

**TRƯỜNG ĐẠI HỌC AN GIANG
KHOA CÔNG NGHỆ THÔNG TIN**

BÁO CÁO ĐỒ ÁN

Chuyên đề Python (COS525)

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ NHÂN VIÊN
VỚI PYTHON, TKINTER VÀ MYSQL**

Giảng viên hướng dẫn: Ths. Nguyễn Ngọc Minh

Sinh viên thực hiện: DPM245439-Nguyễn Văn Khánh-DH25PM-Nhóm4_Tổ2
DPM245408-Danh Ngô Tấn Đạt-DH25PM-Nhóm4_Tổ2

AN GIANG , 12-2025

Mục Lục I. ĐẶT VẤN ĐỀ.....	
1. Tính cần thiết của đề tài.....	
2. Mục tiêu nghiên cứu.....	
II: TỔNG QUAN VÀ COSỞ LÝ THUYẾT.....	
1. Đặt vấn đề.....	
2. Lịch sử giải quyết vấn đề.....	
3. Phạm vi của đề tài.....	
4. Phương pháp nghiên cứu.....	
5. Cơ sở lý thuyết.....	
III. KẾT QUẢ ĐẠT ĐƯỢC VÀ ĐỀ MOCODE.....	

1. Đặt vấn đề

1.1 Tính cần thiết của đề tài

Bối cảnh thực tiễn: Trong xu thế chuyển đổi số và ứng dụng công nghệ thông tin vào quản lý doanh nghiệp, việc quản lý nhân sự hiệu quả là một yếu tố then chốt giúp nâng cao năng suất và duy trì ổn định tổ chức. Tuy nhiên, tại nhiều doanh nghiệp vừa và nhỏ, đặc biệt là các cửa hàng, công ty khởi nghiệp, việc quản lý thông tin nhân viên vẫn còn được thực hiện thủ công thông qua sổ sách, file Excel rời rạc. Phương pháp này không chỉ tốn thời gian, dễ xảy ra sai sót, mà còn gây khó khăn trong việc tra cứu, cập nhật và báo cáo dữ liệu.

Hạn chế của các giải pháp hiện có: Trên thị trường hiện nay tuy có nhiều phần mềm quản lý nhân sự chuyên nghiệp (HRM), nhưng đa phần có chi phí cao, giao diện phức tạp và yêu cầu người dùng phải được đào tạo bài bản. Điều này gây ra rào cản lớn đối với các đơn vị có nguồn lực tài chính và kỹ thuật hạn chế. Mặt khác, các công cụ miễn phí như Google Sheets hay Excel tuy linh hoạt nhưng thiếu tính bảo mật, khó kiểm soát phiên bản và không có khả năng tích hợp xử lý nghiệp vụ chuyên sâu.

Vấn đề cần giải quyết: Xuất phát từ thực tế đó, việc thiết kế và phát triển một ứng dụng quản lý nhân viên đơn giản, dễ sử dụng, chi phí thấp nhưng vẫn đảm bảo đầy đủ tính năng cơ bản là một nhu cầu cấp thiết. Ứng dụng cần cho phép người dùng thực hiện các thao tác quản lý cốt lõi như thêm, sửa, xóa, tìm kiếm và lưu trữ thông tin nhân viên một cách hệ thống, đồng thời có khả năng kết nối với cơ sở dữ liệu để lưu trữ lâu dài và an toàn.

Ý nghĩa của đề tài:

Về mặt lý luận: Đề tài góp phần minh họa cho việc ứng dụng ngôn ngữ lập trình Python cùng các thư viện giao diện (Tkinter) và cơ sở dữ liệu (MySQL) vào bài toán thực tế, từ đó củng cố kiến thức về phát triển phần mềm tích hợp.

Về mặt thực tiễn: Sản phẩm của đề tài có thể được sử dụng ngay trong thực tế bởi các doanh nghiệp nhỏ, cửa hàng, phòng ban có nhu cầu quản lý nhân sự cơ bản, giúp tiết kiệm thời gian, giảm thiểu sai sót và nâng cao hiệu quả quản lý.

2. Mục tiêu nghiên cứu của đề tài

Mục tiêu chung: Xây dựng một ứng dụng quản lý nhân viên hoàn chỉnh chạy trên nền tảng desktop, sử dụng Python làm ngôn ngữ lập trình chính, tích hợp giao diện đồ họa với Tkinter và lưu trữ dữ liệu tập trung trên hệ quản trị cơ sở dữ liệu MySQL.

Mục tiêu cụ thể:

Nghiên cứu và áp dụng thư viện Tkinter: Thiết kế giao diện người dùng thân thiện, trực quan, hỗ trợ đầy đủ các thành phần nhập liệu như textbox, combobox, radio button, date picker và bảng hiển thị dữ liệu.

