

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



**BÁO CÁO ĐỒ ÁN**

**Chuyên đề Python(COS525)**

**Chủ đề 16**

**XÂY DỰNG ỨNG DỤNG QUẢN LÝ XE VÀ LÁI XE  
VỚI PYTHON, TKINTER VÀ SQL SERVER**

Giảng viên hướng dẫn: Ths.Nguyễn Ngọc Minh

Sinh viên thực hiện:

DTH23516 - Nguyễn Văn Chiến - DH24TH1\_ Nhóm1\_ Tô1

DTH235660- Phạm Mạnh Hùng - DH24TH1\_ Nhóm1\_ Tô1

An Giang, tháng 11 năm 2025

<b>MSV</b>	<b>Họ tên</b>	<b>Lớp</b>	<b>Công việc thực hiện</b>	<b>Tỷ lệ tham gia</b>
DTH235617	Nguyễn Văn Chiến	DH24TH1	CSDL, code, thiết kế giao diện	50%
DTH235660	Phạm Mạnh Hùng	DH24TH1	CSDL, code, thiết kế giao diện	50%

# MỤC LỤC

<b>I. GIỚI THIỆU CHUNG .....</b>	<b>4</b>
1. Đặt vấn đề .....	4
2. Tổng quan và cơ sở lý thuyết .....	4
3. Kết quả đạt được và Demo Code .....	5
<b>II. THIẾT KẾ GIAO DIỆN .....</b>	<b>6</b>
1. Giao diện đăng nhập: .....	6
2. Giao diện chính Admin: .....	7
3. Giao diện chính TaiXe: .....	11
<b>III. Các bước xây dựng ứng dụng .....</b>	<b>16</b>
1. Cài đặt các thư viện cần thiết: .....	16
2. Thiết kế cơ sở dữ liệu MySQL .....	16
a. Mô tả Cấu trúc Cơ sở dữ liệu QL_VanTai .....	16
b. Mối liên kết của các bảng: .....	19
3. Xây dựng giao diện người dùng (GUI) bằng Tkinter: .....	19
a. Quản lý Nhân Viên (Bảng NhanVien) .....	19
b. Quản lý Tài Xe (Bảng TaiXe) .....	19
c. Quản lý Xe (Bảng Xe) .....	20
d. Quản lý Tài Khoản (Bảng TaiKhoan) .....	20
e. Quản lý Chuyến Đi (Bảng ChuyenDi) .....	20
f. Quản lý Bảo Trì (Bảng LichSuBaoTri) .....	21
g. Quản lý Nhiên Liệu (Bảng NhatKyNhienLieu) .....	21
<b>IV. Thiết kế và cài đặt chương trình .....</b>	<b>21</b>
1. Cài đặt môi trường .....	21
2. Chuẩn bị CSDL SQL Server .....	21
a. Bảng NhanVien .....	22

b. Bảng TaiXe .....	22
c. Bảng Xe .....	22
d. Bảng ChuyenDi .....	23
e. Bảng LichSuBaoTri .....	23
f. Bảng NhatKyNhienLieu .....	23
g. Bảng TaiKhoan .....	24
3. Code ứng dụng với Tkinker và SQL Server .....	24
a. Kết nối với SQL Server .....	24
b. Form Login .....	25
c. Form Main .....	29
d. Form QuanLyChuyenDi .....	36
e. Form LichSuBaoTri .....	45
f. Form QuanLyNhanVien .....	57
g. Form NhatKyNguyenLieu .....	66
h. Form QuanLyTaiXe .....	82
i. Form QuanLyXe .....	93

## V. TÀI LIỆU THAM KHẢO ..... 108

## I. GIỚI THIỆU CHUNG

### 1. Đặt vấn đề

Tính cần thiết của đề tài: Việc quản lý thủ công (qua sổ sách, excel) đối xe, tài xế, và các chuyến đi trong một công ty vận tải rất phức tạp, tốn thời gian, dễ sai sót và thiếu tính tức thời. Việc xây dựng một hệ thống phần mềm tập trung là cần thiết để số hóa quy trình, giảm thiểu sai sót, và cung cấp cái nhìn tổng quan cho nhà quản lý.

Mục tiêu nghiên cứu: Xây dựng một ứng dụng desktop bằng Python (Tkinter) để giải quyết các bài toán:

- Quản lý tập trung thông tin Xe (quanli\_xe) và Tài xế (quanli\_taixe).
- Quản lý việc điều phối Chuyến đi (quanli\_chuyendi), gán tài xế và xe cho các tuyến.
- Ghi nhận và quản lý Lịch sử bảo trì (quanli\_lichsubaotri) và Nhật ký nhiên liệu (quanli\_nhatkinguyenlieu).
- Phân quyền người dùng (Admin và Tài xế) với chức năng đăng nhập (login) bảo mật.

### 2. Tổng quan và cơ sở lý thuyết

Đặt vấn đề: Đề tài giải quyết bài toán nghiệp vụ về quản lý vận tải, tập trung vào việc số hóa các quy trình CRUD (Thêm, Sửa, Xóa, Xem) và phân quyền truy cập thông tin.

Lịch sử giải quyết vấn đề: Hệ thống này được xây dựng mới hoàn toàn để thay thế các phương pháp quản lý thủ công, chưa được giải quyết bằng phần mềm chuyên dụng trước đó trong phạm vi đề tài.

Phạm vi của đề tài:

- Trong phạm vi: Ứng dụng quản lý 6 module chính: Nhân viên, Tài xế, Xe, Chuyến đi, Nhiên liệu, Bảo trì và Tài khoản. Hệ thống có 2 vai trò chính:
  - + Admin: Toàn quyền xem, thêm, sửa, xóa trên tất cả các module. Có quyền *duyệt/tù chối* các báo cáo nhiên liệu.

+ TaiXe (Tài xế): Chỉ xem được thông tin cá nhân, các chuyến đi và nhật ký nhiên liệu của *chính mình*. Chỉ có quyền *thêm* nhật ký (chờ duyệt) và *cập nhật trạng thái* chuyến đi (Bắt đầu, Hoàn thành, Hủy).

- Ngoài phạm vi: Không bao gồm các chức năng định vị GPS thời gian thực, tối ưu hóa tuyến đường tự động, hoặc tích hợp với ứng dụng di động.

Phương pháp giải quyết:

- Phân tích yêu cầu nghiệp vụ của một hệ thống quản lý vận tải.
- Thiết kế cơ sở dữ liệu quan hệ (CSDL) trên Microsoft SQL Server.
- Sử dụng ngôn ngữ Python và thư viện Tkinter (cùng với ttk và tkcalendar) để xây dựng giao diện người dùng (GUI) desktop.
- Áp dụng lập trình module hóa, chia ứng dụng thành các file chức năng riêng biệt (ví dụ: quanli\_xe.py, quanli\_chuyendi.py, utils.py).

Trình bày tóm tắt cơ sở lý thuyết:

- Python & Tkinter: Sử dụng Python làm ngôn ngữ chính và tkinter để tạo giao diện. Thư viện ttk được dùng để tạo các widget (nút, bảng...) hiện đại, và tkcalendar cho ô nhập ngày tháng.
- SQL Server & pyodbc: Sử dụng SQL Server làm hệ quản trị CSDL. Dùng thư viện pyodbc để thực thi các truy vấn SQL (SELECT, INSERT, UPDATE, DELETE) từ Python.

- Bảo mật & Phân quyền:

- + Sử dụng thư viện hashlib (SHA-256) để băm và lưu trữ mật khẩu an toàn.
- + Logic phân quyền được thực hiện bằng cách: file login.py xác thực người dùng, sau đó dùng subprocess.Popen để mở main.py và truyền USER\_ROLE (vai trò) và USER\_USERNAME (tên đăng nhập) qua sys.argv.
- + File main.py và các file con (ví dụ: quanli\_chuyendi.py) đọc các biến này để quyết định ẩn/hiện các nút (ví dụ: button.pack\_forget()) và lọc dữ liệu SQL (ví dụ: WHERE MaNhanVien = ?).

### **3. Kết quả đạt được và Demo Code**

Phân tích và xây dựng: Đã xây dựng thành công ứng dụng desktop đa giao diện. Ứng dụng có một cửa sổ chính (main.py) chứa thanh điều hướng bên trái và

một khung nội dung chính. Khi nhấp vào các nút trên thanh điều hướng, khung nội dung sẽ được xóa đi và tải trang chức năng tương ứng vào.

Vận dụng kỹ thuật lập trình:

- Tái sử dụng code: Tạo file utils.py dùng chung cho các hàm quan trọng như connect\_db(), setup\_theme(), và các hàm tải combobox (load\_xe\_combobox(), load\_taixe\_combobox()), giúp code gọn gàng và dễ bảo trì.

- Xử lý vai trò (Admin): Admin có thể thấy toàn bộ dữ liệu. Giao diện của Admin có đầy đủ các nút "Thêm", "Sửa", "Xóa", "Lưu", "Hủy". Đặc biệt, trong trang nhiên liệu, Admin có 2 nút "Duyệt" và "Tù chối" để thay đổi TrangThaiDuyet trong CSDL.

- Xử lý vai trò (Tài xế): Giao diện của Tài xế bị ẩn đi các nút và form không cần thiết.

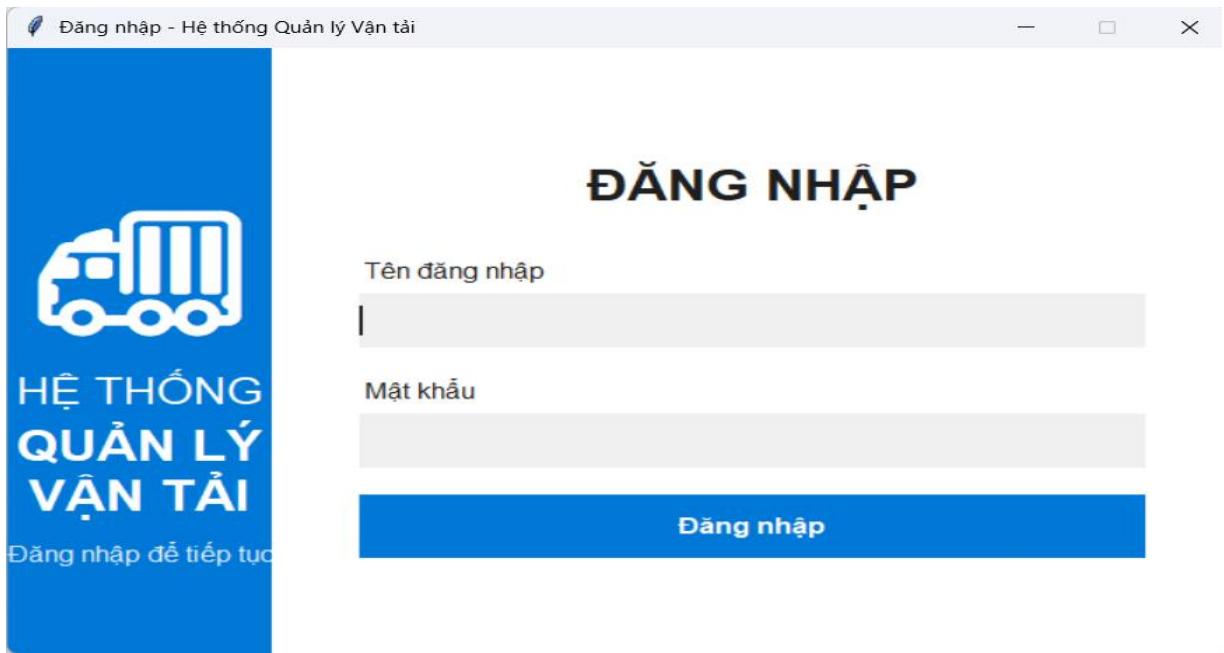
+ Hàm load\_data trong các trang (chuyển đi, nhiên liệu) được sửa đổi để thêm mệnh đề WHERE nk.MaNhanVien = ? vào câu SQL, chỉ lọc các chuyến đi và nhật ký của tài xế đó.

+ Các combobox "Xe" được lọc theo xe mà tài xế đó đang được gán (utils.load\_xe\_by\_manv(manv)).

+ Tài xế có các nút nghiệp vụ riêng, ví dụ: "Bắt đầu thực hiện" (UPDATE TrangThai = 1), "Xác nhận Hoàn Thành" (UPDATE TrangThai = 2, ThoiGianKetThuc = GETDATE()).

## II. THIẾT KẾ GIAO DIỆN

### 1. Giao diện đăng nhập:



Hình 2. 1. Giao diện đăng nhập

## 2. Giao diện chính Admin:

- Trang chủ



Hình 2. 2. Màn hình chính Admin

## - Quản lý xe

Hệ Thống Quản Lý Vận Tải (Vai trò: Admin)

**QUẢN LÝ XE**

Biển số xe:	21A-312.12	Năm sản xuất:	2020
Loại xe:	tỷ	Số khung (VIN):	31231
Hãng sản xuất:	re	Ngày hết hạn ĐK:	2025-12-25
Dòng xe:	gu	Ngày hết hạn BH:	2025-12-25
Mã NV lái:	NV001	Trạng thái:	Hoạt động
<input type="button" value="Thêm"/> <input type="button" value="Lưu"/> <input type="button" value="Sửa"/> <input type="button" value="Hủy"/> <input type="button" value="Xóa"/>			

**Danh sách xe**

Biển số xe	Loại xe	Hãng sản xuất	Năm SX	Trạng thái	Ngày hết hạn ĐK
21A-312.12	tỷ	re	2020	Hoạt động	2025-12-25
29H-678.90	Xe tải 1.5T	Hyundai	2021	Hoạt động	2026-05-01
51F-555.55	7 chỗ	Ford	2020	Bảo trì	2025-12-25
51G-123.45	4 chỗ	Toyota	2022	Hoạt động	2026-11-01

Hình 2. 3. Bảng quản lý xe

## - Quản lý tài xế

Hệ Thống Quản Lý Vận Tải (Vai trò: Admin)

**QUẢN LÝ TÀI XẾ**

Mã nhân viên:	NV001	Hạng bằng lái:	B2
Họ và tên:	Nguyễn Văn An	Ngày hết hạn BL:	2028-10-20
Số điện thoại:	0901112221	Trạng thái NV:	Đang làm việc
Địa chỉ:	456 Đường Tài Xế, Q. Bình Thạnh,	Trạng thái Lái:	Rảnh
<input type="button" value="Thêm"/> <input type="button" value="Lưu"/> <input type="button" value="Sửa"/> <input type="button" value="Hủy"/> <input type="button" value="Xóa"/>			

**Danh sách tài xế**

Mã NV	Họ Tên	Số điện thoại	Hạng Bằng Lái	Trạng thái NV	Trạng thái Lái
NV001	Nguyễn Văn An	0901112221	B2	Đang làm việc	Rảnh
NV002	Trần Thị Bình	0902223332	C	Đang làm việc	Đang lái
NV003	Lê Văn Cường	0903344443	B2	Nghỉ	Rảnh
NV012	Phạm Mạnh Hưng	098765432	C1	Đang làm việc	Đang lái

Hình 2. 4. Bảng quản lý tài xế

## - Quản lý chuyển đi

Mã CD	Tên Tài xế	Biển số xe	Thời gian bắt đầu	Thời gian kết thúc	Trạng thái
4	Phạm Mạnh Hưng	21A-312.12	2025-11-14 00:00	2025-11-14 09:47	Đang thực hiện
1	Nguyễn Văn An	51G-123.45	2025-11-13 00:00	2025-11-13 19:26	Hoàn thành
2	Trần Thị Bình	29H-678.90	2025-11-12 06:11		Đang thực hiện

Hình 2. 5. Bảng quản lý chuyển đi

## - Lịch sử bảo trì

Mã BT	Biển số xe	Ngày bảo trì	Chi phí	Mô tả	Người nhập
4	51F-555.55	2025-11-20	3000000.00	bảo dưỡng định kỳ...	N/A
9	68B-555.55	2025-11-14	500000.00	Đổ xăng...	Nguyễn Văn E
2	51F-555.55	2025-11-11	3600000.00	Bảo dưỡng định kỳ, thay dầu...	N/A

Hình 2. 6. Bảng lịch sử bảo trì

## - Nhật ký nhiên liệu

Hệ Thống Quản Lý Vận Tải (Vai trò: Admin)

**HỆ THỐNG VẬN TẢI**

- ✉ Quản lý Xe
- 👤 Quản lý Tài Xế
- 🚚 Quản lý Chuyển Đổi
- 🕒 Lịch sử Bảo Trì
- 📝 Nhật ký Nhiên Liệu
- 🔗 Quản lý Tài Khoản
- 💻 Quản lý Nhân Viên

👤 Đăng xuất

✖ Thoát

**QUẢN LÝ NHẬT KÝ NHIÊN LIỆU**

Xe:	29H-678.90	Trạng thái:	Từ chối
Ngày đổ:	2025-11-13	Tài xế đổ:	NV001 - Nguyễn Văn An
Số lít:	30.00	Giờ đổ (HH:MM):	23:00
Số Odo (Km):	10	Tổng chi phí:	2500000.00

Duyệt Từ chối

**Danh sách nhật ký nhiên liệu (Sắp xếp mới nhất)**

Mã NK	Biển số xe	Tên Tài xé	Ngày đổ	Số lít	Tổng chi phí	Trạng thái
3	29H-678.90	Nguyễn Văn An	2025-11-13 23:06	30.00	2500000.00	Từ chối
2	29H-678.90	Trần Thị Bình	2025-11-12 08:11	50.00	1250000.00	Đã duyệt
1	51G-123.45	Nguyễn Văn An	2025-11-07 08:11	30.00	750000.00	Đã duyệt

Hình 2. 7. Bảng nhật ký nhiên liệu

## - Quản lý tài khoản

Hệ Thống Quản Lý Vận Tải (Vai trò: Admin)

**HỆ THỐNG VẬN TẢI**

- ✉ Quản lý Xe
- 👤 Quản lý Tài Xế
- 🚚 Quản lý Chuyển Đổi
- 🕒 Lịch sử Bảo Trì
- 📝 Nhật ký Nhiên Liệu
- 🔗 Quản lý Tài Khoản
- 💻 Quản lý Nhân Viên

👤 Đăng xuất

✖ Thoát

**QUẢN LÝ TÀI KHOẢN**

Tên đăng nhập:	admin	Mật khẩu:	*****
Nhân viên:	AD001 - Trần Văn Quản Trị	Vai trò:	Admin

Thêm Lưu Sửa Hủy Xóa

**Danh sách tài khoản (Không hiển thị mật khẩu)**

Tên đăng nhập	Mã NV	Họ Tên Nhân Viên	Vai trò
admin	AD001	Trần Văn Quản Trị	Admin
an.an	NV001	Nguyễn Văn An	Tài Xe
binh.tt	NV002	Trần Thị Bình	Tài Xe
cuong.lv	NV003	Lê Văn Cường	Tài Xe
hungok	NV012	Phạm Mạnh Hưng	Tài Xe

Hình 2. 8. Bảng quản lý tài khoản

## - Quản lý nhân viên

The screenshot shows a Windows application window titled 'Test Quản lý Nhân Viên'. The main title bar is 'QUẢN LÝ TOÀN BỘ NHÂN VIÊN'. Below the title bar, there are four input fields: 'Mã nhân viên' (AD001), 'Địa chỉ' (Cần Thơ), 'Họ và tên' (Admin), and 'Trạng thái NV' (Đang làm việc). Below these fields are four buttons: 'Thêm' (Add), 'Lưu' (Save), 'Sửa' (Edit), and 'Xóa' (Delete). A search bar at the bottom left contains the placeholder 'Tim kiếm (Tên, Mã, SĐT):' followed by a 'Tim kiếm' button.

Danh sách nhân viên (Tất cả)					
Mã NV	Họ Tên	Số điện thoại	Địa chỉ	Trạng thái NV	
AD001	Admin	03393922432	Cần Thơ	Đang làm việc	
AD002	Phạm Mạnh Hùng	02131231321	Long Xuyên, An Giang	Đang làm việc	
NV001	Nguyễn Văn An	0901112221	456 Đường Tài Xế, Q. Bình Thạnh, TP. HCM	Đang làm việc	
NV002	Trần Thị Bình	0902223332	789 Đường Văn Cao, Q. 7, TP. HCM	Đang làm việc	
NV003	Lê Văn Cường	0903334443	101 Đường Cửu, Q. 3, TP. HCM	Nghỉ	
NV004	Nguyễn Văn Chiến	0384772884	Long Xuyên, An Giang	Đang làm việc	
NV005	Trần Thị F	0202838282	An Giang	Đang làm việc	
NV006	Nguyễn Văn E	22132313123	An Giang	Đang làm việc	

Hình 2. 9. Bảng quản lý nhân viên

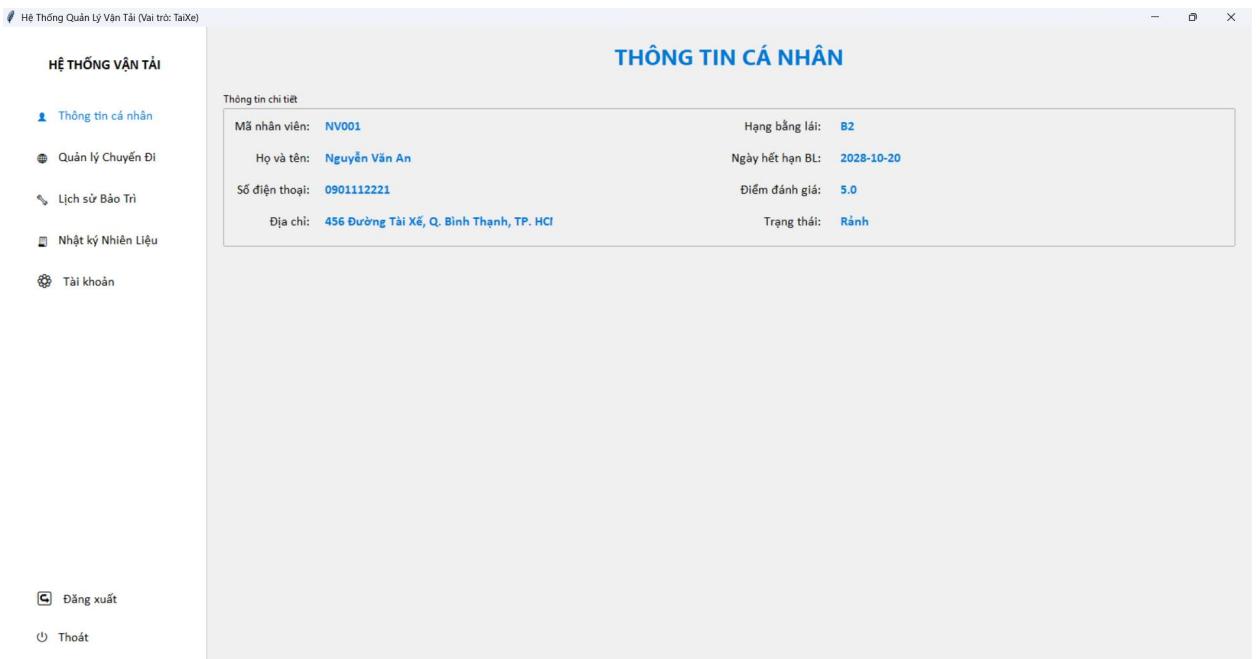
### 3. Giao diện chính TaiXe:

