TRƯỜNG ĐẠI HỌC AN GIANG

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN**

**Chuyên đề Python (COS525)**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ BÁN HÀNG VỚI PYTHON, TKINTER VÀ MYSQL**

Giảng viên hướng dẫn: Ths. Nguyễn Ngọc Minh

Sinh viên thực hiện:

DTH235763.Nguyễn Trần Quốc Thái.DH24TH3\_NhómTH3\_TổTH2

DTH235777.Nguyễn Huỳnh Ngọc Thịnh.DH24TH3\_NhómTH3\_TổTH2

**AN GIANG, 11-2025**

MỤC LỤC Trang

[I. ĐẶT VẤN ĐỀ 1](#_hii0r49so535)

[1. Tính cần thiết của đề tài. 1](#_5wu3lgyybnyp)

[2. Mục tiêu nghiên cứu của đề tài. 1](#_ir1y9hcj0v2z)

[II. TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT 2](#_ur39ph5c5kcr)

[1. Đặt vấn đề. 2](#_c9bbz6hcbd8b)

[2. Lịch sử giải quyết vấn đề. 2](#_luu0xm95yb6i)

[3. Phạm vi và mục tiêu. 2](#_cpkcqy30tsz1)

[4. Phương pháp nghiên cứu/hướng giải quyết vấn đề. 2](#_tvd4h6zcxco3)

[5. Tóm tắt cơ sở lý thuyết. 3](#_jdz6gdslqz0j)

[III. KẾT QUẢ ĐẠT ĐƯỢC VÀ DEMO CODE 4](#_rchh2oedw4rr)

[1. Giao diện hệ thống quản lý bán hàng. 4](#_n03qr5qeomkw)

[2. Các bước xây dựng ứng dụng. 14](#_xczbucc64x9u)

[2.1. Cài đặt các thư viện cần thiết. 14](#_3nz6mvkycmt1)

[2.2. Thiết kế cơ sở dữ liệu MySQL. 15](#_sqwffuq5r02u)

[2.3. Xây dựng giao diện người dùng (GUI) bằng Tkinter. 17](#_2hd556l00grr)

[2.4.Viết mã PyThon kết nối cơ sở dữ liệu. 17](#_bbz6zl5xr4me)

[2.5. Xây dựng giao diện Tkinter. 18](#_zfrcl3vk6oq9)

[3. Hướng dẫn cài đặt và code ứng dụng. 19](#_rkmruauwlkf2)

[3.1.Cài đặt môi trường. 19](#_fwr00d17owf6)

[3.2. Chuẩn bị CSDL MySQL 19](#_k348x7u05jbc)

[3.3.Viết code ứng dụng với Tkinter + MySQL. 26](#_rbap2fcllw7g)

[4. Kết luận 50](#_uquoe6xq87rn)

# I. ĐẶT VẤN ĐỀ

## 1. Tính cần thiết của đề tài.

- Trong các siêu thị hoặc cửa hàng , việc quản lý thông tin hàng hoá, nhân viên, khách hàng và doanh thu thường được thực hiện thủ công, dễ sai sót và tốn thời gian. Vì vậy, việc xây dựng một ứng dụng quản lý bán hàng giúp tự động hóa quy trình quản lý, lưu trữ dữ liệu tập trung, tra cứu nhanh chóng là rất cần thiết và mang tính thực tiễn cao.

- Đề tài nhằm ứng dụng ngôn ngữ Python cùng thư viện Tkinter để xây dựng giao diện quản lý thân thiện, kết hợp MySQL để lưu trữ và truy xuất dữ liệu hiệu quả.

## 2. Mục tiêu nghiên cứu của đề tài.

- Xây dựng một phần mềm có khả năng:

+ Quản lý thông tin hàng hoá (mã hàng, tên hàng, số lượng, đơn giá)

+ Quản lý thông tin nhân viên (mã nhân viên, họ tên, vị trí, ngày vào làm, mức lương).

+ Quản lý khách hàng (mã khách hàng, họ tên, thông tin liên hệ, địa chỉ).

+Theo dõi doanh thu, số lượng hàng bán được.

- Mục tiêu là giúp người dùng quản lý dữ liệu nhanh chóng, trực quan và giảm thiểu sai sót khi nhập liệu thủ công.

# II. TỔNG QUAN VÀ CƠ SỞ LÝ THUYẾT

## 1. Đặt vấn đề.

- Trước đây, việc quản lý hàng, nhân viên và khách hàng thường thực hiện bằng Excel hoặc ghi chép thủ công, dẫn đến việc trùng lặp thông tin, thiếu liên kết giữa các bảng dữ liệu, và khó khăn khi cần truy xuất lịch sử hoạt động. Do đó, cần một hệ thống phần mềm chuyên biệt để quản lý thông tin một cách nhất quán, chính xác và dễ sử dụng.

## 2. Lịch sử giải quyết vấn đề.

- Đã có nhiều phần mềm quản lý bán hàng chuyên nghiệp, tuy nhiên hầu hết là sản phẩm thương mại, phức tạp và không phù hợp với quy mô nhỏ (trường học, trung tâm, doanh nghiệp vừa và nhỏ). Đề tài này hướng đến việc xây dựng một ứng dụng gọn nhẹ, miễn phí, phù hợp nhu cầu cơ bản.

## 3. Phạm vi và mục tiêu.

- Đề tài tập trung vào việc xây dựng ứng dụng desktop (chạy trên Windows) có thể:

+ Quản lý hàng hoá: Lưu trữ và cập nhật thông tin hàng như mã hàng, tên hàng, số lượng, giá.

+ Quản lý nhân viên: Lưu trữ thông tin nhân viên như họ tên, mã nhân viên, chức vụ, mức lương.

+ Quản lý khách hàng: Lưu thông tin khách hàng, hỗ trợ tìm kiếm nhanh.

+ Quản lý doanh thu: thể hiện doanh thu và số lượng hàng bán được.

- Xây dựng một hệ thống giúp người dùng quản lý dữ liệu nhanh chóng, trực quan, hạn chế sai sót khi nhập liệu thủ công và nâng cao hiệu quả trong việc quản lý bán hàng – mặt hàng – nhân viên – khách hàng.

## 4. Phương pháp nghiên cứu/hướng giải quyết vấn đề.

- Ứng dụng ngôn ngữ Python kết hợp thư viện Tkinter để xây dựng giao diện đồ họa thân thiện.

- Sử dụng hệ quản trị cơ sở dữ liệu MySQL để lưu trữ và truy xuất dữ liệu.

- Áp dụng mô hình lập trình hướng đối tượng (OOP) để tổ chức dữ liệu và xử lý nghiệp vụ.

- Kiểm thử chương trình qua các tình huống giả lập thực tế trong môi trường kinh doanh .

## 5. Tóm tắt cơ sở lý thuyết.

- Python: là ngôn ngữ lập trình đa năng, dễ học, có thư viện phong phú hỗ trợ GUI và kết nối cơ sở dữ liệu.

- Tkinter: là thư viện chuẩn của Python dùng để tạo giao diện người dùng (GUI) đơn giản và trực quan.

- ttkbootstrap: là thư viện mở rộng dựa trên Tkinter, cung cấp giao diện hiện đại, đẹp mắt theo phong cách Bootstrap. Thư viện này giúp giao diện thân thiện hơn, chuyên nghiệp hơn so với Tkinter mặc định.

