

Tenho o seguinte projeto 'frontend' feito em 'React':

Arquivo 'Public': contendo imagens;

Diretório 'src', contendo os seguintes arquivos e pastas:

arquivo 'App.jsx' com a seguinte configuração:

```
// src/App.js
import React from 'react';
import AppRouter from './AppRouter';

function App() {
  return (
    <div>
      <AppRouter />
    </div>
  );
}

export default App;
```

arquivo 'AppRouter.jsx' com a seguinte configuração:

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Simule from './components/Simule';
import Resultado from './components/Resultado';

function AppRouter() {
  return (
    <Router>
      <Routes>
        <Route path="/simule" element={<Simule />} />
        <Route path="/resultado" element={<Resultado />} />
      </Routes>
    </Router>
  );
}

export default AppRouter;
```

arquivo 'main.jsx' com a seguinte configuração:

```
import React from 'react'
import ReactDOM from 'react-dom'
import App from './App.jsx'
import './index.css'
```

```
ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  document.getElementById('root')
);
```

arquivo ‘index.html’ com a seguinte configuração:

```
<!doctype html>
<html lang="pt-br">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Educação Financeira</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

pasta ‘components’ com os seguintes arquivos e configurações:

componente ‘Simule.jsx’, com a seguinte configuração:

```
import React, { useState } from 'react';
import { useNavigate } from 'react-router-dom'; // Importe useNavigate

function SimulePage() {
  const navigate = useNavigate(); // Obtenha o navigate

  const [mesAno, setMesAno] = useState([]);
  const [salario, setSalario] = useState([]);

  const adicionarLinha = () => {
    setMesAno([...mesAno, '']);
    setSalario([...salario, '']);
  };

  const handleNavigationToResultado = () => {
    navigate('/resultado'); // Use navigate para ir para a página de resultado
  };

  return (
    <div>
      <h1>Simule: calcule sua aposentadoria aqui!</h1>
      <blockquote className="item">
```


A partir da Emenda Constitucional 103/2019 (https://www.planalto.gov.br/ccivil_03/Constituicao/Emendas/Emc/emc103.htm), que entrou em vigor em 13 de novembro de 2019, o cálculo da aposentadoria para os trabalhadores que ingressaram no sistema previdenciário após a vigência da reforma mudou em relação ao modelo anterior.

Agora, o trabalhador precisará atingir a idade mínima de 62 anos, se mulher, e 65 anos, se homem, além de ter no mínimo 15 anos de contribuição para a Previdência Social, se mulher e 20 anos no caso de homem. Para esses casos, o cálculo do benefício previdenciário passou a ser feito com base na média aritmética simples de todos os salários de contribuição do trabalhador desde julho de 1994, corrigidos mês a mês pelo fator de atualização fornecido pelo governo, com base no inpc acumulado. O valor do benefício, então, será equivalente a 60% da média salarial, acrescido de 2% para cada ano de contribuição que exceder o tempo mínimo de 15 ou 20 anos, respectivamente, se mulher ou homem.

Na seção abaixo, disponibilizamos um recurso que ajudará o beneficiário da previdência a fazer uma simulação de cálculo do seu benefício, devendo para tanto preencher os dados necessários (salários de contribuição, data de nascimento e sexo) e clicar em 'Enviar dados e Simular'.

/* Conteúdo da imagem */

1. Informe o salário recebido correspondente ao mês/ano trabalhado



|--|

```

        <h2>MÊS / ANO</h2>
      </th>
      <th>
        <h2>SALÁRIO (R$)</h2>
      </th>
    </tr>
  </thead>
  <tbody>
    {mesAno.map((value, index) => (
      <tr key={index}>
        <td>
          <input
            type="text"
            name="mesAno"
            value={value}
            onChange={(e) => {
              const updatedMesAno = [...mesAno];
              updatedMesAno[index] = e.target.value;
              setMesAno(updatedMesAno);
            }}
          />
        </td>
        <td>
          <input
            type="text"
            name="salario"
            value={salario[index]}
            onChange={(e) => {
              const updatedSalario = [...salario];
              updatedSalario[index] = e.target.value;
              setSalario(updatedSalario);
            }}
          />
        </td>
      </tr>
    )
  </tbody>
</table>
<table id=""></table>
<button type="button" id="adicionarLinha" onClick={adicionarLinha}>Adicionar
Linha na Tabela</button>
<button type="button" onClick={handleNavigationToResultado}>Ir para
Resultado</button>
</div>
</div>
</form>
</div>
</div>
);
}