- Trang  
chủ



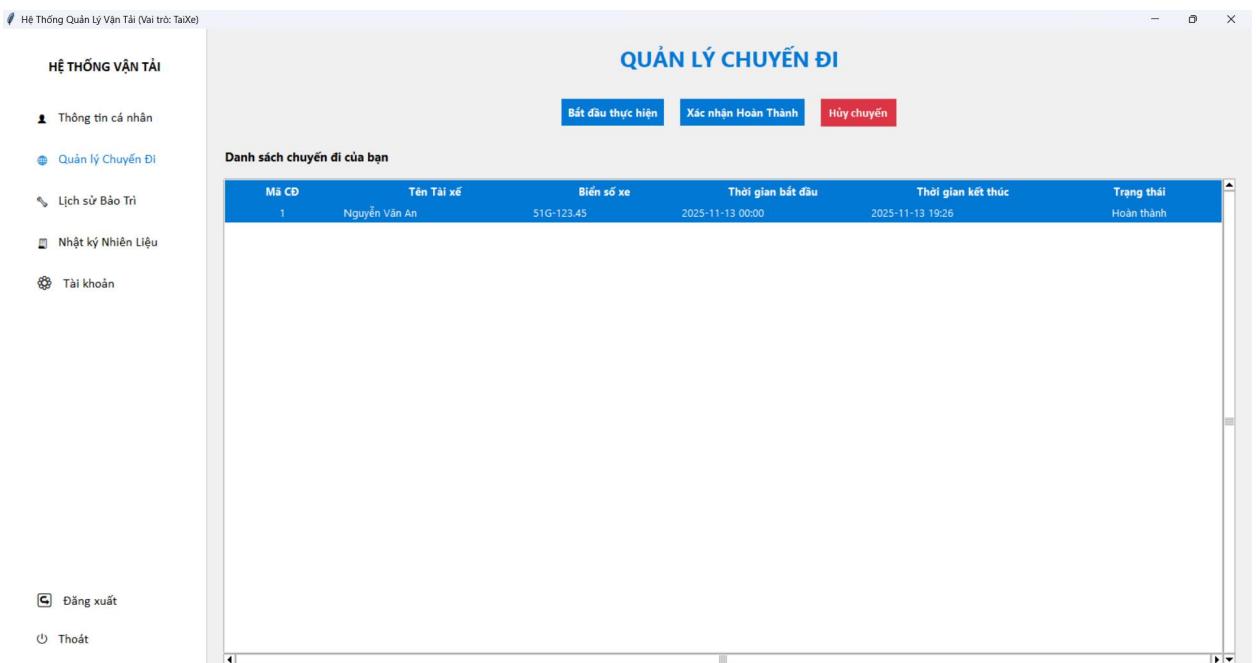
*Hình 2. 10. Màn hình chính Tài xế*

## - Thông tin cá nhân



Hình 2. 11. Bảng thông tin cá nhân

## - Quản lý chuyển đi:



Hình 2. 12. Bảng quản lý chuyển đi

### - Lịch sử bảo trì

**LỊCH SỬ BẢO TRÌ XE CỦA BẠN**

Mã BT	Biển số xe	Ngày bảo trì	Chi phí	Mô tả	Người nhập

Hình 2. 13. Bảng lịch sử bảo trì

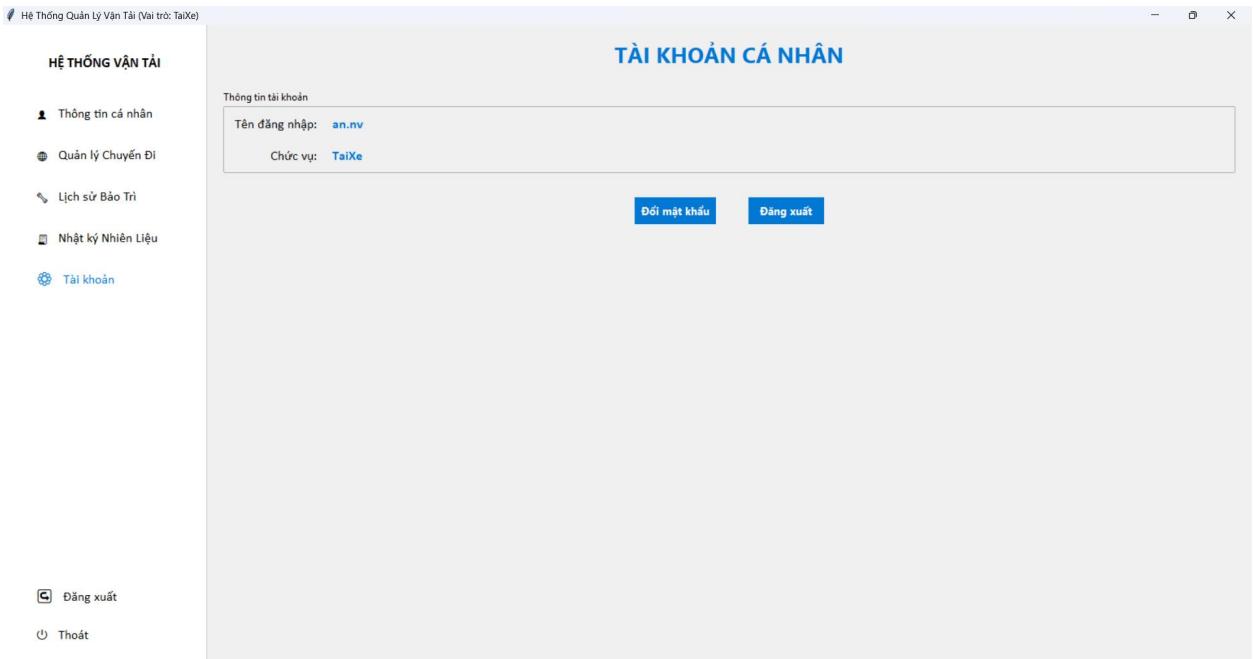
### - Nhật ký nhiên liệu

**QUẢN LÝ NHẬT KÝ NHIÊN LIỆU**

Mã NK	Biển số xe	Tên Tài xế	Ngày đổ	Số lít	Tổng chi phí	Trạng thái
3	29H-678.90	Nguyễn Văn An	2025-11-13 23:06	30.00	2500000.00	Từ chối
1	51G-123.45	Nguyễn Văn An	2025-11-07 08:11	30.00	750000.00	Đã duyệt

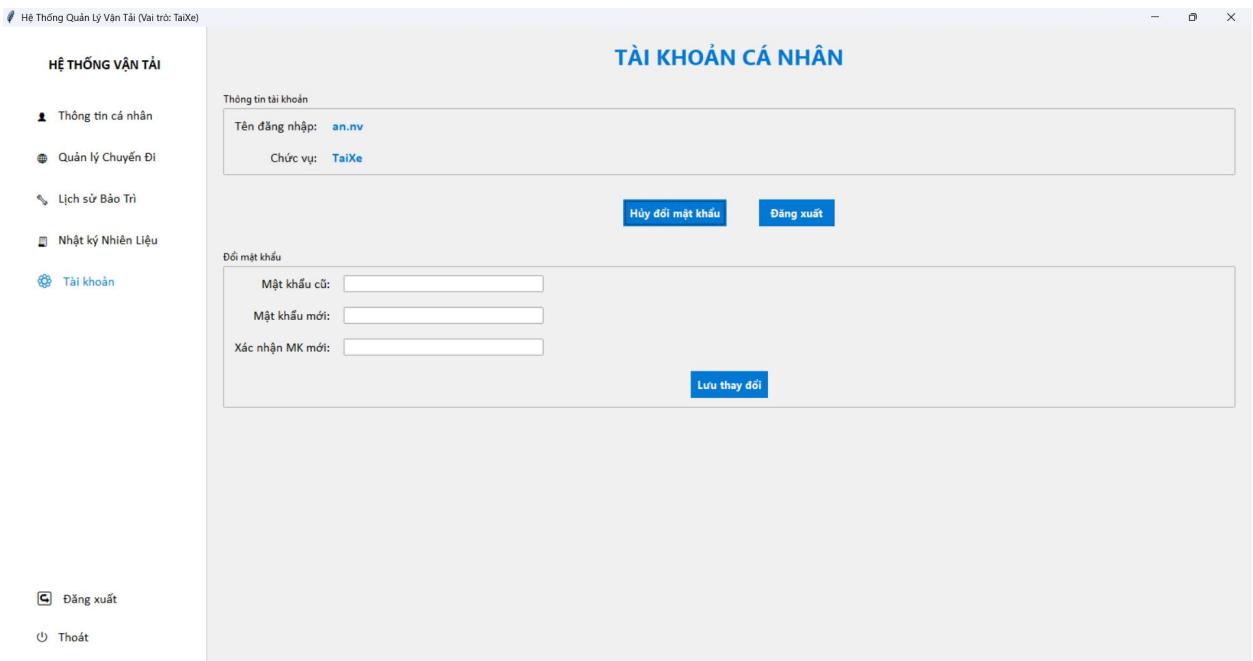
Hình 2. 14. Bảng nhật ký nhiên liệu

## - Tài khoản



Hình 2. 15. Bảng tài khoản

+ Khi ấn “Đổi mật khẩu”



Hình 2. 16. Bảng đổi mật khẩu

### III. Các bước xây dựng ứng dụng

#### 1. Cài đặt các thư viện cần thiết:

- **Python:** Đảm bảo bạn đã cài đặt Python trên máy tính của mình.
- **SQL Server 2022:** Cài đặt máy chủ SQL Server (ví dụ: bản Developer hoặc Express) và tạo một cơ sở dữ liệu cho ứng dụng của bạn.
- **pyodbc:** Cài đặt thư viện này để kết nối Python với cơ sở dữ liệu SQL Server.

```
pip install pyodbc
```

#### 2. Thiết kế cơ sở dữ liệu MySQL

##### a. Mô tả Cấu trúc Cơ sở dữ liệu QL\_VanTai

- Bảng 1: NhanVien

TenCot	KieuDuLieu	ChoPhepNull	DoDaiToiDa	GiaTriMacDinh	KhoaChinh(PK)	KhoaNgoai(FK)
MaNhanVien	nvarchar	NO	10		PK	
HoVaTen	nvarchar	NO	100			
SoDienThoai	varchar	NO	15			
DiaChi	nvarchar	YES	255			
TrangThai	int	NO		((1))		

- Bảng 2: TaiXe

TenCot	KieuDuLieu	ChoPhepNull	DoDaiToiDa	GiaTriMacDinh	KhoaChinh(PK)	KhoaNgoai(FK)
MaNhanVien	nvarchar	NO	10		PK	FK
HangBangLai	varchar	YES	10			
NgayHetHanBangLai	date	YES				
DiemDanhGia	float	YES		((5.0))		
TrangThaiTaiXe	int	NO		((1))		

- Bảng 3: TaiKhoan

TenCot	KieuDuLieu	ChoPhepNull	DoDaiToiDa	GiaTriMacDinh	KhoaChinh(PK)	KhoaNgoai(FK)

TenDangNhap	nvarchar	NO	50		PK	
MatKhau	nvarchar	NO	255			
MaNhanVien	nvarchar	NO	10			FK
VaiTro	nvarchar	NO	20	('TaiXe')		

- Bảng 4: Xe

TenCot	KieuDuLieu	ChoPhepNull	DoDaiTroiDa	GiaTriMacDinh	KhoaChinh(PK)	KhoaNgoai(FK)
BienSoXe	varchar	NO	15		PK	
LoaiXe	nvarchar	YES	50			
HangSanXuat	nvarchar	YES	50			
DongXe	nvarchar	YES	50			
NamSanXuat	int	YES				
SoKhungVIN	varchar	YES	17			
NgayHetHanDangKiem	date	YES				
NgayHetHanBaoHiem	date	YES				
TrangThai	int	NO		((1))		
MaNhanVienHienTai	nvarchar	YES	10			FK

- Bảng 5: ChuyenDi

TenCot	KieuDuLieu	ChoPhepNull	DoDaiToiDa	GiaTriMacDinh	KhoaChinh(PK)	KhoaNgoai(FK)
MaChuyenDi	int	NO			PK	
MaNhanVien	nvarchar	NO	10			FK
BienSoXe	varchar	NO	15			FK
DiemBatDau	nvarchar	YES	-1			
DiemKetThuc	nvarchar	YES	-1			
ThoiGianBatDau	datetime	YES				
ThoiGianKetThuc	datetime	YES				
TrangThai	int	NO		((0))		

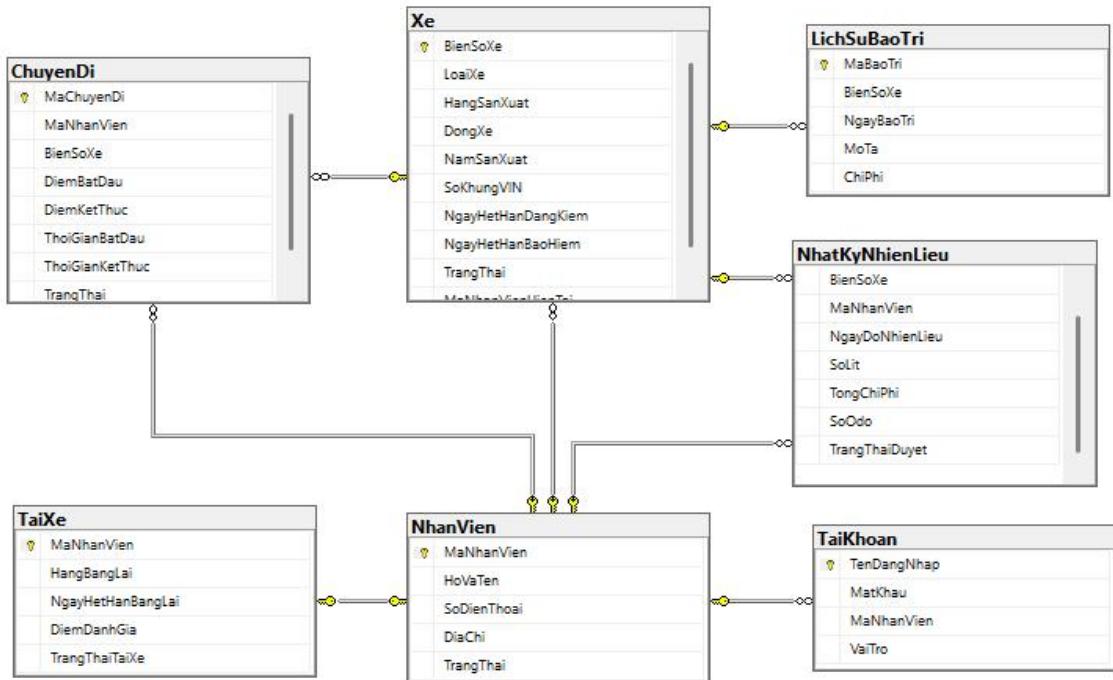
- Bảng 6: LichSuBaoTri

TenCot	KieuDuLieu	ChoPhepNull	DoDaiToiDa	GiaTriMacDinh	KhoaChinh(PK)	KhoaNgoai(FK)
MaBaoTri	int	NO			PK	
BienSoXe	varchar	NO	15			FK
NgayBaoTri	date	NO				
MoTa	nvarchar	YES	-1			
ChiPhi	decimal	YES				
MaNhanVienNhaP	nvarchar	YES	10			FK

- Bảng 7: NhatKiNhienLieu

TenCot	KieuDuLieu	ChoPhepNull	DoDaiToiDa	GiaTriMacDinh	KhoaChinh(PK)	KhoaNgoai(FK)
MaNhatKy	int	NO			PK	
BienSoXe	varchar	NO	15			FK
MaNhanVien	nvarchar	YES	10			FK
NgayDoNhienLieu	datetime	YES		(getdate())		
SoLit	decimal	YES				
TongChiPhi	decimal	YES				
SoOdo	int	YES				
TrangThaiDuyet	int	NO		((0))		

## b. Mối liên kết của các bảng:



Hình 3. 1. Liên kết giữa các bảng

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

### a. Quản lý Nhân Viên (Bảng NhanVien)

- Thêm thông tin Nhân viên mới (Mã NV, Tên, SĐT, Địa chỉ...).
- Lưu thông tin Nhân viên (khi Thêm mới hoặc Sửa).
- Sửa thông tin Nhân viên.
- Hủy thao tác (xóa trắng các ô nhập liệu).
- Xóa thông tin Nhân viên (hoặc cập nhật TrangThai sang "Nghi việc").
- Xem danh sách tất cả Nhân viên.
- Tìm kiếm Nhân viên (theo Tên, Mã NV, SĐT).

### b. Quản lý Tài Xế (Bảng TaiXe)

- Thêm/Cập nhật thông tin Tài xế (chọn từ một Nhân viên, nhập Hạng bằng lái, Ngày hết hạn...).

- Sửa thông tin bằng lái, cập nhật Điểm đánh giá.
- Thay đổi Trạng thái Tài xế (ví dụ: "Sẵn sàng", "Đang chạy", "Nghỉ").
- Xóa thông tin Tài xế (xóa vai trò tài xế khỏi nhân viên).
- Xem danh sách tất cả Tài xế.

#### **c. Quản lý Xe (Bảng Xe)**

- Thêm thông tin Xe mới (Biển số, Loại xe, Hãng sản xuất, Ngày đăng kiểm...).
- Lưu thông tin Xe.
- Sửa thông tin Xe (cập nhật ngày hết hạn bảo hiểm, đăng kiểm...).
- Xóa thông tin Xe.
- Xem danh sách tất cả Xe.
- Tìm kiếm Xe (theo Biển số, Loại xe...).

#### **d. Quản lý Tài Khoản (Bảng TaiKhoan)**

- Thêm Tài khoản mới (Tên đăng nhập, Mật khẩu, Vai trò) và liên kết với một MaNhanVien.
- Sửa thông tin Tài khoản (đặt lại Mật khẩu, thay đổi Vai trò).
- Xóa Tài khoản.
- Xem danh sách Tài khoản.

#### **e. Quản lý Chuyến Đi (Bảng ChuyenDi)**

- Thêm Chuyến đi mới (chọn MaNhanVien (Tài xế), BienSoXe, nhập Điểm đi/đến, Thời gian...).
- Sửa thông tin Chuyến đi.
- Cập nhật TrangThai Chuyến đi (ví dụ: "Chưa bắt đầu", "Đang chạy", "Hoàn thành", "Đã hủy").
- Xóa Chuyến đi (nếu chưa diễn ra).
- Xem danh sách tất cả Chuyến đi.
- Tìm kiếm/Lọc Chuyến đi (theo ngày, theo Tài xế, theo Xe).

#### **f. Quản lý Bảo Trì (Bảng LichSuBaoTri)**

- Thêm Lịch sử Bảo trì mới (chọn BienSoXe, nhập Ngày, Mô tả, Chi phí).
- Sửa thông tin Lịch sử Bảo trì.
- Xóa thông tin Lịch sử Bảo trì.
- Xem danh sách Lịch sử Bảo trì (có thể lọc theo BienSoXe).

#### **g. Quản lý Nhiên Liệu (Bảng NhatKyNhiemLieu)**

- Thêm Nhật ký Nhiên liệu mới (chọn BienSoXe, MaNhanVien (người đổ), nhập Số Lít, Tổng chi phí, Số Odo...).
- Sửa thông tin Nhật ký Nhiên liệu.
- Xóa thông tin Nhật ký Nhiên liệu.
- Cập nhật TrangThaiDuyet (nếu có quy trình duyệt chi phí).
- Xem danh sách Nhật ký Nhiên liệu (lọc theo Xe, theo Nhân viên, theo ngày).

### **IV. Thiết kế và cài đặt chương trình**

#### **1. Cài đặt môi trường**

```
pip install pyodbc tkcalendar
```

##### **Giải thích:**

- pyodbc: là thư viện dùng để kết nối Python với SQL Server.
- tkcalendar: dùng để tạo lịch chọn ngày tháng

#### **2. Chuẩn bị CSDL SQL Server**

### a. Bảng NhanVien

```
CREATE TABLE NhanVien (
    MaNhanVien NVARCHAR(10) NOT NULL PRIMARY KEY,
    HoVaTen NVARCHAR(100) NOT NULL,
    SoDienThoai VARCHAR(15) NOT NULL,
    DiaChi NVARCHAR(255),
    TrangThai INT DEFAULT 1 NOT NULL -- 0=Nghi, 1=Đang làm việc
);
GO
```

Hình 4. 2. 1. Bảng Nhân Viên

### b. Bảng TaiXe

```
CREATE TABLE TaiXe (
    MaNhanVien NVARCHAR(10) NOT NULL PRIMARY KEY,
    HangBangLai VARCHAR(10),
    NgayHetHanBangLai DATE,
    DiemDanhGia FLOAT DEFAULT 5.0,
    TrangThaiTaiXe INT DEFAULT 1 NOT NULL, -- 1=Rảnh, 2=Đang lái
    FOREIGN KEY (MaNhanVien) REFERENCES NhanVien(MaNhanVien)
);
GO
```

Hình 4. 2. 2. Bảng Tài Xe

### c. Bảng Xe

```
CREATE TABLE Xe (
    BienSoXe VARCHAR(15) PRIMARY KEY,
    LoaiXe NVARCHAR(50),
    HangSanXuat NVARCHAR(50),
    DongXe NVARCHAR(50),
    NamSanXuat INT,
    SoKhungVIN VARCHAR(17),
    NgayHetHanDangKiem DATE,
    NgayHetHanBaoHiem DATE,
    TrangThai INT DEFAULT 1 NOT NULL, -- 0=Bảo trì, 1=Hoạt động, 2=Hỏng
    MaNhanVienHienTai NVARCHAR(10) NULL,
    FOREIGN KEY (MaNhanVienHienTai) REFERENCES NhanVien(MaNhanVien)
);
GO
```

Hình 4. 2. 3. Bảng Xe

#### d. Bảng ChuyenDi

```
CREATE TABLE ChuyenDi (
    MaChuyenDi INT PRIMARY KEY IDENTITY(1,1),
    MaNhanVien Nvarchar(10) NOT NULL,
    BienSoXe VARCHAR(15) NOT NULL,
    DiemBatDau NVARCHAR(MAX),
    DiemKetThuc NVARCHAR(MAX),
    ThoiGianBatDau DATETIME,
    ThoiGianKetThuc DATETIME NULL,
    TrangThai INT DEFAULT 0 NOT NULL, -- 0=Đã gán, 1=Đang thực hiện, 2=Hoàn thành, 3=Hủy
    FOREIGN KEY (MaNhanVien) REFERENCES NhanVien(MaNhanVien),
    FOREIGN KEY (BienSoXe) REFERENCES Xe(BienSoXe)
);
GO
```

Hình 4. 2. 4. Bảng Chuyển Di

#### e. Bảng LichSuBaoTri

```
CREATE TABLE LichSuBaoTri (
    MaBaoTri INT PRIMARY KEY IDENTITY(1,1),
    BienSoXe VARCHAR(15) NOT NULL,
    NgayBaoTri DATE NOT NULL,
    MoTa NVARCHAR(MAX),
    ChiPhi DECIMAL(18, 2),
    FOREIGN KEY (BienSoXe) REFERENCES Xe(BienSoXe)
);
GO
```

Hình 4. 2. 5. Bảng Lịch Sử Bảo Trì

#### f. Bảng NhatKyNhienLieu

```
CREATE TABLE NhatKyNhienLieu (
    MaNhatKy INT PRIMARY KEY IDENTITY(1,1),
    BienSoXe VARCHAR(15) NOT NULL,
    MaNhanVien Nvarchar(10) NULL,
    NgayDoNhienLieu DATETIME DEFAULT GETDATE(),
    SoLit DECIMAL(10, 2),
    TongChiPhi DECIMAL(18, 2),
    SoOdo INT,
    FOREIGN KEY (BienSoXe) REFERENCES Xe(BienSoXe),
    FOREIGN KEY (MaNhanVien) REFERENCES NhanVien(MaNhanVien)
);
GO
ALTER TABLE NhatKyNhienLieu
ADD TrangThaiDuyet INT DEFAULT 0 NOT NULL; -- 0=Chờ, 1=Duyệt, 2=Từ chối
```