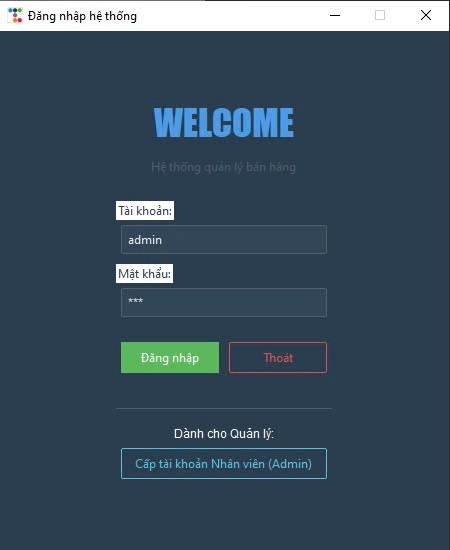
- MySQL: là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến, dễ kết nối với Python thông qua thư viện mysql-connector-python.

- Mô hình CRUD (Create – Read – Update – Delete): là mô hình cơ bản để quản lý dữ liệu trong ứng dụng quản lý.

- Mối liên kết dữ liệu: giữa bảng nhân viên và bảng sản phẩm thông qua khóa ngoại (mỗi nhân viên có thể quản lí cho nhiều sản phẩm khác nhau).

# III. KẾT QUẢ ĐẠT ĐƯỢC VÀ DEMO CODE

## 1. Giao diện hệ thống quản lý bán hàng.



*Hình 1: giao diện đăng nhập hệ thống.*

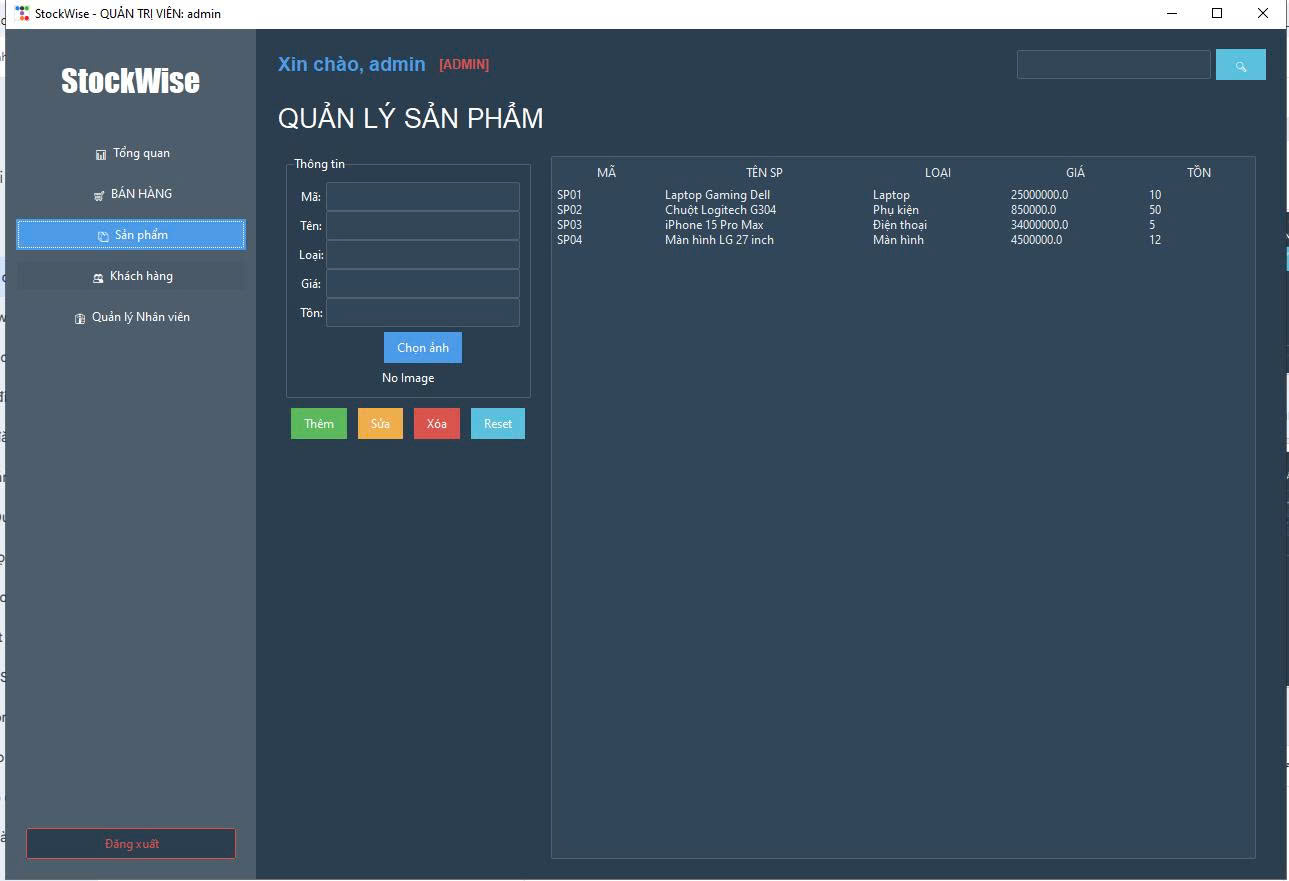
- Đây là màn hình đầu tiên khi khởi chạy phần mềm. Chức năng đăng nhập giúp kiểm soát quyền truy cập, đảm bảo chỉ những người có tài khoản mới được phép sử dụng hệ thống.



*Hình 2: giao diện tổng quan hệ thống.*

- Trang tổng quan hiển thị các chỉ số chính như tổng doanh thu, tổng khách hàng và tổng sản phẩm. Bên dưới có

biểu đồ cột minh họa doanh thu theo tháng.. Menu trái giúp điều hướng sang các chức năng quản lý khác.

**

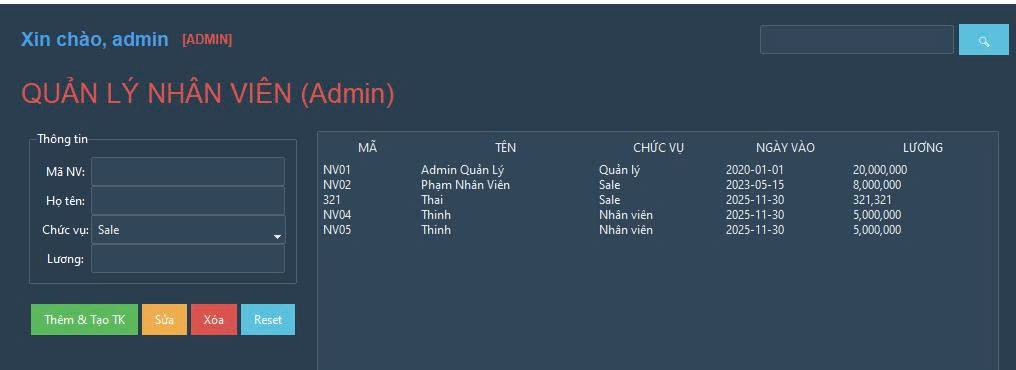
*Hình 3: giao diện quản lý sản phẩm.*

- Hiển thị tổng sản phẩm và danh sách sản phẩm gồm mã sp, loại sp, tên sp, giá và hàng tồn . Có chức năng thêm mới, chỉnh sửa và lọc theo trạng thái. Trang này giúp quản lý tình trạng và lịch sử của các sản phẩm.

**

*Hình 4: Giao diện thêm sp mới.*

- Hiển thị biểu mẫu cho phép nhập thông tin sản phẩm gồm mã, tên, loại, giá gần nhất, tài xế phụ trách và trạng . Cung cấp nút Sửa để chỉnh sửa thông tin sản phẩm và nút Xoá để xoá sản phẩm. Giao diện hỗ trợ quản lý việc cập nhật thông tin sản phẩm một cách nhanh chóng và chính xác.



