

# Scenario Based Challenge - AI Integration

Five (5) full days to complete

This challenge will test your software development skills, understanding of AI and AI tools, and how to leverage AI to solve problems in today's fast-moving world of software services.

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, and save the messages in a database
2. Using any publicly available (open source, or freemium) AI tools on the market, integrate communication AI with your backend so that your application can converse with the user on the Web Chat Widget.
  - a. Your AI bot should be able to understand messages in free text, without using menus or buttons.
  - b. When the user opens the widget, start a new conversation; the system should prompt the user for their name and email address, and should be able to understand the response presented in any number of ways. It should extract the relevant information and save the new user as a lead, or update an existing.
  - c. It should be able to handle general conversation, and fail gracefully, if a message is not clear. The AI bot should remember the context of the conversation, and interpret questions in context
  - d. It should be able to handle specific conversations at least in the following scenarios
    - i. Booking a flight ticket: it should answer questions relating to the process of booking a ticket with any airline specified. If it can't find the airline, or answer questions specific to said airline, it should fail gracefully - failing gracefully implies that the chatbot should be able to recognize when it cannot interpret a prompt, or does not have the appropriate response, and handle this appropriately.
    - ii. **[Pretend]** It should proceed to book a ticket for the user should they choose to buy one, using flight information **scraped** from any flight booking website, or through a publicly available API (**do NOT actually book a flight ticket. Just make a booking record in a database against the lead**)
    - iii. **[Pretend]** Answer questions about their flight booking if they've already booked a flight (**use the fake flight records in your database**)
    - iv. **[Pretend]** Change flight dates or cancel flights if a flight has already been booked (**use the fake flight records in your database**)

**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