Hình 4. 2. 6. Bảng Nhật Ký Nhiệt Liệu

### g. Bảng TaiKhoan

```
CREATE TABLE TaiKhoan (
    TenDangNhap NVARCHAR(50) PRIMARY KEY,
    MatKhau NVARCHAR(255) NOT NULL,
    MaNhanVien Nvarchar(10) NOT NULL,
    VaiTro NVARCHAR(20) DEFAULT 'TaiXe' NOT NULL, -- Ví dụ: 'TaiXe', 'Admin'
    FOREIGN KEY (MaNhanVien) REFERENCES NhanVien(MaNhanVien)
);
GO
```

Hình 4. 2. 7. Bảng Tài Khoản

## 3. Code ứng dụng với Tkinter và SQL Server

### a. Kết nối với SQL Server

```
def connect_db():
    """Hàm kết nối đến CSDL SQL Server."""
    try:
        conn_string = (
            r'DRIVER={SQL Server};'
            r'SERVER=LAPTOP-MKC70SQE\SQLEXPRESS;' # Giữ nguyên server
        của bạn
            r'DATABASE=QL_VanTai;'
            r'Trusted_Connection=yes;'
        )
        conn = pyodbc.connect(conn_string)
        return conn
    except pyodbc.Error as e:
        messagebox.showerror("Lỗi kết nối CSDL", f"Không thể kết nối
đến SQL Server:\n{e}")
        return None
    except Exception as e:
        messagebox.showerror("Lỗi không xác định", f"Lỗi: {str(e)}")
        return None
```

## b. Form Login

```
COLOR_PRIMARY = "#0078D7" # Xanh dương
COLOR_LIGHT_BLUE = "#E0EFFF"
COLOR_DARK_BLUE = "#005a9e"
COLOR_WHITE = "#ffffff"
COLOR_BLACK = "#1f1f1f"
COLOR_GREY = "#a0a0a0"
COLOR_LIGHT_GREY = "#f0f0f0"

FONT_BIG = ("Arial", 24, "bold")
FONT_MEDIUM = ("Arial", 14, "bold")
FONT_NORMAL = ("Arial", 12)
FONT_NORMAL_BOLD = ("Arial", 12, "bold")
```

- Câu hình

- Hàm xử lý logic

```
def hash_password(password):
    """Hàm băm mật khẩu bằng SHA-256."""
    return hashlib.sha256(password.encode('utf-8')).hexdigest()

def check_login(event=None): # Thêm event=None để bắt sự kiện Enter
    username = entry_username.get()
    password = entry_password.get()

    if not username or not password:
        messagebox.showwarning("Thiếu thông tin", "Vui lòng nhập Tên đăng nhập và Mật khẩu.")
        return

    conn = utils.connect_db()
    if conn is None:
        messagebox.showerror("Lỗi Kết Nối", "Không thể kết nối CSDL. Vui lòng kiểm tra utils.py hoặc driver.")
        return

    try:
        cur = conn.cursor()
        sql = "SELECT MatKhau, VaiTro FROM TaiKhoan WHERE TenDangNhap = ?"
        cur.execute(sql, (username,))
        record = cur.fetchone()

        if record:
```

```

# Tên đăng nhập ĐÚNG, kiểm tra mật khẩu
db_hashed_password = record[0]
db_role = record[1]
input_hashed_password = hash_password(password)

if input_hashed_password == db_hashed_password:
    # MẬT KHẨU ĐÚNG
    messagebox.showinfo("Thành công", f"Đăng nhập thành
công với vai trò: {db_role}")
    login_window.destroy()

    # Truyền 'username' và 'role' sang main.py
    open_main_menu(username, db_role)
else:
    messagebox.showerror("Sai thông tin", "Sai Mật khẩu.
Vui lòng thử lại.")

else:
    # Tên đăng nhập SAI
    messagebox.showerror("Sai thông tin", "Không tìm thấy Tên
đăng nhập.")
except pyodbc.Error as e:
    messagebox.showerror("Lỗi SQL", f"Lỗi truy vấn:\n{str(e)}")
except Exception as e:
    messagebox.showerror("Lỗi không xác định", f"Lỗi: {str(e)}")
finally:
    if conn:
        conn.close()

def open_main_menu(username, role):
"""
Hàm này chạy file main.py và truyền Tên đăng nhập, Vai trò vào.
"""
print(f"Mở Main Menu: User={username}, Role={role}")

python_executable = sys.executable
current_dir = os.path.dirname(os.path.abspath(__file__))
main_menu_path = os.path.join(current_dir, "main.py")

if not os.path.exists(main_menu_path):
    messagebox.showerror("Lỗi", "Không tìm thấy file: main.py")
    return

try:
    # Truyền 2 tham số: username và role
    subprocess.Popen([python_executable, main_menu_path, username,
role])
except Exception as e:

```

```
messagebox.showerror("Lỗi khi mở Main Menu", f"Không thể khởi  
chạy main.py:\n{e}")
```

### - Khởi tạo cửa sổ chính

```
login_window = tk.Tk()  
login_window.title("Đăng nhập - Hệ thống Quản lý Vận tải")  
  
# Căn giữa cửa sổ (Lớn hơn)  
w = 700  
h = 450  
ws = login_window.winfo_screenwidth()  
hs = login_window.winfo_screenheight()  
x = (ws/2) - (w/2)  
y = (hs/2) - (h/2)  
login_window.geometry('%dx%d+%d+%d' % (w, h, x, y))  
login_window.resizable(False, False)  
login_window.configure(bg=COLOR_WHITE)
```

### - Tạo layout chính

```
login_window.grid_columnconfigure(0, weight=2) # Cột trái (Graphic)  
login_window.grid_columnconfigure(1, weight=3) # Cột phải (Form)  
login_window.grid_rowconfigure(0, weight=1)  
  
# --- CỘT TRÁI (Graphic) ---  
left_frame = tk.Frame(login_window, bg=COLOR_PRIMARY)  
left_frame.grid(row=0, column=0, sticky="nsew")  
left_frame.pack_propagate(False) # Ngăn co lại  
  
# --- CỘT PHẢI (Form) ---  
right_frame = tk.Frame(login_window, bg=COLOR_WHITE, padx=50, pady=50)  
right_frame.grid(row=0, column=1, sticky="nsew")
```

### - Tạo Widgets

```
# --- Widgets CỘT TRÁI ---  
tk.Label(left_frame, text="ELCOME", font=("Arial", 100), bg=COLOR_PRIMARY,  
fg=COLOR_WHITE).pack(pady=(80, 0))  
tk.Label(left_frame, text="HỆ THỐNG", font=("Arial", 20),  
bg=COLOR_PRIMARY, fg=COLOR_WHITE).pack()  
tk.Label(left_frame, text="QUẢN LÝ\nVẬN TẢI", font=FONT_BIG,  
bg=COLOR_PRIMARY, fg=COLOR_WHITE, justify=tk.CENTER).pack()  
tk.Label(left_frame, text="Đăng nhập để tiếp tục", font=FONT_NORMAL,  
bg=COLOR_PRIMARY, fg=COLOR_LIGHT_BLUE).pack(pady=10)
```

```

# --- Widgets CỘT PHẢI ---
tk.Label(right_frame, text="ĐĂNG NHẬP", font=FONT_BIG, bg=COLOR_WHITE,
fg=COLOR_BLACK).pack(pady=(30, 20))

# Ô Tên đăng nhập
tk.Label(right_frame, text="Tên đăng nhập", font=FONT_NORMAL,
bg=COLOR_WHITE, fg=COLOR_BLACK, anchor="w").pack(fill="x", pady=(10,0))
entry_username = tk.Entry(right_frame, font=FONT_MEDIUM,
bg=COLOR_LIGHT_GREY, bd=0, relief="flat", insertbackground=COLOR_BLACK)
entry_username.pack(fill="x", ipady=8, pady=(5, 10))

# Ô Mật khẩu
tk.Label(right_frame, text="Mật khẩu", font=FONT_NORMAL,
bg=COLOR_WHITE, fg=COLOR_BLACK, anchor="w").pack(fill="x", pady=(10,0))
entry_password = tk.Entry(right_frame, font=FONT_MEDIUM,
bg=COLOR_LIGHT_GREY, bd=0, relief="flat", show="*",
insertbackground=COLOR_BLACK)
entry_password.pack(fill="x", ipady=8, pady=(5, 20))

# Nút đăng nhập
btn_login = tk.Button(right_frame,
                      text="Đăng nhập",
                      font=FONT_NORMAL_BOLD,
                      bg=COLOR_PRIMARY,
                      fg=COLOR_WHITE,
                      activebackground=COLOR_DARK_BLUE,
                      activeforeground=COLOR_WHITE,
                      relief="flat",
                      bd=0,
                      pady=10,
                      cursor="hand2", # Con trỏ hình bàn tay
                      command=check_login)
btn_login.pack(fill="x")

- Gán sự kiện và chạy ứng dụng

# --- Hiệu ứng Hover cho nút ---
btn_login.bind("<Enter>", on_btn_enter)
btn_login.bind("<Leave>", on_btn_leave)

# --- Bắt sự kiện nhấn Enter ---
login_window.bind('<Return>', check_login)

# --- Đặt focus (con trỏ) vào ô username ---
entry_username.focus()

# --- Chạy cửa sổ ---
login_window.mainloop()

```

### c. Form Main

- Khởi tạo biến toàn cục và lấy vai trò

```
# --- Lấy vai trò (role) từ lúc đăng nhập ---
try:
    USER_USERNAME = sys.argv[1]
    USER_ROLE = sys.argv[2]
except IndexError:
    USER_USERNAME = "test_admin" # (dùng để test)
    USER_ROLE = "Admin"
    print("Không thấy vai trò, mặc định là Admin để test.")

print(f"Đang chạy Main Menu: User={USER_USERNAME}, Role={USER_ROLE}")

# --- Biến trạng thái toàn cục ---
current_page_frame = None
current_active_button = None
```

- Các hàm tiện ích và quản lý

```
def show_page(page_creator_func):
    """Xóa frame cũ và hiển thị frame mới trong main_frame."""
    global current_page_frame

    if current_page_frame:
        current_page_frame.destroy()

    # Truyền main_frame làm 'master' cho trang con
    current_page_frame = page_creator_func(main_frame)
    current_page_frame.pack(fill=tk.BOTH, expand=True)

def show_homepage():
    """Hiển thị lại trang chủ (Lời chào)."""
    global current_page_frame
    if current_page_frame:
        current_page_frame.destroy()
        current_page_frame = None

    # create_main_content sẽ được định nghĩa ở dưới
    create_main_content(main_frame)

def do_logout(root, force=False):
    """Đóng cửa sổ chính (root) và mở lại login.py."""

    if not force:
        if not messagebox.askyesno("Xác nhận Đăng xuất", "Bạn có chắc chắn muốn đăng xuất?"):
            return
```

```

try:
    root.destroy() # Đóng cửa sổ main.py

    python_executable = sys.executable
    script_dir = os.path.dirname(os.path.abspath(__file__))
    login_py_path = os.path.join(script_dir, "login.py")

    if not os.path.exists(login_py_path):
        messagebox.showerror("Lỗi", "Không tìm thấy file login.py!")
        return

    subprocess.Popen([python_executable, login_py_path])

except Exception as e:
    messagebox.showerror("Lỗi Đăng xuất", f"Không thể khởi động lại login.py:\n{e}")

def reset_active_button():
    """Trả nút đang active về trạng thái bình thường."""
    global current_active_button
    if current_active_button:
        try:
            # Trả về màu nền/chữ gốc
            current_active_button.config(bg=NAV_BG, fg=NAV_FG)

            if current_active_button == title_btn:
                # Gắn lại hover CHỮ (cho title_btn)
                current_active_button.bind("<Enter>", lambda e:
e.widget.config(bg=NAV_BG, fg=NAV_HOVER_FG))
                current_active_button.bind("<Leave>", lambda e:
e.widget.config(bg=NAV_BG, fg=NAV_FG))
            else:
                # Gắn lại hover NỀN (cho các nút khác)
                current_active_button.bind("<Enter>", lambda e:
e.widget.config(bg=NAV_HOVER_BG, fg=NAV_HOVER_FG))
                current_active_button.bind("<Leave>", lambda e:
e.widget.config(bg=NAV_BG, fg=NAV_FG))
        except tk.TclError:
            pass
        current_active_button = None

def set_active_button(button_widget):
    """Tô màu CHỮ của nút được chọn và gỡ hover."""
    global current_active_button

    # 1. Reset nút cũ trước
    reset_active_button()

```

```

try:
    # 2. Chỉ đổi MÀU CHỮ (fg) thành màu xanh (NAV_HOVER_FG)
    button_widget.config(bg=NAV_BG, fg=NAV_HOVER_FG)

    # 3. Gỡ sự kiện di chuột để nó "dính" màu
    button_widget.unbind("<Enter>")
    button_widget.unbind("<Leave>")

    # 4. Lưu lại nút này là nút active
    current_active_button = button_widget
except TclError:
    pass

def disable_button(btn):
    """Hàm tùy chỉnh để vô hiệu hóa tk.Button (vì 'state' làm xấu)."""
    btn.config(fg=NAV_DISABLED_FG, command=lambda: None)
    btn.unbind("<Enter>")
    btn.unbind("<Leave>")

```

### - Khởi tạo layout chính

```

root = tk.Tk()
root.title(f"Hệ Thống Quản Lý Vận Tải (Vai trò: {USER_ROLE})")
root.state('zoomed')
root.configure(bg=MAIN_BG)

# --- Thanh điều hướng bên trái ---
left_nav_frame = tk.Frame(root, bg=NAV_BG, width=250)
left_nav_frame.pack(side=tk.LEFT, fill=tk.Y)
left_nav_frame.pack_propagate(False)

# --- Viền Phân Cách ---
separator = tk.Frame(root, bg=SEPARATOR_COLOR, width=1)
separator.pack(side=tk.LEFT, fill=tk.Y)

# --- Khung nội dung chính ---
main_frame = tk.Frame(root, bg=MAIN_BG)
main_frame.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

```

### - Tạo thanh điều hướng

```

# --- Hàm helper tạo nút nav ---
def create_nav_button(parent, text, icon, page_command_func):
    btn_text = f" {icon} {text}"
    btn = tk.Button(parent,

```

```

        text=btn_text,
        font=NAV_BUTTON_FONT,
        bg=NAV_BG, fg=NAV_FG,
        relief="flat", borderwidth=0,
        anchor="w", padx=20, pady=10,
        activebackground=NAV_HOVER_BG,
        activeforeground=NAV_HOVER_FG,
        command=lambda: (page_command_func(),
set_active_button(btn))
    )

    btn.bind("<Enter>", lambda e: e.widget.config(bg=NAV_HOVER_BG,
fg=NAV_HOVER_FG))
    btn.bind("<Leave>", lambda e: e.widget.config(bg=NAV_BG,
fg=NAV_FG))

    btn.pack(side=tk.TOP, fill=tk.X, pady=2, padx=10)
    return btn

# --- Nút Tiêu đề (Trang chủ) ---
title_btn = tk.Button(left_nav_frame,
                      text="HỆ THỐNG VẬN TẢI",
                      font=NAV_TITLE_FONT,
                      bg=NAV_BG, fg=NAV_FG,
                      anchor="center",
                      padx=20,
                      relief="flat", borderwidth=0,
                      activebackground=NAV_BG,
                      activeforeground=NAV_HOVER_FG,
                      command=lambda: (show_homepage(),
set_active_button(title_btn))
    )
title_btn.bind("<Enter>", lambda e: e.widget.config(bg=NAV_BG,
fg=NAV_HOVER_FG))
title_btn.bind("<Leave>", lambda e: e.widget.config(bg=NAV_BG,
fg=NAV_FG))
title_btn.pack(side=tk.TOP, fill=tk.X, pady=(30, 20))

# --- Tạo các nút chức năng ---
btn_thongtin = create_nav_button(left_nav_frame, "Thông tin cá nhân",
"👤",
lambda: show_page(lambda
master_frame:thongtin_canh.create_page(master_frame, USER_USERNAME)))
btn_xe = create_nav_button(left_nav_frame, "Quản lý Xe", "🚗",
lambda: show_page(quanli_xe.create_page()))
btn_taixe = create_nav_button(left_nav_frame, "Quản lý Tài Xế", "👤",
lambda:
show_page(quanli_taixe.create_page()))

```

```

btn_chuyendi = create_nav_button(left_nav_frame, "Quản lý Chuyển Đì",
"🌐",
    lambda: show_page(lambda master_frame:
quanli_chuyendi.create_page(
    master_frame,
    USER_ROLE,
    USER_USERNAME
))
)
btn_baotri = create_nav_button(left_nav_frame, "Lịch sử Bảo Trì", "🔧",
    lambda: show_page(lambda master_frame:
quanli_lichsubaotri.create_page(
    master_frame,
    USER_ROLE,
    USER_USERNAME
)))
)
btn_nhienlieu = create_nav_button(left_nav_frame, "Nhật ký Nhiên Liệu",
"📝",
    lambda: show_page(lambda master_frame:
quanli_nhatkinguyenlieu.create_page(
    master_frame,
    USER_ROLE,
    USER_USERNAME
)))
)
btn_taikhoan_user = create_nav_button(left_nav_frame, "Tài khoản",
"👤",
    lambda: show_page(lambda master_frame:
thongtin_taikhoan.create_page(
    master_frame,
    master_frame.winfo_toplevel(),
    USER_USERNAME, # Gửi tên đăng
nhập
    USER_ROLE      # Gửi vai trò
)))
)
# Đây là cửa sổ 'root' chính
btn_taikhoan = create_nav_button(left_nav_frame, "Quản lý Tài Khoản",
"⌚",
    lambda:
show_page(quanli_taikhoan.create_page()))
btn_nhanvien = create_nav_button(left_nav_frame, "Quản lý Nhân Viên",
"👥",
    lambda:
show_page(quanli_nhanvien.create_page()))

# --- Nút Thoát (Pack ở BOTTOM) ---
btn_thoat = tk.Button(left_nav_frame,

```

```

        text="    ⏺ Thoát",
        font=NAV_BUTTON_FONT,
        bg=NAV_BG, fg=NAV_FG,
        relief="flat", borderwidth=0,
        anchor="w", padx=20, pady=10,
        activebackground=NAV_HOVER_BG,
        activeforeground=NAV_EXIT_FG,
        command=root.quit)
btn_thoat.bind("<Enter>", lambda e: e.widget.config(bg=NAV_HOVER_BG,
fg=NAV_EXIT_FG))
btn_thoat.bind("<Leave>", lambda e: e.widget.config(bg=NAV_BG,
fg=NAV_FG))
btn_thoat.pack(side=tk.BOTTOM, fill=tk.X, pady=(0, 20), padx=10)

# --- Nút Đăng xuất (Pack ở BOTTOM, ngay trên nút Thoát) ---
btn_dangxuat = tk.Button(left_nav_frame,
                        text="    🗑 Đăng xuất",
                        font=NAV_BUTTON_FONT,
                        bg=NAV_BG, fg=NAV_FG,
                        relief="flat", borderwidth=0,
                        anchor="w", padx=20, pady=10,
                        activebackground=NAV_HOVER_BG,
                        activeforeground=NAV_HOVER_FG,
                        command=lambda: do_logout(root, force=False))
btn_dangxuat.bind("<Enter>", lambda e: e.widget.config(bg=NAV_HOVER_BG,
fg=NAV_HOVER_FG))
btn_dangxuat.bind("<Leave>", lambda e: e.widget.config(bg=NAV_BG,
fg=NAV_FG))
btn_dangxuat.pack(side=tk.BOTTOM, fill=tk.X, pady=0, padx=10)

```

### - Nội dung trang chủ

```

def create_main_content(parent):
    """Tạo nội dung gốc (Lời chào) cho main_frame."""
    global current_page_frame

    home_frame = tk.Frame(parent, bg=MAIN_BG)

    lbl_title_main = tk.Label(home_frame,
                              text="HỆ THỐNG VẬN TẢI",
                              font=("Calibri", 24, "bold"),
                              bg=MAIN_BG, fg=MAIN_FG)
    lbl_title_main.pack(pady=(40, 20), fill='x', anchor='center')

    lbl_welcome_main = tk.Label(home_frame,
                               text=f"Chào mừng {USER_ROLE}",
                               ({USER_USERNAME})!", # Sửa lại để chào đúng user
                               font=("Calibri", 16),

```

```

        bg=MAIN_BG, fg=MAIN_FG)
lbl_welcome_main.pack(pady=20, fill='x', expand=True,
anchor='center')

lbl_footer_main = tk.Label(home_frame,
                           text="Phát triển bởi [Tên Nhóm Của
Bạn]",
                           font=("Calibri", 10),
                           bg=MAIN_BG, fg=MAIN_FOOTER_FG)
lbl_footer_main.pack(pady=10, side=tk.BOTTOM, anchor='center')

current_page_frame = home_frame
current_page_frame.pack(fill=tk.BOTH, expand=True)

```

### - Hàm phân quyền

```

def apply_permissions(role):
"""
Áp dụng phân quyền: Ẩn các nút không thuộc vai trò (role) này.
"""

# 1. Liệt kê TẤT CẢ các nút cần phân quyền
all_buttons = {
    "thongtin": btn_thongtin,
    "xe": btn_xe,
    "taixe": btn_tai xe,
    "chuyendi": btn_chuyendi,
    "baotri": btn_baotri,
    "nhienlieu": btn_nhienlieu,
    "taikhoan": btn_taikhoan,
    "nhanvien": btn_nhanvien,
    "taikhoan_user": btn_taikhoan_user
}

# 2. Định nghĩa vai trò nào được thấy nút nào
permissions = {
    "Admin": [
        "xe", "tai xe", "chuyendi", "baotri",
        "nhienlieu", "taikhoan", "nhanvien"
    ],
    "TaiXe": [
        "thongtin", "chuyendi", "baotri", "nhienlieu",
    ],
    "Taikhoan_user": [
    ]
    # Thêm vai trò khác ở đây
}

# 3. Lấy danh sách các nút ĐƯỢC PHÉP của vai trò hiện tại

```

```

allowed_keys = permissions.get(role, [])

# 4. Duyệt qua TẤT CẢ các nút và ẩn (pack_forget) nút không được phép
for key, button in all_buttons.items():
    if key not in allowed_keys:
        button.pack_forget()

```

- Chạy ứng dụng

```

apply_permissions(USER_ROLE)      # Áp dụng phân quyền
create_main_content(main_frame)  # Tải trang chủ lần đầu
set_active_button(title_btn)     # Đặt nút tiêu đề làm nút active
root.mainloop()

```

#### d. Form QuanLyChuyenDi

- Hàm quản lý và sự kiện

```

def set_form_state(is_enabled, widgets):
    """Bật (enable) hoặc Tắt (disable) toàn bộ các trường nhập liệu."""
    if is_enabled:
        widgets['entry_hoten'].config(state='normal')
        widgets['entry_sdt'].config(state='normal')
        widgets['entry_diachi'].config(state='normal')
        widgets['cbb_trangthai_nv'].config(state='readonly')
    else:
        widgets['entry_manv'].config(state='disabled')
        widgets['entry_hoten'].config(state='disabled')
        widgets['entry_sdt'].config(state='disabled')
        widgets['entry_diachi'].config(state='disabled')
        widgets['cbb_trangthai_nv'].config(state='disabled')

def clear_input(widgets):
    """(NÚT THÊM) Xóa trắng và Mở khóa các trường nhập liệu (Chế độ Thêm mới)."""
    set_form_state(is_enabled=True, widgets=widgets)
    widgets['entry_manv'].config(state='normal')

    widgets['entry_manv'].delete(0, tk.END)
    widgets['entry_hoten'].delete(0, tk.END)
    widgets['entry_sdt'].delete(0, tk.END)
    widgets['entry_diachi'].delete(0, tk.END)
    widgets['cbb_trangthai_nv'].set("Đang làm việc")

