Lost Package Tracking Chatbot

Tyler Tran

Description

- I built a website chatbot that helps people track their packages when they can't find them
- Uses Google's Gemini LLM to figure out what to say next in the conversation
  - Uses Gemini to understand what the user's responses and find tracking numbers in their messages
- Tells you if your package is still on the way or already delivered
- Accepts 10 tracking numbers for testing purposes (see *Valid Tracking Numbers* section of README.md file)

Design Choices

- Created clean, simple, and easy-to-understand user interface
  - Made bot messages blue and user messages green so you can tell them apart
  - Chatbot shows a "Thinking..." message with animated ellipses while the AI is reading the user's response and deciding what to do next
- Included helpful and friendly predefined responses in the conversation flow to ensure a good customer support experience
- Used Gemini LLM to understand a wide variety of possible user responses and decide which branch in the conversation flow to take
- Made it so the bot knows what to do based on whether the package is in transit or has been delivered already
- Split up the code so each part does one thing

Technical Implementation

- Frontend:
  - Used HTML, CSS, and JavaScript
- AI integration:
  - Used Google Gemini API (*gemini-2.5-flash* model) to quickly understand user responses and support a wide variety of possible user responses
- Architecture:
  - Single-page application with separate functions for different jobs
  - Uses true/false flags to keep track of where you are in the conversation
  - Picks random cities for packages that are in transit to simulate real packages

Challenges Faced

- Creating the entire conversation decision tree
  - Had to think about how to deal with varying user responses (e.g. responses with irrelevant words, different ways of expressing the same idea, etc.)
    - Ultimately decided on using Gemini to process user responses
- Deciding between using predefined chatbot responses or using AI to generate responses to user messages
  - Ultimately decided on using predefined chatbot responses and AI to understand user messages
- Hard to keep track of what the conversation was about across multiple AI calls
- Making it clear when the AI is thinking about the user's response vs. when it is done thinking

How I Would Improve the Chatbot

- Add more branches to the decision tree to support more scenarios and edge cases
- Add support for languages other than English
- Remember conversations between visits
- Add voice input and output
- Rewrite chatbot responses to make them seem more human-like and genuine
- Add multiple chatbot responses for each scenario to increase variety of responses