```

```
export default SimulePage;
```

componente 'Resultado.jsx', com a seguinte configuração:

```
import React from 'react';
import { useNavigate } from 'react-router-dom'; // Importe useNavigate

function Resultado() {
  const navigate = useNavigate(); // Obtenha o navigate

  const handleNavigationToSimule = () => {
    navigate('/simule'); // Use navigate para voltar para a página de simule
  };

  return (
    <div>
      <div className="cabeca">
        <h1>Resultados</h1>
        <div className="clique">
          <div className="botao">
            <button type="submit" onClick={handleNavigationToSimule}>
              Voltar
            </button>
          </div>
          <div className="botao">
            <button type="submit">
              <a href="/">Início</a>
            </button>
          </div>
        </div>
      </div>
      <div>
        <main>
          <div className="bloco">
            <div className="container">
              <div className="titulo">
                <h2><b>Dados Pessoais</b></h2>
              </div>
              <div className="barra">
                <div>
                  <p><i>Gênero:</i> <span>Masculino</span></p>
                </div>
                <div>
                  <p><i>Idade Atual:</i> <span>30 anos</span></p>
                </div>
                <div>
                  <p><i>Tempo de Contribuição:</i> <span>120 meses</span></p>
                </div>
              </div>
            </div>
          </div>
        </main>
      </div>
    </div>
  );
}
```

```

    <div className="titulo">
      <h2><b>Resultados</b></h2>
    </div>
    <div className="barra">
      <div>
        <p><i>Idade da Aposentadoria:</i> <span>65 anos</span></p>
      </div>
      <div>
        <p><i>Mês e ano da Aposentadoria:</i> <span>12 / 2030</span></p>
      </div>
      <div>
        <p><i>Quanto tempo falta para contribuir:</i></p>
        <span>60 meses</span>
      </div>
      <div>
        <p><i>O valor mínimo do meu benefício será:</i> <span
id="valorBeneficio">R$ 1,000.00</span></p>
      </div>
    </div>
    </div>
    <div className="botao">
      <button type="submit" className="envio">
        <a href="/simule" target="_blank">Nova Consulta</a>
      </button>
    </div>
    <div className="imagem">
      
    </div>
  </div>
</main>
</div>
);
}

```

```
export default Resultado;
```

já o arquivo 'package.json', tem a seguinte configuração:

```

{
  "name": "frontendwise",
  "private": true,
  "version": "0.0.0",
  "type": "module",
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings
0",
    "preview": "vite preview"
  }
}

```

```

},
"dependencies": {
  "history": "^5.3.0",
  "react": "^17.0.2",
  "react-dom": "^17.0.2",
  "react-router-dom": "^6.17.0"
},
"devDependencies": {
  "@types/react": "^17.0.2",
  "@types/react-dom": "^18.2.7",
  "@vitejs/plugin-react": "^4.1.0",
  "eslint": "^8.45.0",
  "eslint-plugin-react": "^7.32.2",
  "eslint-plugin-react-hooks": "^4.6.0",
  "eslint-plugin-react-refresh": "^0.4.3",
  "vite": "^4.4.5"
}
}

```

Também possuo uma API que tem os seguintes diretórios, pastas e arquivos:

arquivo 'server.js' na raiz da API com a seguinte configuração:

```

import express from 'express';
import cors from 'cors';
import bodyParser from 'body-parser';
import morgan from 'morgan';
import simulacaoRoutes from './src/routes/simulacaoRoutes.js';

const app = express();

// Middlewares
app.use(cors());
app.use(bodyParser.json());
app.use(morgan('combined'));

// Rotas
app.use('/api', simulacaoRoutes); // Defina um prefixo de rota "/api" para todas as rotas

// Porta para o servidor
const port = process.env.PORT || 3000;
app.listen(port, () => {
  console.log(`Servidor rodando na porta ${port}`);
});