```

```

        widgets['entry_manv'].focus()

        tree = widgets['tree']
        if tree.selection():
            tree.selection_remove(tree.selection()[0])

        widgets["current_mode"] = "ADD"

def on_item_select(event, widgets):
    """(SỰ KIỆN CLICK) Khi click vào Treeview, chỉ đổ dữ liệu lên form
    (ở trạng thái mờ)."""
    tree = widgets['tree']
    selected = tree.selection()
    if not selected: return

    selected_item = tree.item(selected[0])
    manv = selected_item['values'][0]

    conn = utils.connect_db()
    if conn is None: return

    try:
        cur = conn.cursor()
        sql = "SELECT * FROM NhanVien WHERE MaNhanVien = ?"
        cur.execute(sql, (manv,))
        data = cur.fetchone()

        if not data:
            messagebox.showerror("Lỗi", "Không tìm thấy dữ liệu nhân
viên.")
            return

        # 1. Mở khóa form để điền
        set_form_state(is_enabled=True, widgets=widgets)
        widgets['entry_manv'].config(state='normal')

        # 2. Xóa
        widgets['entry_manv'].delete(0, tk.END)
        widgets['entry_hoten'].delete(0, tk.END)
        widgets['entry_sdt'].delete(0, tk.END)
        widgets['entry_diachi'].delete(0, tk.END)

        # 3. Điền
        widgets['entry_manv'].insert(0, data.MaNhanVien)
        widgets['entry_hoten'].insert(0, data.HoVaTen or "")
        widgets['entry_sdt'].insert(0, data.SoDienThoai or "")
        widgets['entry_diachi'].insert(0, data.DiaChi or "")
        widgets['cbb_trangthai_nv'].set("Đang làm việc" if
data.TrangThai == 1 else "Nghỉ")
    
```

```

except Exception as e:
    messagebox.showerror("Lỗi không xác định", f"Lỗi: {str(e)}")
finally:
    if conn: conn.close()
    # 4. Khóa form lại (chế độ View)
    set_form_state(is_enabled=False, widgets=widgets)
    widgets["current_mode"] = "VIEW"

def chon_nhanvien_de_sua(widgets):
    """(NÚT SỬA) Kích hoạt chế độ sửa, Mở khóa form (trừ MaNV)."""
    selected = widgets['tree'].selection()
    if not selected:
        messagebox.showwarning("Chưa chọn", "Hãy chọn một nhân viên
trong danh sách trước khi nhấn 'Sửa'")
        return

    if not widgets['entry_manv'].get():
        messagebox.showwarning("Lỗi", "Không tìm thấy mã nhân viên.
Vui lòng chọn lại.")
        return

    set_form_state(is_enabled=True, widgets=widgets)
    widgets['entry_manv'].config(state='disabled') # Không cho sửa
    khóa chính
    widgets['entry_hoten'].focus()
    widgets["current_mode"] = "EDIT"

```

### - Hàm logic CRUD

```

def load_data(widgets):
    """(READ) Tải TOÀN BỘ dữ liệu từ NhanVien VÀ LÀM MỜ FORM."""
    tree = widgets['tree']
    for i in tree.get_children():
        tree.delete(i)

    conn = utils.connect_db()
    if conn is None:
        set_form_state(is_enabled=False, widgets=widgets)
        return

    try:
        cur = conn.cursor()
        sql = "SELECT MaNhanVien, HoVaTen, SoDienThoai, DiaChi,
TrangThai FROM NhanVien"
        cur.execute(sql)
        rows = cur.fetchall()

        for row in rows:

```

```

        trang_thai_nv_text = "Đang làm việc" if row[4] == 1 else
    "Nghỉ"
        tree.insert("", tk.END, values=(
            row[0], row[1], row[2], row[3], trang_thai_nv_text
        ))

        children = tree.get_children()
        if children:
            # Tự động chọn dòng đầu tiên
            first_item = children[0]
            tree.selection_set(first_item)
            tree.focus(first_item)
            tree.event_generate("<<TreeviewSelect>>")
        else:
            # Nếu bảng trống, bật chế độ "Thêm"
            clear_input(widgets)

    except Exception as e:
        messagebox.showerror("Lỗi tải dữ liệu", f"Lỗi SQL: {str(e)}")
    finally:
        if conn:
            conn.close()

    # Chỉ khóa form và đặt chế độ VIEW nếu có dữ liệu
    if tree.get_children():
        set_form_state(is_enabled=False, widgets=widgets)
        widgets["current_mode"] = "VIEW"
    # (Nếu không có children, clear_input() đã được gọi và set
mode="ADD")

def them_nhanvien(widgets):
    """(CREATE) Thêm một nhân viên mới (chỉ vào bảng NhanVien)."""
    manv = widgets['entry_manv'].get()
    hoten = widgets['entry_hoten'].get()
    sdt = widgets['entry_sdt'].get()
    diachi = widgets['entry_diachi'].get()

    trangthai_nv_text = widgets['cbb_trangthai_nv_var'].get()
    trangthai_nv_value = 1 if trangthai_nv_text == "Đang làm việc"
else 0

    if not manv or not hoten:
        messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập Mã Nhân
Viên và Họ Tên")
        return False

    conn = utils.connect_db()
    if conn is None: return False

    try:

```

```

        cur = conn.cursor()
        sql_nhanvien = "INSERT INTO NhanVien (MaNhanVien, HoVaTen,
SoDienThoai, DiaChi, TrangThai) VALUES (?, ?, ?, ?, ?)"
        cur.execute(sql_nhanvien, (manv, hoten, sdt, diachi,
trangthai_nv_value))
        conn.commit()
        messagebox.showinfo("Thành công", "Đã thêm nhân viên mới thành
công.")
        return True

    except pyodbc.IntegrityError:
        conn.rollback()
        messagebox.showerror("Lỗi Trùng lặp", f"Mã nhân viên '{manv}'"
đã tồn tại.")
        return False
    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể thêm nhân
viên:\n{str(e)}")
        return False
    finally:
        if conn: conn.close()

def luu_nhanvien_da_sua(widgets):
    """(UPDATE) Lưu thay đổi (UPDATE) sau khi sửa (chỉ bảng
NhanVien)."""
    manv = widgets['entry_manv'].get()
    if not manv:
        messagebox.showwarning("Thiếu dữ liệu", "Không có Mã Nhân Viên
để cập nhật")
        return False

    hoten = widgets['entry_hoten'].get()
    sdt = widgets['entry_sdt'].get()
    diachi = widgets['entry_diachi'].get()

    trangthai_nv_text = widgets['cbb_trangthai_nv_var'].get()
    trangthai_nv_value = 1 if trangthai_nv_text == "Đang làm việc"
else 0

    conn = utils.connect_db()
    if conn is None: return False

    try:
        cur = conn.cursor()
        sql_nhanvien = """
UPDATE NhanVien SET
    HoVaTen = ?, SoDienThoai = ?, DiaChi = ?, TrangThai = ?
WHERE MaNhanVien = ?
"""

```

```

        cur.execute(sql_nhanvien, (hoten, sdt, diachi,
trangthai_nv_value, manv))
        conn.commit()
        messagebox.showinfo("Thành công", "Đã cập nhật thông tin nhân
viên")
        return True

    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể cập
nhật: \n{str(e)}")
        return False
    finally:
        if conn: conn.close()

def save_data(widgets):
    """(NÚT LƯU) Tự động kiểm tra xem nên Thêm mới (INSERT) hay Cập
nhật (UPDATE)."""

    success = False

    if widgets.get("current_mode") == "EDIT":
        success = luu_nhanvien_da_sua(widgets)
    elif widgets.get("current_mode") == "ADD":
        success = them_nhanvien(widgets)
    else:
        messagebox.showwarning("Chưa Sửa/Thêm", "Vui lòng nhấn 'Thêm'
hoặc 'Sửa' trước khi Lưu.")
        return

    if success:
        load_data(widgets)

def xoa_nhanvien_vinhvien(widgets):
    """(DELETE) Xóa vĩnh viễn nhân viên và MỌI DỮ LIỆU LIÊN QUAN."""
    selected = widgets['tree'].selection()
    if not selected:
        messagebox.showwarning("Chưa chọn", "Hãy chọn một nhân viên
trong danh sách để xóa")
        return

    manv = widgets['entry_manv'].get()
    if not manv:
        messagebox.showwarning("Lỗi", "Không tìm thấy mã nhân viên.
Vui lòng chọn lại.")
        return

    msg_xacnhan = (
        f"BẠN CÓ CHẮC CHẮN MUỐN XÓA VĨNH VIỄN NHÂN VIÊN '{manv}'?\n\n"
        "CẢNH BÁO: Thao tác này KHÔNG THỂ hoàn tác.\n"
    )

```

```

        "Tất cả dữ liệu liên quan (chuyển đi, nhiên liệu, tài
        khoản,...) sẽ bị XÓA SẠCH."
    )
    if not messagebox.askyesno("XÁC NHẬN XÓA VĨNH VIỄN", msg_xacnhan):
        return

    conn = utils.connect_db()
    if conn is None: return

    try:
        cur = conn.cursor()

        # Xóa/Cập nhật các bảng con/liên quan trước
        cur.execute("DELETE FROM NhatKyNhienLieu WHERE MaNhanVien=?", (manv,))
        cur.execute("DELETE FROM ChuyenDi WHERE MaNhanVien=?", (manv,))
        cur.execute("DELETE FROM TaiKhoan WHERE MaNhanVien=?", (manv,))
        cur.execute("DELETE FROM TaiXe WHERE MaNhanVien=?", (manv,))
        cur.execute("UPDATE Xe SET MaNhanVienHienTai = NULL WHERE
        MaNhanVienHienTai = ?", (manv,))

        # Xóa bảng cha
        cur.execute("DELETE FROM NhanVien WHERE MaNhanVien=?", (manv,))

        conn.commit()
        messagebox.showinfo("Thành công", f"Đã xóa vĩnh viễn nhân viên
        '{manv}' và tất cả dữ liệu liên quan.")
        load_data(widgets)

    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể xóa nhân
        viên:\n{str(e)}")
    finally:
        if conn: conn.close()

```

### - Hàm tạo trang

```

def create_page(master):
    """
    Hàm này được main.py gọi.
    Nó tạo ra toàn bộ nội dung trang và đặt vào 'master' (là
    main_frame).
    """

    # 1. TẠO FRAME CHÍNH VÀ CÀI STYLE
    page_frame = ttk.Frame(master, style="TFrame")
    utils.setup_theme(page_frame)

```

```

# 2. TẠO TIÊU ĐỀ
lbl_title = ttk.Label(page_frame, text="QUẢN LÝ TOÀN BỘ NHÂN VIÊN",
style="Title.TLabel")
lbl_title.pack(pady=15)

# 3. KHUNG FORM NHẬP LIỆU
frame_info = ttk.Frame(page_frame, style="TFrame")
frame_info.pack(pady=10, padx=20, fill="x")

    ttk.Label(frame_info, text="Mã nhân viên:").grid(row=0, column=0,
padx=5, pady=8, sticky="w")
    entry_manv = ttk.Entry(frame_info, width=30)
    entry_manv.grid(row=0, column=1, padx=5, pady=8, sticky="w")

    ttk.Label(frame_info, text="Họ và tên:").grid(row=1, column=0,
padx=5, pady=8, sticky="w")
    entry_hoten = ttk.Entry(frame_info, width=30)
    entry_hoten.grid(row=1, column=1, padx=5, pady=8, sticky="w")

    ttk.Label(frame_info, text="Số điện thoại:").grid(row=2, column=0,
padx=5, pady=8, sticky="w")
    entry_sdt = ttk.Entry(frame_info, width=30)
    entry_sdt.grid(row=2, column=1, padx=5, pady=8, sticky="w")

    ttk.Label(frame_info, text="Địa chỉ:").grid(row=0, column=2,
padx=15, pady=8, sticky="w")
    entry_diachi = ttk.Entry(frame_info, width=40)
    entry_diachi.grid(row=0, column=3, padx=5, pady=8, sticky="w")

    ttk.Label(frame_info, text="Trạng thái NV:").grid(row=1, column=2,
padx=15, pady=8, sticky="w")
    trangthai_nv_options = ["Nghỉ", "Đang làm việc"]
    cbb_trangthai_nv_var = tk.StringVar()
    cbb_trangthai_nv = ttk.Combobox(frame_info,
textvariable=cbb_trangthai_nv_var, values=trangthai_nv_options,
width=38, state='readonly')
    cbb_trangthai_nv.grid(row=1, column=3, padx=5, pady=8, sticky="w")
    cbb_trangthai_nv.set("Đang làm việc")

    frame_info.columnconfigure(1, weight=1)
    frame_info.columnconfigure(3, weight=1)

# 4. KHUNG CÁC NÚT CHỨC NĂNG
frame_btn = ttk.Frame(page_frame, style="TFrame")
frame_btn.pack(pady=10)

# 5. KHUNG BẢNG DANH SÁCH (TREEVIEW)
lbl_ds = ttk.Label(page_frame, text="Danh sách nhân viên (Tất cả)",
style="Header.TLabel")
lbl_ds.pack(pady=(10, 5), padx=20, anchor="w")

```

```

frame_tree = ttk.Frame(page_frame, style="TFrame")
frame_tree.pack(pady=10, padx=20, fill="both", expand=True)

scrollbar_y = ttk.Scrollbar(frame_tree, orient=tk.VERTICAL,
style="Vertical.TScrollbar")
scrollbar_x = ttk.Scrollbar(frame_tree, orient=tk.HORIZONTAL,
style="Horizontal.TScrollbar")

columns = ("manv", "hoten", "sdt", "diachi", "trangthai_nv")
tree = ttk.Treeview(frame_tree, columns=columns, show="headings",
height=10,
            yscrollcommand=scrollbar_y.set,
            xscrollcommand=scrollbar_x.set)

scrollbar_y.config(command=tree.yview)
scrollbar_x.config(command=tree.xview)
scrollbar_y.pack(side=tk.RIGHT, fill=tk.Y)
scrollbar_x.pack(side=tk.BOTTOM, fill=tk.X)

tree.heading("manv", text="Mã NV")
tree.column("manv", width=80, anchor="center")
tree.heading("hoten", text="Họ Tên")
tree.column("hoten", width=150)
tree.heading("sdt", text="Số điện thoại")
tree.column("sdt", width=120)
tree.heading("diachi", text="Địa chỉ")
tree.column("diachi", width=200)
tree.heading("trangthai_nv", text="Trạng thái NV")
tree.column("trangthai_nv", width=120, anchor="center")

tree.pack(fill="both", expand=True)

# 6. TẠO TỪ ĐIỂN 'widgets' ĐỂ TRUYỀN ĐI
widgets = {
    "tree": tree,
    "entry_manv": entry_manv,
    "entry_hoten": entry_hoten,
    "entry_sdt": entry_sdt,
    "entry_diachi": entry_diachi,
    "cbb_trangthai_nv": cbb_trangthai_nv,
    "cbb_trangthai_nv_var": cbb_trangthai_nv_var,
    "current_mode": "VIEW" # Thêm biến trạng thái
}

# 7. GÁN LỆNH CHO CÁC NÚT
btn_them = ttk.Button(frame_btn, text="Thêm", width=8,
command=lambda: clear_input(widgets))
btn_them.grid(row=0, column=0, padx=10)

```

```

        btn_luu = ttk.Button(frame_btn, text="Lưu", width=8,
command=lambda: save_data(widgets))
        btn_luu.grid(row=0, column=1, padx=10)
        btn_sua = ttk.Button(frame_btn, text="Sửa", width=8,
command=lambda: chon_nhanvien_de_sua(widgets))
        btn_sua.grid(row=0, column=2, padx=10)
        btn_huy = ttk.Button(frame_btn, text="Hủy", width=8,
command=lambda: load_data(widgets))
        btn_huy.grid(row=0, column=3, padx=10)
        btn_xoa = ttk.Button(frame_btn, text="Xóa", width=8,
command=lambda: xoa_nhanvien_vinhvien(widgets))
        btn_xoa.grid(row=0, column=4, padx=10)

# 8. KẾT NỐI BINDING (SỰ KIỆN CLICK)
tree.bind("<<TreeviewSelect>>", lambda event: on_item_select(event,
widgets))

# 9. TÀI DỮ LIỆU LẦN ĐẦU
load_data(widgets)

# 10. TRẢ VỀ FRAME CHÍNH
return page_frame

```

### e. Form LichSuBaoTri

- Hàm tiện ích

```

def load_xe_combobox(user_role, user_username):
    """Tải danh sách xe (BienSoXe) vào Combobox, LỌC theo tài xế."""
    conn = utils.connect_db()
    if conn is None: return []

    try:
        cur = conn.cursor()
        sql = "SELECT BienSoXe FROM Xe"
        params = []

        # Nếu là Tài xế, chỉ tải xe của tài xế đó
        if user_role == "TaiXe":
            manv = get_manv_from_username(user_username)
            if manv:
                sql += " WHERE MaNhanVienHienTai = ?"
                params.append(manv)
            else:
                sql += " WHERE 1=0" # Không tìm thấy tài xế, trả về
rỗng
    
```

```

        sql += " ORDER BY BienSoXe"
        cur.execute(sql, params)
        rows = cur.fetchall()
        return [row[0] for row in rows]
    except Exception as e:
        print(f'Lỗi tải combobox xe: {e}')
        return []
    finally:
        if conn: conn.close()

def get_mav_from_username(username):
    """Lấy MaNhanVien từ TenDangNhap."""
    conn = utils.connect_db()
    if conn is None: return None
    try:
        cur = conn.cursor()
        cur.execute("SELECT MaNhanVien FROM TaiKhoan WHERE TenDangNhap = ?", (username,))
        row = cur.fetchone()
        return row[0] if row else None
    except Exception as e:
        print(f'Lỗi get_mav_from_username: {e}')
        return None
    finally:
        if conn: conn.close()

```

- Hàm quản lý và sự kiện

```

def set_form_state(is_enabled, widgets):
    """Bật (enable) hoặc Tắt (disable) toàn bộ các trường nhập liệu."""
    widgets['entry_mabaotri'].config(state='disabled')

    if is_enabled:
        widgets['cbb_xe'].config(state='readonly')
        widgets['date_ngaybaotri'].config(state='normal')
        widgets['entry_chiphi'].config(state='normal')
        widgets['entry_mota'].config(state='normal',
                                     bg=utils.theme_colors["bg_entry"], fg=utils.theme_colors["text"])
    else:
        widgets['cbb_xe'].config(state='disabled')
        widgets['date_ngaybaotri'].config(state='disabled')
        widgets['entry_chiphi'].config(state='disabled')
        widgets['entry_mota'].config(state='disabled',
                                     bg=utils.theme_colors["disabled_bg"],
                                     fg=utils.theme_colors["text_disabled"])

def clear_input(widgets):

```

```

"""(NÚT THÊM) Xóa trăng và Mở khóa các trường nhập liệu (Chế độ
Thêm mới)."""
    set_form_state(is_enabled=True, widgets=widgets)

    widgets['entry_mabaotri'].config(state='normal')
    widgets['entry_mabaotri'].delete(0, tk.END)
    widgets['entry_mabaotri'].config(state='disabled')

    widgets['cbb_xe'].set("")
    widgets['entry_chiphi'].delete(0, tk.END)
    widgets['entry_mota'].delete("1.0", tk.END)

    widgets['date_ngaybaotri'].set_date(datetime.now().strftime("%Y-
%m-%d"))

    widgets['cbb_xe'].focus()

    tree = widgets['tree']
    if tree.selection():
        tree.selection_remove(tree.selection()[0])

    widgets["current_mode"] = "ADD" # Đặt chế độ

def on_item_select(event, widgets):
    """(SỰ KIỆN CLICK) Khi click vào Treeview, đổ dữ liệu đầy đủ lên
form (ở trạng thái mờ)."""
    tree = widgets['tree']
    selected = tree.selection()
    if not selected: return

    selected_item = tree.item(selected[0])
    mabaotri = selected_item['values'][0]

    conn = utils.connect_db()
    if conn is None: return

    try:
        cur = conn.cursor()
        sql = "SELECT * FROM LichSuBaoTri WHERE MaBaoTri = ?"
        cur.execute(sql, (mabaotri,))
        data = cur.fetchone()

        if not data:
            messagebox.showerror("Lỗi", "Không tìm thấy dữ liệu.")
            return

        # 1. Mở khóa form để điền
        set_form_state(is_enabled=True, widgets=widgets)
        widgets['entry_mabaotri'].config(state='normal')

```

```

# 2. Xóa
widgets['entry_mabaotri'].delete(0, tk.END)
widgets['cbb_xe'].set("")
widgets['entry_chiphi'].delete(0, tk.END)
widgets['entry_mota'].delete("1.0", tk.END)

# 3. Đień
widgets['entry_mabaotri'].insert(0, data.MaBaoTri)
widgets['cbb_xe'].set(data.BienSoXe or "")

if data.NgayBaoTri:
    widgets['date_ngaybaotri'].set_date(str(data.NgayBaoTri))

widgets['entry_chiphi'].insert(0, str(data.ChiPhi or ""))
widgets['entry_mota'].insert("1.0", data.MoTa or "")

except Exception as e:
    messagebox.showerror("Lỗi không xác định", f"Lỗi: {str(e)}")
finally:
    if conn: conn.close()

# 4. Khóa form lại (chế độ View)
widgets['entry_mabaotri'].config(state='disabled')
set_form_state(is_enabled=False, widgets=widgets)
widgets["current_mode"] = "VIEW" # Đặt chế độ

def chon_baotri_de_sua(widgets):
    """(NÚT SỬA) Kích hoạt chế độ sửa, Mở khóa form (trừ MaBaoTri)."""
    selected = widgets['tree'].selection()
    if not selected:
        messagebox.showwarning("Chưa chọn", "Hãy chọn một mục trong danh sách trước khi nhấn 'Sửa'")
        return

    if not widgets['entry_mabaotri'].get():
        messagebox.showwarning("Lỗi", "Không tìm thấy mã bảo trì. Vui lòng chọn lại.")
        return

    set_form_state(is_enabled=True, widgets=widgets)
    widgets['entry_mabaotri'].config(state='disabled') # Không cho sửa khóa chính
    widgets['cbb_xe'].focus()
    widgets["current_mode"] = "EDIT" # Đặt chế độ

```

- Hàm logic CRUD

```
def load_data(widgets, user_role, user_username):
```

```

"""(READ) Tải dữ liệu Bảo trì (LỌC THEO VAI TRÒ) VÀ LÀM MỜ
FORM."""
tree = widgets['tree']
for i in tree.get_children():
    tree.delete(i)

conn = utils.connect_db()
if conn is None:
    set_form_state(is_enabled=False, widgets=widgets)
    return

try:
    cur = conn.cursor()

    sql = """
        SELECT
            bt.MaBaoTri, bt.BienSoXe, bt.NgayBaoTri, bt.ChiPhi,
            bt.MoTa,
            nv.HoVaTen AS NguoiNhap
        FROM LichSuBaoTri AS bt
        LEFT JOIN NhanVien AS nv ON bt.MaNhanVienNhap =
            nv.MaNhanVien
    """
    params = []

    if user_role == "TaiXe":
        manv = get_mav_from_username(user_username)
        if manv:
            sql += """
                WHERE bt.BienSoXe IN (
                    SELECT BienSoXe FROM Xe WHERE
                    MaNhanVienHienTai = ?
                )
            """
            params.append(mav)
        else:
            sql += " WHERE 1=0"

    sql += " ORDER BY bt.NgayBaoTri DESC"

    cur.execute(sql, params)
    rows = cur.fetchall()

    for row in rows:
        ma_bt = row[0]
        bienso = row[1]
        ngay_bt = str(row[2]) if row[2] else "N/A"
        chiph = row[3]
        mota = (row[4] or "")[:50] + "..."
        nguoi_nhph = row[5] or "N/A"

