BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

# Second Report

# Foundation Internship

## *Project Title: Smart Shop AI Assistant*

| | |
|---|---|
| Instructor: | Kim Ngoc Bach |
| Student Name : | Le Tran Quoc Bao |
| Student ID : | B22DCVT050 |
| Lớp : | E22CQCN05-B |

*Hà Nội - 2024*

# INTERNSHIP BASE REPORT - WEEK 2

During the second week of my foundation internship, I dedicated my efforts to advancing the development of the "Smart Shop AI Assistant" project under the guidance of my instructor, Kim Ngọc Bách. This week marked significant progress as I successfully completed several critical tasks that form the backbone of the e-commerce platform. These tasks included integrating an online payment API, setting up an automated email notification system, and incorporating a chatbot to enhance customer interaction. Below, I provide an exhaustive account of the work accomplished, challenges encountered, and my plans moving forward.

## 1. Payment API Integration

One of the primary focuses this week was integrating a robust online payment system to enable seamless transactions within the "Smart Shop AI Assistant" platform. This involved multiple steps, each requiring careful consideration and execution:

- **Selection of Payment Gateway:** My first task was to research and select an appropriate payment gateway suitable for the project's needs. After evaluating several options—such as Stripe, PayPal, and local solutions—I opted for VietQR to generate QR codes for payments and the MBBank library to monitor transaction histories. This decision was driven by VietQR's widespread use in Vietnam, its simplicity for end-users, and MBBank's reliable API for transaction verification, both of which align well with the project's technical requirements and local context.
- **Configuration of the Payment Processing System:** Once the tools were chosen, I proceeded to configure the entire payment workflow. This process entailed:
    - **Order Creation:** I developed a mechanism to generate orders based on the user's cart contents, ensuring that product details, quantities, and total costs were accurately reflected in the system.
    - **QR Code Generation:** Using VietQR, I implemented a feature that dynamically generates a unique QR code for each transaction, which users can scan using their mobile banking apps to complete payments.
    - **Payment Status Verification:** I integrated the MBBank library to periodically check the transaction history and confirm whether

    payments were successfully processed. This step required establishing secure API calls and handling responses appropriately.
  - **Order Confirmation:** Upon successful payment verification, the system updates the order status in the database and notifies the user of the transaction's completion.
- **Testing and Validation:** To ensure the payment system's reliability, I conducted extensive testing across various scenarios. These included successful payments, failed transactions due to insufficient funds, interrupted network connections, and timeouts. The goal was to identify and address any potential weaknesses in the system, ensuring a smooth experience for users under real-world conditions.

**Challenges Encountered:** Despite the progress, I encountered a notable issue: the system does not yet automatically reduce the product quantity in the inventory database after a successful payment. This oversight means that the stock levels remain unchanged, which could lead to overselling if not addressed. I plan to resolve this in the upcoming week by implementing a stock update function triggered by payment confirmation.

### 2. Automated Email Sending

Another key achievement this week was the development of an automated email notification system to improve user communication and operational efficiency. This task required careful planning and execution to ensure reliability and user satisfaction. Here's a detailed breakdown:

- **SMTP Configuration:** I chose Gmail's SMTP service as the backbone for email delivery due to its widespread availability, robust infrastructure, and straightforward integration process. To set this up, I configured the SMTP server settings, including the host (smtp.gmail.com), port (587), and security protocol (TLS). I also generated an app-specific password for authentication, as Gmail's standard login credentials are insufficient for third-party applications due to security restrictions. This step ensured a secure and stable connection between the system and Gmail's servers.
- **Development of Email Functionalities:** With the SMTP foundation in place, I developed three distinct email functionalities tailored to different user needs:
  - **Order Confirmation Emails:** These emails are sent to customers immediately after a successful payment, containing detailed