Thiết kế và triển khai cơ sở dữ liệu: Xây dựng cấu trúc bảng nhanvien trong MySQL để lưu trữ các thông tin cơ bản của nhân viên bao gồm mã số, họ lót, tên, giới tính, ngày sinh và chức vụ.

Phát triển các chức năng nghiệp vụ cốt lõi (CRUD):

- +Thêm nhân viên mới vào hệ thống.
- +Hiển thị danh sách nhân viên dưới dạng bảng.
- +Cập nhật thông tin nhân viên hiện có.
- +Xóa nhân viên khỏi hệ thống.
- +Tìm kiếm nhân viên theo tiêu chí.
- +Hủy bỏ thao tác và làm mới form nhập liệu.

Kết nối và xử lý dữ liệu thời gian thực: Sử dụng thư viện mysql-connector-python để thực hiện kết nối, truy vấn và đồng bộ dữ liệu giữa ứng dụng và cơ sở dữ liệu một cách ổn định.

Đảm bảo tính khả thi và dễ triển khai: Ứng dụng phải có hướng dẫn cài đặt rõ ràng, mã nguồn được tổ chức khoa học, dễ bảo trì và mở rộng trong tương lai.

Đề tài hoàn toàn khả thi về mặt kỹ thuật do sử dụng các công nghệ mã nguồn mở, phổ biến và tài liệu phong phú. Về mặt thời gian và nhân lực, phạm vi đề tài được giới hạn rõ ràng, phù hợp với khả năng của sinh viên trong một học phần đồ án.

2. Tổng quan và cơ sở lý thuyết

2.1. Đặt vấn đề

Trong thời đại công nghệ số, việc ứng dụng tin học vào công tác quản lý nhân sự không còn là lựa chọn mà trở thành yêu cầu tất yếu. Tuy nhiên, việc phát triển một giải pháp phần mềm vừa đảm bảo tính hiệu quả, vừa phù hợp với nguồn lực hạn chế của các tổ chức nhỏ vẫn là một bài toán thực tế. Đề tài này đặt ra yêu cầu xây dựng một hệ thống quản lý nhân viên tối ưu, kết hợp sức mạnh của ngôn ngữ lập trình Python, giao diện người dùng đồ họa Tkinter và hệ quản trị cơ sở dữ liệu MySQL.

2.2. Lịch sử giải quyết vấn đề

Trên thế giới: Các hệ thống quản lý nhân sự (HRM) đã phát triển từ nhiều thập kỷ, từ các phần mềm độc quyền lớn như SAP, Oracle HRM đến các giải pháp mã nguồn mở như Odoo, OrangeHRM. Các nền tảng này thường tích hợp nhiều module phức tạp, phù hợp với doanh nghiệp lớn.

Tại Việt Nam: Nhiều công ty phần mềm trong nước đã phát triển các giải pháp quản lý nhân sự như MISA, FAST, BRAVO, tập trung vào thị trường doanh nghiệp vừa và nhỏ. Tuy nhiên, các sản phẩm này thường có chi phí triển khai và vận hành không nhỏ.

Trong học thuật: Nhiều đồ án, khóa luận của sinh viên ngành Công nghệ thông tin đã thực hiện các ứng dụng quản lý nhân viên sử dụng các công nghệ khác nhau như Java Swing, C# .NET, PHP MySQL. Tuy nhiên, việc sử dụng bộ ba Python-Tkinter-MySQL cho bài toán này còn chưa thực sự phổ biến trong các tài liệu hướng dẫn chuẩn mực.

Tồn tại: Các giải pháp hiện có cho đối tượng người dùng là cá nhân hoặc tổ chức siêu nhỏ thường thiếu tính ổn định, khó tùy chỉnh hoặc không có kết nối cơ sở dữ liệu chuyên nghiệp. Đề tài này hướng đến việc lấp đầy khoảng trống đó bằng một ứng dụng mã nguồn mở, đơn giản nhưng vẫn duy trì được kiến trúc rõ ràng và dễ phát triển.

2.3. Phạm vi của đề tài

Phạm vi chức năng: Đề tài tập trung vào việc xây dựng các chức năng quản lý nhân viên cơ bản (CRUD - Create, Read, Update, Delete) và tìm kiếm. Các chức năng nâng cao như tính lương, chấm công, đánh giá KPI, báo cáo phức tạp sẽ không thuộc phạm vi thực hiện.

Phạm vi công nghệ: Ứng dụng được phát triển bằng Python 3.x, sử dụng thư viện Tkinter cho giao diện và MySQL làm hệ quản trị cơ sở dữ liệu. Các thư viện hỗ trợ khác bao gồm mysql-connector-python để kết nối và tkcalendar để chọn ngày.