```

```

        tree.insert("", tk.END, values=(ma_bt, bienso, ngay_bt,
chiphii, mota, nguoi_nhap))

    children = tree.get_children()
    if children:
        # Tự động chọn dòng đầu tiên
        first_item = children[0]
        tree.selection_set(first_item)
        tree.focus(first_item)
        tree.event_generate("<<TreeviewSelect>>")
    else:
        clear_input(widgets) # Đặt chế độ Thêm nếu bảng trống

except Exception as e:
    messagebox.showerror("Lỗi tải dữ liệu", f"Lỗi SQL: {str(e)}")
finally:
    if conn:
        conn.close()

    if tree.get_children():
        set_form_state(is_enabled=False, widgets=widgets)
        widgets["current_mode"] = "VIEW"
    # (Nếu bảng trống, clear_input() đã set mode="ADD")

def them_baotri(widgets):
    """(CREATE) Thêm một lịch sử bảo trì mới."""
    try:
        bienso = widgets['cbb_xe_var'].get()
        ngay_bt = widgets['date_ngaybaotri'].get_date().strftime('%Y-%m-%d')
        mota = widgets['entry_mota'].get("1.0", tk.END).strip()
        chiphii = widgets['entry_chiphii'].get()

        if not bienso or not ngay_bt:
            messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập Xe và Ngày bảo trì")
            return False

        chiphii_dec = float(chiphii) if chiphii else 0.0

        user_username = widgets.get("user_username")
        manv_nhap = get_manv_from_username(user_username)
        if not manv_nhap:
            messagebox.showerror("Lỗi", "Không thể xác định người dùng. Vui lòng đăng nhập lại.")
            return False

    except Exception as e:

```

```

        messagebox.showerror("Lỗi định dạng", f"Chi phí không hợp lệ: {e}")
    return False

    conn = utils.connect_db()
    if conn is None: return False

    try:
        cur = conn.cursor()
        sql = "INSERT INTO LichSuBaoTri (BienSoXe, NgayBaoTri, MoTa, ChiPhi, MaNhanVienNhap) VALUES (?, ?, ?, ?, ?, ?)"
        cur.execute(sql, (bienso, ngay_bt, mota, chiphid, manv_nhap))
        conn.commit()
        messagebox.showinfo("Thành công", "Đã thêm lịch sử bảo trì thành công")
    return True

    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể thêm:\n{str(e)}")
    return False
finally:
    if conn: conn.close()

def luu_baotri_da_sua(widgets):
    """(UPDATE) Lưu thay đổi (UPDATE) sau khi sửa."""
    mabaotri = widgets['entry_mabaotri'].get()
    if not mabaotri:
        messagebox.showwarning("Thiếu dữ liệu", "Không có Mã bảo trì để cập nhật")
    return False

    try:
        bienso = widgets['cbb_xe_var'].get()
        ngay_bt = widgets['date_ngaybaotri'].get_date().strftime('%Y-%m-%d')
        mota = widgets['entry_mota'].get("1.0", tk.END).strip()
        chiphid = widgets['entry_chiphid'].get()

        if not bienso or not ngay_bt:
            messagebox.showwarning("Thiếu dữ liệu", "Xe và Ngày bảo trì không được rỗng")
        return False

        chiphid = float(chiphid) if chiphid else 0.0

    except Exception as e:
        messagebox.showerror("Lỗi định dạng", f"Chi phí không hợp lệ: {e}")

```

```

        return False

    conn = utils.connect_db()
    if conn is None: return False

    try:
        cur = conn.cursor()
        sql = """
            UPDATE LichSuBaoTri SET
                BienSoXe = ?, NgayBaoTri = ?, MoTa = ?, ChiPhi = ?
            WHERE MaBaoTri = ?
        """
        cur.execute(sql, (bienso, ngay_bt, mota, chiphi_dec, mabaotri))
        conn.commit()
        messagebox.showinfo("Thành công", "Đã cập nhật lịch sử bảo
trì")
        return True

    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể cập
nhật:\n{str(e)}")
        return False
    finally:
        if conn: conn.close()

def save_data(widgets):
    """(NÚT LƯU) Tự động kiểm tra xem nên Thêm mới (INSERT) hay Cập
nhật (UPDATE)."""

    user_role = widgets.get("user_role", "Admin")
    user_username = widgets.get("user_username", "")

    success = False
    if widgets.get("current_mode") == "EDIT":
        success = luu_baotri_da_sua(widgets)
    elif widgets.get("current_mode") == "ADD":
        success = them_baotri(widgets)
    else:
        # Dự phòng nếu 'current_mode' không được set
        if widgets['entry_mabaotri'].get():
            success = luu_baotri_da_sua(widgets)
        else:
            success = them_baotri(widgets)

    if success:
        load_data(widgets, user_role, user_username)

def xoa_baotri(widgets):
    """(DELETE) Xóa lịch sử bảo trì được chọn."""

```

```

selected = widgets['tree'].selection()
if not selected:
    messagebox.showwarning("Chưa chọn", "Hãy chọn một mục để xóa")
    return

mabaotri = widgets['entry_mabaotri'].get()

if not mabaotri:
    messagebox.showwarning("Lỗi", "Không tìm thấy mã bảo trì. Vui lòng chọn lại.")
    return

if not messagebox.askyesno("Xác nhận", f"Bạn có chắc chắn muốn xóa Lịch sử Mã: {mabaotri}?"):
    return

conn = utils.connect_db()
if conn is None: return

try:
    cur = conn.cursor()
    cur.execute("DELETE FROM LichSuBaoTri WHERE MaBaoTri=?", (mabaotri,))
    conn.commit()
    messagebox.showinfo("Thành công", "Đã xóa lịch sử bảo trì thành công")

    user_role = widgets.get("user_role", "Admin")
    user_username = widgets.get("user_username", "")
    load_data(widgets, user_role, user_username)

except Exception as e:
    conn.rollback()
    messagebox.showerror("Lỗi SQL", f"Không thể xóa:\n{str(e)}")
finally:
    if conn: conn.close()

```

- Hàm tạo trang

```

def create_page(master, user_role, user_username):
    """
    Hàm này được main.py gọi.
    Nó tạo ra toàn bộ nội dung trang và đặt vào 'master' (là main_frame).
    """

    # 1. TẠO FRAME CHÍNH VÀ CÀI STYLE
    page_frame = ttk.Frame(master, style="TFrame")
    utils.setup_theme(page_frame)

```

```

# 2. TẠO TIÊU ĐỀ
lbl_title_text = "QUẢN LÝ LỊCH SỬ BẢO TRÌ"
if user_role == "TaiXe":
    lbl_title_text = "LỊCH SỬ BẢO TRÌ XE CỦA BẠN"

lbl_title = ttk.Label(page_frame, text=lbl_title_text,
style="Title.TLabel")
lbl_title.pack(pady=15)

# 3. KHUNG FORM NHẬP LIỆU
frame_info = ttk.Frame(page_frame, style="TFrame")
frame_info.pack(pady=5, padx=20, fill="x")

# --- Hàng 1 ---
ttk.Label(frame_info, text="Mã bảo trì:").grid(row=0, column=0,
padx=5, pady=8, sticky="w")
entry_mabaotri = ttk.Entry(frame_info, width=30, state='disabled')
entry_mabaotri.grid(row=0, column=1, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Ngày bảo trì:").grid(row=0, column=2,
padx=15, pady=8, sticky="w")
date_entry_style_options = {
    'width': 38, 'date_pattern': 'yyyy-MM-dd',
    'background': utils.theme_colors["bg_entry"],
    'foreground': utils.theme_colors["text"],
    'disabledbackground': utils.theme_colors["disabled_bg"],
    'disabledforeground': utils.theme_colors["text_disabled"],
    'bordercolor': utils.theme_colors["bg_entry"],
    'headersbackground': utils.theme_colors["accent"],
    'headersforeground': utils.theme_colors["accent_text"],
    'selectbackground': utils.theme_colors["accent"],
    'selectforeground': utils.theme_colors["accent_text"]
}
date_ngaybaotri = DateEntry(frame_info, **date_entry_style_options)
date_ngaybaotri.grid(row=0, column=3, padx=5, pady=8, sticky="w")

# --- Hàng 2 ---
ttk.Label(frame_info, text="Xe:").grid(row=1, column=0, padx=5,
pady=8, sticky="w")
cbb_xe_var = tk.StringVar()
cbb_xe = ttk.Combobox(frame_info, textvariable=cbb_xe_var,
width=28, state='readonly')
cbb_xe.grid(row=1, column=1, padx=5, pady=8, sticky="w")

# Lọc combobox theo tài xe
cbb_xe['values'] = load_xe_combobox(user_role, user_username)

ttk.Label(frame_info, text="Chi phí:").grid(row=1, column=2,
padx=15, pady=8, sticky="w")

```

```

entry_chiphi = ttk.Entry(frame_info, width=40)
entry_chiphi.grid(row=1, column=3, padx=5, pady=8, sticky="w")

# --- Hàng 3 (Mô tả) ---
ttk.Label(frame_info, text="Mô tả công việc:").grid(row=2,
column=0, padx=5, pady=8, sticky="nw")
entry_mota = tk.Text(frame_info, width=60, height=4,
font=("Segoe UI", 10),
bg=utils.theme_colors["bg_entry"],
fg=utils.theme_colors["text"],
relief="solid",
borderwidth=1,
insertbackground=utils.theme_colors["text"],
highlightthickness=1,
highlightbackground="#ACACAC",
highlightcolor=utils.theme_colors["accent"])
)
entry_mota.grid(row=2, column=1, columnspan=3, padx=5, pady=8,
sticky="w")

frame_info.columnconfigure(1, weight=1)
frame_info.columnconfigure(3, weight=1)

# 4. KHUNG CÁC NÚT CHỨC NĂNG
frame_btn = ttk.Frame(page_frame, style="TFrame")
frame_btn.pack(pady=10)

# 5. KHUNG BẢNG DANH SÁCH (TREEVIEW)
lbl_ds = ttk.Label(page_frame, text="Danh sách bảo trì (Sắp xếp
mới nhất)", style="Header TLabel")
lbl_ds.pack(pady=(10, 5), padx=20, anchor="w")

frame_tree = ttk.Frame(page_frame, style="TFrame")
frame_tree.pack(pady=10, padx=20, fill="both", expand=True)

scrollbar_y = ttk.Scrollbar(frame_tree, orient=tk.VERTICAL,
style="Vertical.TScrollbar")
scrollbar_x = ttk.Scrollbar(frame_tree, orient=tk.HORIZONTAL,
style="Horizontal.TScrollbar")

columns = ("ma_bt", "bienso", "ngay_bt", "chiphi", "mota",
"nguoinhanh")
tree = ttk.Treeview(frame_tree, columns=columns, show="headings",
height=10,
yscrollcommand=scrollbar_y.set,
xscrollcommand=scrollbar_x.set)

scrollbar_y.config(command=tree.yview)
scrollbar_x.config(command=tree.xview)
scrollbar_y.pack(side=tk.RIGHT, fill=tk.Y)

```

```

scrollbar_x.pack(side=tk.BOTTOM, fill=tk.X)

tree.heading("ma_bt", text="Mã BT")
tree.column("ma_bt", width=60, anchor="center")
tree.heading("bienso", text="Biển số xe")
tree.column("bienso", width=100)
tree.heading("ngay_bt", text="Ngày bảo trì")
tree.column("ngay_bt", width=100)
tree.heading("chiph", text="Chi phí")
tree.column("chiph", width=100, anchor="e")
tree.heading("mota", text="Mô tả")
tree.column("mota", width=250)
tree.heading("nguoinhap", text="Người nhập")
tree.column("nguoinhap", width=120)

tree.pack(fill="both", expand=True)

# 6. TẠO TỪ ĐIỂN 'widgets' ĐỂ TRUYỀN ĐI
widgets = {
    "tree": tree,
    "entry_mabaotri": entry_mabaotri,
    "date_ngaybaotri": date_ngaybaotri,
    "cbb_xe": cbb_xe,
    "entry_chiph": entry_chiph,
    "entry_mota": entry_mota,
    "cbb_xe_var": cbb_xe_var,
    "user_role": user_role,
    "user_username": user_username,
    "current_mode": "VIEW" # Thêm biến trạng thái
}

# 7. GÁN LỆNH CHO CÁC NÚT
btn_them = ttk.Button(frame_btn, text="Thêm", width=8,
command=lambda: clear_input(widgets))
btn_them.grid(row=0, column=0, padx=10)
btn_luu = ttk.Button(frame_btn, text="Lưu", width=8,
command=lambda: save_data(widgets))
btn_luu.grid(row=0, column=1, padx=10)
btn_sua = ttk.Button(frame_btn, text="Sửa", width=8,
command=lambda: chon_baotri_de_sua(widgets))
btn_sua.grid(row=0, column=2, padx=10)

btn_huy = ttk.Button(frame_btn, text="Hủy", width=8,
                     command=lambda: load_data(widgets, user_role,
user_username))
btn_huy.grid(row=0, column=3, padx=10)

btn_xoa = ttk.Button(frame_btn, text="Xóa", width=8,
command=lambda: xoa_baotri(widgets))
btn_xoa.grid(row=0, column=4, padx=10)

```

```

# Phân quyền nút cho Tài xế
if user_role == "TaiXe":
    btn_sua.config(state="disabled")
    btn_xoa.config(state="disabled")

# 8. KẾT NỐI BINDING
tree.bind("<>TreeviewSelect>>", lambda event: on_item_select(event, widgets))

# 9. TẢI DỮ LIỆU LẦN ĐẦU
load_data(widgets, user_role, user_username)

# 10. TRẢ VỀ FRAME CHÍNH
return page_frame

```

## f. Form QuanLyNhanVien

- Hàm quản lý và sự kiện

```

def set_form_state(is_enabled, widgets):
    """Bật (enable) hoặc Tắt (disable) toàn bộ các trường nhập liệu."""
    if is_enabled:
        widgets['entry_hoten'].config(state='normal')
        widgets['entry_sdt'].config(state='normal')
        widgets['entry_diachi'].config(state='normal')
        widgets['cbb_trangthai_nv'].config(state='readonly')
    else:
        widgets['entry_manv'].config(state='disabled')
        widgets['entry_hoten'].config(state='disabled')
        widgets['entry_sdt'].config(state='disabled')
        widgets['entry_diachi'].config(state='disabled')
        widgets['cbb_trangthai_nv'].config(state='disabled')

def clear_input(widgets):
    """(NÚT THÊM) Xóa trắng và Mở khóa các trường nhập liệu (Chế độ Thêm mới)."""
    set_form_state(is_enabled=True, widgets=widgets)
    widgets['entry_manv'].config(state='normal')

    widgets['entry_manv'].delete(0, tk.END)
    widgets['entry_hoten'].delete(0, tk.END)
    widgets['entry_sdt'].delete(0, tk.END)
    widgets['entry_diachi'].delete(0, tk.END)
    widgets['cbb_trangthai_nv'].set("Đang làm việc")

```

```

        widgets['entry_manv'].focus()

        tree = widgets['tree']
        if tree.selection():
            tree.selection_remove(tree.selection()[0])

        widgets["current_mode"] = "ADD"

def on_item_select(event, widgets):
    """(SỰ KIỆN CLICK) Khi click vào Treeview, chỉ đổ dữ liệu lên form
    (ở trạng thái mờ)."""
    tree = widgets['tree']
    selected = tree.selection()
    if not selected: return

    selected_item = tree.item(selected[0])
    manv = selected_item['values'][0]

    conn = utils.connect_db()
    if conn is None: return

    try:
        cur = conn.cursor()
        sql = "SELECT * FROM NhanVien WHERE MaNhanVien = ?"
        cur.execute(sql, (manv,))
        data = cur.fetchone()

        if not data:
            messagebox.showerror("Lỗi", "Không tìm thấy dữ liệu nhân
viên.")
            return

        # 1. Mở khóa form để điền
        set_form_state(is_enabled=True, widgets=widgets)
        widgets['entry_manv'].config(state='normal')

        # 2. Xóa
        widgets['entry_manv'].delete(0, tk.END)
        widgets['entry_hoten'].delete(0, tk.END)
        widgets['entry_sdt'].delete(0, tk.END)
        widgets['entry_diachi'].delete(0, tk.END)

        # 3. Điền
        widgets['entry_manv'].insert(0, data.MaNhanVien)
        widgets['entry_hoten'].insert(0, data.HoVaTen or "")
        widgets['entry_sdt'].insert(0, data.SoDienThoai or "")
        widgets['entry_diachi'].insert(0, data.DiaChi or "")
        widgets['cbb_trangthai_nv'].set("Đang làm việc" if
data.TrangThai == 1 else "Nghỉ")
    
```

```

        except Exception as e:
            messagebox.showerror("Lỗi không xác định", f"Lỗi: {str(e)}")
    finally:
        if conn: conn.close()
        # 4. Khóa form lại (chế độ View)
        set_form_state(is_enabled=False, widgets=widgets)
        widgets["current_mode"] = "VIEW"

def chon_nhanvien_de_sua(widgets):
    """(NÚT SỬA) Kích hoạt chế độ sửa, Mở khóa form (trừ MaNV)."""
    selected = widgets['tree'].selection()
    if not selected:
        messagebox.showwarning("Chưa chọn", "Hãy chọn một nhân viên
trong danh sách trước khi nhấn 'Sửa'")
        return

    if not widgets['entry_manv'].get():
        messagebox.showwarning("Lỗi", "Không tìm thấy mã nhân viên.
Vui lòng chọn lại.")
        return

    set_form_state(is_enabled=True, widgets=widgets)
    widgets['entry_manv'].config(state='disabled') # Không cho sửa
    khóa chính
    widgets['entry_hoten'].focus()
    widgets["current_mode"] = "EDIT"

```

### - Hàm logic CRUD

```

def load_data(widgets):
    """(READ) Tải TOÀN BỘ dữ liệu từ NhanVien VÀ LÀM MỜ FORM."""
    tree = widgets['tree']
    for i in tree.get_children():
        tree.delete(i)

    conn = utils.connect_db()
    if conn is None:
        set_form_state(is_enabled=False, widgets=widgets)
        return

    try:
        cur = conn.cursor()
        sql = "SELECT MaNhanVien, HoVaTen, SoDienThoai, DiaChi,
TrangThai FROM NhanVien"
        cur.execute(sql)
        rows = cur.fetchall()

        for row in rows:

```

```

        trang_thai_nv_text = "Đang làm việc" if row[4] == 1 else
    "Nghỉ"
        tree.insert("", tk.END, values=(
            row[0], row[1], row[2], row[3], trang_thai_nv_text
        ))

        children = tree.get_children()
        if children:
            # Tự động chọn dòng đầu tiên
            first_item = children[0]
            tree.selection_set(first_item)
            tree.focus(first_item)
            tree.event_generate("<<TreeviewSelect>>")
        else:
            # Nếu bảng trống, bật chế độ "Thêm"
            clear_input(widgets)

    except Exception as e:
        messagebox.showerror("Lỗi tải dữ liệu", f"Lỗi SQL: {str(e)}")
    finally:
        if conn:
            conn.close()

    # Chỉ khóa form và đặt chế độ VIEW nếu có dữ liệu
    if tree.get_children():
        set_form_state(is_enabled=False, widgets=widgets)
        widgets["current_mode"] = "VIEW"
    # (Nếu không có children, clear_input() đã được gọi và set
mode="ADD")

def them_nhanvien(widgets):
    """(CREATE) Thêm một nhân viên mới (chỉ vào bảng NhanVien)."""
    manv = widgets['entry_manv'].get()
    hoten = widgets['entry_hoten'].get()
    sdt = widgets['entry_sdt'].get()
    diachi = widgets['entry_diachi'].get()

    trangthai_nv_text = widgets['cbb_trangthai_nv_var'].get()
    trangthai_nv_value = 1 if trangthai_nv_text == "Đang làm việc"
else 0

    if not manv or not hoten:
        messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập Mã Nhân
Viên và Họ Tên")
        return False

    conn = utils.connect_db()
    if conn is None: return False

    try:

```

```

        cur = conn.cursor()
        sql_nhanvien = "INSERT INTO NhanVien (MaNhanVien, HoVaTen,
SoDienThoai, DiaChi, TrangThai) VALUES (?, ?, ?, ?, ?)"
        cur.execute(sql_nhanvien, (manv, hoten, sdt, diachi,
trangthai_nv_value))
        conn.commit()
        messagebox.showinfo("Thành công", "Đã thêm nhân viên mới thành
công.")
        return True

    except pyodbc.IntegrityError:
        conn.rollback()
        messagebox.showerror("Lỗi Trùng lặp", f"Mã nhân viên '{manv}'"
đã tồn tại.")
        return False
    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể thêm nhân
viên:\n{str(e)}")
        return False
    finally:
        if conn: conn.close()

def luu_nhanvien_da_sua(widgets):
    """(UPDATE) Lưu thay đổi (UPDATE) sau khi sửa (chỉ bảng
NhanVien)."""
    manv = widgets['entry_manv'].get()
    if not manv:
        messagebox.showwarning("Thiếu dữ liệu", "Không có Mã Nhân Viên
để cập nhật")
        return False

    hoten = widgets['entry_hoten'].get()
    sdt = widgets['entry_sdt'].get()
    diachi = widgets['entry_diachi'].get()

    trangthai_nv_text = widgets['cbb_trangthai_nv_var'].get()
    trangthai_nv_value = 1 if trangthai_nv_text == "Đang làm việc"
else 0

    conn = utils.connect_db()
    if conn is None: return False

    try:
        cur = conn.cursor()
        sql_nhanvien = """
UPDATE NhanVien SET
    HoVaTen = ?, SoDienThoai = ?, DiaChi = ?, TrangThai = ?
WHERE MaNhanVien = ?
"""

```

```

        cur.execute(sql_nhanvien, (hoten, sdt, diachi,
trangthai_nv_value, manv))
        conn.commit()
        messagebox.showinfo("Thành công", "Đã cập nhật thông tin nhân
viên")
        return True

    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể cập
nhật: \n{str(e)}")
        return False
    finally:
        if conn: conn.close()

def save_data(widgets):
    """(NÚT LƯU) Tự động kiểm tra xem nên Thêm mới (INSERT) hay Cập
nhật (UPDATE)."""

    success = False

    if widgets.get("current_mode") == "EDIT":
        success = luu_nhanvien_da_sua(widgets)
    elif widgets.get("current_mode") == "ADD":
        success = them_nhanvien(widgets)
    else:
        messagebox.showwarning("Chưa Sửa/Thêm", "Vui lòng nhấn 'Thêm'
hoặc 'Sửa' trước khi Lưu.")
        return

    if success:
        load_data(widgets)

def xoa_nhanvien_vinhvien(widgets):
    """(DELETE) Xóa vĩnh viễn nhân viên và MỌI DỮ LIỆU LIÊN QUAN."""
    selected = widgets['tree'].selection()
    if not selected:
        messagebox.showwarning("Chưa chọn", "Hãy chọn một nhân viên
trong danh sách để xóa")
        return

    manv = widgets['entry_manv'].get()
    if not manv:
        messagebox.showwarning("Lỗi", "Không tìm thấy mã nhân viên.
Vui lòng chọn lại.")
        return

    msg_xacnhan = (
        f"BẠN CÓ CHẮC CHẮN MUỐN XÓA VĨNH VIỄN NHÂN VIÊN '{manv}'?\n\n"
        "CẢNH BÁO: Thao tác này KHÔNG THỂ hoàn tác.\n"
    )

