

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



Weekly Report
Foundation Internship
ChippyCash
Smart Chatbot for Easier Expense Management

Instructor: Kim Ngoc Bach
Student Name : Dinh Manh Hung
Student ID : B22DCCN359
Class : E22CQCN05-B

I. PROJECT OVERVIEW

→ Project Title: ChippyCash – Personal Finance Management Application

→ Rationale:

Managing personal expenses is essential in modern life. With the development of AI, personal finance management applications should not only store transactions but also analyze and provide intelligent advice to help users manage finances more effectively.

→ Objectives:

- Build a web-based personal finance management platform.
- Integrate AI to provide financial advice and spending optimization.
- Support data analysis and visualization via charts.

→ Scope:

- Web interface using HTML, CSS, Bootstrap.
- Backend using PHP (transactions) and Python (chatbot).
- Data stored in MySQL.
- Integration with OpenAI API and LlamaIndex.

II. SYSTEM FEATURES

1. User-side features:

- Register, login, logout.
- Manage income and expenses: add, edit, delete transactions.
- Categorize transactions.
- View financial statistics and charts.
- Chat with AI for financial advice.

2. System-side features:

- Manage transaction categories.
- Analyze income/expense ratio.
- Suggest savings plans based on the 50/30/20 model.
- Store chat history and analyze spending patterns.

III. TECHNOLOGIES USED

→ Frontend:

- HTML, CSS, JavaScript
- PHP
- Bootstrap, Chart.js

→ Backend:

- PHP: manage transactions and users
- Python: implement AI and NLP
- MySQL: data storage
- FastAPI: for chatbot API

→ AI:

- OpenAI API (GPT-4)
- LlamaIndex for vector storage and intelligent suggestions

→ Tools:

- VS Code, XAMPP
- Git & GitHub

IV. PROJECT IMPLEMENTATION PLAN

- Week 1–2: Analysis and Design

- + UI design using Figma
- + Database design (MySQL): Users, Transactions, Categories, ChatHistory
- Week 3–5: Backend Development
 - + PHP API for transactions and users
 - + FastAPI Python: integrate OpenAI API, LlamaIndex
- Week 6–8: Frontend Development
 - + Build dashboard, charts, chatbot interfaces
 - + Integrate backend APIs using AJAX/Axios
- Week 9–10: Finalization and Testing
 - + Optimize performance and fix bugs
 - + Write user and developer documentation

V. AI CHATBOT DEPLOYMENT

→ Main functions:

- Analyze income and expenses, suggest savings
- Answer finance-related questions
- Recommend better spending habits

→ Implementation:

- Use FastAPI to connect OpenAI API
- Store chat data using LlamaIndex vector store
- Communicate via REST API between frontend and chatbot

VI. TESTING PLAN

→ Unit Testing:

- Test APIs: authentication, transaction processing
- Test category management and statistics functions

→ Integration Testing:

- Test chatbot operation in frontend
- Test data synchronization between modules

VII. DEPLOYMENT PLAN

→ Requirements:

- PHP + MySQL web server (Apache/XAMPP)
- Python server for FastAPI and AI services

→ Steps:

- Configure database and environment
- Deploy frontend and backend
- Perform overall testing and go live

VIII. SECURITY CONSIDERATIONS

→ API:

- Store API keys in .env files
- Authenticate access to AI endpoints

→ User:

- Passwords encrypted with bcrypt
- Validate session logins
- Role-based access control

→ Data:

- Regular backups
- Prevent SQL injection