Phạm vi đối tượng: Ứng dụng hướng đến người dùng cuối là quản trị viên hoặc nhân viên văn phòng tại các doanh nghiệp nhỏ, cửa hàng, phòng ban

có nhu cầu quản lý thông tin nhân sự cơ bản với số lượng vài chục đến vài trăm người.

Phạm vi triển khai: Ứng dụng chạy trên môi trường desktop Windows, có thể mở rộng cho các hệ điều hành khác do Python hỗ trợ đa nền tảng. Việc triển khai mạng nội bộ (LAN) hoặc web là phạm vi mở rộng sau này

2.4. Phương pháp nghiên cứu/hướng giải quyết vấn đề

Phương pháp nghiên cứu lý thuyết: Tổng hợp, phân tích tài liệu về ngôn ngữ Python, lập trình giao diện với Tkinter, thiết kế cơ sở dữ liệu quan hệ với MySQL và mô hình phát triển phần mềm.

Phương pháp khảo sát và phân tích yêu cầu: Tham khảo các mẫu ứng dụng quản lý nhân viên có sẵn và nhu cầu thực tế từ các tổ chức nhỏ để xác định các chức năng tối thiểu cần thiết.

Thực thi: Lập trình hệ thống dựa trên Python, sử dụng thư viện Tkinter để xây dựng giao diện và thư viện để thao tác dữ liệu.

Kiểm thử: Thực hiện kiểm thử đơn vị và kiểm thử chức năng để đảm bảo hệ thống hoạt động đúng theo yêu cầu.

2.5. Trình bày tóm tắt cơ sở lý thuyết

Ngôn ngữ lập trình Python: Là ngôn ngữ thông dịch, cấp cao, có cú pháp rõ ràng và thư viện phong phú. Python được chọn vì tính dễ học, dễ đọc, phù hợp cho việc phát triển ứng dụng nhanh (Rapid Application Development) và có cộng đồng hỗ trợ lớn.

Thư viện Tkinter: Là thư viện giao diện người dùng đồ họa (GUI) tiêu chuẩn và được tích hợp sẵn với Python. Tkinter cung cấp bộ công cụ (widgets) đa dạng để xây dựng giao diện như cửa sổ, nhãn, ô nhập, nút bấm, bảng (Treeview). Ưu điểm là nhẹ, đơn giản và phù hợp cho các ứng dụng desktop quy mô nhỏ đến trung bình.

Hệ quản trị cơ sở dữ liệu MySQL: Là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến nhất thế giới. MySQL được chọn vì tính ổn định, hiệu suất cao, dễ cài đặt và quản lý, đồng thời hoàn toàn miễn phí. Nó cho phép lưu trữ dữ liệu một cách có cấu trúc, đảm bảo tính toàn vẹn và nhất quán.

Kiến trúc ứng dụng Client-Server (mô hình 2 lớp): Ứng dụng tuân theo mô hình này, trong đó:

Client (Front-end): Là giao diện Tkinter chạy trên máy người dùng, đảm nhận việc tiếp nhận thao tác và hiển thị dữ liệu.

Server (Back-end): Là cơ sở dữ liệu MySQL (có thể cài trên cùng máy hoặc máy chủ riêng), đảm nhận việc lưu trữ, xử lý và trả về dữ liệu theo yêu cầu.

Kết nối giữa hai lớp được thực hiện thông qua thư viện mysql-connector-python, sử dụng các câu lệnh SQL chuẩn để giao tiếp.

3. Kết quả đạt được và Demo Code

A. THIẾT KẾ GIAO DIỆN QUẢN LÝ NHÂN VIÊN

Maxe	Tenxe	Hangsx	Giaban	Soluong
(1,	'Wave	Alpha',	'Honda',	Decimal('18000000.00'),
(2,	'Vision',	'Honda',	Decimal('30000000.00'),	5)
(3,	'Exciter',	'Yamaha',	Decimal('40000000.00'),	3)
(4,	'SH	Mode',	'Honda',	Decimal('35000000.00'),

Quản lý cửa hàng xe máy

Quản lý Nhân viên Quản lý Xe máy

Nhập / Sửa Xe máy

Mã xe (4, Tên xe 'SH Hãng SX 'Mode', Giá bán 'Honda', Số lượng Decimal('3500

Thêm Sửa (nạp) Lưu (cập nhật) Xóa Hủy Tải lại

Maxe	Tenxe	Hangsx	Giaban	Soluong
(1, 'Wave	Alpha',	'Honda',	Decimal('18000000.00'),	
(2, 'Vision',	'Honda',	Decimal('30000000.00'),	5)	
(3, 'Exciter',	'Yamaha',	Decimal('40000000.00'),	3)	
(4, 'SH	Mode',	'Honda',	Decimal('35000000.00'),	

