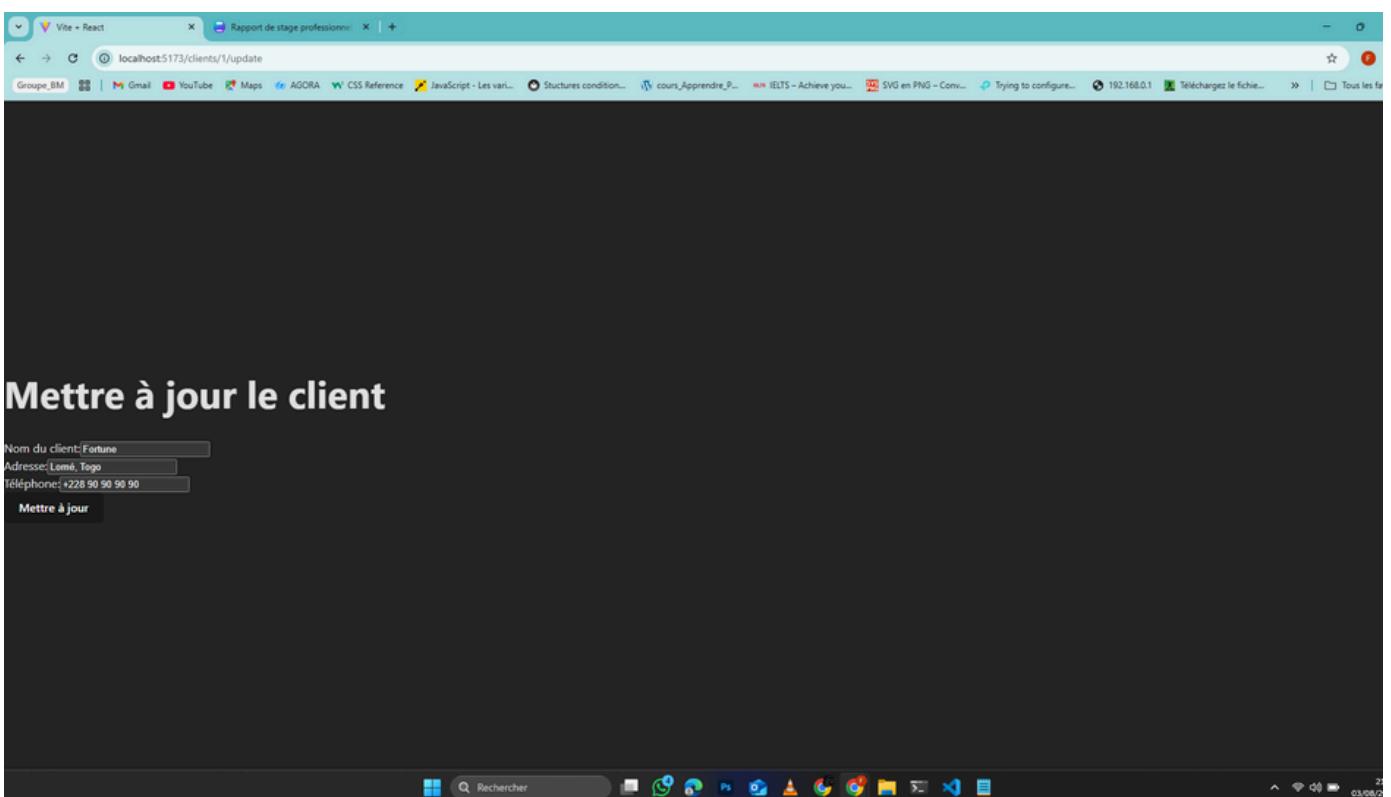
3. IMPLÉMENTATION DE LA MISE À JOUR ET DE LA SUPPRESSION D'UN CLIENT

- 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).