```

```

        "Tất cả dữ liệu liên quan (chuyển đi, nhiên liệu, tài
        khoản,...) sẽ bị XÓA SẠCH."
    )
    if not messagebox.askyesno("XÁC NHẬN XÓA VĨNH VIỄN", msg_xacnhan):
        return

    conn = utils.connect_db()
    if conn is None: return

    try:
        cur = conn.cursor()

        # Xóa/Cập nhật các bảng con/liên quan trước
        cur.execute("DELETE FROM NhatKyNhienLieu WHERE MaNhanVien=?", (manv,))
        cur.execute("DELETE FROM ChuyenDi WHERE MaNhanVien=?", (manv,))
        cur.execute("DELETE FROM TaiKhoan WHERE MaNhanVien=?", (manv,))
        cur.execute("DELETE FROM TaiXe WHERE MaNhanVien=?", (manv,))
        cur.execute("UPDATE Xe SET MaNhanVienHienTai = NULL WHERE
        MaNhanVienHienTai = ?", (manv,))

        # Xóa bảng cha
        cur.execute("DELETE FROM NhanVien WHERE MaNhanVien=?", (manv,))

        conn.commit()
        messagebox.showinfo("Thành công", f"Đã xóa vĩnh viễn nhân viên
        '{manv}' và tất cả dữ liệu liên quan.")
        load_data(widgets)

    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể xóa nhân
        viên:\n{str(e)}")
    finally:
        if conn: conn.close()

```

### - Hàm tạo trang

```

def create_page(master):
    """
    Hàm này được main.py gọi.
    Nó tạo ra toàn bộ nội dung trang và đặt vào 'master' (là
    main_frame).
    """

    # 1. TẠO FRAME CHÍNH VÀ CÀI STYLE
    page_frame = ttk.Frame(master, style="TFrame")
    utils.setup_theme(page_frame)

```

```

# 2. TẠO TIÊU ĐỀ
lbl_title = ttk.Label(page_frame, text="QUẢN LÝ TOÀN BỘ NHÂN VIÊN",
style="Title.TLabel")
lbl_title.pack(pady=15)

# 3. KHUNG FORM NHẬP LIỆU
frame_info = ttk.Frame(page_frame, style="TFrame")
frame_info.pack(pady=10, padx=20, fill="x")

    ttk.Label(frame_info, text="Mã nhân viên:").grid(row=0, column=0,
padx=5, pady=8, sticky="w")
    entry_manv = ttk.Entry(frame_info, width=30)
    entry_manv.grid(row=0, column=1, padx=5, pady=8, sticky="w")

    ttk.Label(frame_info, text="Họ và tên:").grid(row=1, column=0,
padx=5, pady=8, sticky="w")
    entry_hoten = ttk.Entry(frame_info, width=30)
    entry_hoten.grid(row=1, column=1, padx=5, pady=8, sticky="w")

    ttk.Label(frame_info, text="Số điện thoại:").grid(row=2, column=0,
padx=5, pady=8, sticky="w")
    entry_sdt = ttk.Entry(frame_info, width=30)
    entry_sdt.grid(row=2, column=1, padx=5, pady=8, sticky="w")

    ttk.Label(frame_info, text="Địa chỉ:").grid(row=0, column=2,
padx=15, pady=8, sticky="w")
    entry_diachi = ttk.Entry(frame_info, width=40)
    entry_diachi.grid(row=0, column=3, padx=5, pady=8, sticky="w")

    ttk.Label(frame_info, text="Trạng thái NV:").grid(row=1, column=2,
padx=15, pady=8, sticky="w")
    trangthai_nv_options = ["Nghỉ", "Đang làm việc"]
    cbb_trangthai_nv_var = tk.StringVar()
    cbb_trangthai_nv = ttk.Combobox(frame_info,
textvariable=cbb_trangthai_nv_var, values=trangthai_nv_options,
width=38, state='readonly')
    cbb_trangthai_nv.grid(row=1, column=3, padx=5, pady=8, sticky="w")
    cbb_trangthai_nv.set("Đang làm việc")

    frame_info.columnconfigure(1, weight=1)
    frame_info.columnconfigure(3, weight=1)

# 4. KHUNG CÁC NÚT CHỨC NĂNG
frame_btn = ttk.Frame(page_frame, style="TFrame")
frame_btn.pack(pady=10)

# 5. KHUNG BẢNG DANH SÁCH (TREEVIEW)
lbl_ds = ttk.Label(page_frame, text="Danh sách nhân viên (Tất cả)",
style="Header.TLabel")
lbl_ds.pack(pady=(10, 5), padx=20, anchor="w")

```

```

frame_tree = ttk.Frame(page_frame, style="TFrame")
frame_tree.pack(pady=10, padx=20, fill="both", expand=True)

scrollbar_y = ttk.Scrollbar(frame_tree, orient=tk.VERTICAL,
style="Vertical.TScrollbar")
scrollbar_x = ttk.Scrollbar(frame_tree, orient=tk.HORIZONTAL,
style="Horizontal.TScrollbar")

columns = ("manv", "hoten", "sdt", "diachi", "trangthai_nv")
tree = ttk.Treeview(frame_tree, columns=columns, show="headings",
height=10,
            yscrollcommand=scrollbar_y.set,
            xscrollcommand=scrollbar_x.set)

scrollbar_y.config(command=tree.yview)
scrollbar_x.config(command=tree.xview)
scrollbar_y.pack(side=tk.RIGHT, fill=tk.Y)
scrollbar_x.pack(side=tk.BOTTOM, fill=tk.X)

tree.heading("manv", text="Mã NV")
tree.column("manv", width=80, anchor="center")
tree.heading("hoten", text="Họ Tên")
tree.column("hoten", width=150)
tree.heading("sdt", text="Số điện thoại")
tree.column("sdt", width=120)
tree.heading("diachi", text="Địa chỉ")
tree.column("diachi", width=200)
tree.heading("trangthai_nv", text="Trạng thái NV")
tree.column("trangthai_nv", width=120, anchor="center")

tree.pack(fill="both", expand=True)

# 6. TẠO TỪ ĐIỂN 'widgets' ĐỂ TRUYỀN ĐI
widgets = {
    "tree": tree,
    "entry_manv": entry_manv,
    "entry_hoten": entry_hoten,
    "entry_sdt": entry_sdt,
    "entry_diachi": entry_diachi,
    "cbb_trangthai_nv": cbb_trangthai_nv,
    "cbb_trangthai_nv_var": cbb_trangthai_nv_var,
    "current_mode": "VIEW" # Thêm biến trạng thái
}

# 7. GÁN LỆNH CHO CÁC NÚT
btn_them = ttk.Button(frame_btn, text="Thêm", width=8,
command=lambda: clear_input(widgets))
btn_them.grid(row=0, column=0, padx=10)

```

```

        btn_luu = ttk.Button(frame_btn, text="Lưu", width=8,
command=lambda: save_data(widgets))
        btn_luu.grid(row=0, column=1, padx=10)
        btn_sua = ttk.Button(frame_btn, text="Sửa", width=8,
command=lambda: chon_nhanvien_de_sua(widgets))
        btn_sua.grid(row=0, column=2, padx=10)
        btn_huy = ttk.Button(frame_btn, text="Hủy", width=8,
command=lambda: load_data(widgets))
        btn_huy.grid(row=0, column=3, padx=10)
        btn_xoa = ttk.Button(frame_btn, text="Xóa", width=8,
command=lambda: xoa_nhanvien_vinhvien(widgets))
        btn_xoa.grid(row=0, column=4, padx=10)

# 8. KẾT NỐI BINDING (SỰ KIỆN CLICK)
tree.bind("<<TreeviewSelect>>", lambda event: on_item_select(event,
widgets))

# 9. TẢI DỮ LIỆU LẦN ĐẦU
load_data(widgets)

# 10. TRẢ VỀ FRAME CHÍNH
return page_frame

```

## g. Form NhatKyNguyenLieu

- Hàm tiện ích

```

def get_manv_from_username(username):
    """Lấy MaNhanVien từ TenDangNhap để lọc dữ liệu."""
    conn = utils.connect_db()
    if conn is None: return (None, None)
    try:
        cur = conn.cursor()
        cur.execute("SELECT NV.MaNhanVien, NV.HoVaTen FROM TaiKhoan TK
JOIN NhanVien NV ON TK.MaNhanVien = NV.MaNhanVien WHERE TK.TenDangNhap
= ?", (username,))
        row = cur.fetchone()
        return (row[0], f"{row[0]} - {row[1]})" if row else (None,
None) # Trả về (MaNhanVien, TenHienThi)
    except Exception as e:
        print(f'Lỗi get_manv_from_username: {e}')
        return (None, None)
    finally:
        if conn: conn.close()

def get_manhatky_from_tree(widgets):
    """Lấy MaNhatKy từ hàng đang được chọn trong Treeview."""
    tree = widgets['tree']
    selected = tree.selection()

```

```

if not selected: return None
# Giá trị MANK là cột đầu tiên (index 0) trong values
return tree.item(selected[0])['values'][0]

```

### - Hàm quản lý và sự kiện

```

def set_form_state(is_enabled, widgets, user_role):
    """Bật (enable) hoặc Tắt (disable) toàn bộ các trường nhập liệu."""
    if is_enabled:
        # === TÀI XẾ CẦN BẬT CÁC Ô NÀY ĐỂ THÊM MỚI/SỬA ===
        widgets['cbb_xe'].config(state='readonly')
        widgets['date_ngaydo'].config(state='normal')
        widgets['entry_giodo'].config(state='normal')
        widgets['entry_solit'].config(state='normal')
        widgets['entry_tongchiph'].config(state='normal')
        widgets['entry_soodo'].config(state='normal')

        if user_role == "Admin":
            widgets['cbb_taixe'].config(state='readonly')
            widgets['cbb_trangthai'].config(state='readonly')
        else: # Nếu là TaiXe
            widgets['cbb_taixe'].config(state='disabled')
            widgets['cbb_trangthai'].config(state='disabled')
    else:
        # Tắt tất cả
        widgets['cbb_xe'].config(state='disabled')
        widgets['cbb_taixe'].config(state='disabled')
        widgets['date_ngaydo'].config(state='disabled')
        widgets['entry_giodo'].config(state='disabled')
        widgets['entry_solit'].config(state='disabled')
        widgets['entry_tongchiph'].config(state='disabled')
        widgets['entry_soodo'].config(state='disabled')
        widgets['cbb_trangthai'].config(state='disabled')

def clear_input(widgets, user_role, user_username):
    """(NÚT THÊM) Xóa trắng và Mở khóa các trường nhập liệu (Chế độ Thêm mới)."""
    if user_role == "Admin":
        messagebox.showwarning("Thông báo", "Admin chỉ có thể Duyệt hoặc Tù chối.")
        return

    set_form_state(is_enabled=True, widgets=widgets,
user_role=user_role)

    widgets['cbb_xe'].set("")
    widgets['entry_solit'].delete(0, tk.END)

```

```

        widgets['entry_tongchiph'].delete(0, tk.END)
        widgets['entry_soodo'].delete(0, tk.END)

        now = datetime.now()
        widgets['date_ngaydo'].set_date(now.strftime("%Y-%m-%d"))
        widgets['entry_giodo'].delete(0, tk.END)
        widgets['entry_giodo'].insert(0, now.strftime("%H:%M"))

        widgets['cbb_trangthai'].set("Chờ duyệt")

    # TỰ ĐỘNG ĐIỀN THÔNG TIN TÀI XẾ
    if user_role == "TaiXe":
        manv, ten_hien_thi = get_mav_from_username(user_username)
        if ten_hien_thi:
            widgets['cbb_taixe'].set(ten_hien_thi)
    else:
        widgets['cbb_taixe'].set("")

    widgets['cbb_xe'].focus()

    tree = widgets['tree']
    if tree.selection():
        tree.selection_remove(tree.selection()[0])

def on_item_select(event, widgets):
    """(SỰ KIỆN CLICK) Đổ dữ liệu lên form (phân quyền)."""
    tree = widgets['tree']
    selected = tree.selection()
    if not selected: return

    selected_item = tree.item(selected[0])
    manhatky = selected_item['values'][0]

    user_role = widgets.get("user_role", "Admin")

    conn = utils.connect_db()
    if conn is None: return

    try:
        cur = conn.cursor()
        sql = """
        SELECT nk.*, nv.HoVaTen
        FROM NhatKyNhienLieu nk
        LEFT JOIN NhanVien nv ON nk.MaNhanVien = nv.MaNhanVien
        WHERE nk.MaNhatKy = ?
        """
        cur.execute(sql, (manhatky,))
        data = cur.fetchone()

        if not data:

```

```

        messagebox.showerror("Lỗi", "Không tìm thấy dữ liệu nhật
ký.")
    return

    # === Admin không cần bật form ===
    if user_role == "TaiXe":
        set_form_state(is_enabled=True, widgets=widgets,
user_role=user_role)

        # Xóa các fields khác
        widgets['cbb_xe'].set("")
        widgets['cbb_taixe'].set("")
        widgets['entry_giodo'].delete(0, tk.END)
        widgets['entry_solist'].delete(0, tk.END)
        widgets['entry_tongchiphi'].delete(0, tk.END)
        widgets['entry_soodo'].delete(0, tk.END)

        # Đìền
        widgets['cbb_xe'].set(data.BienSoXe or "")

    if data.MaNhanVien:
        cbb_taixe_val = f"{data.MaNhanVien} - {data.HoVaTen}"
        widgets['cbb_taixe'].set(cbb_taixe_val)

    if data.NgayDoNhienLieu:
        widgets['date_ngaydo'].set_date(data.NgayDoNhienLieu)
        widgets['entry_giodo'].insert(0,
data.NgayDoNhienLieu.strftime("%H:%M"))

        widgets['entry_solist'].insert(0, str(data.SoLit or ""))
        widgets['entry_tongchiphi'].insert(0, str(data.TongChiPhi or ""))
        widgets['entry_soodo'].insert(0, str(data.SoOdo or ""))

        trangthai_map = {0: "Chờ duyệt", 1: "Đã duyệt", 2: "Từ chối"}
        trangthai_text = trangthai_map.get(data.TrangThaiDuyet, "Chờ
duyệt")
        widgets['cbb_trangthai'].set(trangthai_text)

    # === KHÓA FORM NẾU LÀ TÀI XE VÀ ĐÃ DUYỆT ===
    if user_role == "TaiXe" and data.TrangThaiDuyet != 0: # 0 =
Chờ duyệt
        set_form_state(is_enabled=False, widgets=widgets,
user_role=user_role)

    except Exception as e:
        messagebox.showerror("Lỗi không xác định", f"Lỗi: {str(e)}")
    finally:
        if conn: conn.close()

```

```

# Tắt form ở cuối (nếu không ở chế độ sửa)
btn_sua = widgets.get("btn_sua") # Nút này chỉ Tài xế mới có
if (user_role == "Admin") or (btn_sua and
btn_sua.cget('text') != 'Hủy Sửa'):
    set_form_state(is_enabled=False, widgets=widgets,
user_role=user_role)

def chon_nhienlieu_de_sua(widgets, user_role):
    """(NÚT SỬA) Kích hoạt chế độ sửa (phân quyền)."""
    if user_role == "Admin":
        messagebox.showwarning("Thông báo", "Admin chỉ có thể Duyệt
hoặc Từ chối.")
        return

    selected = widgets['tree'].selection()
    if not selected:
        messagebox.showwarning("Chưa chọn", "Hãy chọn một nhặt ký
trong danh sách trước khi nhấn 'Sửa'")
        return

    manhatky = get_manhatky_from_tree(widgets)
    if not manhatky:
        messagebox.showwarning("Lỗi", "Không tìm thấy mã nhặt ký. Vui
lòng chọn lại.")
        return

    trangthai_text = widgets['cbb_trangthai_var'].get()
    if trangthai_text != "Chờ duyệt":
        messagebox.showwarning("Không thể sửa", "Tài xế chỉ có thể sửa
nhặt ký đang 'Chờ duyệt'.")
        return

    set_form_state(is_enabled=True, widgets=widgets,
user_role=user_role)
    widgets['cbb_xe'].focus()

```

### - Hàm logic CRUD

```

def load_data(widgets, user_role, user_username):
    """(READ) Tải dữ liệu lọc theo vai trò."""
    tree = widgets['tree']
    for i in tree.get_children():
        tree.delete(i)

    conn = utils.connect_db()
    if conn is None:
        set_form_state(is_enabled=False, widgets=widgets,
user_role=user_role)
        return

```

```

try:
    cur = conn.cursor()
    sql = """
        SELECT
            nk.MaHatKy, nk.BienSoXe, nv.HoVaTen,
            nk.NgayDoNghienLieu, nk.SoLit, nk.TongChiPhi,
nk.TrangThaiDuyet
        FROM NhatKyNghienLieu AS nk
        LEFT JOIN NhanVien AS nv ON nk.MaHanVien = nv.MaHanVien
"""

    # === LỌC SQL THEO TÀI XẾ ===
    params = []
    if user_role == "TaiXe":
        manv, _ = get_mav_from_username(user_username)
        if manv:
            sql += " WHERE nk.MaHanVien = ?"
            params.append(mav)
    else:
        sql += " WHERE 1=0" # Không thấy gì

    sql += " ORDER BY nk.NgayDoNghienLieu DESC"

    cur.execute(sql, params)
    rows = cur.fetchall()

    trangthai_map = { 0: "Chờ duyệt", 1: "Đã duyệt", 2: "Từ chối" }

    for row in rows:
        ma_nk = row[0]
        bienso = row[1]
        ten_tx = row[2] or "N/A"
        ngay_do = row[3].strftime("%Y-%m-%d %H:%M") if row[3] else
"N/A"
        so_lit = row[4]
        tong_cp = row[5]
        trangthai_text = trangthai_map.get(row[6], "Không rõ")

        tree.insert("", tk.END, values=(ma_nk, bienso, ten_tx,
ngay_do, so_lit, tong_cp, trangthai_text))

        children = tree.get_children()
        if children:
            first_item = children[0]
            tree.selection_set(first_item)
            tree.focus(first_item)
            tree.event_generate("<<TreeviewSelect>>")
        elif user_role == "TaiXe":
            # Nếu là Tài xế và bảng trống, bật chế độ Thêm

```

```

        set_form_state(is_enabled=True, widgets=widgets,
user_role=user_role)
        clear_input(widgets, user_role, user_username)

    except Exception as e:
        messagebox.showerror("Lỗi tải dữ liệu", f'Lỗi SQL: {str(e)}')
    finally:
        if conn:
            conn.close()

def them_nhienlieu(widgets, user_role, user_username):
    """(CREATE) Thêm một nhật ký nhiên liệu mới (chỉ Tài xế)."""
    try:
        bienso = widgets['cbb_xe_var'].get()

        manv = ""
        if user_role == "TaiXe":
            manv, _ = get_manv_from_username(user_username)
            if not manv:
                messagebox.showerror("Lỗi", "Không tìm thấy mã nhân
viên của bạn.")
                return False
        else:
            messagebox.showerror("Lỗi", "Admin không thể thêm nhật
ký.")
            return False

        ngay_do_str = widgets['date_ngaydo'].get_date().strftime('%Y-
%m-%d')
        gio_do_str = widgets['entry_giodo'].get() or "00:00"
        tg_nhienlieu = f'{ngay_do_str} {gio_do_str}:00'

        solit = widgets['entry_solit'].get()
        tongchiph = widgets['entry_tongchiph'].get()
        soodo = widgets['entry_soodo'].get()

        trangthai_value = 0 # Tài xế thêm luôn là "Chờ duyệt"

        if not bienso or not manv or not solit or not tongchiph:
            messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập Xe,
Số lít và Chi phí")
            return False

        solit_dec = float(solit) if solit else 0.0
        tongchiph_dec = float(tongchiph) if tongchiph else 0.0
        soodo_int = int(soodo) if soodo else None

    except Exception as e:
        messagebox.showerror("Lỗi định dạng", f'Dữ liệu số (lít, chi
phí, odo) không hợp lệ: {e}')

```

```

        return False

    conn = utils.connect_db()
    if conn is None: return False

    try:
        cur = conn.cursor()
        sql = """
            INSERT INTO NhatKyNhienLieu (
                BienSoXe, MaNhanVien, NgayDoNhienLieu, SoLit,
                TongChiPhi, SoOdo, TrangThaiDuyet
            ) VALUES (?, ?, ?, ?, ?, ?, ?, ?)
        """
        cur.execute(sql, (bienso, manv, tg_nhienlieu, solit_dec,
tongchiphidec, soodo_int, trangthai_value))
        conn.commit()
        messagebox.showinfo("Thành công", "Đã thêm nhật ký nhiên liệu
(chờ duyệt).")
        return True

    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể thêm nhật
ký:{\n{str(e)}}")
        return False
    finally:
        if conn: conn.close()

def luu_nhienlieu_da_sua(widgets, user_role, user_username):
    """(UPDATE) Lưu thay đổi (UPDATE) sau khi sửa (chỉ Tài xế)."""
    if user_role == "Admin":
        messagebox.showwarning("Thông báo", "Admin chỉ có thể Duyệt
hoặc Từ chối.")
        return False

    manhatky = get_manhatky_from_tree(widgets)
    if not manhatky:
        messagebox.showwarning("Thiếu dữ liệu", "Không có Mã nhật ký
để cập nhật")
        return False

    try:
        bienso = widgets['cbb_xe_var'].get()
        manv, _ = get_manv_from_username(user_username)

        ngay_do_str = widgets['date_ngaydo'].get_date().strftime('%Y-
%m-%d')
        gio_do_str = widgets['entry_giodo'].get() or "00:00"
        tg_nhienlieu = f"{ngay_do_str} {gio_do_str}:00"

```

```

solit = widgets['entry_solit'].get()
tongchiph = widgets['entry_tongchiph'].get()
soodo = widgets['entry_sodo'].get()

trangthai_value = 0 # Tài xế sửa thì luôn là "Chờ duyệt"

if not bienso or not manv or not solit or not tongchiph:
    messagebox.showwarning("Thiếu dữ liệu", "Xe, Tài xế, Số
lít và Chi phí không được rỗng")
    return False

solit_dec = float(solit) if solit else 0.0
tongchiph_dec = float(tongchiph) if tongchiph else 0.0
soodo_int = int(soodo) if soodo else None

except Exception as e:
    messagebox.showerror("Lỗi định dạng", f"Dữ liệu nhập không hợp
lệ: {e}")
    return False

conn = utils.connect_db()
if conn is None: return False

try:
    cur = conn.cursor()

    # Kiểm tra lần nữa: Tài xế chỉ được sửa khi "Chờ duyệt"
    cur.execute("SELECT TrangThaiDuyet FROM NhatKyNhienLieu WHERE
MaNhatKy = ?", (manhatky,))
    status = cur.fetchone()
    if status and status[0] != 0:
        messagebox.showwarning("Không thể sửa", "Tài xế chỉ có thể
sửa nhật ký đang 'Chờ duyệt'.")
        return False

    sql = """
UPDATE NhatKyNhienLieu SET
    BienSoXe = ?, MaNhanVien = ?, NgayDoNhiemLieu = ?,
    SoLit = ?, TongChiPhi = ?, SoOdo = ?, TrangThaiDuyet = ?
WHERE MaNhatKy = ?
"""
    cur.execute(sql, (
        bienso, manv, tg_nhienlieu,
        solit_dec, tongchiph_dec, soodo_int, trangthai_value,
        manhatky
    ))
    conn.commit()
    messagebox.showinfo("Thành công", "Đã cập nhật nhật ký nhiên
liệu")
    return True