B. CÁC BƯỚC XÂY DỰNG ỨNG DỤNG

Để xây dựng ứng dụng quản lý sản phẩm bằng Python, Tkinter và MySQL, cần cài đặt các thư viện cần thiết, thiết kế cơ sở dữ liệu MySQL để lưu trữ thông tin sản phẩm, sử dụng Tkinter để tạo giao diện người dùng, và kết hợp Python với thư viện mysql.connector để thực hiện các thao tác như thêm, xóa, sửa, và hiển thị danh sách Nhân Viên

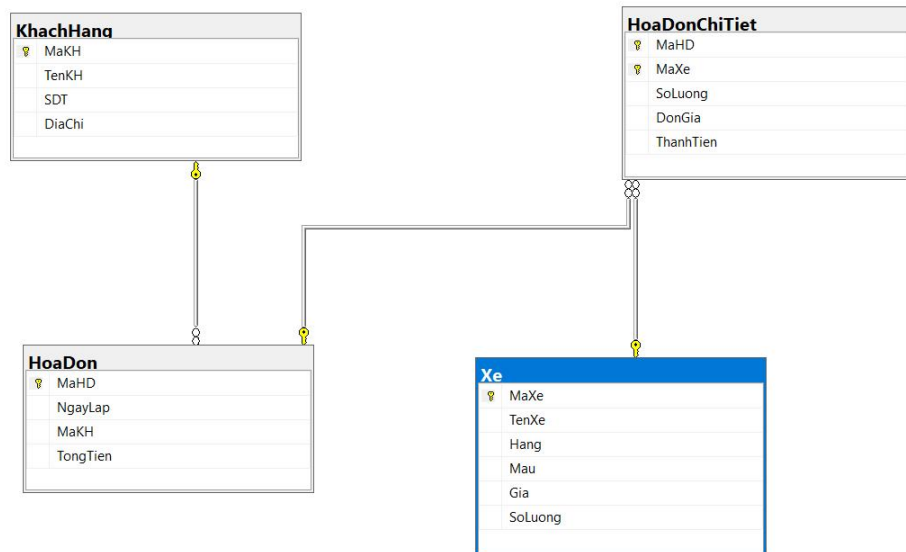
1. Cài đặt các thư viện cần thiết:

Python: Đảm bảo bạn đã cài đặt Python trên máy tính của mình.

MySQL: Cài đặt máy chủ MySQL và tạo một cơ sở dữ liệu cho ứng dụng của bạn.

mysql-connector-python: Cài đặt thư viện này để kết nối Python với cơ sở dữ liệu MySQL (pip install mysql-connector-python)

2. Thiết kế cơ sở dữ liệu MySQL



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

Để đáp ứng yêu cầu quản lý nhân viên, giao diện ứng dụng được xây dựng bằng thư viện Tkinter của Python, chia thành 3 khu vực chức năng chính để đảm bảo tính trực quan và dễ sử dụng

C. CÀI ĐẶT CODE DEMO


```

import tkinter as tk
from tkinter import ttk, messagebox
from tkcalendar import DateEntry
import pyodbc

SERVER = "localhost\\KHANH"
DATABASE = "CuaHangXeMay"
CONN_STR = f"DRIVER={{SQL Server}};SERVER={SERVER};DATABASE={DATABASE};Trusted_Connection=yes;"

#
def connect_db(): 8 usages
    try:
        return pyodbc.connect(CONN_STR)
    except Exception as e:
        messagebox.showerror(title="Lỗi kết nối DB", str(e))
        return None

def load_nhanvien(tree): 5 usages
    tree.delete(*tree.get_children())
    conn = connect_db()
    if not conn: return
    cur = conn.cursor()
    try:
        cur.execute("SELECT maso, holot, ten, phai, ngaysinh, chucvu FROM nhanvien ORDER BY maso")
        for row in cur.fetchall():
            tree.insert("", tk.END, values=row)
    except Exception as e:
        messagebox.showerror(title="Lỗi SQL", str(e))
    finally:
        cur.close(); conn.close()

def them_nhanvien(tree, e_maso, e_holot, e_ten, e_phai, e_ngaysinh, e_chucvu): 1 usage
    maso = e_maso.get().strip()
    holot = e_holot.get().strip()
    ten = e_ten.get().strip()
    phai = e_phai.get().strip()
    ngaysinh = e_ngaysinh.get()
    chucvu = e_chucvu.get().strip()

    if not (maso and holot and ten):
        messagebox.showwarning(title="Thiếu dữ liệu", message="Nhập mã, họ lót, tên.")
        return

    try:
        maso_i = int(maso)
    except:
        messagebox.showerror(title="Lỗi", message="Mã phải là số nguyên.")
        return

    conn = connect_db()
    if not conn: return
    cur = conn.cursor()
    try:
        cur.execute(sql="INSERT INTO nhanvien (maso, holot, ten, phai, ngaysinh, chucvu) VALUES (?, ?, ?, ?, ?, ?)",
            *params: (maso_i, holot, ten, phai, ngaysinh, chucvu))
        conn.commit()
        load_nhanvien(tree)
        clear_nv_inputs(e_maso, e_holot, e_ten, e_phai, e_ngaysinh, e_chucvu)
        messagebox.showinfo(title="OK", message="Đã thêm nhân viên.")

