Deeto Frontend Challenge: Chatbot Application

Description

Build an application using **React** and **TypeScript** that implements a **functional chatbot**.

The goal is to simulate a basic **ChatGPT-like** chat experience:

- Display initial messages loaded from our API.
- Allow the user to send new messages.
- Display the chatbot's responses, using polling until the full response is received.

Functional Requirements

- **Display initial messages** provided in `settings.messages[]` when the application starts.
- Provide a **text input** for the user to send messages.
- **Render conversation messages** on the screen.
- Use a **global state management** solution (your choice: **useContext**, **Zustand**, **Redux**, etc.) to persist chat data and settings across the application.
- **Apply custom styles** based on the variables returned from the API in `settings.styles`

Endpoints

`vendorId = c91c8550-8c5b-48ae-8be5-80522fd34dcd`

1. Fetch chat configuration and start conversation

`GET https://dev-api.deeto.ai/v2/chatbot/{vendorId}`

> This endpoint will return initial messages in `settings.messages[]` along with additional style configuration variables.

---

2. Send a message

`POST https://dev-api.deeto.ai/v2/chat`

Body:

```
{
  "async": false,
  "message": "Your message here",
  "vendorId": "{vendorId}"
}
```

Technical Considerations

- Components should be **reusable**.
- Use a **global state management** library of your choice.
- Implement **customizable styles** based on the API-provided variables.
- Ensure a **clean and stable chat flow** (handle loading and error states properly).
- Project structure should be **organized and scalable**.

Evaluation Criteria

- Correct **implementation of the chat flow**.
- **Code quality** (readability, best practices, clear TypeScript typing).
- **Global state usage** to store settings and chat messages.
- **Component reusability** for a flexible design.
- **Style customization** using API variables.
- Proper **asynchronous handling** (polling for responses).
- **Smooth and responsive UX**.