Olvidaste el proyecto base   
ARCHIvos relevantes. Por favor toma se objetivo y crea este proyecto muy bien, no te olvides de los stylos, es muy importante, deben verse y funcionar igual   
con las dependencias y versiones par que funcione

Entonces conecta la api ya creada a este proyecto cliente. Gracias te respeto

import { FC, createContext, useState } from 'react';

import PocketBase from 'pocketbase';

import './style.css';

export interface Tareas {

  id: string;

  collectionId: string;

  collectionName: string;

  created: string;

  updated: string;

  nombre: string;

  estado: 'creada' | 'iniciada' | 'completada';

}

type ContextoContextType = {

  autenticar: () => Promise<void>;

  obtenerTareas: () => Promise<void>;

  eliminarTarea: (id: string) => void;

  agregarTarea: (textoTarea: string) => Promise<void>;

  actualizarTarea: (id: string, nuevoEstado: 'creada' | 'iniciada' | 'completada') => Promise<void>;

  tareas: Tareas[];

  pb: PocketBase;

};

export const ContextoContext = createContext<ContextoContextType>({} as ContextoContextType);

export const ContextoProvider: FC<{ children: React.ReactNode }> = ({ children }) => {

  const [tareas, setTareas] = useState<Tareas[]>([]);

  const pb = new PocketBase('http://127.0.0.1:8090');

  pb.autoCancellation(false);

  const autenticar = async () => {

    try {

      const authData = await pb.collection('users').authWithPassword('usuario', 'K-0ppbvY2HNIe6S');

      console.log('Autenticación exitosa:', authData);

    } catch (error) {

      console.error('Error de autenticación:', error);

    }

  };

  const obtenerTareas = async () => {

    try {

      const listaTareas = await pb.collection<Tareas>('Tareasv2').getFullList();

      setTareas(listaTareas);

    } catch (error) {

          console.error('Error al obtener las tareas:', error);

    }

  };

  const eliminarTarea = async (id: string) => {

    await pb.collection<Tareas>('Tareasv2').delete(id);

    obtenerTareas();

  };

  const agregarTarea = async (textoTarea: string) => {

    const nuevaTarea = await pb.collection<Tareas>('Tareasv2').create({ nombre: textoTarea, estado: 'creada' });

    if (nuevaTarea) {

      setTareas([...tareas, nuevaTarea]);

    }

  };

  const actualizarTarea = async (id: string, nuevoEstado: 'creada' | 'iniciada' | 'completada') => {

    const tareaActualizada = await pb.collection<Tareas>('Tareasv2').update(id, { estado: nuevoEstado });

    if (tareaActualizada) {

      const nuevasTareas = tareas.map((tarea) => (tarea.id === id ? tareaActualizada : tarea));

      setTareas(nuevasTareas);

    }

  };

  return (

    <ContextoContext.Provider

      value={{ autenticar, eliminarTarea, obtenerTareas, agregarTarea, actualizarTarea, tareas, pb }}

    >

      {children}

    </ContextoContext.Provider>

  );

};

body {

  background-color: #757579;

  font-family: Arial, sans-serif;

}

.contenedor-principal {

  background-color: rgb(66, 66, 64);

  border: 2px solid rgb(191, 196, 188);

  border-radius: 5px;

  padding: 20px;

  margin: 20px auto;

  max-width: 100%;

  text-decoration: rgb(248, 248, 2);

  }

.listas {

  display: flex;

  justify-content: center;

  flex-wrap: wrap;

}

.lista-1,

.lista-2,

.lista-3 {

  width: 20%;

  border: 2px solid #ccc;

  border-radius: 5px;

  padding: 10px;

  margin: 10px;

}

.lista-1 {

  background-color: #2b37e7;

}

.lista-2 {

  background-color: #b8b807;

}

.lista-3 {

  background-color: #285e28;

}

h1,

h3 {

  text-align: center;

  text-decoration: beige;

}

ul {

  list-style-type: none;

  padding: 0;

  flex-wrap: wrap;

}

li {

  background-color: #f0f8ff;

  border: 1px solid #4682b4;

  border-radius: 5px;

  padding: 20px;

  margin: 5px 0;

  display: flex;

  justify-content: space-between;

  align-items: center;

  flex-wrap: wrap;

}

form {

  display: flex;

  justify-content: center;

  margin-bottom: 20px;

   align-items: center;

    justify-content: center;

  }

input[type="text"] {

  padding: 25px;

  border-radius: 5px;

  border: 1px solid #ccc;

  flex: 0.5;

  margin-right: 10px;

  width: 150%;

  background-color: rgba(192, 199, 184, 0.897);