```

```

        messagebox.showinfo(title="OK", message="Đã thêm nhân viên.")
    except Exception as e:
        messagebox.showerror(title="Lỗi khi thêm", str(e))
    finally:
        cur.close(); conn.close()

def xoa_nhanvien(tree): 1 usage
    sel = tree.selection()
    if not sel:
        messagebox.showwarning(title="Chưa chọn", message="Chọn nhân viên để xóa.")
        return
    maso = tree.item(sel[0])['values'][0]
    if not messagebox.askyesno(title="Xóa", message=f"Xác nhận xóa nhân viên mã {maso}?"):
        return
    conn = connect_db()
    if not conn: return
    cur = conn.cursor()
    try:
        cur.execute(sql: "DELETE FROM nhanvien WHERE maso = ?", *params: (maso,))
        conn.commit()
        load_nhanvien(tree)
        messagebox.showinfo(title="Xong", message="Đã xóa.")
    except Exception as e:
        messagebox.showerror(title="Lỗi xóa", str(e))
    finally:
        cur.close(); conn.close()

def sua_nhanvien(tree, e_maso, e_holot, e_ten, e_phai, e_ngaysinh, e_chucvu): 1 usage
    sel = tree.selection()
    if not sel:
        messagebox.showwarning(title="Chưa chọn", message="Chọn nhân viên để sửa (nạp).")
        return
    messagebox.showwarning(title="Chưa chọn", message="Chọn nhân viên để sửa (nạp).")
    return
    vals = tree.item(sel[0])['values']
    e_maso.delete(0, tk.END); e_maso.insert(0, vals[0])
    e_holot.delete(0, tk.END); e_holot.insert(0, vals[1])
    e_ten.delete(0, tk.END); e_ten.insert(0, vals[2])
    e_phai.set(vals[3] if vals[3] else "Nam")
    try:
        e_ngaysinh.set_date(vals[4])
    except:
        pass
    e_chucvu.delete(0, tk.END); e_chucvu.insert(0, vals[5])

def luu_nhanvien(tree, e_maso, e_holot, e_ten, e_phai, e_ngaysinh, e_chucvu): 1 usage
    maso = e_maso.get().strip()
    holot = e_holot.get().strip()
    ten = e_ten.get().strip()
    phai = e_phai.get().strip()
    ngaysinh = e_ngaysinh.get()
    chucvu = e_chucvu.get().strip()

    if not maso:
        messagebox.showwarning(title="Thiếu", message="Mã không được để trống.")
        return
    try:
        maso_i = int(maso)
    except:
        messagebox.showerror(title="Lỗi", message="Mã phải là số.")
        return

```

```

def luu_nhanvien(tree, e_maso, e_holot, e_ten, e_phai, e_ngaysinh, e_chucvu): 1 usage
    conn = connect_db()
    if not conn: return
    cur = conn.cursor()
    try:
        cur.execute(sql: "UPDATE nhanvien SET holot=?, ten=?, phai=?, ngaysinh=?, chucvu=? WHERE maso=?",
            *params: (holot, ten, phai, ngaysinh, chucvu, maso_i))
        conn.commit()
        load_nhanvien(tree)
        clear_nv_inputs(e_maso, e_holot, e_ten, e_phai, e_ngaysinh, e_chucvu)
        messagebox.showinfo(title: "OK", message: "Đã cập nhật.")
    except Exception as e:
        messagebox.showerror(title: "Lỗi update", str(e))
    finally:
        cur.close(); conn.close()

def clear_nv_inputs(e_maso, e_holot, e_ten, e_phai, e_ngaysinh, e_chucvu): 3 usages
    e_maso.delete(0, tk.END)
    e_holot.delete(0, tk.END)
    e_ten.delete(0, tk.END)
    e_phai.set("Nam")
    try:
        e_ngaysinh.set_date("2000-01-01")
    except:
        pass
    e_chucvu.delete(0, tk.END)

# ----- XE MÁY: CRUD -----

def load_xemay(tree): 5 usages
    tree.delete(*tree.get_children())
    conn = connect_db()
    if not conn: return
    cur = conn.cursor()
    try:
        cur.execute("SELECT maxe, tenxe, hangsx, giaban, soluong FROM xemay ORDER BY maxe")
        for row in cur.fetchall():
            tree.insert("", tk.END, values=row)
    except Exception as e:
        messagebox.showerror(title: "Lỗi SQL", str(e))
    finally:
        cur.close(); conn.close()

def them_xemay(tree, e_maxe, e_tenxe, e_hangsx, e_giaban, e_soluong): 1 usage
    maxe = e_maxe.get().strip()
    tenxe = e_tenxe.get().strip()
    hangsx = e_hangsx.get().strip()
    giaban = e_giaban.get().strip()
    soluong = e_soluong.get().strip()
    if not (maxe and tenxe and giaban and soluong):
        messagebox.showwarning(title: "Thiếu", message: "Nhập mã, tên, giá, số lượng.")
        return
    try:
        maxe_i = int(maxe); giaban_f = float(giaban); soluong_i = int(soluong)
    except:
        messagebox.showerror(title: "Lỗi", message: "Kiểm tra kiểu dữ liệu (số).")
        return
    conn = connect_db()
    if not conn: return
    cur = conn.cursor()
    try:
        cur.execute(sql: "INSERT INTO xemay (maxe, tenxe, hangsx, giaban, soluong) VALUES (?, ?, ?, ?, ?)",
            *params: (maxe_i, tenxe, hangsx, giaban_f, soluong_i))