```

arquivo previdenciaaquidb.sql, também na raiz da API com a seguinte configuração:

```

-- MySQL Script generated by MySQL Workbench
-- Thu May 18 16:59:19 2023
-- Model: New Model    Version: 1.0

```

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR
_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

--
-- Schema previdenciaaquidb
--
DROP SCHEMA IF EXISTS `previdenciaaquidb`;
--
-- Schema previdenciaaquidb
--
CREATE SCHEMA IF NOT EXISTS `previdenciaaquidb` DEFAULT CHARACTER SET utf8 ;
USE `previdenciaaquidb` ;

--
-- Table `previdenciaaquidb`.`simulacao_beneficio`
--
CREATE TABLE IF NOT EXISTS `previdenciaaquidb`.`simulacao_beneficio` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `genero` ENUM('m', 'f') NOT NULL,
  `data_nascimento` DATE NOT NULL,
  `idade` INT NULL,
  `tempo_contribuicao_mes` INT NULL,
  `idade_aposentadoria` INT NULL,
  `mes_aposentadoria` INT NULL,
  `ano_aposentadoria` INT NULL,
  `tempo_contribuicao_pendente_mes` INT NULL,
  `valor_beneficio` DECIMAL(10,2) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;
```

arquivo package.json, também na raiz da API com a seguinte configuração:

```
{
  "name": "api",
  "version": "0.0.1",
  "description": "Api do nosso projeto CashWise. Feito com express seguindo o modelo
MVC com programação modular.",
  "main": "server.js",
  "type": "module",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "dev": "node server.js"
  },
  "keywords": [
    "express",
```



```

    "mysql2",
    "mvc",
    "api",
    "node",
    "json"
  ],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.2",
    "cors": "^2.8.5",
    "express": "^4.18.2",
    "morgan": "^1.10.0",
    "mysql2": "^3.6.1"
  }
}

```

Diretório 'src' com os seguintes arquivos:

arquivo 'app.js' com a seguinte configuração:

```

const express = require('express');
const cors = require('cors');
const bodyParser = require('body-parser');
const morgan = require('morgan');
const simulaRoutes = require('./routes/simulacaoRoutes');

const app = express();

// Middlewares
app.use(cors());
app.use(bodyParser.json());
app.use(morgan('combined'));

// Rotas
app.use('/api', simulaRoutes); // Defina um prefixo de rota "/api" para todas as rotas

// Porta para o servidor
const port = process.env.PORT || 3000;
app.listen(port, () => {
  console.log(`Servidor rodando na porta ${port}`);
});

```

pasta config com o arquivo 'database.config.js' com a seguinte configuração:

```

const mysql = require('mysql2/promise');

```

```
const poolSimule = mysql.createPool({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'previdenciaaquidb',
});

module.exports = {
  poolSimule
};
```

pasta 'controllers com o arquivo 'simulacaoControllers.js' com a seguinte configuração:

```
import express from 'express';
import SimulacaoBeneficio from '../models/SimulacaoBeneficio.js';
const router = express.Router();
const simulacaoModel = new SimulacaoBeneficio();

// Rota para criar uma nova simulação
router.post('/simulacoes', async (req, res) => {
  try {
    const simulacao = req.body; // Certifique-se de que o corpo da solicitação contém os dados da simulação
    const result = await simulacaoModel.criar(simulacao);
    res.json(result);
  } catch (error) {
    res.status(500).json({ success: false, message: 'Erro ao criar a simulação.', error: error.message });
  }
});

// Rota para obter todas as simulações
router.get('/simulacoes', async (req, res) => {
  try {
    const result = await simulacaoModel.listarTodos();
    res.json(result);
  } catch (error) {
    res.status(500).json({ success: false, message: 'Erro ao listar as simulações.', error: error.message });
  }
});