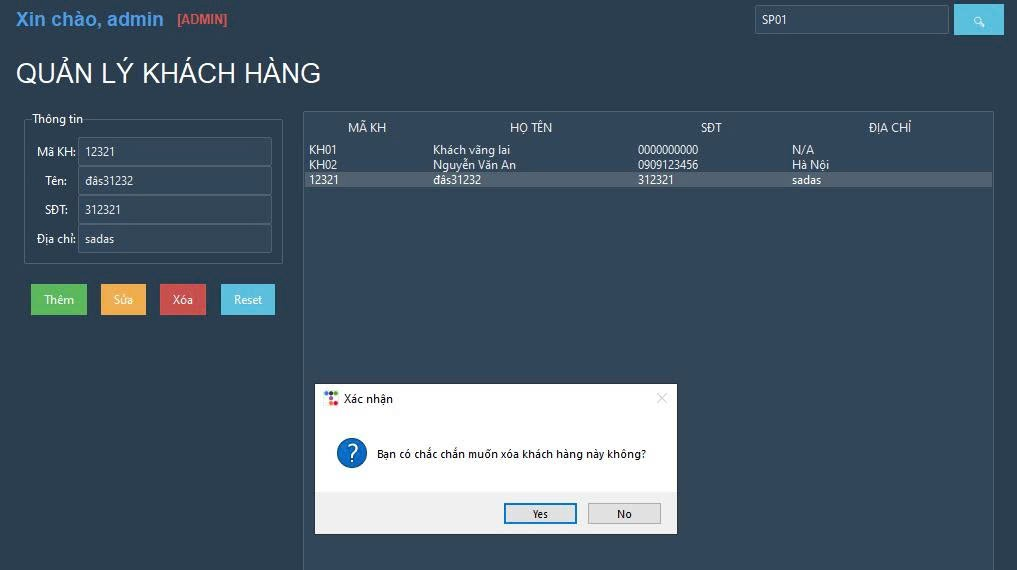
*Hình 5 : giao diện quản lý nhân viên.*

- Trang hiển thị số lượng nhân viên và danh sách chi tiết như họ tên, mã nhân viên, chức vụ, ngày vào và lương. Người dùng có thể thêm, sửa hoặc xoá nhân viên. Giao diện hỗ trợ tìm kiếm và lọc thông tin.



*Hình 6: giao diện thêm nhân viên mới.*

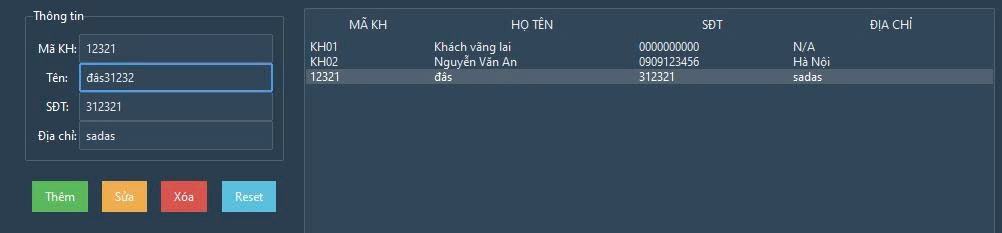
- Đây là form nhập dữ liệu nhân viên mới như họ tên, mã nhân viên, chức vụ, ngày vào và lương. Giao diện được thiết kế để nhập nhanh, kèm nút Sửa và Xoá thao tác.



*Hình 7: giao diện quản lý khách hàng.*

- Trang thống kê và liệt kê khách hàng. Danh sách bao gồm mã khách hàng, họ tên,

số điện thoại và địa chỉ. Người dùng có thể thêm, sửa hoặc xoá thông tin khách hàng.



*Hình 8: giao diện thêm khách hàng mới.*

- Hiển thị biểu mẫu nhập thông tin khách hàng gồm họ tên, số điện thoại và địa chỉ. Cung cấp nút Sửa để chỉnh sửa thông tin khách hàng vào hệ thống và nút Xoá để xoá dữ liệu khách hàng. Giao diện hỗ trợ

quản lý danh sách khách hàng một cách thuận tiện và chính xác

## 2. Các bước xây dựng ứng dụng.

### 2.1. Cài đặt các thư viện cần thiết.

- Python: Đảm bảo bạn đã cài đặt Python trên máy tính (khuyến nghị phiên bản 3.8 trở lên).

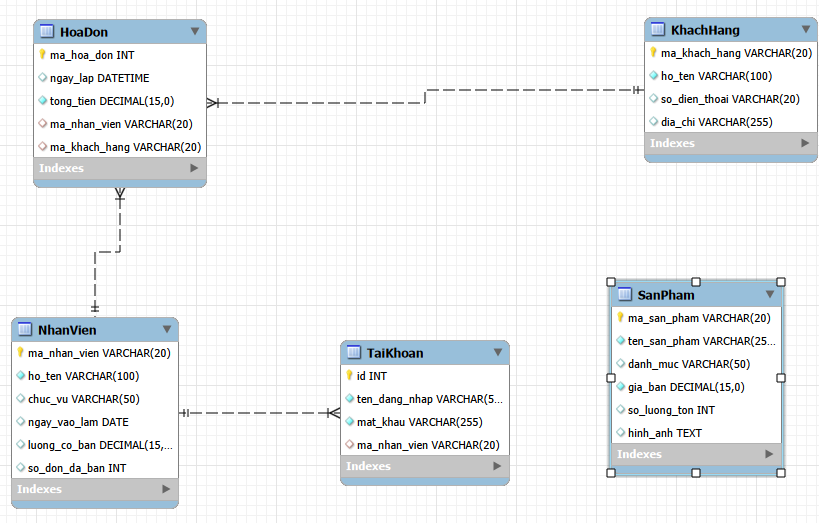
- Tkinter: Thư viện giao diện của Python (thường được cài sẵn). Trường hợp chưa có, cài bằng: pip install tk.

- ttkbootstrap: là thư viện mở rộng dựa trên Tkinter.

- mysql-connector-python: Thư viện để kết nối Python với cơ sở dữ liệu MySQL. (*pip install mysql-connector-python)*

- MySQL: Cài đặt phần mềm MySQL Workbench và tạo database *qlsinhvien* để lưu trữ dữ liệu tài xế, sản phẩm, chuyến đi và khách hàng.

### 2.2. Thiết kế cơ sở dữ liệu MySQL.



*Hình 9: Sơ đồ quan hệ các bảng.*

- Sơ đồ mô tả quan hệ giữa các bảng: khách hàng, nhân viên, sản phẩm, hoá đơn và tài khoản quản trị. Các bảng

liên kết với nhau qua khóa chính – khóa ngoại, thể hiện cấu trúc logic của hệ thống. ERD giúp hiểu tổng thể cấu

trúc dữ liệu và mối liên hệ nghiệp vụ.

### 2.3. Xây dựng giao diện người dùng (GUI) bằng Tkinter.

- Sử dụng thư viện Tkinter và ttkbootstrap để tạo các cửa sổ, trường nhập liệu, nút bấm và bảng hiển thị thông tin của hệ thống quản lý.

- Thiết kế giao diện cho phép người dùng thực hiện các chức năng sau:

Quản lý sản phẩm:

+ Thêm thông tin sản phẩm.  
     + Lưu và cập nhật thông tin sản phẩm.  
     + Sửa thông tin sản phẩm.  
     + Xóa sản phẩm.  
     + Hiển thị danh sách tất cả các sản phẩm.  
     + Tìm kiếm phương tiện theo mã sản phẩm.

Quản lý nhân viên:

+ Thêm nhân viên mới.  
     + Lưu và chỉnh sửa thông tin nhân viên.  
     + Xóa nhân viên.  
     + Sản phẩmm danh sách nhân viên.  
     + Tìm kiếm nhân viên theo mã nhân viên.