```

```

        *params: (maxe_i, tenxe, hangsx, giaban_f, soluong_i))
    conn.commit()
    load_xemay(tree)
    clear_xe_inputs(e_maxe, e_tenxe, e_hangsx, e_giaban, e_soluong)
    messagebox.showinfo(title: "OK", message: "Đã thêm xe.")
except Exception as e:
    messagebox.showerror(title: "Lỗi thêm", str(e))
finally:
    cur.close(); conn.close()

def xoa_xemay(tree): 1 usage
    sel = tree.selection()
    if not sel:
        messagebox.showwarning(title: "Chưa chọn", message: "Chọn xe để xóa.")
        return
    maxe = tree.item(sel[0])['values'][0]
    if not messagebox.askyesno(title: "Xóa", message: f"Xác nhận xóa xe mã {maxe}?"):
        return
    conn = connect_db()
    if not conn: return
    cur = conn.cursor()
    try:
        cur.execute(sql: "DELETE FROM xemay WHERE maxe = ?", *params: (maxe,))
        conn.commit()
        load_xemay(tree)
        messagebox.showinfo(title: "OK", message: "Đã xóa.")
    except Exception as e:
        messagebox.showerror(title: "Lỗi xóa", str(e))
    finally:
        cur.close(); conn.close()

def sua_xemay(tree, e_maxe, e_tenxe, e_hangsx, e_giaban, e_soluong): 1 usage
    sel = tree.selection()
    if not sel:
        messagebox.showwarning(title: "Chưa chọn", message: "Chọn xe để sửa (nạp).")
        return
    vals = tree.item(sel[0])['values']
    e_maxe.delete(0, tk.END); e_maxe.insert(0, vals[0])
    e_tenxe.delete(0, tk.END); e_tenxe.insert(0, vals[1])
    e_hangsx.delete(0, tk.END); e_hangsx.insert(0, vals[2])
    e_giaban.delete(0, tk.END); e_giaban.insert(0, vals[3])
    e_soluong.delete(0, tk.END); e_soluong.insert(0, vals[4])

def luu_xemay(tree, e_maxe, e_tenxe, e_hangsx, e_giaban, e_soluong): 1 usage
    maxe = e_maxe.get().strip()
    tenxe = e_tenxe.get().strip()
    hangsx = e_hangsx.get().strip()
    giaban = e_giaban.get().strip()
    soluong = e_soluong.get().strip()
    if not maxe:
        messagebox.showwarning(title: "Thiếu", message: "Mã xe không được rỗng.")
        return
    try:
        maxe_i = int(maxe)
        giaban_f = float(giaban) if giaban else 0.0
        soluong_i = int(soluong) if soluong else 0
    except:
        messagebox.showerror(title: "Lỗi", message: "Kiểm tra kiểu dữ liệu.")
        return
    conn = connect_db()