// Rota para obter uma simulação pelo ID
router.get('/simulacoes/:id', async (req, res) => {
  try {
    const id = req.params.id;
    const result = await simulacaoModel.consultarPorId(id);
    res.json(result);
  } catch (error) {
```

```

    res.status(500).json({ success: false, message: 'Erro ao obter a simulação.',
error: error.message });
  }
});

// Rota para excluir uma simulação pelo ID
router.delete('/simulacoes/:id', async (req, res) => {
  try {
    const id = req.params.id;
    const result = await simulacaoModel.deletar(id);
    res.json(result);
  } catch (error) {
    res.status(500).json({ success: false, message: 'Erro ao excluir a simulação.',
error: error.message });
  }
});

export default router;

```

pasta 'models' com o arquivo 'SimulacaoBeneficio' com a seguinte configuração:

```

import mysql from 'mysql2/promise';

class SimulacaoBeneficio {
  constructor() {
    this.connection = mysql.createPool({
      host: 'localhost',
      user: 'root',
      password: '',
      database: 'previdenciaaquidb'
    });
  }

  async criar(simulacao) {
    try {
      const sql = `INSERT INTO simulacao_beneficio (tempo_contribuicao_mes,
idade_aposentadoria, mes_aposentadoria, ano_aposentadoria,
tempo_contribuicao_pendente_mes) VALUES (?, ?, ?, ?, ?)`;
      const values = [
        simulacao.tempoContribuicaoMes,
        simulacao.idadeAposentadoria,
        simulacao.mesAposentadoria,
        simulacao.anoAposentadoria,
        simulacao.tempoContribuicaoPendenteMes,
      ];

      const [rows] = await this.connection.execute(sql, values);

      if (rows.affectedRows === 1) {
        return { success: true, message: 'Registro criado com sucesso.' };
      } else {

```

```

    return { success: false, message: 'Não foi possível criar o registro.' };
  }
} catch (error) {
  return { success: false, message: 'Erro ao criar o registro.', error };
}
}

async listarTodos() {
  try {
    const sql = `SELECT * FROM simulacao_beneficio`;
    const [rows] = await this.connection.query(sql);

    return { success: true, data: rows };
  } catch (error) {
    return { success: false, message: 'Erro ao listar os registros.', error };
  }
}

async consultarPorId(id) {
  try {
    const sql = `SELECT * FROM simulacao_beneficio WHERE id = ?`;
    const [rows] = await this.connection.execute(sql, [id]);

    if (rows.length === 1) {
      return { success: true, data: rows[0] };
    } else {
      return { success: false, message: 'Registro não encontrado.' };
    }
  } catch (error) {
    return { success: false, message: 'Erro ao consultar o registro.', error };
  }
}

async deletar(id) {
  try {
    const sql = `DELETE FROM simulacao_beneficio WHERE id = ?`;
    const [rows] = await this.connection.execute(sql, [id]);

    if (rows.affectedRows === 1) {
      return { success: true, message: 'Registro deletado com sucesso.' };
    } else {
      return { success: false, message: 'Registro não encontrado.' };
    }
  } catch (error) {
    return { success: false, message: 'Erro ao deletar o registro.', error };
  }
}

export default SimulacaoBeneficio;

```

e por fim a pasta 'routes' com o arquivo 'simulacaoRoutes.js' com a seguinte configuração:

```
import express from 'express';
import simulacaoControllers from '../controllers/simulacaoControllers.js';

const router = express.Router();

router.use('/api', simulacaoControllers);

// Middleware de tratamento de erros
router.use((err, req, res, next) => {
  console.error(err.stack); // Registra o erro no console para fins de depuração

  // Verifique se a exceção possui uma mensagem personalizada
  if (err.message) {
    return res.status(500).json({ error: err.message });
  }

  // Caso contrário, trate como um erro interno genérico
  res.status(500).json({ error: 'Erro interno do servidor' });
});

export default router;
```

Qual a melhor forma de conectar o frontend acima que foi desenvolvido em React + Vite a essa API backend desenvolvida em NodeJs com Express e base de dados?