```

```

        except Exception as e:
            conn.rollback()
            messagebox.showerror("Lỗi SQL", f"Không thể cập nhật:{\n{str(e)}}")
            return False
        finally:
            if conn: conn.close()

def save_data(widgets, user_role, user_username):
    """(NÚT LƯU) Tự động kiểm tra xem nên Thêm mới (INSERT) hay Cập nhật (UPDATE)."""
    # Nếu có item đang được chọn trong cây, ta giả định là Sửa
    if get_manhatky_from_tree(widgets):
        success = luu_nhienlieu_da_sua(widgets, user_role, user_username)
    else:
        # Nếu không có item nào được chọn, ta giả định là Thêm
        success = them_nhienlieu(widgets, user_role, user_username)

    if success:
        load_data(widgets, user_role, user_username)

def xoa_nhienlieu(widgets, user_role, user_username):
    """(DELETE) Xóa nhật ký được chọn (chỉ Tài xế)."""
    if user_role == "Admin":
        messagebox.showwarning("Thông báo", "Admin chỉ có thể Duyệt hoặc Từ chối.")
        return

    selected = widgets['tree'].selection()
    if not selected:
        messagebox.showwarning("Chưa chọn", "Hãy chọn một nhật ký để xóa")
        return

    manhatky = get_manhatky_from_tree(widgets)
    if not manhatky:
        messagebox.showwarning("Lỗi", "Không tìm thấy mã nhật ký. Vui lòng chọn lại.")
        return

    trangthai_text = widgets['cbb_trangthai_var'].get()
    if trangthai_text != "Chờ duyệt":
        messagebox.showwarning("Không thể xóa", "Tài xế chỉ có thể xóa nhật ký đang 'Chờ duyệt'.")
        return

    if not messagebox.askyesno("Xác nhận", f"Bạn có chắc chắn muốn xóa Nhật ký Mã: {manhatky}"):

```

```

        return

    conn = utils.connect_db()
    if conn is None: return

    try:
        cur = conn.cursor()
        cur.execute("DELETE FROM NhatKyNhienLieu WHERE MaNhatKy=?", (manhatky,))
        conn.commit()

        messagebox.showinfo("Thành công", "Đã xóa nhật ký thành công")
        load_data(widgets, user_role, user_username)

    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể xóa:\n{str(e)}")
    finally:
        if conn: conn.close()

def duyet_nhienlieu(widgets, user_role, user_username, new_status):
    """(ADMIN UPDATE) Cập nhật trạng thái nhật ký (1 = Duyệt, 2 = Từ chối)."""

    manhatky = get_manhatky_from_tree(widgets)
    if not manhatky:
        messagebox.showwarning("Chưa chọn", "Vui lòng chọn một nhật ký từ danh sách.")
        return

    trangthai_hientai = widgets['cbb_trangthai_var'].get()

    if new_status == 1: # Duyệt
        if trangthai_hientai == "Đã duyệt":
            messagebox.showinfo("Thông báo", "Nhật ký này đã được duyệt rồi.")
            return
        action_text = "DUYỆT"
    else: # Từ chối
        if trangthai_hientai == "Từ chối":
            messagebox.showinfo("Thông báo", "Nhật ký này đã bị từ chối rồi.")
            return
        action_text = "TỪ CHỐI"

    if not messagebox.askyesno("Xác nhận", f"Bạn có chắc chắn muốn {action_text} nhật ký Mã: {manhatky}?"):
        return

    conn = utils.connect_db()

```

```

if conn is None: return

try:
    cur = conn.cursor()
    sql = "UPDATE NhatKyNhienLieu SET TrangThaiDuyet = ? WHERE MaNhatKy = ?"
    cur.execute(sql, (new_status, manhatky))
    conn.commit()
    messagebox.showinfo("Thành công", f"Đã {action_text} nhật ký.")
except Exception as e:
    conn.rollback()
    messagebox.showerror("Lỗi SQL", f"Không thể cập nhật:\n{str(e)}")
finally:
    if conn: conn.close()
    load_data(widgets, user_role, user_username)

```

### - Hàm tạo trang

```

def create_page(master, user_role, user_username):
    """
    Hàm này tạo ra toàn bộ nội dung trang.
    """

    # 1. TẠO FRAME CHÍNH VÀ CÀI STYLE
    page_frame = ttk.Frame(master, style="TFrame")
    utils.setup_theme(page_frame)

    # === Xử lý LỌC XE theo Tài xế ===
    manv_logged_in, _ = get_manv_from_username(user_username)
    xe_options = []
    if user_role == "Admin":
        xe_options = utils.load_all_xe_combobox() # Hàm này cần được
    tạo trong utils.py
    elif manv_logged_in:
        xe_options = utils.load_xe_by_manv(mav_logged_in) # Giả định
    utils có hàm này

    # === Tải Combobox Tài xế (chỉ dùng cho Admin) ===
    taixe_options = []
    if user_role == "Admin":
        taixe_options = utils.load_taixe_combobox() # Giả định utils
    có hàm này

    # 2. TẠO TIÊU ĐỀ
    lbl_title = ttk.Label(page_frame, text="QUẢN LÝ NHẬT KÝ NHIÊN
LIỆU", style="Title.TLabel")
    lbl_title.pack(pady=15)

```

```

# 3. KHUNG FORM NHẬP LIỆU
frame_info = ttk.Frame(page_frame, style="TFrame")
frame_info.pack(pady=5, padx=20, fill="x")

date_entry_style_options = {
    'width': 28, 'date_pattern': 'yyyy-MM-dd',
    'background': utils.theme_colors["bg_entry"],
    'foreground': utils.theme_colors["text"],
    'disabledbackground': utils.theme_colors["disabled_bg"],
    'disabledforeground': utils.theme_colors["text_disabled"],
    'bordercolor': utils.theme_colors["bg_entry"],
    'headersbackground': utils.theme_colors["accent"],
    'headersforeground': utils.theme_colors["accent_text"],
    'selectbackground': utils.theme_colors["accent"],
    'selectforeground': utils.theme_colors["accent_text"]
}

# --- Hàng 0 (Xe và Trạng thái) ---
ttk.Label(frame_info, text="Xe:").grid(row=0, column=0, padx=5,
pady=8, sticky="w")
cbb_xe_var = tk.StringVar()
cbb_xe = ttk.Combobox(frame_info, textvariable=cbb_xe_var,
values=xe_options, width=28, state='readonly')
cbb_xe.grid(row=0, column=1, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Trạng thái:").grid(row=0, column=2,
padx=15, pady=8, sticky="w")
trangthai_options = ["Chờ duyệt", "Đã duyệt", "Từ chối"]
cbb_trangthai_var = tk.StringVar()
cbb_trangthai = ttk.Combobox(frame_info,
textvariable=cbb_trangthai_var, values=trangthai_options, width=38,
state='readonly')
cbb_trangthai.grid(row=0, column=3, padx=5, pady=8, sticky="w")
cbb_trangthai.set("Chờ duyệt")

# --- Hàng 1 (Ngày đổ và Tài xế đổ) ---
ttk.Label(frame_info, text="Ngày đổ:").grid(row=1, column=0,
padx=5, pady=8, sticky="w")
date_ngaydo = DateEntry(frame_info, **date_entry_style_options)
date_ngaydo.grid(row=1, column=1, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Tài xế đổ:").grid(row=1, column=2,
padx=15, pady=8, sticky="w")
cbb_taixe_var = tk.StringVar()
cbb_taixe = ttk.Combobox(frame_info, textvariable=cbb_taixe_var,
values=taixe_options, width=38, state='readonly')
cbb_taixe.grid(row=1, column=3, padx=5, pady=8, sticky="w")

# --- Hàng 2 (Số lít và Giờ đổ) ---

```

```

        ttk.Label(frame_info, text="Số lít:").grid(row=2, column=0, padx=5,
pady=8, sticky="w")
        entry_solist = ttk.Entry(frame_info, width=30)
        entry_solist.grid(row=2, column=1, padx=5, pady=8, sticky="w")

        ttk.Label(frame_info, text="Giờ đổ (HH:MM):").grid(row=2, column=2,
padx=15, pady=8, sticky="w")
        entry_giodo = ttk.Entry(frame_info, width=40)
        entry_giodo.grid(row=2, column=3, padx=5, pady=8, sticky="w")

# --- Hàng 3 (Số Odo và Tổng chi phí) ---
        ttk.Label(frame_info, text="Số Odo (Km):").grid(row=3, column=0,
padx=5, pady=8, sticky="w")
        entry_soodo = ttk.Entry(frame_info, width=30)
        entry_soodo.grid(row=3, column=1, padx=5, pady=8, sticky="w")

        ttk.Label(frame_info, text="Tổng chi phí:").grid(row=3, column=2,
padx=15, pady=8, sticky="w")
        entry_tongchiph = ttk.Entry(frame_info, width=40)
        entry_tongchiph.grid(row=3, column=3, padx=5, pady=8, sticky="w")

frame_info.columnconfigure(1, weight=1)
frame_info.columnconfigure(3, weight=1)

# 4. KHUNG CÁC NÚT CHỨC NĂNG
frame_btn = ttk.Frame(page_frame, style="TFrame")
frame_btn.pack(pady=10)

# 5. KHUNG BẢNG DANH SÁCH (TREEVIEW)
lbl_ds_text = "Danh sách nhật ký nhiên liệu (Sắp xếp mới nhất)"
if user_role == "TaiXe":
    lbl_ds_text = "Danh sách nhật ký của bạn"
lbl_ds = ttk.Label(page_frame, text=lbl_ds_text,
style="Header.TLabel")
lbl_ds.pack(pady=(10, 5), padx=20, anchor="w")

frame_tree = ttk.Frame(page_frame, style="TFrame")
frame_tree.pack(pady=10, padx=20, fill="both", expand=True)

scrollbar_y = ttk.Scrollbar(frame_tree, orient=tk.VERTICAL,
style="Vertical.TScrollbar")
scrollbar_x = ttk.Scrollbar(frame_tree, orient=tk.HORIZONTAL,
style="Horizontal.TScrollbar")
columns = ("ma_nk", "bienso", "ten_tx", "ngay_do", "so_lit",
"tong_cp", "trangthai")
tree = ttk.Treeview(frame_tree, columns=columns, show="headings",
height=10,
            yscrollcommand=scrollbar_y.set,
            xscrollcommand=scrollbar_x.set)
scrollbar_y.config(command=tree.yview)

```

```

        scrollbar_x.config(command=tree.xview)
        scrollbar_y.pack(side=tk.RIGHT, fill=tk.Y)
        scrollbar_x.pack(side=tk.BOTTOM, fill=tk.X)

        tree.heading("ma_nk", text="Mã NK")
        tree.column("ma_nk", width=60, anchor="center")
        tree.heading("bienso", text="Biển số xe")
        tree.column("bienso", width=100)
        tree.heading("ten_tx", text="Tên Tài xế")
        tree.column("ten_tx", width=150)
        tree.heading("ngay_do", text="Ngày đổ")
        tree.column("ngay_do", width=150)
        tree.heading("so_lit", text="Số lít")
        tree.column("so_lit", width=80, anchor="e")
        tree.heading("tong_cp", text="Tổng chi phí")
        tree.column("tong_cp", width=100, anchor="e")
        tree.heading("trangthai", text="Trạng thái")
        tree.column("trangthai", width=100, anchor="center")

    tree.pack(fill="both", expand=True)

# 6. TẠO TỪ ĐIỂN 'widgets' ĐỂ TRUYỀN ĐI
widgets = {
    "tree": tree,
    "cbb_xe": cbb_xe,
    "cbb_taixe": cbb_taixe,
    "date_ngaydo": date_ngaydo,
    "entry_giodo": entry_giodo,
    "entry_solid": entry_solid,
    "entry_tongchiph": entry_tongchiph,
    "entry_soodo": entry_soodo,
    "cbb_trangthai": cbb_trangthai,
    "cbb_xe_var": cbb_xe_var,
    "cbb_taixe_var": cbb_taixe_var,
    "cbb_trangthai_var": cbb_trangthai_var,
    "user_role": user_role,
    "user_username": user_username
}

# 7. GÁN LỆNH CHO CÁC NÚT (THEO VAI TRÒ)
if user_role == "Admin":
    # Admin chỉ thấy nút Duyệt và Từ chối
    btn_duyet = ttk.Button(frame_btn, text="Duyệt", width=10,
style="Accent.TButton",
                           command=lambda: duyet_nhienlieu(widgets,
user_role, user_username, 1))
    btn_duyet.grid(row=0, column=0, padx=10)

    try:
        style = ttk.Style()

```

```

        danger_color = "#DC3545"
        danger_active = "#B02A37"
        style.configure("Danger.TButton", font=("Calibri", 11,
"bold"), background=danger_color, foreground="white", relief="flat",
padding=5)
        style.map("Danger.TButton", background=[('active',
danger_active)], relief=[('pressed', 'sunken')])
    except Exception:
        pass

        btn_tuchoi = ttk.Button(frame_btn, text="Tù chối", width=10,
style="Danger.TButton",
                           command=lambda:
duyet_nhienlieu(widgets, user_role, user_username, 2))
        btn_tuchoi.grid(row=0, column=1, padx=10)

    else: # Nếu là TaiXe
        # Tài xế thấy Thêm, Lưu, Sửa, Hủy, Xóa
        btn_them = ttk.Button(frame_btn, text="Thêm", width=8,
command=lambda: clear_input(widgets, user_role, user_username))
        btn_them.grid(row=0, column=0, padx=10)

        btn_luu = ttk.Button(frame_btn, text="Lưu", width=8,
command=lambda: save_data(widgets, user_role, user_username))
        btn_luu.grid(row=0, column=1, padx=10)

        btn_sua = ttk.Button(frame_btn, text="Sửa", width=8,
command=lambda: chon_nhienlieu_de_sua(widgets, user_role))
        btn_sua.grid(row=0, column=2, padx=10)

        btn_huy = ttk.Button(frame_btn, text="Hủy", width=8,
command=lambda: load_data(widgets, user_role, user_username))
        btn_huy.grid(row=0, column=3, padx=10)

        btn_xoa = ttk.Button(frame_btn, text="Xóa", width=8,
command=lambda: xoa_nhienlieu(widgets, user_role, user_username))
        btn_xoa.grid(row=0, column=4, padx=10)

        widgets["btn_sua"] = btn_sua # Gán nút Sửa vào widgets

# 8. KẾT NỐI BINDING
tree.bind("<<TreeviewSelect>>", lambda event: on_item_select(event,
widgets))

# 9. TẢI DỮ LIỆU LẦN ĐẦU
load_data(widgets, user_role, user_username)

# 10. TRẢ VỀ FRAME CHÍNH
return page_frame

```

## **h. Form QuanLyTaiXe**

- Hàm quản lý và sự kiện

```
def set_form_state(is_enabled, widgets):
    """Bật (enable) hoặc Tắt (disable) toàn bộ các trường nhập liệu."""
    if is_enabled:
        widgets['entry_hoten'].config(state='normal')
        widgets['entry_sdt'].config(state='normal')
        widgets['entry_diachi'].config(state='normal')
        widgets['entry_banglai'].config(state='normal')
        widgets['date_banglai'].config(state='normal')
        widgets['cbb_trangthai_nv'].config(state='readonly')
        widgets['cbb_trangthai_lx'].config(state='readonly')
    else:
        # Mã NV luôn bị khóa khi ở chế độ xem
        widgets['entry_manv'].config(state='disabled')
        widgets['entry_hoten'].config(state='disabled')
        widgets['entry_sdt'].config(state='disabled')
        widgets['entry_diachi'].config(state='disabled')
        widgets['entry_banglai'].config(state='disabled')
        widgets['date_banglai'].config(state='disabled')
        widgets['cbb_trangthai_nv'].config(state='disabled')
        widgets['cbb_trangthai_lx'].config(state='disabled')

def clear_input(widgets):
    """(NÚT THÊM) Xóa trắng và Mở khóa các trường nhập liệu (Chế độ Thêm mới)."""
    set_form_state(is_enabled=True, widgets=widgets)
    widgets['entry_manv'].config(state='normal') # Mở khóa Mã NV

    widgets['entry_manv'].delete(0, tk.END)
    widgets['entry_hoten'].delete(0, tk.END)
    widgets['entry_sdt'].delete(0, tk.END)
    widgets['entry_diachi'].delete(0, tk.END)
    widgets['entry_banglai'].delete(0, tk.END)

    widgets['date_banglai'].set_date("2025-01-01")
    widgets['cbb_trangthai_nv'].set("Đang làm việc")
    widgets['cbb_trangthai_lx'].set("Rảnh")

    widgets['entry_manv'].focus()

    # Bỏ chọn item trên Treeview
    tree = widgets['tree']
    if tree.selection():
        tree.selection_remove(tree.selection()[0])
```

```

def on_item_select(event, widgets):
    """(SỰ KIỆN CLICK) Khi click vào Treeview, đổ dữ liệu đầy đủ lên
form (ở trạng thái mờ)."""
    tree = widgets['tree']
    selected = tree.selection()
    if not selected: return

    selected_item = tree.item(selected[0])
    manv = selected_item['values'][0]

    conn = utils.connect_db()
    if conn is None: return

    try:
        cur = conn.cursor()
        sql = "SELECT * FROM NhanVien nv JOIN TaiXe tx ON
nv.MaNhanVien = tx.MaNhanVien WHERE nv.MaNhanVien = ?"
        cur.execute(sql, (manv,))
        data = cur.fetchone()

        if not data:
            messagebox.showerror("Lỗi", "Không tìm thấy dữ liệu tài
xế.")
            return

        # 1. Mở khóa form để điền dữ liệu
        set_form_state(is_enabled=True, widgets=widgets)
        widgets['entry_mav'].config(state='normal')

        # 2. Xóa
        widgets['entry_mav'].delete(0, tk.END)
        widgets['entry_hoten'].delete(0, tk.END)
        widgets['entry_sdt'].delete(0, tk.END)
        widgets['entry_diachi'].delete(0, tk.END)
        widgets['entry_banglai'].delete(0, tk.END)

        # 3. Điền
        widgets['entry_mav'].insert(0, data.MaNhanVien)
        widgets['entry_hoten'].insert(0, data.HoVaTen or "")
        widgets['entry_sdt'].insert(0, data.SoDienThoai or "")
        widgets['entry_diachi'].insert(0, data.DiaChi or "")
        widgets['cbb_trangthai_nv'].set("Đang làm việc" if
data.TrangThai == 1 else "Nghỉ")
        widgets['entry_banglai'].insert(0, data.HangBangLai or "")
        if data.NgayHetHanBangLai:
            widgets['date_banglai'].set_date(data.NgayHetHanBangLai)
        else:
            widgets['date_banglai'].set_date("2025-01-01")

```

```

        widgets['cbb_trangthai_lx'].set("Rảnh" if data.TrangThaiTaiXe
== 1 else "Đang lái")

    except pyodbc.Error as e:
        messagebox.showerror("Lỗi SQL", f"Không thể lấy dữ liệu tài
xế:\n{str(e)}")
    except Exception as e:
        messagebox.showerror("Lỗi không xác định", f"Lỗi: {str(e)}")
    finally:
        if conn: conn.close()
        # 4. Khóa form lại (chế độ View)
        set_form_state(is_enabled=False, widgets=widgets)

def chon_taixe_de_sua(widgets):
    """(NÚT SỬA) Kích hoạt chế độ sửa, Mở khóa form (trừ MaNV)."""
    selected = widgets['tree'].selection()
    if not selected:
        messagebox.showwarning("Chưa chọn", "Hãy chọn một tài xế trong
danh sách trước khi nhấn 'Sửa'")
        return

    if not widgets['entry_manv'].get():
        messagebox.showwarning("Lỗi", "Không tìm thấy mã nhân viên.
Vui lòng chọn lại.")
        return

    set_form_state(is_enabled=True, widgets=widgets)
    widgets['entry_manv'].config(state='disabled') # Không cho sửa
    khóa chính
    widgets['entry_hoten'].focus()

```

### - Hàm logic CRUD

```

def load_data(widgets):
    """(READ) Tải TOÀN BỘ dữ liệu tài xế (JOIN) VÀ LÀM MỜ FORM."""
    tree = widgets['tree']
    for i in tree.get_children():
        tree.delete(i)

    conn = utils.connect_db()
    if conn is None:
        set_form_state(is_enabled=False, widgets=widgets)
        return

    try:
        cur = conn.cursor()
        sql = """
        SELECT
            nv.MaNhanVien, nv.HoVaTen, nv.SoDienThoai,

```

```

        tx.HangBangLai, nv.TrangThai, tx.TrangThaiTaiXe
    FROM NhanVien AS nv
    JOIN TaiXe AS tx ON nv.MaNhanVien = tx.MaNhanVien
    """
    cur.execute(sql)
    rows = cur.fetchall()

    for row in rows:
        trang_thai_nv_text = "Đang làm việc" if row[4] == 1 else
    "Nghỉ"
        trang_thai_lx_text = "Rảnh" if row[5] == 1 else "Đang lái"

        tree.insert("", tk.END, values=(
            row[0], row[1], row[2], row[3],
            trang_thai_nv_text, trang_thai_lx_text
        ))

        children = tree.get_children()
        if children:
            # Tự động chọn dòng đầu tiên
            first_item = children[0]
            tree.selection_set(first_item)
            tree.focus(first_item)
            tree.event_generate("<<TreeviewSelect>>")
        else:
            # Nếu không có dữ liệu, bật chế độ Thêm
            clear_input(widgets)

    except pyodbc.Error as e:
        messagebox.showerror("Lỗi tải dữ liệu", f"Lỗi SQL: {str(e)}")
    except Exception as e:
        messagebox.showerror("Lỗi không xác định", f"Lỗi: {str(e)}")
    finally:
        if conn:
            conn.close()

        # Nếu có dữ liệu, khóa form. Nếu không, để mở (theo logic của
        clear_input)
        if tree.get_children():
            set_form_state(is_enabled=False, widgets=widgets)

def them_taixe(widgets):
    """(CREATE) Thêm một tài xế mới (vào cả 2 bảng NhanVien và
    TaiXe)."""
    manv = widgets['entry_manv'].get()
    hoten = widgets['entry_hoten'].get()
    sdt = widgets['entry_sdt'].get()
    diachi = widgets['entry_diachi'].get()
    banglai = widgets['entry_banglai'].get()
    ngay_bl = widgets['date_banglai'].get_date().strftime('%Y-%m-%d')

```

```

        trangthai_nv_text = widgets['cbb_trangthai_nv_var'].get()
        trangthai_nv_value = 1 if trangthai_nv_text == "Đang làm việc"
else 0

        trangthai_lx_text = widgets['cbb_trangthai_lx_var'].get()
        trangthai_lx_value = 1 if trangthai_lx_text == "Rảnh" else 2

        if not manv or not hoten or not banglai:
            messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập Mã NV,
Họ Tên và Hạng Bằng Lái")
            return False

        conn = utils.connect_db()
        if conn is None: return False

        try:
            cur = conn.cursor()

            sql_nhanvien = "INSERT INTO NhanVien (MaNhanVien, HoVaTen,
SoDienThoai, DiaChi, TrangThai) VALUES (?, ?, ?, ?, ?)"
            cur.execute(sql_nhanvien, (manv, hoten, sdt, diachi,
trangthai_nv_value))

            sql_taixe = "INSERT INTO TaiXe (MaNhanVien, HangBangLai,
NgayHetHanBangLai, TrangThaiTaiXe) VALUES (?, ?, ?, ?)"
            cur.execute(sql_taixe, (manv, banglai, ngay_bl,
trangthai_lx_value))

            conn.commit()
            messagebox.showinfo("Thành công", "Đã thêm tài xế mới thành
công")
            return True

        except pyodbc.IntegrityError:
            conn.rollback()
            messagebox.showerror("Lỗi Trùng lặp", f"Mã nhân viên '{manv}'"
đã tồn tại.")
            return False
        except pyodbc.Error as e:
            conn.rollback()
            messagebox.showerror("Lỗi SQL", f"Không thể thêm tài
xế:\n{str(e)}")
            return False
        except Exception as e:
            conn.rollback()
            messagebox.showerror("Lỗi không xác định", f"Lỗi: {str(e)}")
            return False
        finally:
            if conn: conn.close()