The screenshot shows the Visual Studio Code interface with the 'UpdateClient.jsx' file open in the center editor tab. The file contains React code for updating a client. It uses useState and useEffect hooks, axios for API calls, and a form with three input fields (name, address, phone) to update a client's information in a database. The code is well-structured with comments explaining the logic.

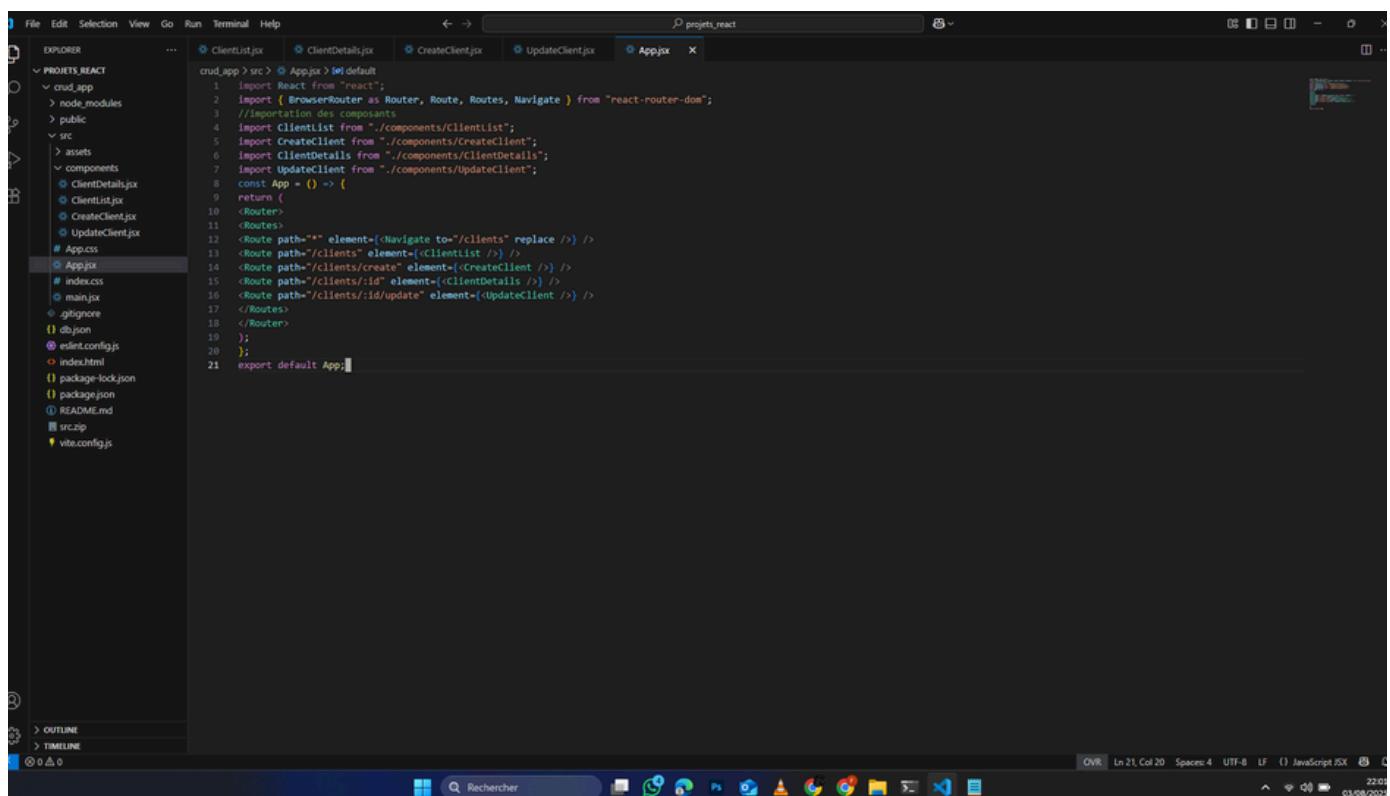
```
1 // UpdateClient.jsx
2 import React, { useState, useEffect } from 'react';
3 import { useParams, useNavigate } from 'react-router-dom';
4 import axios from 'axios';
5
6 const UpdateClient = () => [
7   const { id } = useParams(); // récupération id du client à modifier
8   const [client, setClient] = 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 })} /><br/>
30         <label>Adresse:</label>
31         <input type="text" value={client.adresse} onChange={(e) => setClient({ ...client, adresse: e.target.value })} /><br/>
32         <label>Téléphone:</label>