Quản lý khách hàng:  
     + Thêm khách hàng mới.  
     + Sửa hoặc cập nhật thông tin khách hàng.  
     + Xóa khách hàng.  
     + Sản phẩmm danh sách khách hàng.  
     + Tìm kiếm khách hàng theo mã khách hàng.

### 2.4.Viết mã PyThon kết nối cơ sở dữ liệu.

- Sử dụng mysql.connector để thiết lập kết nối với cơ sở dữ liệu MySQL.

- Viết các hàm xử lý dữ liệu cho từng chức năng trong hệ thống:

+ Đăng nhập vào hệ thống: Kiểm tra tài khoản, xác định vai trò và đăng nhập vào giao diện phù hợp.

+ Quản lý sản phẩm: Lấy danh sách sản phẩm, thêm sản phẩm mới, sửa thông tin sản phẩm, xóa sản phẩm và tìm kiếm theo mã sản phẩm.

+ Thêm sản phẩm: Nhập dữ liệu từ giao diện và chèn vào bảng *SanPham*.

+ Sửa thông tin sản phẩm: Cập nhật thông tin sản phẩm trong cơ sở dữ liệu.

+ Xóa sản phẩm: Xóa sản phẩm khỏi cơ sở dữ liệu theo mã sản phẩm.

+ Tìm kiếm sản phẩm: Tìm sản phẩm theo mã sản phẩm.

+ Quản lý nhân viên: Hiển thị danh sách nhân viên, thêm, sửa, xóa và tìm kiếm theo tên hoặc mã nhân viên.

+ Thêm nhân viên: Lấy dữ liệu từ giao diện và chèn vào bảng *NhanVien*.

+ Sửa thông tin nhân viên: Cập nhật thông tin nhân viên trong cơ sở dữ liệu.

+ Xóa nhân viên: Xóa nhân viên theo mã nhân viên.

+ Tìm kiếm nhân viên: Tìm theo họ tên hoặc mã nhân viên.

+ Quản lý khách hàng: Hiển thị danh sách khách hàng, thêm mới, sửa, xóa và tìm kiếm theo tên hoặc mã khách hàng.

+ Thêm khách hàng: Chèn khách hàng mới vào bảng *KhachHang*.

+ Sửa thông tin khách hàng: Cập nhật dữ liệu trong cơ sở dữ liệu.

+ Xóa khách hàng: Xóa khách hàng theo mã khách hàng.

+ Tìm kiếm khách hàng: Tìm theo họ tên hoặc mã khách hàng.

### 2.5. Xây dựng giao diện Tkinter.

- Liên kết các nút bấm trên giao diện Tkinter với các hàm Python tương ứng để xử lý dữ liệu và tương tác với cơ sở dữ liệu.

- Các Chức Năng:

+ Đăng nhập vào hệ thống: Người dùng chọn một trong hai vai trò là *quản trị viên* hoặc *nhân viên quản lý* , sau đó nhập mã đăng nhập và truy cập vào hệ thống quản lý bán hàng.

+ Quản lý sản phẩm: Khi người dùng mở cửa sổ ứng dụng, mã Python sẽ truy vấn cơ sở dữ liệu, lấy tất cả thông tin sản phẩm (mã sản phẩm, loại sản phẩm, tên sản phẩm, tồn kho, đơn giá) và hiển thị chúng dưới dạng một danh sách hoặc bảng trên giao diện Tkinter.

+ Thêm sản phẩm: Khi người dùng nhập các thông tin như mã sản phẩm, loại sản phẩm và nhấn nút “Thêm”, mã Python sẽ lấy các giá trị này và thêm một bản ghi mới vào bảng sản phẩm trong MySQL.

+ Sửa thông tin sản phẩm: Khi người dùng chọn một sản phẩm từ danh sách, chỉnh sửa các trường thông tin rồi nhấn “Sửa”, mã Python sẽ cập nhật các giá trị mới vào bảng sản phẩm. Dữ liệu sau khi cập nhật sẽ hiển thị lại trên giao diện.

+ Xóa sản phẩm: Khi người dùng chọn một sản phẩm và nhấn “Xóa”, chương trình sẽ gửi lệnh xóa bản ghi tương ứng trong MySQL. Sau khi xóa thành công, danh sách sản phẩm trên giao diện sẽ tự động được cập nhật.

+ Tìm kiếm sản phẩm: Khi người dùng nhập mã sản phẩm hoặc từ khóa liên quan và nhấn “Tìm kiếm”, Python sẽ thực hiện truy vấn MySQL để lọc ra những sản phẩm phù hợp và hiển thị kết quả ngay trên giao diện.

## 3. Hướng dẫn cài đặt và code ứng dụng.

### 3.1.Cài đặt môi trường.

- Trước tiên cài đặt các thư viện cần thiết:  
 + pip install ttkbootstrap  
 + pip install mysql-connector-python

- Giải thích:

- ttkbootstrap:  
 Dùng để tạo giao diện nâng cao dựa trên Tkinter.  
 + Cung cấp nhiều theme đẹp theo phong cách Bootstrap.  
 + Bổ sung các widget hiện đại như Button, Entry, Frame, Table,

Notebook.

+ Giúp giao diện quản lý sản phẩm, nhân viên và khách hàng

trực quan và dễ sử dụng.

- mysql-connector-python:  
 + Kết nối đến database quản lý sản phẩm.  
 + Gửi câu lệnh SQL (SELECT, INSERT, UPDATE, DELETE).  
 + Nhận dữ liệu trả về từ MySQL để hiển thị lên giao diện.

- Tkinter:  
 + Tạo cửa sổ giao diện chính.  
 + Tạo các nút bấm, ô nhập liệu, bảng hiển thị,…  
 +ttkbootstrap hoạt động dựa trên Tkinter.

### 3.2. Chuẩn bị CSDL MySQL

drop database qlbanhang;

CREATE DATABASE IF NOT EXISTS QLBANHANG

CHARACTER SET utf8mb4 COLLATE utf8mb4\_unicode\_ci;

USE QLBANHANG;

--- Bảng 1: TaiKhoanQuanTri (Dùng cho đăng nhập hệ thống/Admin) ---

CREATE TABLE IF NOT EXISTS `TaiKhoanQuanTri` (

`id\_quan\_tri` INT NOT NULL AUTO\_INCREMENT,

`ten\_dang\_nhap` VARCHAR(50) NOT NULL UNIQUE,

`mat\_khau\_hash` VARCHAR(255) NOT NULL,

`ho\_ten` VARCHAR(100) NULL,

`vai\_tro` ENUM('admin', 'manager', 'accountant') NOT NULL DEFAULT 'manager',

`ngay\_tao` TIMESTAMP NULL DEFAULT CURRENT\_TIMESTAMP,

PRIMARY KEY (`id\_quan\_tri`)

) ENGINE=InnoDB;

--- Bảng 2: KhachHang ---

CREATE TABLE IF NOT EXISTS `KhachHang` (

`ma\_khach\_hang` VARCHAR(10) NOT NULL UNIQUE,

`ho\_ten` VARCHAR(100) NOT NULL,

`so\_dien\_thoai` VARCHAR(15) NOT NULL UNIQUE,

`hang\_thanh\_vien` ENUM('Đồng', 'Bạc', 'VIP') NOT NULL DEFAULT 'Đồng',

`dia\_chi` varchar(100) not null,

PRIMARY KEY (`ma\_khach\_hang`)

) ENGINE=InnoDB;

