

# Scenario Based Challenge - Chat app Hiring Test

Five (5) full days to complete

This challenge will test your software development skills, problem-solving skills and software architecture and design patterns, including solutioning, finding the right tools, and also leveraging today's booming Artificial Intelligence chatbot landscape, which is becoming increasingly relevant in the software development space.

At our company, we use Postgres-Monodb-Express/NestJS-React-Node (P-MERN stack? 🧐). We recommend choosing from these technologies, but a choice of any other language/stack will be equally accepted.

1. Build a very simple Web chat widget that can be plugged into a web page. A simple script for a widget will do, so that the chat window toggles open and closed when the chat bubble is clicked. Your widget should be able to communicate with a simple backend API in any language of your choice to send and receive messages. This widget will be used by customers
  - a. Each conversation session should have a timeout period of 5 mins. That is the amount of time after which the session will end if no messages have been exchanged
  - b. When the widget is toggled open,
    - i. if there is no currently active session, the widget should display a form for the customer to enter their name and email. After filling the form, the user should then be allowed to proceed and initiate a conversation.
    - ii. If the timeout period has not expired, display the last conversation history and allow the customer to continue the conversation where they left off
2. Build a simple chat application where messages from customers will be displayed and replied to. This application will be used by organizations (support agents).
  - a. Provide two (2) options so that users are able to chat with a chatbot, or with a live agent. In the settings, the organization should be able to switch the settings at any time, and messages will automatically be routed to either a GPT chatbot or a live agent. See diagram below. If the choice is a live agent, implement a round-robin algorithm to assign the conversation to the least burdened or otherwise randomly selected agent

☐ Route messages to chatbot

chatbot URL

Auth token

☐ Route messages to live agents

- b. An agent is allowed to be assigned, and to chat with multiple customers at the same time. Display each customer that the agent is chatting with, or has chatted with, in a separate panel, by name. See diagram below

Customers

Allen Bore

Kate Johnson

Jim Francis

Send

**Important:**

1. Host your code base on any public, free repositories, and include the links in your email replies. **Do not** send me zipped code files
2. Host the application on any public, free application hosting websites, and share the URLs in your email replies. I will not download and run applications on my local computer