BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Proposal 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

Table of Contents

Introduction

In the era of rapid digital transformation, e-commerce has become an indispensable part of modern life, particularly with the growing demand for premium technology products such as iPhone, iPad, MacBook, Mac Mini, cases, Apple Watch, and many other Apple products. However, the vast array of models, configurations, and price points often poses challenges for customers in making informed purchasing decisions. To address this issue, the **Smart Shop AI Assistant** project is proposed with the goal of developing an intelligent e-commerce platform that integrates artificial intelligence (AI) to deliver a convenient, personalized, and efficient shopping experience.

The project not only focuses on providing a specialized online store with a user-friendly interface but also incorporates advanced features such as an AI-powered chatbot for product consultation, automated payment processing via VietQR and MBBank API, and automated email notifications through SMTP. Notably, the system is designed with a flexible architecture, enabling administrators to easily manage products, revenue, and user information while ensuring scalability to meet future growth demands.

By leveraging modern technologies such as Pure PHP, MySQL, FastAPI, Flask, Gemini API, and popular payment services in Vietnam, **Smart Shop AI Assistant** aims to create a comprehensive e-commerce solution that not only meets customer needs but also enhances operational efficiency for businesses. This proposal will provide a detailed overview of the project's objectives, significance, theoretical foundation, technologies used, technical requirements, and implementation plan, offering a clear and thorough understanding of the practical value the project delivers.

Smart Shop AI Assistant Project Proposal

1. Project Introduction

Project Objectives

The **Smart Shop AI Assistant** project is designed to build a specialized e-commerce platform focused on selling Apple products (iPhone, iPad, MacBook, Mac Mini, cases, Apple Watch, etc.) with integrated artificial intelligence (AI) to enhance user experience. The primary objectives include:

- Developing an e-commerce website with a user-friendly, intuitive interface, enabling customers to easily search for and purchase suitable products.
- Integrating an intelligent AI chatbot to provide product consultation, answer queries, and offer personalized product recommendations based on user needs.
- Automating payment and notification processes using VietQR and MBBank API to ensure fast, secure transactions.
- Creating an efficient administrative system to support product, revenue, and user management.
- Building a flexible, scalable platform capable of accommodating an increasing number of users in the future.

Significance, Rationale, and Importance

- **Significance**:
  In the context of the booming e-commerce landscape in Vietnam and globally, the demand for high-tech products, particularly Apple products, is on the rise. However, customers often struggle to select the right product due to the variety of models, configurations, and price points. This project is significant in providing an AI-integrated solution that simplifies the shopping process, enhances customer satisfaction, and optimizes business performance.
- **Rationale**:
  The application of AI in e-commerce is not only a trend but also a

breakthrough solution for improving customer experience. Technologies such as AI chatbots, automated payments, and intelligent data management reduce manual processing time, enhance accuracy, and improve operational efficiency. Additionally, Vietnam's market, with the widespread adoption of QR code payments and digital banking services like MBBank, provides a favorable environment for integrating these technological solutions.

- **Importance**:
  The project not only addresses modern shopping needs but also contributes to the advancement of e-commerce in Vietnam. By incorporating cutting-edge technologies such as AI, payment APIs, and smart management systems, it enables businesses to compete more effectively in the market while delivering practical value to customers through convenience, speed, and personalization. The system also lays the foundation for future innovations, such as advanced product recommendation algorithms or expansion to other product categories.

2. Theoretical Foundation and Technologies Used

Theoretical Foundation

The project is grounded in the following theoretical principles:

- **E-commerce**: Analysis of modern e-commerce models, focusing on user experience (UX), user interface (UI), and factors influencing purchasing decisions.
- **Artificial Intelligence and Natural Language Processing (NLP)**: Leveraging NLP techniques in the Gemini API to develop a chatbot capable of understanding and responding to user queries naturally while recommending products based on user needs.
- **Database Management**: Applying relational database management system (RDBMS) principles to efficiently store and retrieve user, product, order, and transaction data.