information such as the order ID, purchased items, total amount, and estimated delivery date. I designed a clean, professional template to enhance readability and user trust.
    - **Password Reset Emails:** To support account security, I implemented a feature that sends a temporary password reset link to users who request it. This involved generating unique, time-sensitive tokens and embedding them in the email content for seamless account recovery.
    - **Promotional Notifications:** I also created a system to send marketing emails, such as discount announcements or new product launches, to engage customers and drive sales. These emails were designed with customizable templates to allow future flexibility.
- **Testing and Optimization:** To guarantee the system's effectiveness, I conducted rigorous testing. I sent sample emails to various accounts, checking delivery times, formatting consistency across email clients (e.g., Gmail, Outlook), and ensuring they landed in the inbox rather than the spam folder. To avoid spam flags, I optimized email subjects, avoided suspicious keywords, and adhered to best practices for email deliverability. The tests confirmed that the system works reliably, though I noted areas for further refinement, such as adding personalized greetings.

This email system lays a strong foundation for future enhancements, such as order status updates or customer feedback requests.

### 3. Chatbot Integration

The third major task was integrating an AI-powered chatbot to provide real-time customer support and enhance the shopping experience. This component is central to the "Smart Shop AI Assistant" vision, and I approached it with a focus on usability and functionality:

- **Chatbot Platform Selection:** After exploring options like Dialog Flow, Rasa, and Google AI Studio, I selected Google AI Studio API for its advanced natural language processing capabilities, ease of integration, and extensive documentation. This choice allowed me to leverage pre-trained models while still customizing the chatbot to suit the project's specific needs.

- **Conversation Flow Design:** I meticulously designed the chatbot's conversation flows to address common customer interactions. These included:
    - Responding to product-related inquiries (e.g., price, availability, specifications).
    - Offering personalized product recommendations based on user input or past behavior.
    - Assisting with order tracking and basic troubleshooting (e.g., payment issues).

      I created a series of predefined intents and responses, ensuring the chatbot could handle a variety of queries naturally and efficiently.
- **Website Integration:** I embedded the chatbot into the website using a JavaScript-based interface, positioning it as a floating chat window in the bottom-right corner of the screen. This placement ensures accessibility without obstructing the user's browsing experience. The integration involved connecting the chatbot's backend API to the front-end, enabling real-time communication.
- **Testing and Evaluation:** I tested the chatbot extensively by simulating customer interactions, such as asking "Is this product in stock?" or "How do I reset my password?" The chatbot responded accurately in most cases, though it occasionally struggled with ambiguous or complex queries. These tests helped me refine its responses and identify areas for improvement, such as expanding its knowledge base.

The chatbot is now functional and adds significant value to the platform, though further tuning will enhance its intelligence and versatility.

## 4. Plan for Next Week

Building on this week's accomplishments, I have outlined an ambitious yet achievable plan for Week 3 to advance the project further:

- **Front-End Design and Development:** I will design and build the e-commerce website's user interface using HTML for structure, CSS for styling, and JavaScript for interactivity. The focus will be on creating an intuitive, visually appealing layout that includes a homepage, product listings, a shopping cart, and a checkout page. I aim to ensure responsiveness across devices (desktop, tablet, mobile) for a consistent user experience.

- **Front-End to Backend Integration:** I will connect the front-end interface to the existing backend system to enable core functionalities, including:
    - **User Registration and Login:** Implementing secure sign-up and authentication processes with proper session management.
    - **Cart Management:** Allowing users to add, remove, and update items in their cart, with real-time synchronization to the backend database.
- **Performance and Security Optimization:** To safeguard the platform, I will implement measures to protect against common vulnerabilities:
    - **SQL Injection:** Using parameterized queries to prevent malicious database manipulation.
    - **XSS (Cross-Site Scripting):** Sanitizing and encoding user inputs to block script injections.
    - **CSRF (Cross-Site Request Forgery):** Adding anti-CSRF tokens to forms and API requests for secure transactions.

      Additionally, I will address the pending issue of updating product quantities post-payment to maintain accurate inventory levels.