**Documentation**

**Use css variables with Styled –components :**

Import themeprovider :

// Theme provider for css variable styled component

import { ThemeProvider } from "styled-components";

dans App.jsx or tsx

const App = () => {

  return (

    <>

      <ThemeProvider theme={theme}>

        <BrowserRouter>

          <Routes>

            <Route path="/" element={<Home />} />

            <Route path="/home" element={<Home />} />

          </Routes>

        </BrowserRouter>

      </ThemeProvider>

      <GlobalStyle />

    </>

  );

};

**Dans le fichier global-styles.ts, definer le props theme :**

import { createGlobalStyle } from "styled-components";

const GlobalStyle = createGlobalStyle`

@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@100;300;400;500;600&display=swap');

\*{

    margin: 0;

    padding: 0;

    box-sizing: border-box;

    font-family: 'Poppins', sans-serif;

    text-decoration: none;

}

`;

export const theme = {

  colors: {

    GrayBgColor: "#f3f3f3",

  },

};

export default GlobalStyle;

**Appel dans votre wrapper.ts comme props**

export const FirstFooterWrapper = styled.div`

  padding: 3rem 9%;

  /\* background-color: #f3f3f3; \*/

  background-color: ${({ theme }) => theme.colors.GrayBgColor};

  .box-container {

    display: flex;

    align-items: center;

    justify-content: space-around;

  }

**React pagination :**

**Installation :**

npm i react-paginate

**Importation :**

import ReactPaginate from "react-paginate";

**Utilisation :**

<ReactPaginate

          previousLabel={"Précédent"}

          nextLabel={"Suivant"}

          breakLabel={"..."}

          pageCount={10}

          marginPagesDisplayed={4}

          onPageChange={handlePageClick}

          containerClassName={"pagination"}

          pageClassName={"page-item"}

          pageLinkClassName={"page-link"}

          previousClassName={"page-ps"}

          previousLinkClassName={"page-link"}

          nextClassName={"page-ps"}

          nextLinkClassName={"page-link"}

          breakClassName={"page-item"}

          breakLinkClassName={"page-link"}

          activeClassName={"active"}

          disabledClassName={"unactive"}

        />

**FakeRestAPI :**

https://jsonplaceholder.typicode.com/

**installation :**

npm install -g json-server

**Alternative port**

You can start JSON Server on other ports with the --port flag:

$ json-server --watch db.json --port 3004

**Remove duplicate values :**

One way to acheive that could be to use the [Set](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Set) API after mapping your array :

{Array.from(new Set(this.state.statistic\_table.map(obj => obj.period))).map(period => {

return <option value={period}>{period}</option>

})}

And then converting it back to an array using Array.from to map your component with it