--- Bảng 3: NhanVien ---

CREATE TABLE IF NOT EXISTS `NhanVien` (

`ma\_nhan\_vien` VARCHAR(10) NOT NULL UNIQUE COMMENT 'Mã nhân viên (VD: NV001)',

`ho\_ten` VARCHAR(100) NOT NULL,

`so\_dien\_thoai` VARCHAR(15) NOT NULL UNIQUE,

`chuc\_vu` VARCHAR(50) NOT NULL DEFAULT 'Nhân viên bán hàng',

`ngay\_vao\_lam` DATE NOT NULL,

PRIMARY KEY (`ma\_nhan\_vien`)

) ENGINE=InnoDB;

--- Bảng 4: SanPham ---

CREATE TABLE IF NOT EXISTS `SanPham` (

`ma\_san\_pham` VARCHAR(20) NOT NULL UNIQUE COMMENT 'Mã SKU/BarCode của sản phẩm',

`ten\_san\_pham` VARCHAR(255) NOT NULL,

`don\_vi\_tinh` VARCHAR(20) NOT NULL COMMENT 'VD: Cái, Chiếc, Hộp...',

`gia\_ban` DECIMAL(12, 0) NOT NULL,

`so\_luong\_ton` INT NOT NULL DEFAULT 0,

`trang\_thai` ENUM('Đang bán', 'Ngừng kinh doanh', 'Hết hàng') NOT NULL DEFAULT 'Đang bán',

`ma\_nhan\_vien\_quan\_ly` VARCHAR(10) NULL COMMENT 'Nhân viên chịu trách nhiệm quản lý kho',

PRIMARY KEY (`ma\_san\_pham`),

INDEX `fk\_SanPham\_NhanVien\_idx` (`ma\_nhan\_vien\_quan\_ly` ASC) VISIBLE,

CONSTRAINT `fk\_SanPham\_NhanVien`

FOREIGN KEY (`ma\_nhan\_vien\_quan\_ly`)

REFERENCES `NhanVien` (`ma\_nhan\_vien`)

ON DELETE SET NULL

ON UPDATE CASCADE

) ENGINE=InnoDB;

--- Bảng 5: HoaDon ---

CREATE TABLE IF NOT EXISTS `HoaDon` (

`id\_hoa\_don` INT NOT NULL AUTO\_INCREMENT,

`ma\_khach\_hang` VARCHAR(10) NOT NULL UNIQUE,

`ma\_nhan\_vien` VARCHAR(10) NOT NULL COMMENT 'Nhân viên thực hiện giao dịch',

`ma\_san\_pham` VARCHAR(20) NOT NULL UNIQUE,

`ten\_san\_pham` VARCHAR(255) NOT NULL,

`ngay\_lap\_hoa\_don` TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,

`tong\_tien\_phai\_tra` DECIMAL(12, 0) NOT NULL,

`tien\_khach\_thanh\_toan` DECIMAL(12, 0) NOT NULL,

`phuong\_thuc\_thanh\_toan` ENUM('Tiền mặt', 'Chuyển khoản', 'Thẻ', 'Ví điện tử') NOT NULL DEFAULT 'Tiền mặt',

`trang\_thai\_thanh\_toan` ENUM('Đã thanh toán', 'Chưa thanh toán', 'Đã hủy') NOT NULL DEFAULT 'Đã thanh toán',

PRIMARY KEY (`id\_hoa\_don`),

INDEX `fk\_HoaDon\_KhachHang\_idx` (`ma\_khach\_hang` ASC) VISIBLE,

INDEX `fk\_HoaDon\_NhanVien\_idx` (`ma\_nhan\_vien` ASC) VISIBLE,

CONSTRAINT `fk\_HoaDon\_KhachHang`

FOREIGN KEY (`ma\_khach\_hang`)

REFERENCES `KhachHang` (`ma\_khach\_hang`)

ON DELETE NO ACTION

ON UPDATE CASCADE,

CONSTRAINT `fk\_HoaDon\_NhanVien`

FOREIGN KEY (`ma\_nhan\_vien`)

REFERENCES `NhanVien` (`ma\_nhan\_vien`)

ON DELETE NO ACTION

ON UPDATE CASCADE

) ENGINE=InnoDB;

--- Dữ liệu Tài khoản Quản trị ---

INSERT INTO `TaiKhoanQuanTri` (`ten\_dang\_nhap`, `mat\_khau\_hash`, `ho\_ten`, `vai\_tro`)

VALUES

('admin', '202cb962ac59075b964b07152d234b70', 'Quản Trị Viên Chính', 'admin'),

('manager01', '202cb962ac59075b964b07152d234b70', 'Nguyễn Huỳnh Ngọc Thịnh', 'manager')

ON DUPLICATE KEY UPDATE mat\_khau\_hash='202cb962ac59075b964b07152d234b70';

--- Dữ liệu Nhân viên ---

INSERT INTO `NhanVien` (`ma\_nhan\_vien`, `ho\_ten`, `so\_dien\_thoai`, `chuc\_vu`, `ngay\_vao\_lam`)

VALUES

('NV001', 'Nguyễn Văn An', '0901234567', 'Quản lý Kho', '2025-10-04'),

('NV002', 'Trần Văn Bình', '0912345678', 'Nhân viên bán hàng', '2023-06-12'),

('NV003', 'Lê Thị Cẩm', '0923456789', 'Kế toán', '2023-12-22')

ON DUPLICATE KEY UPDATE ma\_nhan\_vien=ma\_nhan\_vien;

--- Dữ liệu Khách hàng (Tái sử dụng) ---

INSERT INTO `KhachHang` (`ma\_khach\_hang`,`ho\_ten`, `so\_dien\_thoai`, `hang\_thanh\_vien`, `dia\_chi`)

VALUES

('C001','Nguyễn Thị An', '0901111222', 'Bạc', 'An Giang'),

('C002','Trần Văn Bình', '0912222333', 'Bạc', 'Cần Thơ'),

('C003','Phạm Văn Cường', '0987654321', 'VIP', 'Đồng Tháp')

ON DUPLICATE KEY UPDATE ma\_khach\_hang=ma\_khach\_hang;

--- Dữ liệu Sản phẩm ---

INSERT INTO `SanPham` (`ma\_san\_pham`, `ten\_san\_pham`, `don\_vi\_tinh`, `gia\_ban`, `so\_luong\_ton`, `trang\_thai`, `ma\_nhan\_vien\_quan\_ly`)

VALUES

('LAPTOP-001', 'Laptop Dell XPS 13', 'Chiếc', 25000000, 15, 'Đang bán', 'NV001'),

('PHONE-005', 'Điện thoại Samsung S24', 'Cái', 18000000, 50, 'Đang bán', 'NV002'),

('ACC-101', 'Chuột không dây Logitech', 'Cái', 450000, 150, 'Đang bán', 'NV001'),

('MONITOR-27', 'Màn hình Dell 27 inch', 'Cái', 6500000, 5, 'Hết hàng', NULL)

ON DUPLICATE KEY UPDATE ma\_san\_pham=ma\_san\_pham;

--- Dữ liệu Hóa đơn ---

INSERT INTO `HoaDon` (`id\_hoa\_don`,`ma\_khach\_hang`, `ma\_nhan\_vien`,`ma\_san\_pham`,`ten\_san\_pham`, `tong\_tien\_phai\_tra`, `tien\_khach\_thanh\_toan`, `phuong\_thuc\_thanh\_toan`, `trang\_thai\_thanh\_toan`, `ngay\_lap\_hoa\_don`)

VALUES

(1,'C001', 'NV002','LAPTOP-001','Laptop Dell XPS 13', 25000000, 25450000, 'Thẻ', 'Đã thanh toán', '2025-11-10 10:30:00'),

(2,'C002', 'NV002','PHONE-005', 'Điện thoại Samsung S24', 18000000, 18450000, 'Chuyển khoản', 'Đã thanh toán', '2025-11-11 14:00:00'),