33         <input type="text" value={client.tel} onChange={(e) => setClient({ ...client, tel: e.target.value })} /><br/>
34         <button onClick={handleUpdate}>Mettre à jour</button>
35       </form>
36     </div>
37   );
38 }
39
40 export default UpdateClient;
```



The screenshot shows a web browser window titled 'Rapport de stage professionnel' with the URL 'localhost:5173/clients/1/update'. The page displays a form titled 'Mettre à jour le client' for updating a client named 'Fortune'. The form has three input fields: 'Nom du client' (value: 'Fortune'), 'Adresse' (value: 'Lomé, Togo'), and 'Téléphone' (value: '+228 90 90 90'). A 'Mettre à jour' button is at the bottom of the form. The browser's status bar shows the URL and the current date and time.

5. DÉFINITION DES ROUTES ET RÉSOLUTION 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.



```
File Edit Selection View Go Run Terminal Help ← → ⌘ projets_react
EXPLORER ... ClientList.jsx ClientDetails.jsx CreateClient.jsx UpdateClient.jsx App.jsx
PROJETS_REACT
crud_app
node_modules
public
src
assets
components
ClientDetails.jsx
ClientList.jsx
CreateClient.jsx
UpdateClient.jsx
App.css
App.jsx
index.css
main.jsx
.githignore
db.json
eslint.config.js
index.html
package-lock.json
package.json
README.md
src.zip
vite.config.js

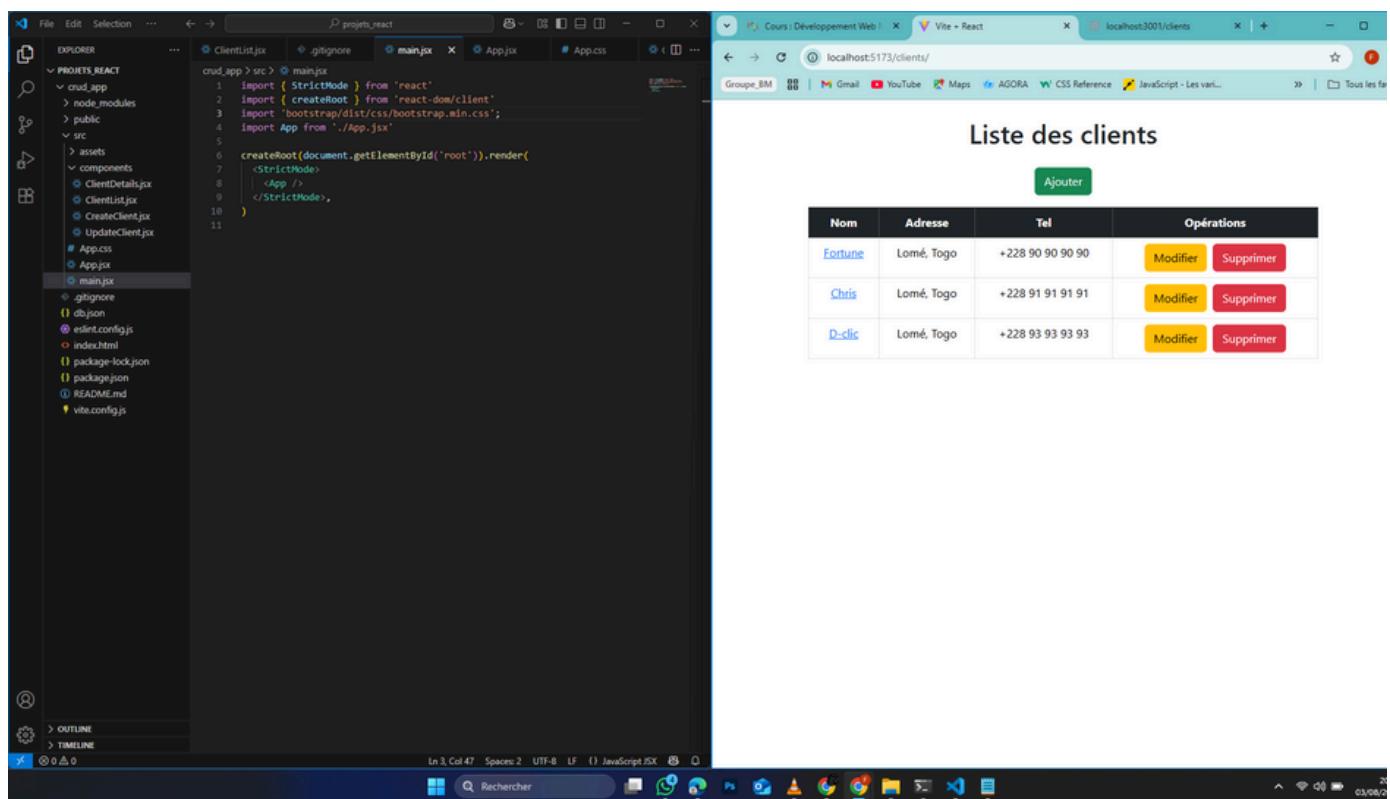
OVR Ln 21, Col 20 Spaces: 4 UTF-8 LF ⓘ JavaScript JSX ⌘ D 22:03
0 0 0
OUTLINE
TIMELINE
Rechercher
C:\Users\FORTUNE\Desktop\projets_react\crud_app\src\App.jsx:5:23
[plugin:vite:import-analysis] Failed to resolve import
"./components/ClientList" from "src/App.jsx". Does the file exist?
C:/Users/FORTUNE/Desktop/projets_react/crud_app/src/App.jsx:5:23
17 | import React from "react";
18 | import { BrowserRouter as Router, Route, Routes, Navigate } from "react-router-dom";
19 | import ClientList from "./components/ClientList";
20 | import CreateClient from "./components/CreateClient";
21 | import ClientDetails from "./components/ClientDetails";
```

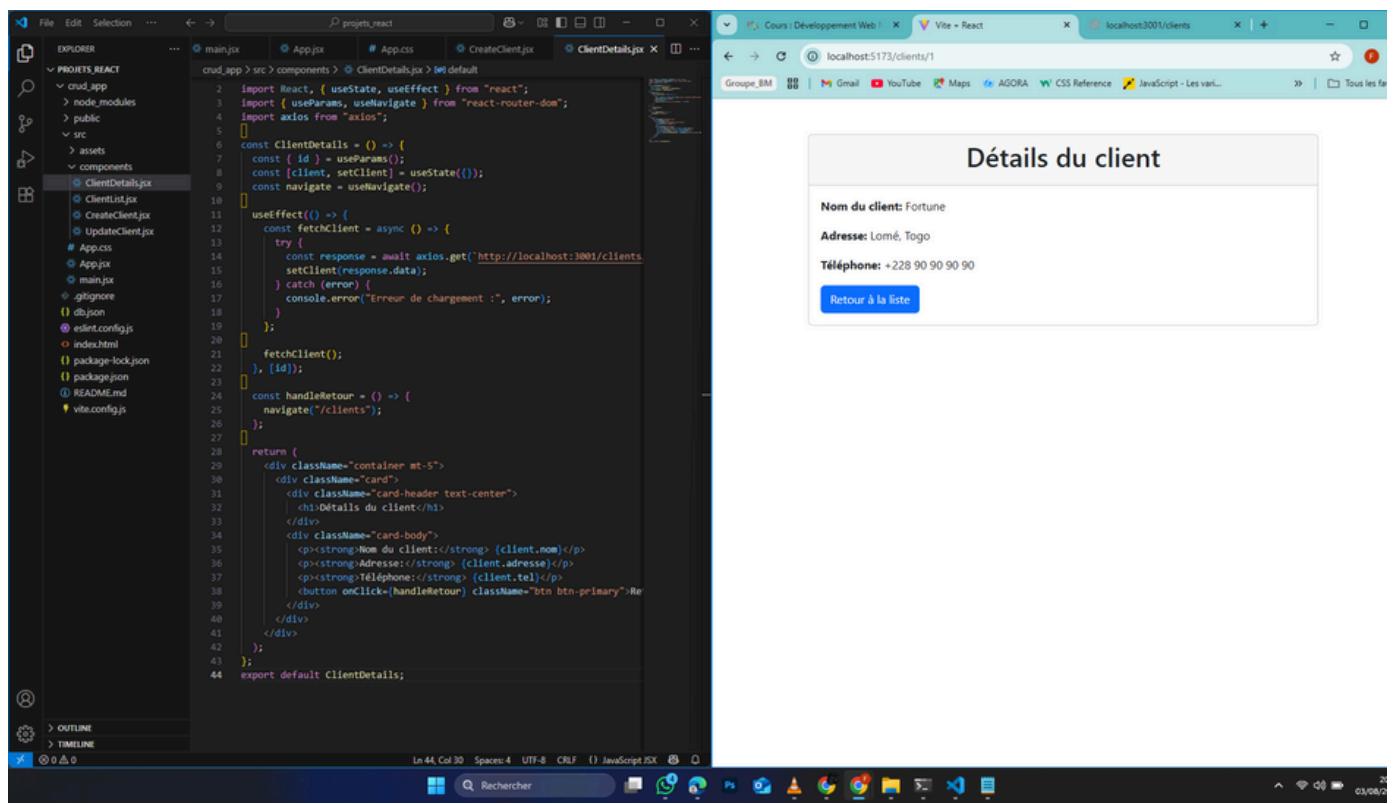
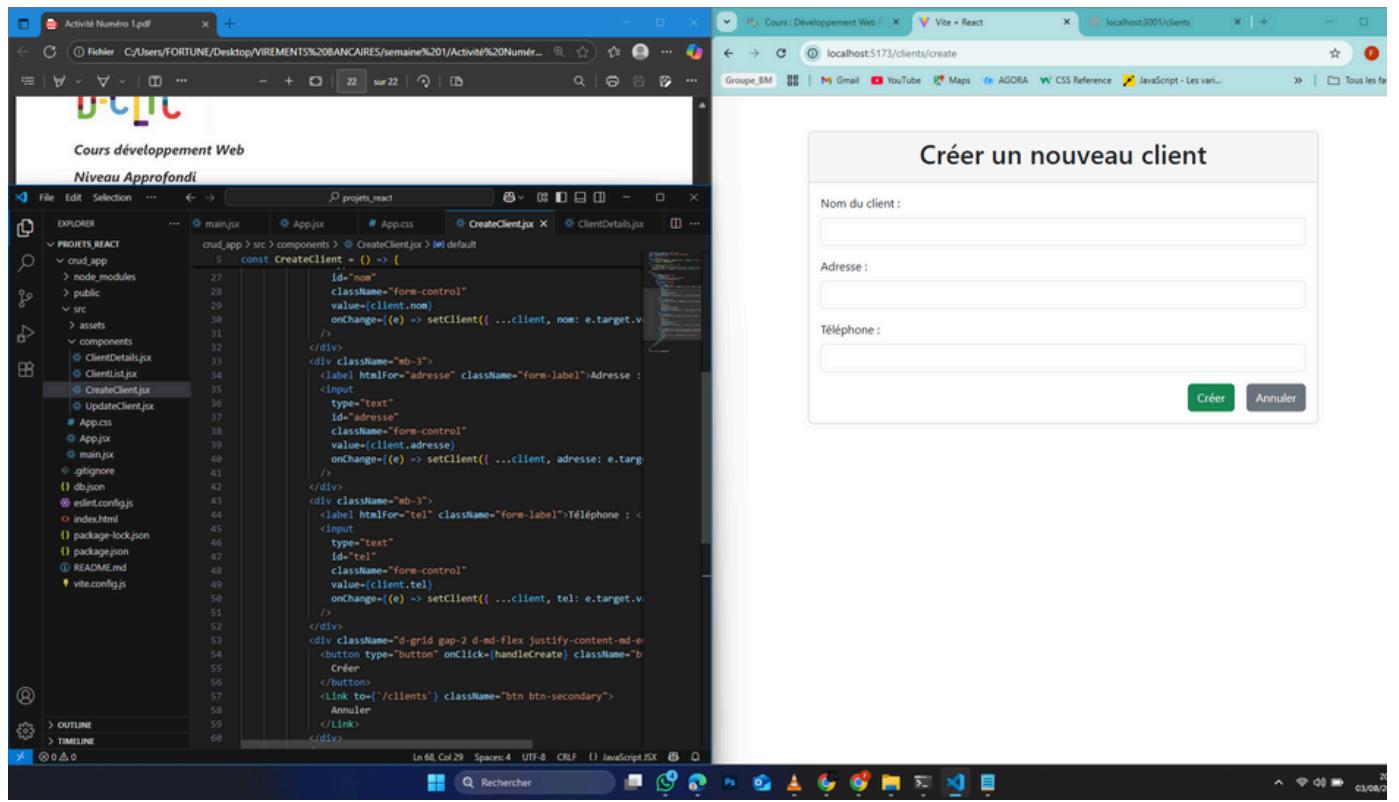
```
