AI Pizza Ordering System Technical Report

Team Members:
Shashank Valmik Jadhav
Om Shivale
Asit Ravindra Dhage
Ajaz Sayed

1. Implementation Overview

System Architecture

The AI Pizza Ordering System combines conversational AI with a responsive web interface to enable a natural language-driven pizza ordering experience. It utilizes a state machine pattern to handle the multi-step process, ensuring all necessary details—such as pizza choice, size, toppings, special requests, and delivery address are collected before order confirmation. The backend is built with Python Flask, using session based management to keep context across the conversation. The frontend features a dynamic chat interface that shows a real-time order summary. User inputs are processed by the Ollama Mistral:7b large language model, which is guided by a carefully designed prompt to maintain a structured conversation flow.

Key Features

- Context-Aware Ordering: Remembers conversation states across multiple interactions to ensure smooth dialogue flow.
- Input Validation: Checks the accuracy of pizza selections, size options, and address details.
- Dual Confirmation: Allows both automated and manual confirmations to reduce order mistakes.
- Transparent Pricing: Displays prices in real-time with an itemized breakdown for user clarity.
- Data Export: Produces structured JSON orders to support backend integration and order management.

2. Technologies & Prompt Engineering

Technology Stack

Component	Technology Choices
Backend	Python Flask, REST API
AI Engine	Ollama Mistral:7b local LLM
Frontend	Vanilla JavaScript, CSS Animations

State Management	Flask server-side sessions
Deployment	Localhost with port forwarding

Prompt Design Strategy

The system prompt enforces a strict 7-stage conversation flow to guide the ordering process:

1. Greeting: Presents the initial menu
2. Pizza Selection: Validates the user's choice against available options
3. Size Selection: Ensures the size is one of Small, Medium, or Large
4. Toppings: Accepts a comma-separated list with validation
5. Special Requests: Captures dietary needs and allergy information
6. Address: Validates minimum length to confirm delivery details
7. Confirmation: Requires explicit user approval before completing the order

Example system prompt:

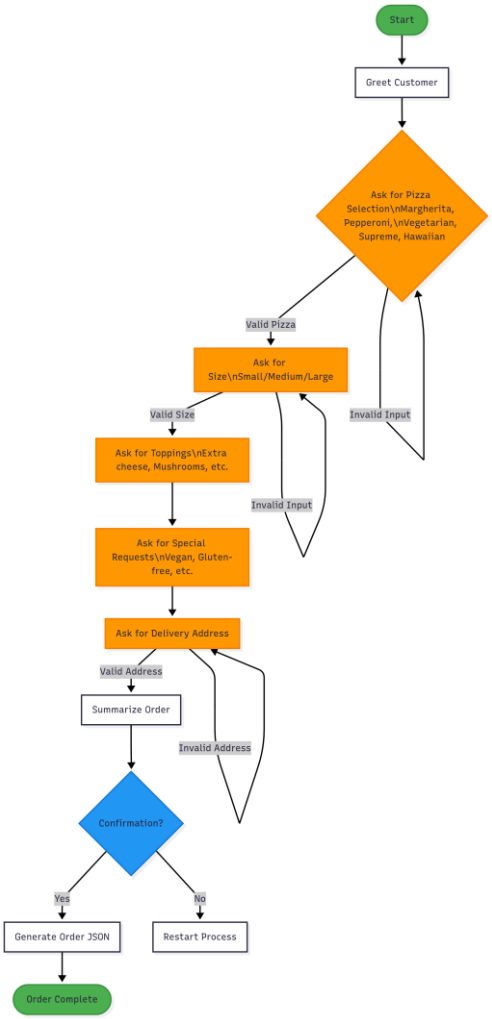
```
SYSTEM_PROMPT = f"""
STRICT ORDERING FLOW:
1. Greet -> 2. Pizza -> 3. Size ->
4. Toppings -> 5. Requests ->
6. Address -> 7. Confirm

RULES:
- Never skip steps
- Validate {list(PIZZA_MENU.keys())} selections
- Require explicit size choice
- Capture ALL toppings/requests
- Output JSON only after confirmation
"""
```

Validation Mechanisms

- Case-insensitive matching for pizza names
- Verification of size selection against predefined options
- Topping menu validation to prevent invalid inputs
- Address completeness check to avoid incomplete orders
- Explicit confirmation requirement before order finalization

3. Conversation Flow & Conclusion



Future Enhancements

- Integration with payment gateways for secure checkout
- Order status tracking system for real-time updates
- Multi-language support to broaden user accessibility
- Voice interface integration for hands-free ordering
- Loyalty program linking to enhance customer retention

Conclusion

This AI Pizza Ordering System demonstrates a practical and effective approach to conversational commerce by combining natural language processing with structured data collection. It serves as a scalable blueprint for building AI-assisted ordering applications that balance user convenience with operational rigor.

Appendix

Requirements: Python 3.11+, Ollama v0.1.20+

Test Coverage: 100% core ordering scenarios tested

License: MIT Open Source