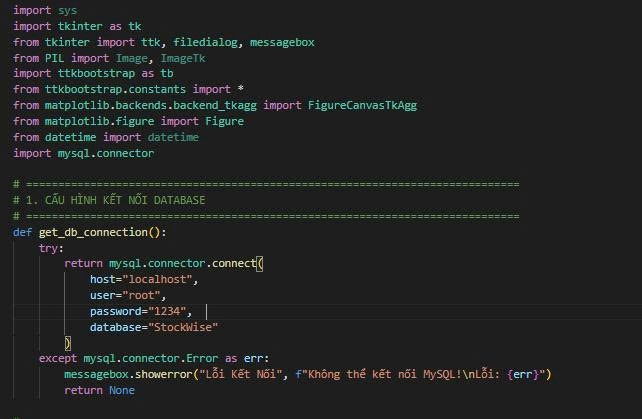
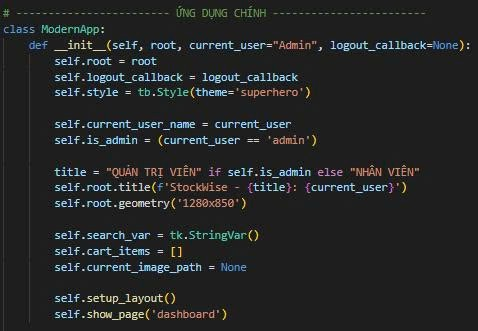
(3,'C003', 'NV001','ACC-101', 'Chuột không dây Logitech', 450000, 500000, 'Tiền mặt', 'Đã thanh toán', '2025-11-12 11:00:00'),

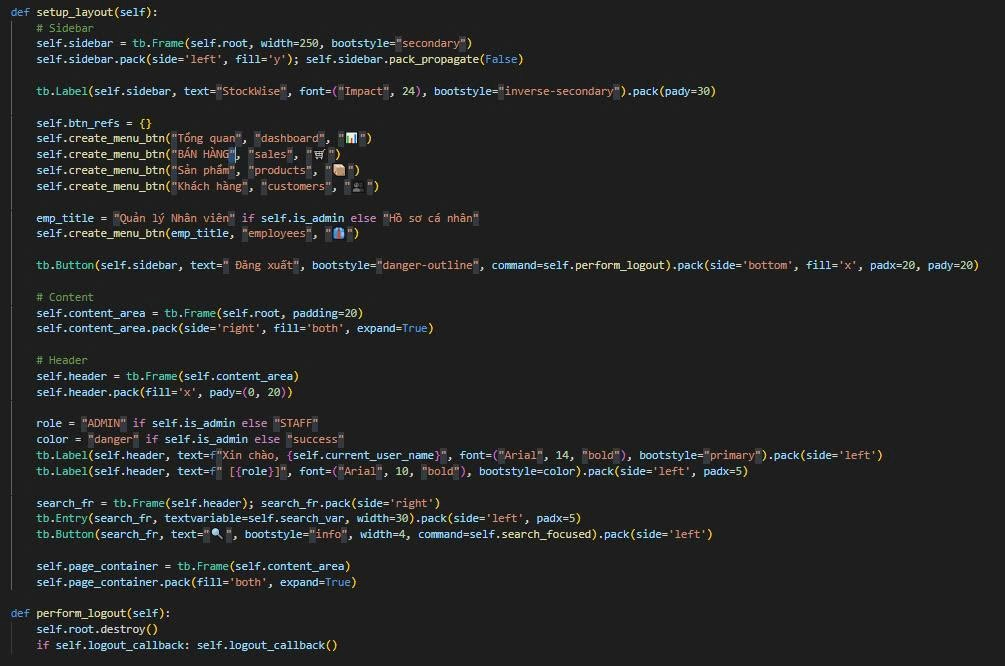
(4,'C001', 'NV002','PHONE-005', 'Điện thoại Samsung S24', 18000000, 18000000, 'Thẻ', 'Đã thanh toán', '2025-11-14 16:00:00'),

(5,'C003', 'NV003','MONITOR-27', 'Màn hình Dell 27 inch', 6500000, 6500000, 'Chuyển khoản', 'Đã hủy', '2025-11-15 09:00:00')

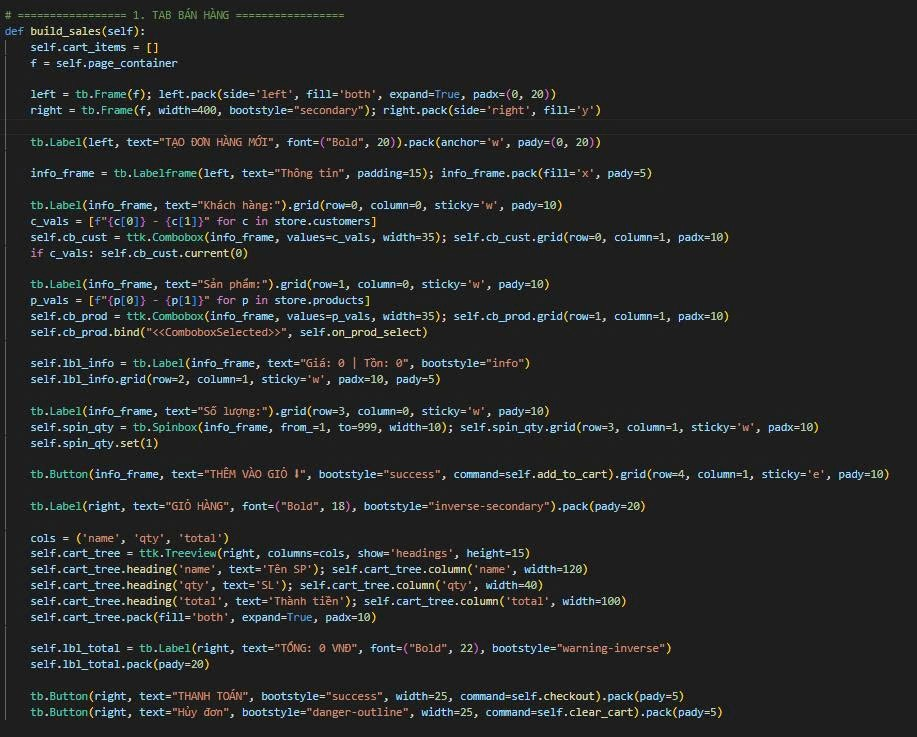
ON DUPLICATE KEY UPDATE id\_hoa\_don=id\_hoa\_don;

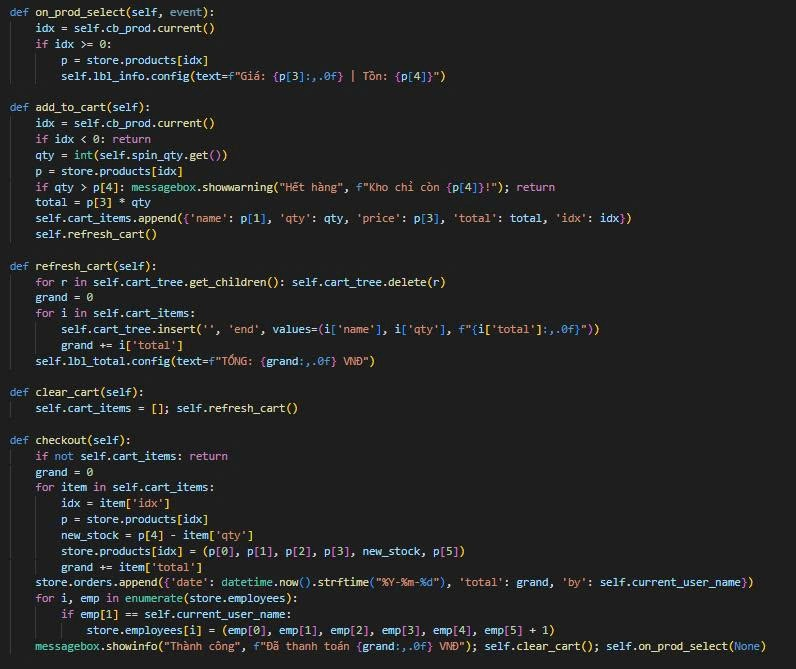
Kết nối python với database (db\_connect.py):

