Software Engineer (Frontend) Technical Exercise

Technical Exercise

Overview

Maihem helps companies automate performance, QA and security testing of enterprise-grade AI applications. An important component of this involves enabling our customers to view real-time monitored and simulated interactions between their customers and their conversational AI agents (i.e. chatbots).

While many of our users are engineers, Maihem is used heavily by non-technical users to review these conversations, so the interface needs to feel clean, responsive and familiar.

Exercise

Requirements

Design and build a simple real-time chat interface for Maihem customers to interact directly with their conversational AI agent with the following requirements:

1. Maihem user can view the conversation in real-time (think Whatsapp / Telegram / Intercom style chat)
2. Maihem user can send a message to the AI agent and receive a reply
3. AI agent messages should be accompanied by an "evaluation result" for the Maihem helpfulness metric on each message, including:
    - Result: Passed or Failed
    - Score: 0% - 100% (80%+ = Passed)
    - Explanation: human readable explanation (e.g. "The agent's response was helpful/not helpful because....")

Bonus points

If you have time and/or you're enjoying the exercise:

- Add a "conversation quality summary" component that keeps a running conversation score based on the evaluation results
- Add a "user feedback" component that allows the Maihem user to give positive/negative feedback on the quality of the "evaluation result"
- Deploy the app to Vercel (or similar)

Notes

- We'd suggest spending **no more than 4 hours** on the exercise in total
- You can mock/randomise the helpfulness evaluation results

Design specifications

This is totally up to you! We want you to use your creativity and understanding of the use case to design the interface while taking inspiration from products you know and love.

Technical specifications

Tech stack	**React.js/Next.js**, **Tailwind** and your **preferred UI components** (we use shadcn/ui but ch you like!)
AI chat API	Azure OpenAI completions endpoint (details below)

You should have received an API key in our email.

Here's a snippet to help get you started with the OpenAI client:

```
import { AzureOpenAI } from "openai"; export default async function
main() { const openai = new AzureOpenAI({ endpoint: "https://maihem-
openai-uswest.openai.azure.com", apiKey: "<api key provided by
Maihem>", apiVersion: "2024-08-01-preview", deployment: "maihem-
mini", }); const result = await openai.chat.completions.create({
model: "gpt-4o-mini", messages: [{ role: "user", content: "Say
hello!" }], }); console.log(result.choices[0]!.message?.content); }
main();
```

Deliverables

1. Github/Gitlab repo with your completed app
2. README.md about your app and design decisions

Questions?

Email simon@maihem.ai with any questions before, during or after the exercise!