

Controle 2 dev 203 / Calcule

1. App.js - Point d'entrée principal

Ce composant est le **point central** de l'application. Il gère les **états globaux** et le **routage** entre les pages.

Code

```
import React, { useState } from "react";
import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";
import Accueil from "./Accueil";
import Liste from "./Liste";

function App() {
  // État pour stocker le compteur (valeur actuelle)
  const [compteur, setCompteur] = useState(0);

  // État pour stocker l'historique des opérations
  const [operations, setOperations] = useState([]);

  // Fonction pour effectuer un calcul
  const calculer = (nombre, operation) => {
    const valeurNombre = parseFloat(nombre); // Convertit en nombre réel

    // Calcul du nouveau compteur
    const nouveauCompteur =
      operation === "+" ? compteur + valeurNombre : compteur - valeurNombre;

    setCompteur(nouveauCompteur); // Met à jour le compteur

    // Enregistre l'opération dans l'historique
    const nouvelleOperation = {
      operation,
      nombre: valeurNombre,
      compteur: nouveauCompteur,
    };
    setOperations([...operations, nouvelleOperation]); // Ajoute au tableau des opérations
  };

  return (
    <Router>
    <div>
      {/* Barre de navigation */}
      <nav className="navbar navbar-expand-lg navbar-dark bg-primary">
        <div className="container-fluid">
          <Link className="navbar-brand" to="/">
            Calculeur
          </Link>
          <ul className="navbar-nav me-auto">
```

```

        <li className="nav-item">
          <Link className="nav-link" to="/">
            Accueil
          </Link>
        </li>
        <li className="nav-item">
          <Link className="nav-link" to="/liste">
            Liste
          </Link>
        </li>
      </ul>
    </div>
  </nav>

  { /* Contenu des pages */ }
  <div className="container mt-4">
    <Routes>
      <Route
        path="/"
        element={ <Accueil compteur={compteur} calculer={calculer} /> }
      />
      <Route
        path="/liste"
        element={ <Liste operations={operations} /> }
      />
    </Routes>
  </div>
</div>
</Router>
);
}

export default App;

```

Logique

1. Routage :

- Utilisation de **React Router** pour naviguer entre deux pages :
 - **/** : Page d'accueil (formulaire pour effectuer des calculs).
 - **/liste** : Page d'historique (affiche les opérations effectuées).
- Les liens dans la barre de navigation utilisent **<Link>** pour naviguer sans recharger la page.

2. État Global :

- **compteur** : Stocke la valeur actuelle du compteur.
- **operations** : Stocke un tableau d'objets décrivant chaque opération.

3. Fonction **calculer** :

- Reçoit un nombre et un type d'opération (+ ou -).

- Calcule la nouvelle valeur du compteur et l'ajoute à l'historique des opérations.
- Met à jour les états `compteur` et `operations`.

2. `Accueil.js` - Page principale

Ce composant affiche un formulaire permettant d'effectuer des calculs et d'afficher le compteur actuel.

Code

```
import React, { useState } from "react";

function Accueil({ compteur, calculer }) {
  const [nombre, setNombre] = useState(""); // État pour le nombre saisi
  const [operation, setOperation] = useState("+"); // État pour l'opération choisie

  // Fonction appelée lors du clic sur "Calculer"
  const handleCalculer = () => {
    if (nombre === "") {
      alert("Veuillez entrer un nombre.");
      return;
    }
    calculer(nombre, operation); // Appelle la fonction calculer du parent
    setNombre(""); // Réinitialise le champ après le calcul
  };

  return (
    <div>
      <h1 className="text-center mb-4">Calculateur</h1>
      <div className="card p-4 shadow">
        {/* Saisie du nombre */}
        <div className="mb-3">
          <label className="form-label">Nombre :</label>
          <input
            type="number"
            className="form-control"
            value={nombre}
            onChange={(e) => setNombre(e.target.value)}
          />
        </div>

        {/* Boutons radio pour l'opération */}
        <div className="mb-3">
          <label className="form-label me-2">Opération :</label>
          <div className="form-check form-check-inline">
            <input
              className="form-check-input"
              type="radio"
              value="+"
              checked={operation === "+"}
              onChange={() => setOperation("+")}
            />

```

```

        />
        <label className="form-check-label">+</label>
      </div>
      <div className="form-check form-check-inline">
        <input
          className="form-check-input"
          type="radio"
          value="-"
          checked={operation === "-"}
          onChange={() => setOperation("-")}
        />
        <label className="form-check-label">-</label>
      </div>
    </div>

    {/* Affichage du compteur */}
    <div className="mb-3">
      <h5>
        Valeur du Compteur : <span className="text-success">{compteur}</span>
      </h5>
    </div>

    {/* Bouton pour calculer */}
    <button className="btn btn-primary" onClick={handleCalculer}>
      Calculer
    </button>
  </div>
</div>
);
}

export default Accueil;

```

Logique

1. États Locaux :

- **nombre** : Stocke le nombre saisi par l'utilisateur.
- **operation** : Stocke le type d'opération (+ ou -).

2. Fonction **handleCalculer** :

- Vérifie que le champ **nombre** est rempli.
- Appelle la fonction **calculer** (transmise par le composant parent).
- Réinitialise le champ de saisie après calcul.

3. Affichage :

- Un formulaire avec un champ de saisie (**nombre**) et des boutons radio pour choisir l'opération.
- La valeur actuelle du compteur est affichée en temps réel.

3. Liste.js - Historique des opérations

Ce composant affiche l'historique des opérations sous forme de tableau.

Code

```
import React from "react";

function Liste({ operations }) {
  return (
    <div>
      <h2 className="text-center mb-4">Historique des Opérations</h2>
      <div className="table-responsive">
        <table className="table table-bordered table-striped">
          <thead className="table-dark">
            <tr>
              <th>Opération</th>
              <th>Nombre</th>
              <th>Valeur du Compteur</th>
            </tr>
          </thead>
          <tbody>
            {operations.map((op, index) => (
              <tr key={index}>
                <td>{op.operation}</td>
                <td>{op.nombre}</td>
                <td>{op.compteur}</td>
              </tr>
            ))}
          </tbody>
        </table>
      </div>
    </div>
  );
}

export default Liste;
```

Logique

1. Propriété **operations** :

- Reçoit l'historique des opérations via les **props**.

2. Affichage Dynamique :

- Parcourt le tableau **operations** avec **map**.
 - Affiche chaque opération comme une ligne dans un tableau.
-

Cycle Global

1. Page **Accueil** :

- L'utilisateur saisit un nombre, sélectionne une opération, et clique sur **Calculer**.
- Le compteur et l'historique sont mis à jour dans **App.js**.

2. Page **Liste** :

- L'utilisateur consulte l'historique sous forme de tableau.
-