```

```

if not conn: return
cur = conn.cursor()
try:
    cur.execute('sql: "UPDATE xemay SET tenxe=?, hangsx=?, giaban=?, soluong=? WHERE maxe=?",
                *params: (tenxe, hangsx, giaban_f, soluong_i, maxe_i))
    conn.commit()
    load_xemay(tree)
    clear_xe_inputs(e_maxe, e_tenxe, e_hangsx, e_giaban, e_soluong)
    messagebox.showinfo( title: "OK", message: "Đã cập nhật.")
except Exception as e:
    messagebox.showerror( title: "Lỗi cập nhật", str(e))
finally:
    cur.close(); conn.close()

def clear_xe_inputs(e_maxe, e_tenxe, e_hangsx, e_giaban, e_soluong): 3 usages
    e_maxe.delete(0, tk.END)
    e_tenxe.delete(0, tk.END)
    e_hangsx.delete(0, tk.END)
    e_giaban.delete(0, tk.END)
    e_soluong.delete(0, tk.END)

# ----- GIAO DIỆN CHÍNH -----
def center_window(win, w=1000, h=650): 1 usage
    ws = win.winfo_screenwidth(); hs = win.winfo_screenheight()
    x = (ws // 2) - (w // 2); y = (hs // 2) - (h // 2)
    win.geometry(f"{w}x{h}+{x}+{y}")

root = tk.Tk()
root.title("Quản lý cửa hàng xe máy")
center_window(root)
root.resizable( width: False, height: False)

center_window(root)
root.resizable( width: False, height: False)

notebook = ttk.Notebook(root)
notebook.pack(fill="both", expand=True, padx=8, pady=8)

# ----- Tab NHÂN VIÊN -----
tab_nv = tk.Frame(notebook); notebook.add(tab_nv, text="Quản lý Nhân viên")
frame_nv_input = tk.LabelFrame(tab_nv, text="Nhập / Sửa nhân viên"); frame_nv_input.pack(fill="x", padx=10, pady=6)

tk.Label(frame_nv_input, text="Mã số").grid(row=0, column=0, padx=5, pady=5, sticky="w")
e_nv_maso = tk.Entry(frame_nv_input, width=12); e_nv_maso.grid(row=0, column=1, padx=5, pady=5)
tk.Label(frame_nv_input, text="Họ lót").grid(row=0, column=2, padx=5, pady=5, sticky="w")
e_nv_holot = tk.Entry(frame_nv_input, width=30); e_nv_holot.grid(row=0, column=3, padx=5, pady=5)
tk.Label(frame_nv_input, text="Tên").grid(row=0, column=4, padx=5, pady=5, sticky="w")
e_nv_ten = tk.Entry(frame_nv_input, width=18); e_nv_ten.grid(row=0, column=5, padx=5, pady=5)

tk.Label(frame_nv_input, text="Phái").grid(row=1, column=0, padx=5, pady=5, sticky="w")
e_nv_phai = ttk.Combobox(frame_nv_input, values=["Nam", "Nu"], width=10); e_nv_phai.grid(row=1, column=1, padx=5, pady=5); e_nv_phai.set("Nam")
tk.Label(frame_nv_input, text="Ngày sinh").grid(row=1, column=2, padx=5, pady=5, sticky="w")
e_nv_ngaysinh = DateEntry(frame_nv_input, width=14, date_pattern="yyyy-mm-dd"); e_nv_ngaysinh.grid(row=1, column=3, padx=5, pady=5)
tk.Label(frame_nv_input, text="Chức vụ").grid(row=1, column=4, padx=5, pady=5, sticky="w")
e_nv_chucvu = tk.Entry(frame_nv_input, width=25); e_nv_chucvu.grid(row=1, column=5, padx=5, pady=5)

frame_nv_btn = tk.Frame(tab_nv); frame_nv_btn.pack(pady=6)
tk.Button(frame_nv_btn, text="Thêm", width=10, command=lambda: them_nhanvien(tree_nv, e_nv_maso, e_nv_holot, e_nv_ten, e_nv_phai, e_nv_ngaysinh, e_nv_chucvu)).grid(row=0, column=0, padx=5, pady=5)
tk.Button(frame_nv_btn, text="Sửa (ngp)", width=12, command=lambda: sua_nhanvien(tree_nv, e_nv_maso, e_nv_holot, e_nv_ten, e_nv_phai, e_nv_ngaysinh, e_nv_chucvu)).grid(row=0, column=1, padx=5, pady=5)
tk.Button(frame_nv_btn, text="Lưu (cập nhật)", width=12, command=lambda: luu_nhanvien(tree_nv, e_nv_maso, e_nv_holot, e_nv_ten, e_nv_phai, e_nv_ngaysinh, e_nv_chucvu)).grid(row=0, column=2, padx=5, pady=5)
tk.Button(frame_nv_btn, text="Xóa", width=10, command=lambda: xoa_nhanvien(tree_nv)).grid(row=0, column=3, padx=5, pady=5)
tk.Button(frame_nv_btn, text="Hủy", width=10, command=lambda: clear_nv_inputs(e_nv_maso, e_nv_holot, e_nv_ten, e_nv_phai, e_nv_ngaysinh, e_nv_chucvu)).grid(row=0, column=4, padx=5, pady=5)
tk.Button(frame_nv_btn, text="Tải lại", width=10, command=lambda: load_nhanvien(tree_nv)).grid(row=0, column=5, padx=5, pady=5)