- **API Integration**: Utilizing RESTful protocols to integrate third-party services such as VietQR, MBBank API, and SMTP, ensuring compatibility and security.
- **Process Automation**: Implementing automation principles for payment processing, email notifications, and order management to minimize errors and enhance efficiency.

Technologies Used and Rationale for Selection

- **Pure PHP**:
    - **Rationale**: PHP is a widely used, powerful, and cost-effective programming language for web development. It excels in handling backend tasks (e.g., registration, login, product management) and frontend tasks (e.g., dynamic interface rendering). Additionally, PHP has a large community and extensive library support, which accelerates development.
    - **Application**: Building backend logic, user interfaces, and dynamic pages such as product listings, product details, and the admin dashboard.
- **MySQL**:
    - **Rationale**: MySQL is an open-source, reliable, high-performance relational database management system that integrates seamlessly with PHP. It is ideal for storing complex data such as user information, products, orders, and chatbot conversation histories.
    - **Application**: Storing and managing user, product, transaction, and chatbot data, ensuring fast retrieval and security.
- **VietQR**:
    - **Rationale**: VietQR is a popular QR code payment service in Vietnam, supported by multiple banks. It simplifies the payment process, aligning with the digital banking habits of Vietnamese users.
    - **Application**: Generating QR codes for each order, displayed on the payment page for quick and secure bank transfers.
- **FastAPI and Flask**:

    - **Rationale**: FastAPI and Flask are lightweight, high-performance Python frameworks ideal for building RESTful APIs. FastAPI offers fast processing and support for asynchronous applications, while Flask is simple and easy to deploy. Both are compatible with the Gemini API and other services.
    - **Application**: Developing APIs for product management, order processing, and AI chatbot integration.
- **SMTP**:
    - **Rationale**: SMTP is a standard protocol for sending emails, reliable, and easy to integrate. It enables automated email notifications, enhancing customer experience and professionalism.
    - **Application**: Sending emails for order confirmations, payment verifications, and promotional campaigns.
- **MBBank API**:
    - **Rationale**: MBBank API provides capabilities for retrieving transaction histories and verifying payments automatically, meeting the project's need for automated payment processing. As one of Vietnam's major banks, MBBank ensures reliability.
    - **Application**: Checking transaction statuses and updating orders automatically after customer payments.
- **Gemini API and MySQL Connector**:
    - **Rationale**: The Gemini API offers advanced natural language processing, enabling the chatbot to understand and respond to user queries intelligently. The MySQL Connector allows the chatbot to efficiently query product data from the database, ensuring accurate responses.
    - **Application**: Building a chatbot for product consultation, answering questions, and recommending products based on user needs.

3. Technical Requirements

The project must meet the following technical requirements to ensure quality and performance:

- **User Interface (UI/UX)**:
    - The interface must be user-friendly, intuitive, and support multiple platforms (desktop, mobile).
    - Page load times must be under 2 seconds to ensure a smooth experience.
    - Support fast search, product filtering, and clear display of detailed product information.
- **Security**:
    - Encrypt user data (e.g., email, password) using secure algorithms such as bcrypt.
    - Ensure secure payment transactions via VietQR and MBBank API.
- **Technology Integration**:
    - Integrate VietQR API to generate accurate payment QR codes, clearly displayed on the payment page.
    - Integrate MBBank API to verify transactions within 5 seconds of customer payment.
    - The AI chatbot, powered by Gemini API and MySQL Connector, must accurately respond to at least 95% of user queries.
    - Automate email sending via SMTP with a success rate above 99%.
- **Scalability**:
    - The system must support adding new product categories without requiring major structural changes.
    - A modular architecture to enable integration of new features, such as machine learning-based product recommendations, in the future.
    - The database must scale to handle millions of records without performance degradation.
- **Administrative Capabilities**:
    - The admin interface must enable management of products, users, and revenue with visual charts.
    - Support real-time moderation of user reviews and updates to store policies.

