Ce projet a été conçu dans le but d'apprendre React, notamment le concept de state et de props.

MUi est utilisé pour la stylisation

# 1. Installation

- npm create vite@latest TOP\_cv\_builder -- --template react

- cd vers le dossier et npm install

- npm install @mui/material @emotion/react @emotion/styled

- npm install modern-css-reset

# 2. Explication syntaxique

## 2.1 Spread Operator

{

...ancienObjet, // copie toutes les propriétés existantes

propriete: nouvelleValeur ou (...objet) // remplace ou ajoute seulement ce qui a changé

}

EX1 :

const person = { name: "John", age: 30 };

// Créer une copie avec un champ modifié

const updatedPerson = { ...person, age: 31 };

console.log(updatedPerson);

// { name: "John", age: 31 }

EX2 :

const updates = { age: 31 };

const merged = { ...person, ...updates };

console.log(merged);

// { name: "John", age: 31 }

## 2.2 Object.entries(obj)

transforme un objet en tableau de paires [clé, valeur].

Exemple :

const personal\_details = {

full\_name: "Diano",

email: "diano@gmail.com"

};

Object.entries(personal\_details);

// [["full\_name", "Diano"], ["email", "diano@gmail.com"]]

Chaque élément du tableau est un sous-tableau [clé, valeur].

Sans crochets { key: value }

const key = "full\_name";

const value = "Diano";

const obj = { key: value };

console.log(obj);

Résultat :

{ key: "Diano" }

Pourquoi ?

Dans un objet { ... }, le nom avant : est littéral, pas une variable.

JavaScript prend exactement ce que tu écris comme nom de propriété → "key"

Il ne regarde pas si tu as une variable key dans le scope

2️⃣ Avec crochets [key]

const key = "full\_name";

const value = "Diano";

const obj = { [key]: value };

console.log(obj);

Résultat :

{ full\_name: "Diano" }

Pourquoi ?

Les crochets [ ] disent à JavaScript : “évalue l’expression ici et utilise sa valeur comme nom de propriété”

Donc [key] devient "full\_name" et l’objet est correct