Technical Assessment: Forward Deployed Engineer - Restaurant Booking System

Overview

You are tasked to build a conversational agent that helps customers make and manage restaurant bookings. This assessment evaluates your ability to design and implement an intelligent agent system that can handle real-world booking scenarios.

Core Requirements

1. Objective

Create a chat interface that allows customers to:

- Ask questions about restaurant availability and policies
- Make new bookings
- Check existing booking details
- Modify or cancel reservations

Interface:

- **Minimum Requirement**: Terminal/console-based chat interface
- **Bonus Points**: Web-based chat interface

Booking API Integration:

- Connect to the provided mock booking server
- Use the existing API endpoints for all booking operations
- Handle authentication properly with the provided bearer token
- Gracefully handle API errors and invalid requests
- Ensure booking parameters are valid before API calls

Agent framework:

- Use any popular agent framework (LangGraph, CrewAI, Autogen, etc.) or create a custom framework
- Leverage additional open-source tools and frameworks
- Maintain conversation state across a multi-turn conversation

- Utilise closed-source APIs or locally hosted models (ollama, vllm, llama cpp, etc.)

Selection Criteria: Choose the approach that best demonstrates your innovation, creativity, and production-readiness thinking. Feel free to use any AI assistant development tools (Cursor, GitHub Copilot, etc.)

2. User stories

- **Check availability**
  - User story: As a user, I would like to check availability for a specific date and time so I can see if a reservation can be made.
  - *Example*: "Can you show me availability for this weekend?"
- **Book a reservation**
  - User story: As a user, I would like to book a reservation by specifying the date, time, and number of people so I can secure my table.
  - *Example*: "I'd like to book a table for 4 people next Friday at 7pm."
- **Check reservation information**
  - User story: As a user, I would like to check my reservation information using my booking reference
  - *Example*: "What time is my reservation on Saturday?"
- **Modify a reservation**
  - User story: As a user, I would like to modify my reservation by changing the date, time, or cancelling it entirely so I can update my booking details
  - *Example*: "I need to change my booking from 6pm to 8pm."
  - *Example*: "Please cancel my reservation for tomorrow."

3. Deliverables

1. **Source Code**: GitHub repository URL containing your implementation with a clear README. The documentation should include:
   a. Overview
   b. Getting started instructions
   c. Design rationale
      i. Why you chose specific frameworks/tools
      ii. Design decisions and trade-offs considered
      iii. How your solution would scale in production
      iv. Identified limitations and potential improvements
      v. Security considerations and implementation strategies
2. **Demo video**: Demo video showcasing agent functionality for each user story

Technical Specifications

API Integration

- Base URL: http://localhost:8547
- All endpoints require Bearer token authentication
- Content-Type: application/x-www-form-urlencoded
- Restaurant name: TheHungryUnicorn

Getting Started

1. Visit this GitHub repository: https://github.com/AppellaAI/Restaurant-Booking-Mock-API-Server
2. Review the provided README.md
3. Start the mock booking server: python -m app
4. Test API endpoints to understand the booking system
5. Expand the backend server to work with your chosen agent framework if needed
6. Design your conversational interface
7. Implement and test your solution

Evaluation Process

Your submission will be evaluated based on:

- **Functionality**: Does it meet the core requirements?
- **Innovation & Creativity**: Novel approaches and thoughtful problem-solving
- **Technical Excellence**: Code quality, architecture, and documentation
- **Rationale**: Clear explanation of design choices and trade-offs

We particularly value candidates who demonstrate:

- Understanding of real-world production constraints
- Ability to articulate technical decisions clearly
- Creative problem-solving within practical boundaries
- Consideration of user experience and business requirements

Support

If you encounter any issues with the mock booking server or have questions about the requirements, please reach out via email for clarification.

Good luck!