4. Implementation Plan

Weeks 1–2: Analysis and UML Design

- **Objective**: Establish the theoretical foundation and detailed system design.
- **Activities**:
    - **Requirement Analysis**: Gather requirements from customers and administrators, identifying key workflows (registration, purchasing, payment, chatbot consultation).
    - **UML Design**:
        - Create a **Use Case Diagram** to describe user and administrator interactions with the system.
        - Create a **Class Diagram** to define key entities (users, products, orders, transactions).
        - Create **Sequence Diagrams** and **Activity Diagrams** to outline workflows for registration, purchasing, and chatbot consultation.
    - **Database Design**:
        - Design a database schema with key tables.
        - Ensure normalization to optimize queries and avoid data redundancy.
- **Deliverables**: Complete UML documentation and database schema, approved before implementation.

Weeks 3–4: Frontend Development (Pure PHP)

- **Objective**: Build a user-friendly, responsive interface.
- **Activities**:
    - **Project Setup**:
        - Create a PHP project directory structure, including folders.
        - Set up a PHP development environment (e.g., XAMPP).
    - **Interface Development**:
        - **Homepage**: Display product categories, featured products, and promotional banners.

        - **Product Page**: Show product details (description, specifications, price, images), with buttons for adding to cart and reviews.
        - **Cart Page**: Allow users to edit quantities, remove items, and view the total cost.
        - **Registration/Login Pages**: Design forms with input validation.
        - **Virtual Assistant Page**: Integrate a chatbot interface with a text input field and response display.
        - **Admin Page**: Create a dashboard with tabs for managing users, products, and revenue.
    - **Interface Testing**:
        - Test UI/UX across browsers (Chrome, Firefox, Safari) and mobile devices.
        - Ensure responsive design and page load times under 2 seconds.
- **Deliverables**: A complete frontend interface, ready for backend integration.

Weeks 5–6: Backend Development (PHP, FastAPI, Flask, MySQL)

- **Objective**: Develop backend logic and integrate third-party services.
- **Activities**:
    - **Backend Setup**:
        - Install MySQL, FastAPI, and Flask environments.
        - Connect PHP and FastAPI/Flask to the MySQL database.
    - **API Development**:
        - **Product Management API**: Support adding, editing, deleting, and retrieving product lists.
        - **Order Processing API**: Create orders and update statuses (pending payment, paid, shipping).
        - **Chatbot API**:送 consultation requests to the Gemini API and receive responses.
    - **VietQR and MBBank Integration**:

        - Call the VietQR API to generate payment QR codes, store them in the database, and display them on the payment page.
        - Integrate the MBBank API to check transaction histories and update order statuses automatically.
    - **SMTP Configuration**:
        - Set up an SMTP server (e.g., Gmail SMTP) to send emails for order confirmations, payment verifications, and promotions.
        - Verify email delivery success rate.
    - **API Integration with Frontend**:
        - Use PHP's cURL to call APIs from FastAPI/Flask.
        - Display product, order, and chatbot response data on the frontend.
- **Deliverables**: A fully functional backend integrated with VietQR, MBBank, SMTP, and chatbot services.

Weeks 7–8: Finalization, Testing, and Deployment

- **Objective**: Optimize the system, conduct comprehensive testing, and deploy to a live environment.
- **Activities**:
    - **Advanced Feature Enhancements**:
        - Optimize the chatbot: Improve response accuracy and store conversation histories in the database.
        - Add periodic promotional email functionality based on user purchase history.
        - Enhance the admin interface: Add revenue charts (using Chart.js) and product filtering.
    - **Comprehensive Testing**:
        - **Functional Testing**: Ensure registration, purchasing, payment, and chatbot consultation work correctly.
        - **Performance Testing**: Verify page load times, transaction processing, and database query performance.

        - **Security Testing**: Audit API security, data encryption, and user authentication.
    - **Deployment**:
        - Configure an Apache/Nginx server to host the website.
        - Verify functionality in the live environment to ensure no errors.
    - **Documentation and Summary**:
        - Write user and admin manuals.
        - Outline a maintenance plan and propose future features (e.g., machine learning-based product recommendations).
- **Deliverables**: A fully optimized, deployed system ready for operation.