

Internship Assignment: Build a Conversational Voice Agent for Booking with Error Handling

Objective:

The goal of this task is to assess the intern's ability to develop a conversational voice agent capable of handling user interactions for a booking system. The agent should be able to take input for user details, including their name, email, and preferred date, while handling incorrect or malformed inputs intelligently. The primary focus is on providing an optimal user experience with minimal latency and correct error handling in real-world scenarios.

Task Overview:

You will design and implement a voice-based conversational agent that:

1. Takes input from the user: Name, Email, and Date.
2. Validates the inputs to ensure correct data formats (email format, valid date, and name).
3. Handles incorrect or malformed inputs gracefully by prompting the user to re-enter information without causing frustration.

The task should involve the following key components:

- **Voice Input/Output:** Using speech-to-text (STT) for receiving user input and text-to-speech (TTS) for responding.
 - **Data Validation:** Verifying user-provided information (email, date, name).
 - **Error Handling:** Ensuring the system can handle incorrect or incomplete inputs by prompting for corrections.
 - **Minimal Latency:** Ensuring that the agent operates with low response time for a smooth user experience.
-

Detailed Task Breakdown:

1. Voice-Based Interaction (Input/Output):

- **Voice Input:** Use speech-to-text (STT) technology to recognize and transcribe the user's spoken input. The agent will receive input from the user in natural language. For example:
 - User: "I want to book an event."
 - System: "Sure! What is your name?"
- **Voice Output:** Use text-to-speech (TTS) to respond to the user in a conversational tone. Responses should feel natural and prompt the user clearly to enter the necessary details.

2. Information Gathering (Name, Email, and Date):

- **Name Input:**
 - The system will prompt the user for their name.
 - Example interaction:
 - System: "What is your full name?"
 - User: "John Doe"
 - Validate that the input is a string of alphabetic characters, and handle cases where the user enters numbers, symbols, or gibberish.
- **Email Input:**
 - The system will prompt the user to provide their email.
 - Example interaction:
 - System: "Please provide your email address."
 - User: "john.doe@gmail.com"
 - Validate the email to ensure it follows the standard email format (e.g., `user@example.com`). If the email is invalid, the system should ask the user to re-enter it.
- **Date Input:**
 - The system will ask for a preferred date for the booking.
 - Example interaction:

- System: "What date would you prefer?"
- User: "April 15th, 2025"
- Validate the date format (e.g., YYYY-MM-DD) and ensure it's a future date. If the user provides an invalid or past date, prompt them to provide a valid future date.

3. Error Handling & Corrections:

- **Invalid Name Input:** If the user enters anything other than alphabetic characters (e.g., "1234" or special characters like "\$#@!"), the agent will say:
 - "Sorry, that doesn't look like a valid name. Can you please provide your full name using only letters?"
- **Invalid Email Input:** If the user provides an invalid email (e.g., "john.doe" or "john@.com"), the agent should respond:
 - "That doesn't appear to be a valid email address. Can you please re-enter it in the format user@example.com?"
- **Invalid Date Input:** If the user gives a date in the wrong format (e.g., "15/04/2025") or chooses a past date, the agent should respond:
 - "The date you entered seems incorrect or in the past. Please provide a valid future date, for example, 'April 15th, 2025'."

4. Confirmation & Final Booking:

- Once all the inputs (Name, Email, Date) have been successfully validated:
 - The system should confirm the booking details:
 - "To confirm, you are booking with the name John Doe, email john.doe@gmail.com, and the date April 15th, 2025. Is that correct?"
 - The system should allow the user to confirm or make changes:
 - If correct: "Great! Your booking has been confirmed."
 - If incorrect: "Please let me know which part you would like to change."

5. Latency & Optimization:

- The system should be designed to ensure that there is minimal delay between user input and system responses. The voice recognition and synthesis systems should be optimized to provide near-instant feedback, ideally with latencies under 500ms between spoken input and system response.
-

Deliverables:

1. **Source Code:** Complete code for the voice-based agent, including:
 - Input processing (STT).
 - Validation logic for name, email, and date.
 - Conversational flow (TTS responses).
 - Error handling.
 2. **Demo:** A short demo showing how the agent works in action, from input to confirmation.
 3. **Documentation:** A README file with:
 - Instructions for setting up and running the system.
 - Any dependencies or services used (e.g., APIs for STT, TTS).
 4. **Design Considerations:** Briefly explain any design choices made to optimize the system for low latency and smooth user experience.
-

Submission Deadline:

- **1 Weeks** from the date of assignment distribution.
-

Good luck!