```



```

tree_nv = ttk.Treeview(tab_nv, columns=cols_nv, show="headings", height=14)
for c in cols_nv:
    tree_nv.heading(c, text=c.capitalize())
tree_nv.column("maso", width=80, anchor="center")
tree_nv.column("holot", width=240)
tree_nv.column("ten", width=120)
tree_nv.column("phai", width=80, anchor="center")
tree_nv.column("ngaysinh", width=110, anchor="center")
tree_nv.column("chucvu", width=150)
tree_nv.pack(fill="both", expand=True, padx=10, pady=6)
load_nhanvien(tree_nv)

# ----- Tab XE MÁY -----
tab_xe = tk.Frame(notebook); notebook.add(tab_xe, text="Quản lý Xe máy")
frame_xe_input = tk.LabelFrame(tab_xe, text="Nhập / Sửa Xe máy"); frame_xe_input.pack(fill="x", padx=10, pady=6)

tk.Label(frame_xe_input, text="Mã xe").grid(row=0, column=0, padx=5, pady=5, sticky="w")
e_xe_maxe = tk.Entry(frame_xe_input, width=12); e_xe_maxe.grid(row=0, column=1, padx=5, pady=5)
tk.Label(frame_xe_input, text="Tên xe").grid(row=0, column=2, padx=5, pady=5, sticky="w")
e_xe_tenxe = tk.Entry(frame_xe_input, width=30); e_xe_tenxe.grid(row=0, column=3, padx=5, pady=5)
tk.Label(frame_xe_input, text="Hãng SX").grid(row=0, column=4, padx=5, pady=5, sticky="w")
e_xe_hangsx = tk.Entry(frame_xe_input, width=20); e_xe_hangsx.grid(row=0, column=5, padx=5, pady=5)

tk.Label(frame_xe_input, text="Giá bán").grid(row=1, column=0, padx=5, pady=5, sticky="w")
e_xe_giaban = tk.Entry(frame_xe_input, width=20); e_xe_giaban.grid(row=1, column=1, padx=5, pady=5)
tk.Label(frame_xe_input, text="Số lượng").grid(row=1, column=2, padx=5, pady=5, sticky="w")
e_xe_soluong = tk.Entry(frame_xe_input, width=12); e_xe_soluong.grid(row=1, column=3, padx=5, pady=5)

frame_xe_btn = tk.Frame(tab_xe); frame_xe_btn.pack(pady=6)
tk.Button(frame_xe_btn, text="Thêm", width=10, command=lambda: them_xemay(tree_xe, e_xe_maxe, e_xe_tenxe, e_xe_hangsx, e_xe_giaban, e_xe_soluong)).grid(row=0, column=0,
tk.Button(frame_xe_btn, text="Sửa (ngp)", width=12, command=lambda: sua_xemay(tree_xe, e_xe_maxe, e_xe_tenxe, e_xe_hangsx, e_xe_giaban, e_xe_soluong)).grid(row=0, column=1,
tk.Button(frame_xe_btn, text="Lưu (cập nhật)", width=12, command=lambda: luu_xemay(tree_xe, e_xe_maxe, e_xe_tenxe, e_xe_hangsx, e_xe_giaban, e_xe_soluong)).grid(row=0, column=2,
tk.Button(frame_xe_btn, text="Xóa", width=10, command=lambda: xoa_xemay(tree_xe)).grid(row=0, column=3, padx=6)
tk.Button(frame_xe_btn, text="Hủy", width=10, command=lambda: clear_xe_inputs(e_xe_maxe, e_xe_tenxe, e_xe_hangsx, e_xe_giaban, e_xe_soluong)).grid(row=0, column=4, padx=6)
tk.Button(frame_xe_btn, text="Tải lại", width=10, command=lambda: load_xemay(tree_xe)).grid(row=0, column=5, padx=6)

cols_xe = ("maxe", "tenxe", "hangsx", "giaban", "soluong")
tree_xe = ttk.Treeview(tab_xe, columns=cols_xe, show="headings", height=14)
for c in cols_xe:
    tree_xe.heading(c, text=c.capitalize())
tree_xe.column("maxe", width=80, anchor="center")
tree_xe.column("tenxe", width=100)
tree_xe.column("hangsx", width=140)
tree_xe.column("giaban", width=120, anchor="e")
tree_xe.column("soluong", width=100, anchor="center")
tree_xe.pack(fill="both", expand=True, padx=10, pady=6)
load_xemay(tree_xe)

root.mainloop()

```