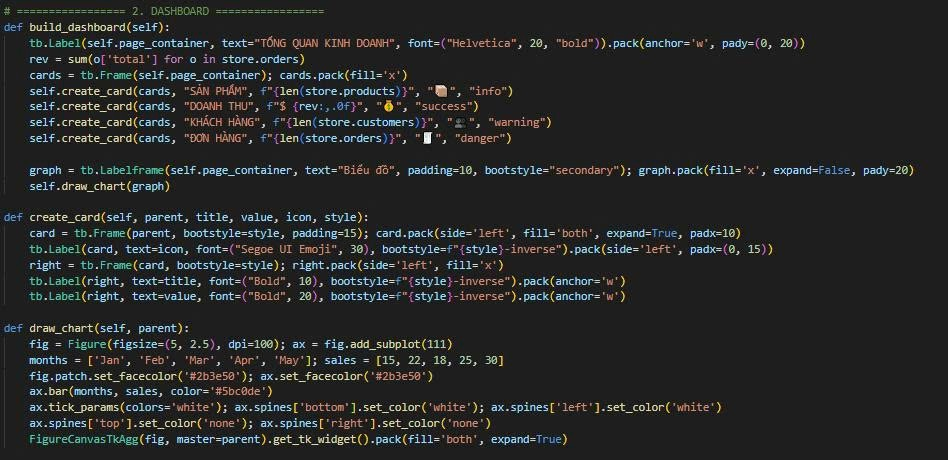
  


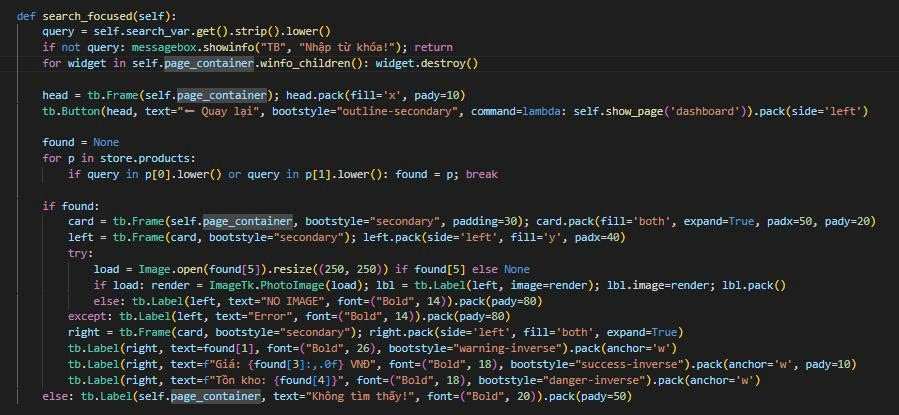


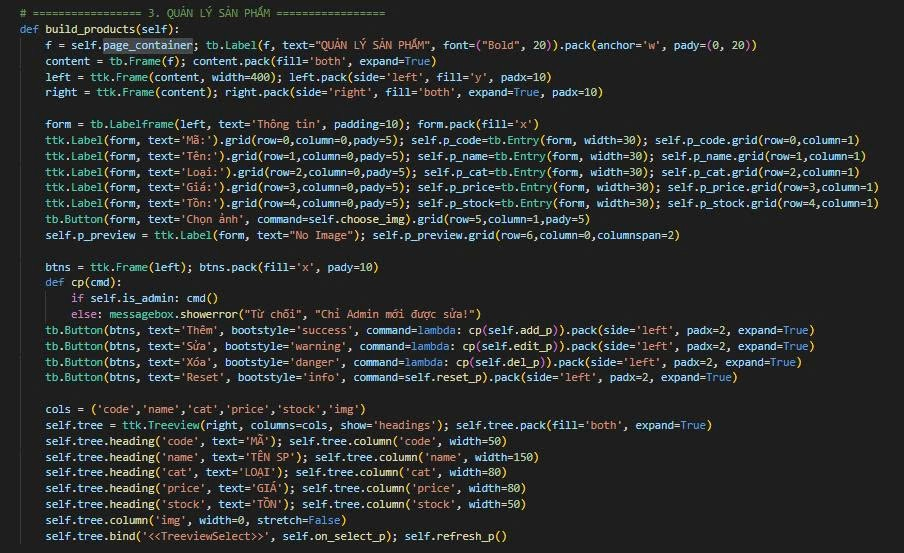


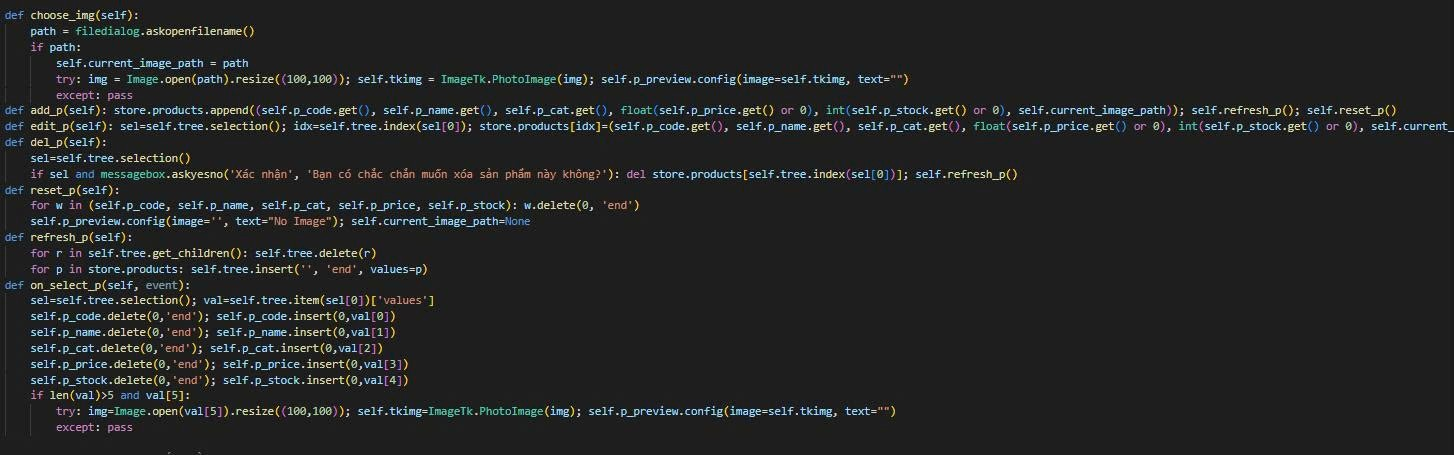


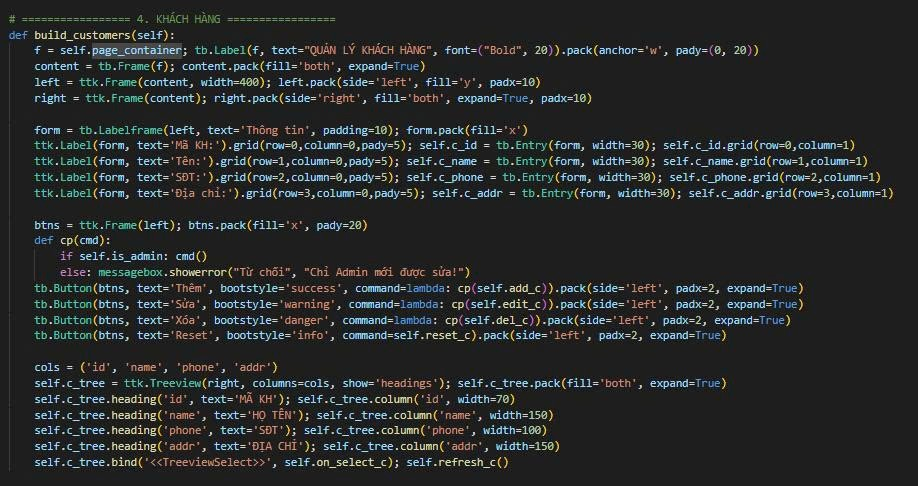


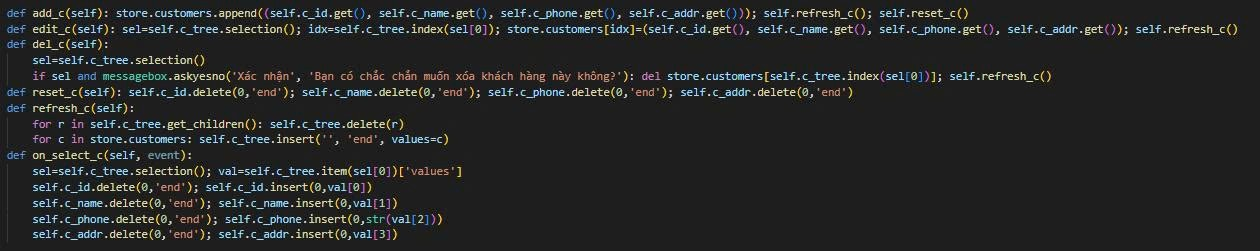




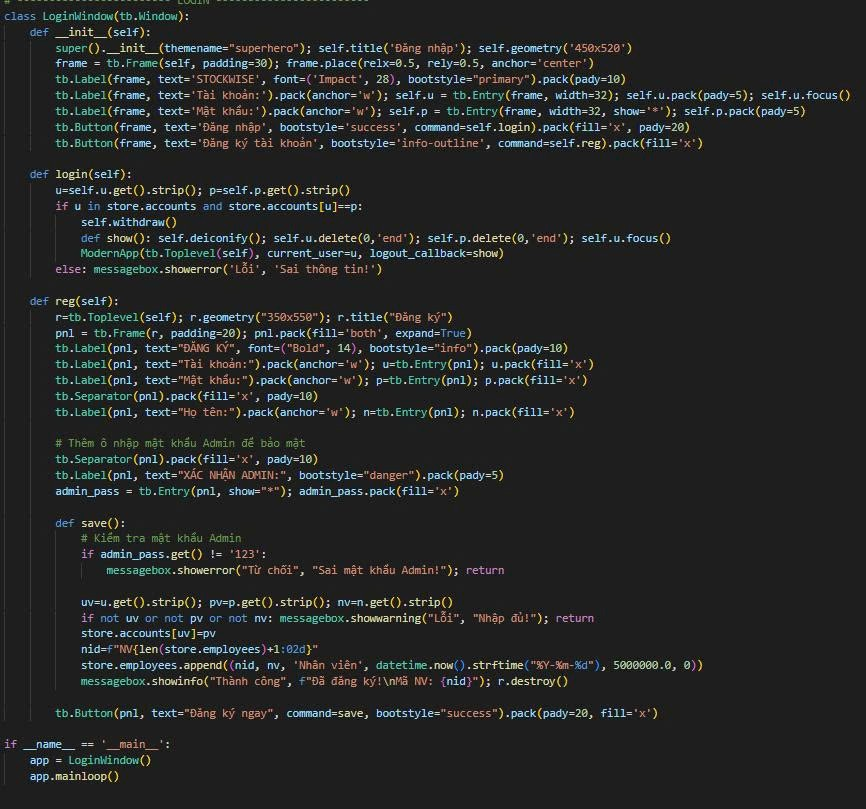








## 



## 4. Kết luận

- Đề tài *“Xây dựng ứng dụng quản lý bán hàng với Python, Tkinter và MySQL”* được nhóm thực hiện với số lượng 2 thành viên và đã hoàn thành đầy đủ các mục tiêu đề ra. Ứng dụng cho phép quản lý các dữ liệu quan trọng của hệ thống bán hàng thông qua các chức năng như: thêm mới, chỉnh sửa, xóa, tìm kiếm và sản phẩmm danh sách theo từng mục. Các phân hệ chính bao gồm: quản lý sản phẩm, quản lý nhân viên, quản lý khách hàng và quản lý hóa đơn. Việc sử dụng Tkinter kết hợp ttkbootstrap giúp giao diện trực quan và dễ thao tác, trong khi MySQL đảm bảo khả năng lưu trữ dữ liệu chính xác, đồng bộ và ổn định.

- Trong quá trình phát triển, nhóm cũng gặp nhiều khó khăn, đặc biệt là ở việc thiết kế giao diện với Tkinter — vốn cần xử lý nhiều bố cục, phân chia frame và cập nhật dữ liệu thời gian thực. Bên cạnh đó, các thao tác kết nối, truy vấn và ràng buộc dữ liệu MySQL đôi lúc phát sinh lỗi cú pháp, lỗi khóa ngoại hoặc lỗi hiển thị khiến tiến độ nhóm bị gián đoạn. Cả hai thành viên phải chủ động tìm hiểu thêm tài liệu, sản phẩmm ví dụ mẫu, thử nghiệm nhiều cách bố trí giao diện và tối ưu các câu truy vấn để ứng dụng chạy ổn định hơn.

