BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Weekly Report

Foundation Internship

Project Title: Smart Shop AI Assistant

Instructor: Kim Ngoc Bach

Student Name : Le Tran Quoc Bao

Student ID : B22DCVT050

Class : E22CQCN05-B

Hà Nội - 2025

INTERNSHIP BASE REPORT - WEEK 7

WEEK 7 INTERNSHIP REPORT

1. Overview of This Week's Work

During the seventh week of the internship, I focused on enhancing the chatbot system to improve user interaction quality and ensure a professional, user-friendly experience. Additionally, I optimized the backend to improve the storage of customer information in the database, specifically in the `orders` table. These enhancements aim to improve system performance, data reliability, and user experience.

2. Work Completed

2.1. Chatbot Enhancements

I made significant improvements to the chatbot to ensure professional and efficient user interactions. Key updates include:

- **Enhanced Politeness and Tone**: Modified the chatbot's prompt to adopt a polite tone, with the chatbot referring to itself as "Shop" and addressing users as "bạn" (translated as "you"). For example, responses now begin with "Xin chào bạn" ("Hello, you") for standalone or initial queries, such as "Hello, you, Shop has iPhone 14 - 128GB priced at 20,990,000 VNĐ."
- **Improved Contact Information Responses**: Reworked responses for queries about store address, email, phone number, and working hours to be more engaging and inviting. For instance, instead of "Contact phone number: 0988888888," the chatbot now responds with "Hello, you, Shop's contact phone number is 0988888888. Feel free to call during working hours!"
- **Upgraded Intent Detection**: Added new intents (`ask_address`, `ask_email`, `ask_phone`, `ask_working_hours`) to accurately identify and handle contact-related queries without relying on the Gemini API, reducing latency and improving response accuracy.

- **Optimized Prompt Structure**: Updated the Gemini API prompt to include clear instructions for maintaining politeness, focusing on the queried product or contact information, and avoiding irrelevant suggestions (e.g., not suggesting iPhone 15 when asked about iPhone 14).
- **Improved Fallback Responses**: Enhanced fallback responses to remain polite and helpful, ensuring users receive meaningful replies even when the chatbot cannot fully understand the query, e.g., "Hello, you, Shop didn't quite understand your question. Could you provide more details?"

2.2. Backend Improvements

I optimized the backend to enhance the efficiency and reliability of storing customer information in the database. Specific updates include:

- **Optimized Order Storage**: Improved the logic for handling the `orders` table to ensure customer information (such as email, total order value, and status) is stored accurately and consistently. Added input validation to prevent errors during storage, e.g., verifying valid email addresses before insertion.
- **Improved Database Performance**: Optimized SQL queries in the order storage function to reduce execution time, particularly when handling large volumes of orders. Ensured database transactions are managed efficiently to avoid conflicts.
- **Chat History Storage**: Verified and optimized the storage of chat history in the `chat_history` table, ensuring user queries, chatbot responses, and related product IDs are accurately recorded to support contextual understanding in subsequent interactions.
- **Error Handling**: Strengthened error handling in the chatbot and order-related backend functions to log and report issues (e.g., Gemini API failures or database errors) more effectively, reducing the risk of undetected errors.

3. Technologies & Tools Used

- **Backend**: Python (FastAPI), MySQL
- **Frontend**: (No changes; previously used HTML, CSS, JavaScript)
- **Chatbot Integration**: Gemini API (Python)
- **Other**: Logging for debugging

4. Outcomes

- The chatbot now delivers polite, professional, and contextually accurate responses, significantly improving user satisfaction.
- Contact information queries are handled with engaging and inviting responses, enhancing the user experience.
- Customer information is stored accurately and efficiently in the `orders` table, ensuring stable order management system performance.
- The system is more robust, user-friendly, and reliable, moving closer to a production-ready state with improved chatbot performance and order management.