Para implementar una funcionalidad de autocompletado en tiempo real en tu aplicación front-end construida con TypeScript y conectada a una base de datos Oracle SQL, puedes seguir estos pasos:

1. **Crear un endpoint en tu backend**: Este endpoint se encargará de recibir las consultas del front-end y devolver las posibles coincidencias desde la base de datos.
2. **Configurar el front-end para hacer solicitudes a este endpoint**: A medida que el usuario escribe en el campo de texto, se harán solicitudes asíncronas (AJAX) al backend para obtener las coincidencias.
3. **Actualizar el campo de texto con las coincidencias**: Mostrar las sugerencias al usuario para que pueda seleccionarlas.

A continuación, te detallo cada uno de estos pasos:

**1. Crear un endpoint en el backend**

Aquí asumo que tienes un backend que puede manejar solicitudes HTTP y conectarse a tu base de datos Oracle SQL. Puedes usar Node.js con Express y un cliente de Oracle como oracledb para este propósito.

javascript

Copiar código

const express = require('express');

const oracledb = require('oracledb');

const app = express();

const port = 3000;

async function getDatabaseConnection() {

return await oracledb.getConnection({

user: 'your\_username',

password: 'your\_password',

connectString: 'your\_connection\_string'

});

}

app.get('/autocomplete', async (req, res) => {

const searchTerm = req.query.q;

if (!searchTerm) {

return res.status(400).send('Query parameter "q" is required');

}

let connection;

try {

connection = await getDatabaseConnection();

const result = await connection.execute(

`SELECT name FROM your\_table WHERE name LIKE :searchTerm`,

[`%${searchTerm}%`]

);

res.json(result.rows.map(row => row[0]));

} catch (err) {

console.error(err);

res.status(500).send('Internal Server Error');

} finally {

if (connection) {

try {

await connection.close();

} catch (err) {

console.error(err);

}

}

}

});

app.listen(port, () => {

console.log(`Server running on http://localhost:${port}`);

});

**2. Configurar el front-end para hacer solicitudes al backend**

En tu aplicación front-end, puedes usar fetch o una biblioteca como axios para hacer solicitudes al backend mientras el usuario escribe en el campo de texto. Aquí tienes un ejemplo usando fetch:

typescript

Copiar código

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

const Autocomplete = () => {

const [searchTerm, setSearchTerm] = useState('');

const [suggestions, setSuggestions] = useState<string[]>([]);

useEffect(() => {

if (searchTerm) {

const fetchSuggestions = async () => {

const response = await fetch(`/autocomplete?q=${searchTerm}`);

const data = await response.json();

setSuggestions(data);

};

fetchSuggestions();

} else {

setSuggestions([]);

}

}, [searchTerm]);

return (

<div>

<input

type="text"

value={searchTerm}

onChange={(e) => setSearchTerm(e.target.value)}

placeholder="Type a name..."

/>

{suggestions.length > 0 && (

<ul>

{suggestions.map((suggestion, index) => (

<li key={index}>{suggestion}</li>

))}

</ul>

)}

</div>

);

};

export default Autocomplete;

**3. Actualizar el campo de texto con las coincidencias**

El componente Autocomplete actualiza el estado con las sugerencias recibidas del backend y muestra una lista desplegable con esas sugerencias debajo del campo de texto.

**Conexión del Front-end y el Backend**

Asegúrate de que tu backend y front-end están configurados para poder comunicarse entre sí. Esto puede implicar configurar CORS (Cross-Origin Resource Sharing) si tu front-end y back-end están en diferentes dominios.

Puedes añadir las siguientes líneas a tu configuración de Express para permitir solicitudes desde tu front-end:

javascript

Copiar código

const cors = require('cors');

app.use(cors());

**Resumen**

1. **Backend**: Crear un endpoint que reciba las consultas y devuelva las coincidencias desde la base de datos.
2. **Front-end**: Configurar el campo de texto para hacer solicitudes al backend y mostrar las coincidencias en tiempo real.

Siguiendo estos pasos, deberías ser capaz de implementar una funcionalidad de autocompletado en tiempo real en tu aplicación.