- Dù gặp không ít trở ngại, nhưng nhờ sự trao đổi thường xuyên, phân chia nhiệm vụ hợp lý và cùng nhau xử lý các vấn đề kỹ thuật phát sinh, nhóm đã từng bước hoàn thiện toàn bộ hệ thống. Thông qua quá trình này, nhóm rút ra nhiều bài học quan trọng như: cần lập kế hoạch rõ ràng trước khi lập trình, kiểm thử từng module nhỏ để dễ phát hiện lỗi, xây dựng cấu trúc thư mục – mã nguồn theo mô hình phù hợp, và đặc biệt là kỹ năng phối hợp nhóm, hỗ trợ lẫn nhau trong quá trình giải quyết vấn đề.

- Tóm lại, đề tài không chỉ mang lại một sản phẩm phần mềm có tính ứng dụng thực tiễn trong hoạt động quản lý vận tải nhỏ và vừa, mà còn giúp cả nhóm nâng cao kỹ năng lập trình Python, kỹ năng thiết kế giao diện với Tkinter, khả năng làm việc với cơ sở dữ liệu MySQL, cùng với kinh nghiệm xử lý lỗi, tổ chức dự án và tinh thần hợp tác trong phát triển phần mềm.

**Tài liệu tham khảo:**

- Huỳnh Phước Hải (chủ biên), Nguyễn Văn Hóa (2020). *Giáo trình Lập Trình Python*. Nhà xuất bản Đại học Quốc gia TP. Hồ Chí Minh.

- OpenAI (2025). *ChatGPT – Hệ thống hỗ trợ trả lời tự động ứng dụng trí tuệ nhân tạo*. Truy cập từ: <https://chat.openai.com>

- Google DeepMind (2025). *Gemini – Mô hình trí tuệ nhân tạo đa phương thức*. Truy cập từ: <https://gemini.google.com/>