 }

button {

  padding: 8px 16px;

  border-radius: 5px;

  border: none;

  cursor: pointer;

}

button.btn-agregar {

  background-color: #87ceeb;

  color: white;

  margin-left: 175px;

  padding: 20px;

  cursor: pointer;

}

button.btn-eliminar {

  background-color: #eba3a3;

  color: white;

  border: 1px solid #c40404;

  cursor: pointer;

  }

  .btn-agregar:hover {

    background-color: #4682b4;

    color: white;

  }

  .btn-eliminar:hover {

    background-color: #f11111;

    color: white;

  }

select {

  padding: 6px;

  border-radius: 5px;

  border: 1px solid #ccc;

  margin-left: 10px;

  background-color: #fff;

  color: #333;

  cursor: pointer;

  width: 40%;

}

import React, { useContext, useState } from 'react';

import { ContextoContext } from './Contexto';

const TareaForm: React.FC = () => {

  const { agregarTarea } = useContext(ContextoContext);

  const [textoTarea, setTextoTarea] = useState('');

  const handleChange = (e: React.ChangeEvent<HTMLInputElement>) => {

    setTextoTarea(e.target.value);

  };

  const handleSubmit = async (e: React.FormEvent<HTMLFormElement>) => {

    e.preventDefault();

    if (!textoTarea.trim()) return;

    try {

      await agregarTarea(textoTarea);

      setTextoTarea('');

    } catch (error) {

      console.error('Error al agregar la tarea:', error);

    }

  };

  return (

    <form className="d-flex align-items-stretch" onSubmit={handleSubmit}>

      <div className='caja'>

      <input type="text" placeholder="Ingresar Nueva tarea" value={textoTarea} onChange={handleChange} />

      </div>

      <button className="btn-agregar"  type="submit">+</button>

    </form>

  );

};

export default TareaForm;

import React, { useContext, useEffect } from 'react';

import { ContextoContext, Tareas } from './Contexto';

interface TareaListaProps {

  estado: 'creada' | 'iniciada' | 'completada';

}

const TareaLista: React.FC<TareaListaProps> = ({ estado }) => {

  const { eliminarTarea, obtenerTareas, actualizarTarea, tareas, autenticar, pb } = useContext(ContextoContext);

  useEffect(() => {

    if (!pb.authStore.isValid) {

      autenticar();

      obtenerTareas();

    } else {

      obtenerTareas();

    }

  }, []);

  const handleSelectChange = async (tarea: Tareas, nuevoEstado: 'creada' | 'iniciada' | 'completada') => {

    await actualizarTarea(tarea.id, nuevoEstado);

    obtenerTareas();

  };

  return (

   <ul >

      {tareas.filter((tarea) => tarea.estado === estado).map((tarea) => (

        <li key={tarea.id}>

          {tarea.nombre}

          <select className=''

            value={tarea.estado}

            onChange={(e) => handleSelectChange(tarea, e.target.value as 'creada' | 'iniciada' | 'completada')}

          >

            <option  value="creada">creada</option>

            <option value="iniciada">iniciada</option>

            <option value="completada">completada</option>

          </select>

          <button className="btn-eliminar" onClick={() => eliminarTarea(tarea.id)}>X</button>

        </li>

      ))}

    </ul>

  );

};

export default TareaLista;

import { FC } from "react"

import App from "../App"

const Wrapper:FC = () => {

    return(

            <App/>

    )

}

export {Wrapper}

import React from 'react';

import { render, screen } from '@testing-library/react';

import App from './App';

test('renders learn react link', () => {

  render(<App />);

  const linkElement = screen.getByText(/learn react/i);

  expect(linkElement).toBeInTheDocument();

});

import React, { useContext, useEffect, useState } from 'react';

*// import './style.css';*

import TareaForm from './components/TareaForm';

import TareaLista from './components/TareaLista';

import { ContextoProvider } from './components/Contexto';

const App: React.FC = () => {

  return (

    <ContextoProvider>

      <div className="contenedor-principal">

        <h1>Lista de Tareas: Mayte</h1>

        <TareaForm />

        <div className="listas">

          <div className="lista-1">

            <h3>Tarea Creada</h3>

            <TareaLista estado="creada" />

          </div>

          <div className="lista-2">

            <h3>Tarea Iniciada</h3>

            <TareaLista estado="iniciada" />

          </div>

          <div className="lista-3">

            <h3>Tarea completada</h3>

            <TareaLista estado="completada" />

          </div>

        </div>

      </div>

    </ContextoProvider>

  );

};

export default App;

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

ReactDOM.render(

  <React.StrictMode>

    <App />

  </React.StrictMode>,

  document.getElementById('root'))