

DESKTECH Pizza Assistant

Team Members:

Ali Nouredine (100001779)

Muhammad Adnan (100001588)

1. Introduction

The DESKTECH Pizza Assistant is a Python application with a graphical user interface (GUI) built using Tkinter. It integrates Google's Gemini AI model to provide a multilingual, interactive pizza ordering assistant. The assistant handles natural language conversations, processes orders, and saves structured order data in JSON format.

2. Code Overview

- **GUI Elements:**
The interface includes language selection (English, Arabic, German, Russian), a chat display window, an input entry box, and a send button.
 - **Gemini AI Integration:**
Upon language selection, the app initializes a Gemini model (gemini-1.5-flash) with a system instruction prompt adapted to the selected language, starting a chat session.
 - **Message Handling:**
User input is sent to Gemini. Responses are cleaned and displayed in the chat window. The assistant detects order summary texts, extracts customer details (name, phone, address), pizza details (types, sizes, crusts, toppings), dietary preferences (halal, allergies), payment method, and totals using regular expressions.
 - **Order Storage:**
Extracted structured orders are saved as timestamped JSON files locally, with success/failure messages displayed.
 - **Error Handling:**
Errors during API calls or file operations are caught and shown in the chat.
-

3. Flowchart Diagram Explanation

The flowchart outlines the main application workflow:

- **Initialization:** GUI setup and language selection by the user.
- **Chat Initialization:** Starting the Gemini AI chat session with the appropriate language prompt.
- **User Interaction:** User sends messages, which the app forwards to Gemini AI.
- **Response Processing:** Responses are displayed, and if they contain order summaries, relevant details are parsed and saved.

- **Order Management:** Orders are stored as JSON files for further use.
- **Error Handling:** Any exceptions in sending messages or saving orders are displayed to the user.

This logical structure ensures a smooth user experience while handling the complexities of natural language input and order extraction.

4. Prompt and Multilingual Support

The assistant uses a multilingual prompt system (`prompt_multilingual`) that provides system instructions tailored to the user's language choice. This enables seamless conversations in multiple languages without restarting the app. The system prompt guides the Gemini model on how to behave and respond in the context of pizza ordering, dietary requirements, payment options, and order summaries.

5. Conclusion

This project successfully demonstrates integrating advanced AI conversational models with a user-friendly Python GUI to create a real-world pizza ordering assistant. The structured flowchart clarifies the application's internal processes, and the multilingual prompt design enhances accessibility.