[plugin:vite:import-analysis] Failed to resolve import
"./components/ClientList" from "src/App.jsx". Does the file exist?
C:/Users/FORTUNE/Desktop/projets_react/crud_app/src/App.jsx:5:23
17 | import React from "react";
18 | import { BrowserRouter as Router, Route, Routes, Navigate } from "react-router-dom";
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/src/App.jsx:5:23)
at TransformPluginContext.error (file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/node_modules/vite/dist/node_modules/vite-plugin-import/lib/index.js:105:11)
at normalizeUrl (file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/node_modules/vite/dist/node_modules/vite-plugin-import/lib/index.js:105:11)
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_modules/vite-plugin-import/lib/index.js:105:11
at async Promise.all (index 5)
at async TransformPluginContext.transform (file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/src/App.jsx:5:23)
at async EnvironmentPluginContainer.transform (file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/src/App.jsx:5:23)
at async loadAndTransform (file:///C:/Users/FORTUNE/Desktop/projets_react/crud_app/src/App.jsx:5:23)
```

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.





The screenshot shows a development environment with two main windows:

- Code Editor:** On the left, the file `db.json` is open in a code editor. It contains a JSON object with three clients:

```
1 {
  "clients": [
    {
      "id": "1",
      "nom": "Fortune",
      "adresse": "Lomé, Togo",
      "tel": "+228 90 90 90 90"
    },
    {
      "id": "2",
      "nom": "Chris",
      "adresse": "Lomé, Togo",
      "tel": "+228 91 91 91 91"
    },
    {
      "id": "3",
      "nom": "D-clic",
      "adresse": "Lomé, Togo",
      "tel": "+228 93 93 93 93"
    }
  ]
}
```

- Browser Preview:** On the right, a browser window displays a form titled "Mettre à jour le client". The form fields are:
 - Nom du client: Fortune
 - Adresse: Lomé, Togo
 - Téléphone: +228 90 90 90 90A blue button labeled "Mettre à jour" is at the bottom.