```

```

def luu_taixe_da_sua(widgets):
    """(UPDATE) Lưu thay đổi (UPDATE) sau khi sửa (vào cả 2 bảng)."""
    manv = widgets['entry_manv'].get()
    if not manv:
        messagebox.showwarning("Thiếu dữ liệu", "Không có Mã Nhân Viên để cập nhật")
        return False

    hoten = widgets['entry_hoten'].get()
    sdt = widgets['entry_sdt'].get()
    diachi = widgets['entry_diachi'].get()
    banglai = widgets['entry_banglai'].get()
    ngay_bl = widgets['date_banglai'].get_date().strftime('%Y-%m-%d')

    trangthai_nv_text = widgets['cbb_trangthai_nv_var'].get()
    trangthai_nv_value = 1 if trangthai_nv_text == "Đang làm việc"
    else 0

    trangthai_lx_text = widgets['cbb_trangthai_lx_var'].get()
    trangthai_lx_value = 1 if trangthai_lx_text == "Rảnh" else 2

    if not hoten or not banglai:
        messagebox.showwarning("Thiếu dữ liệu", "Họ Tên và Hạng Bằng Lái không được rỗng")
        return False

    conn = utils.connect_db()
    if conn is None: return False

    try:
        cur = conn.cursor()

        sql_nhanvien = "UPDATE NhanVien SET HoVaTen = ?, SoDienThoai = ?, DiaChi = ?, TrangThai = ? WHERE MaNhanVien = ?"
        cur.execute(sql_nhanvien, (hoten, sdt, diachi,
        trangthai_nv_value, manv))

        sql_taixe = "UPDATE TaiXe SET HangBangLai = ?, NgayHetHanBangLai = ?, TrangThaiTaiXe = ? WHERE MaNhanVien = ?"
        cur.execute(sql_taixe, (banglai, ngay_bl, trangthai_lx_value,
        manv))

        conn.commit()
        messagebox.showinfo("Thành công", "Đã cập nhật thông tin tài xế")
        return True

    except pyodbc.Error as e:
        conn.rollback()

```

```

        messagebox.showerror("Lỗi SQL", f"Không thể cập nhật tài
xế:\n{str(e)}")
        return False
    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi không xác định", f'Lỗi: {str(e)}')
        return False
    finally:
        if conn: conn.close()

def save_data(widgets):
    """(NÚT LƯU) Tự động kiểm tra xem nên Thêm mới (INSERT) hay Cập
nhật (UPDATE)."""
    if widgets['entry_manv'].cget('state') == 'disabled':
        # Nếu MaNV bị khóa, nghĩa là đang Sửa
        success = luu_taixe_da_sua(widgets)
    else:
        # Nếu MaNV đang mở, nghĩa là đang Thêm
        success = them_taixe(widgets)

    if success:
        load_data(widgets)

def xoa_taixe_vinhvien(widgets):
    """(DELETE) Xóa vĩnh viễn tài xế và MỌI DỮ LIỆU LIÊN QUAN."""
    selected = widgets['tree'].selection()
    if not selected:
        messagebox.showwarning("Chưa chọn", "Hãy chọn một tài xế trong
danh sách để xóa")
        return

    manv = widgets['entry_manv'].get()

    if not manv:
        messagebox.showwarning("Lỗi", "Không tìm thấy mã nhân viên.
Vui lòng chọn lại.")
        return

    msg_xacnhan = (
        f"BẠN CÓ CHẮC CHẮN MUỐN XÓA VĨNH VIỄN TÀI XẾ '{manv}'?\n\n"
        "CẢNH BÁO: Thao tác này KHÔNG THỂ hoàn tác.\n"
        "Tất cả Lịch sử chuyến đi, Lịch sử nhiên liệu, Tài khoản, và
thông tin Tài xế/Nhân viên sẽ bị XÓA SẠCH."
    )
    if not messagebox.askyesno("XÁC NHẬN XÓA VĨNH VIỄN", msg_xacnhan):
        return

    conn = utils.connect_db()
    if conn is None: return

```

```

try:
    cur = conn.cursor()

    # Xóa/Cập nhật các bảng con/liên quan trước
    cur.execute("DELETE FROM NhatKyNhienLieu WHERE MaNhanVien=?", (manv,))
    cur.execute("DELETE FROM ChuyenDi WHERE MaNhanVien=?", (manv,))
    cur.execute("DELETE FROM TaiKhoan WHERE MaNhanVien=?", (manv,))
    cur.execute("UPDATE Xe SET MaNhanVienHienTai = NULL WHERE MaNhanVienHienTai = ?", (manv,))

    # Xóa các bảng chính
    cur.execute("DELETE FROM TaiXe WHERE MaNhanVien=?", (manv,))
    cur.execute("DELETE FROM NhanVien WHERE MaNhanVien=?", (manv,))

    conn.commit()

    messagebox.showinfo("Thành công", f"Đã xóa vĩnh viễn tài xế '{manv}' và tất cả dữ liệu liên quan.")
    load_data(widgets)

except pyodbc.Error as e:
    conn.rollback()
    messagebox.showerror("Lỗi SQL", f"Không thể xóa tài xế: {str(e)}")
except Exception as e:
    conn.rollback()
    messagebox.showerror("Lỗi không xác định", f"Lỗi: {str(e)}")
finally:
    if conn: conn.close()

```

### - Hàm tạo trang

```

def create_page(master):
    """
    Hàm này được main.py gọi.
    Nó tạo ra toàn bộ nội dung trang và đặt vào 'master' (là main_frame).
    """

    # 1. TẠO FRAME CHÍNH VÀ CÀI STYLE
    page_frame = ttk.Frame(master, style="TFrame")
    utils.setup_theme(page_frame)

    # 2. TẠO TIÊU ĐỀ
    lbl_title = ttk.Label(page_frame, text="QUẢN LÝ TÀI XẾ",
                          style="Title.TLabel")
    lbl_title.pack(pady=15)

```

```

# 3. KHUNG FORM NHẬP LIỆU
frame_info = ttk.Frame(page_frame, style="TFrame")
frame_info.pack(pady=5, padx=20, fill="x")

# --- Cột 1 ---
ttk.Label(frame_info, text="Mã nhân viên:").grid(row=0, column=0,
padx=5, pady=8, sticky="w")
entry_mav = ttk.Entry(frame_info, width=30)
entry_mav.grid(row=0, column=1, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Họ và tên:").grid(row=1, column=0,
padx=5, pady=8, sticky="w")
entry_hoten = ttk.Entry(frame_info, width=30)
entry_hoten.grid(row=1, column=1, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Số điện thoại:").grid(row=2, column=0,
padx=5, pady=8, sticky="w")
entry_sdt = ttk.Entry(frame_info, width=30)
entry_sdt.grid(row=2, column=1, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Địa chỉ:").grid(row=3, column=0,
padx=5, pady=8, sticky="w")
entry_diachi = ttk.Entry(frame_info, width=30)
entry_diachi.grid(row=3, column=1, padx=5, pady=8, sticky="w")

# --- Cột 2 ---
ttk.Label(frame_info, text="Hạng bằng lái:").grid(row=0, column=2,
padx=15, pady=8, sticky="w")
entry_banglai = ttk.Entry(frame_info, width=30)
entry_banglai.grid(row=0, column=3, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Ngày hết hạn BL:").grid(row=1,
column=2, padx=15, pady=8, sticky="w")
date_banglai = DateEntry(frame_info, width=28, date_pattern='yyyy-
MM-dd',
background=utils.theme_colors["bg_entry"],
foreground=utils.theme_colors["text"],
disabledbackground=utils.theme_colors["disabled_bg"],
disabledforeground=utils.theme_colors["text_disabled"],
bordercolor=utils.theme_colors["bg_entry"],
headersbackground=utils.theme_colors["accent"],
headersforeground=utils.theme_colors["accent_text"],
selectbackground=utils.theme_colors["accent"],
selectforeground=utils.theme_colors["accent_text"])
)
date_banglai.grid(row=1, column=3, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Trạng thái NV:").grid(row=2, column=2,
padx=15, pady=8, sticky="w")
trangthai_nv_options = ["Nghỉ", "Đang làm việc"]

```

```

cbb_trangthai_nv_var = tk.StringVar()
cbb_trangthai_nv = ttk.Combobox(frame_info,
textvariable=cbb_trangthai_nv_var, values=trangthai_nv_options,
width=28, state='readonly')
cbb_trangthai_nv.grid(row=2, column=3, padx=5, pady=8, sticky="w")
cbb_trangthai_nv.set("Đang làm việc")

ttk.Label(frame_info, text="Trạng thái Lái:").grid(row=3, column=2,
padx=15, pady=8, sticky="w")
trangthai_lx_options = ["Rảnh", "Đang lái"]
cbb_trangthai_lx_var = tk.StringVar()
cbb_trangthai_lx = ttk.Combobox(frame_info,
textvariable=cbb_trangthai_lx_var, values=trangthai_lx_options,
width=28, state='readonly')
cbb_trangthai_lx.grid(row=3, column=3, padx=5, pady=8, sticky="w")
cbb_trangthai_lx.set("Rảnh")

frame_info.columnconfigure(1, weight=1)
frame_info.columnconfigure(3, weight=1)

# 4. KHUNG CÁC NÚT CHỨC NĂNG
frame_btn = ttk.Frame(page_frame, style="TFrame")
frame_btn.pack(pady=10)

# 5. KHUNG BẢNG DANH SÁCH (TREEVIEW)
lbl_ds = ttk.Label(page_frame, text="Danh sách tài xế",
style="Header.TLabel")
lbl_ds.pack(pady=(10, 5), padx=20, anchor="w")

frame_tree = ttk.Frame(page_frame, style="TFrame")
frame_tree.pack(pady=10, padx=20, fill="both", expand=True) # expand=True ở cuối

scrollbar_y = ttk.Scrollbar(frame_tree, orient=tk.VERTICAL,
style="Vertical.TScrollbar")
scrollbar_x = ttk.Scrollbar(frame_tree, orient=tk.HORIZONTAL,
style="Horizontal.TScrollbar")

columns = ("manv", "hoten", "sdt", "banglai", "trangthai_nv",
"trangthai_lx")
tree = ttk.Treeview(frame_tree, columns=columns, show="headings",
height=10,
yscrollcommand=scrollbar_y.set,
xscrollcommand=scrollbar_x.set)

scrollbar_y.config(command=tree.yview)
scrollbar_x.config(command=tree.xview)
scrollbar_y.pack(side=tk.RIGHT, fill=tk.Y)
scrollbar_x.pack(side=tk.BOTTOM, fill=tk.X)

```

```

tree.heading("manv", text="Mã NV")
tree.column("manv", width=80, anchor="center")
tree.heading("hoten", text="Họ Tên")
tree.column("hoten", width=150)
tree.heading("sdt", text="Số điện thoại")
tree.column("sdt", width=120)
tree.heading("banglai", text="Hạng Bằng Lái")
tree.column("banglai", width=100, anchor="center")
tree.heading("trangthai_nv", text="Trạng thái NV")
tree.column("trangthai_nv", width=120, anchor="center")
tree.heading("trangthai_lx", text="Trạng thái Lái")
tree.column("trangthai_lx", width=120, anchor="center")

tree.pack(fill="both", expand=True)

# 6. TẠO TỪ ĐIỂN 'widgets' ĐỂ TRUYỀN ĐI
widgets = {
    "tree": tree,
    "entry_manv": entry_manv,
    "entry_hoten": entry_hoten,
    "entry_sdt": entry_sdt,
    "entry_diachi": entry_diachi,
    "entry_banglai": entry_banglai,
    "date_banglai": date_banglai,
    "cbb_trangthai_nv": cbb_trangthai_nv,
    "cbb_trangthai_lx": cbb_trangthai_lx,
    "cbb_trangthai_nv_var": cbb_trangthai_nv_var,
    "cbb_trangthai_lx_var": cbb_trangthai_lx_var
}

# 7. GÁN LỆNH CHO CÁC NÚT
btn_them = ttk.Button(frame_btn, text="Thêm", width=8,
command=lambda: clear_input(widgets))
btn_them.grid(row=0, column=0, padx=10)
btn_luu = ttk.Button(frame_btn, text="Lưu", width=8,
command=lambda: save_data(widgets))
btn_luu.grid(row=0, column=1, padx=10)
btn_sua = ttk.Button(frame_btn, text="Sửa", width=8,
command=lambda: chon_taixe_de_sua(widgets))
btn_sua.grid(row=0, column=2, padx=10)
btn_huy = ttk.Button(frame_btn, text="Hủy", width=8,
command=lambda: load_data(widgets))
btn_huy.grid(row=0, column=3, padx=10)
btn_xoa = ttk.Button(frame_btn, text="Xóa", width=8,
command=lambda: xoa_taixe_vinhvien(widgets))
btn_xoa.grid(row=0, column=4, padx=10)

# 8. KẾT NỐI BINDING
tree.bind("<>TreeviewSelect>>", lambda event: on_item_select(event,
widgets))

```

```

# 9. TẢI DỮ LIỆU LẦN ĐẦU
load_data(widgets)

# 10. TRẢ VỀ FRAME CHÍNH
return page_frame

```

## i. Form QuanLyXe

- Hàm tiện ích

```

def _load_driver_data():
    """Hàm helper để tải danh sách tài xế và tạo map tra cứu."""
    driver_id_to_name = {} # Map để load (Vd: 'NV002' -> 'Trần Văn B')
    driver_name_to_id = {} # Map để save (Vd: 'Trần Văn B' -> 'NV002')
    driver_display_list = [""] # Thêm một lựa chọn trống (không chọn
ai)

    conn = utils.connect_db()
    if conn is None:
        messagebox.showerror("Lỗi Tải Tài Xế", "Không thể kết nối CSDL
để tải danh sách tài xế.")
        return driver_id_to_name, driver_name_to_id,
driver_display_list

    try:
        cur = conn.cursor()

        # ===== SỬA CÂU SQL TẠI ĐÂY (DÙNG JOIN VÀ TÊN CỘT ĐÚNG) =====
        # Lấy các nhân viên còn làm việc (TrangThai=1) VÀ
        # có tồn tại trong bảng TaiXe (dùng INNER JOIN)
        sql = """
            SELECT
                NV.MaNhanVien,
                NV.HoVaTen
            FROM
                NhanVien AS NV
            INNER JOIN
                TaiXe AS TX ON NV.MaNhanVien = TX.MaNhanVien
            WHERE
                NV.TrangThai = 1
            ORDER BY
                NV.HoVaTen
        """
        # ===== KẾT THÚC SỬA SQL =====

        cur.execute(sql)
        rows = cur.fetchall()

        for row in rows:
            driver_id_to_name[row[0]] = row[1]
            driver_name_to_id[row[1]] = row[0]

    finally:
        conn.close()

```

```

for row in rows:
    # Dùng tên cột đúng từ file SQL của bạn
    ma_nv = row.MaNhanVien
    ten_nv = row.HoVaTen

    if ma_nv and ten_nv:
        driver_id_to_name[ma_nv] = ten_nv
        driver_name_to_id[ten_nv] = ma_nv
        driver_display_list.append(ten_nv)

except Exception as e:
    messagebox.showerror("Lỗi SQL (Tài Xế)", f"Không thể tải danh sách tài xế:\n{e}")
finally:
    if conn:
        conn.close()

return driver_id_to_name, driver_name_to_id, driver_display_list

```

- Hàm quản lý và sự kiện

```

def set_form_state(is_enabled, widgets):
    """Bật (enable) hoặc Tắt (disable) toàn bộ các trường nhập liệu."""
    entry_state = "normal" if is_enabled else "disabled"
    cbb_state = "readonly" if is_enabled else "disabled" # Combobox dùng 'readonly'

    widgets["entry_loaixe"].config(state=entry_state)
    widgets["entry_hangsx"].config(state=entry_state)
    widgets["entry_dongxe"].config(state=entry_state)
    widgets["entry_namsx"].config(state=entry_state)
    widgets["entry_vin"].config(state=entry_state)
    widgets["date_dangkiem"].config(state=entry_state)
    widgets["date_baohiem"].config(state=entry_state)

    widgets["cbb_manv"].config(state=cbb_state)
    widgets["cbb_trangthai"].config(state=cbb_state)

    # Ô Biển số luôn bị khóa khi TẮT (is_enabled=False)
    if not is_enabled:
        widgets["entry_bienso"].config(state="disabled")

def clear_input(widgets):
    """(NÚT THÊM) Xóa trắng và Mở khóa các trường nhập liệu (Chế độ Thêm mới)."""
    set_form_state(is_enabled=True, widgets=widgets)

```

```

widgets["entry_bienso"].config(state="normal") # Mở khóa biến số

widgets["entry_bienso"].delete(0, tk.END)
widgets["entry_loaixe"].delete(0, tk.END)
widgets["entry_hangsx"].delete(0, tk.END)
widgets["entry_dongxe"].delete(0, tk.END)
widgets["entry_namsx"].delete(0, tk.END)
widgets["entry_vin"].delete(0, tk.END)
widgets["cbb_manv"].set("") # Reset combobox tài xế

default_date = datetime.now() + dt.timedelta(days=365)
widgets["date_dangkiem"].set_date(default_date)
widgets["date_baohiem"].set_date(default_date)
widgets["cbb_trangthai"].set("Hoạt động")

widgets["entry_bienso"].focus()

# Bỏ chọn item trên Treeview
tree = widgets["tree"]
if tree.selection():
    tree.selection_remove(tree.selection()[0])

widgets["current_mode"] = "ADD"

def on_item_select(event, widgets):
    """(SỰ KIỆN CLICK) Khi click vào Treeview, đổ dữ liệu đầy đủ lên
    form (ở trạng thái mờ)."""
    tree = widgets["tree"]
    selected = tree.selection()
    if not selected: return

    selected_item = tree.item(selected[0])
    bienso = selected_item["values"][0]

    conn = utils.connect_db()
    if conn is None: return

    try:
        cur = conn.cursor()
        cur.execute("SELECT * FROM Xe WHERE BienSoXe=?", (bienso,))
        data = cur.fetchone()
        if not data:
            messagebox.showerror("Lỗi", "Không tìm thấy dữ liệu xe.")
            return

        # 1. Mở khóa form để điền
        set_form_state(is_enabled=True, widgets=widgets)

        # 2. Xóa và điền dữ liệu
        widgets["entry_bienso"].config(state="normal")

```

```

        widgets["entry_bienso"].delete(0, tk.END)
        widgets["entry_bienso"].insert(0, data.BienSoXe)

        widgets["entry_loaixe"].delete(0, tk.END)
        widgets["entry_hangsx"].delete(0, tk.END)
        widgets["entry_dongxe"].delete(0, tk.END)
        widgets["entry_namsx"].delete(0, tk.END)
        widgets["entry_vin"].delete(0, tk.END)

        widgets["entry_loaixe"].insert(0, data.LoaiXe or "")
        widgets["entry_hangsx"].insert(0, data.HangSanXuat or "")
        widgets["entry_dongxe"].insert(0, data.DongXe or "")
        widgets["entry_namsx"].insert(0, str(data.NamSanXuat or ""))
        widgets["entry_vin"].insert(0, data.SoKhungVIN or "")

    # Lấy Tên tài xế từ ID
    driver_id_to_name = widgets["driver_id_to_name"]
    manv_id = data.MaNhanVienHienTai # Vd: 'NV002'
    driver_name = driver_id_to_name.get(manv_id, "") # Lấy tên,
nếu không có thì trả về rỗng
    widgets["cbb_manv"].set(driver_name)

    if data.NgayHetHanDangKiem:
        widgets["date_dangkiem"].set_date(data.NgayHetHanDangKiem)
    if data.NgayHetHanBaoHiem:
        widgets["date_baohiem"].set_date(data.NgayHetHanBaoHiem)

    trang_thai_text = "Hoạt động"
    if data.TrangThai == 0: trang_thai_text = "Bảo trì"
    elif data.TrangThai == 2: trang_thai_text = "Hỗng"
    widgets["cbb_trangthai"].set(trang_thai_text)

except Exception as e:
    messagebox.showerror("Lỗi không xác định", f'Lỗi: {str(e)}')
finally:
    if conn: conn.close()

    # 3. Khóa form lại (chế độ View)
    set_form_state(is_enabled=False, widgets=widgets)
    widgets["current_mode"] = "VIEW"

def chon_xe_de_sua(widgets):
    """(NÚT SỬA) Kích hoạt chế độ sửa, Mở khóa form (trừ Biển số)."""
    selected = widgets["tree"].selection()
    if not selected:
        messagebox.showwarning(
            "Chưa chọn", "Hãy chọn một xe trong danh sách trước khi
nhấn 'Sửa'"
        )
    return

```

```

if not widgets["entry_bienso"].get():
    messagebox.showwarning("Lỗi", "Không tìm thấy biển số xe. Vui lòng chọn lại.")
    return

set_form_state(is_enabled=True, widgets=widgets)
widgets["entry_bienso"].config(state="disabled") # Không cho sửa khóa chính
widgets["entry_loaixe"].focus()
widgets["current_mode"] = "EDIT"

```

### - Hàm logic CRUD

```

def load_data(widgets):
    """(READ) Tải TOÀN BỘ dữ liệu Xe VÀ LÀM MỜ FORM."""
    tree = widgets["tree"]
    for i in tree.get_children():
        tree.delete(i)

    conn = utils.connect_db()
    if conn is None:
        set_form_state(is_enabled=False, widgets=widgets)
        return

    try:
        cur = conn.cursor()
        cur.execute(
            "SELECT BienSoXe, LoaiXe, HangSanXuat, NamSanXuat,
TrangThai, NgayHetHanDangKiem FROM Xe"
        )
        rows = cur.fetchall()

        for row in rows:
            trang_thai_text = "Hoạt động"
            if row[4] == 0:
                trang_thai_text = "Bảo trì"
            elif row[4] == 2:
                trang_thai_text = "Hỏng"
            ngay_dk = str(row[5]) if row[5] else "N/A"
            tree.insert(
                "",
                tk.END,
                values=(row[0], row[1], row[2], row[3],
trang_thai_text, ngay_dk),
            )

    children = tree.get_children()

```

```

        if children:
            # Tự động chọn dòng đầu tiên
            first_item = children[0]
            tree.selection_set(first_item)
            tree.focus(first_item)
            tree.event_generate("<<TreeviewSelect>>")
        else:
            # Nếu không có xe nào, bật chế độ Thêm
            clear_input(widgets)

    except Exception as e:
        messagebox.showerror("Lỗi tải dữ liệu", f"Lỗi SQL: {str(e)}")
    finally:
        if conn:
            conn.close()

        # Nếu có dữ liệu, khóa form lại
        if tree.get_children():
            set_form_state(is_enabled=False, widgets=widgets)
            widgets["current_mode"] = "VIEW"

def them_xe(widgets):
    """(CREATE) Thêm một xe mới vào CSDL."""
    bienso = widgets["entry_bienso"].get()
    loaixe = widgets["entry_loaixe"].get()
    hangsx = widgets["entry_hangsx"].get()
    dongxe = widgets["entry_dongxe"].get()
    namsx = widgets["entry_namsx"].get()
    vin = widgets["entry_vin"].get()
    ngay_dk = widgets["date_dangkiem"].get_date().strftime("%Y-%m-%d")
    ngay_bh = widgets["date_baohiem"].get_date().strftime("%Y-%m-%d")
    trangthai_text = widgets["cbb_trangthai_var"].get()

    driver_name_to_id = widgets["driver_name_to_id"]
    selected_name = widgets["cbb_manv_var"].get()
    manv = driver_name_to_id.get(selected_name, None)

    trangthai_value = 1 # Hoạt động
    if trangthai_text == "Bảo trì": trangthai_value = 0
    elif trangthai_text == "Hỗn": trangthai_value = 2

    if not bienso or not loaixe:
        messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập Biển số xe và Loại xe")
        return False

    try:
        namsx_int = int(namsx) if namsx else None
    except ValueError:

```

```

        messagebox.showwarning("Sai định dạng", "Năm sản xuất phải là
số")
        return False

    conn = utils.connect_db()
    if conn is None: return False

    try:
        cur = conn.cursor()
        sql = """
        INSERT INTO Xe (
            BienSoXe, LoaiXe, HangSanXuat, DongXe, NamSanXuat,
            SoKhungVIN, NgayHetHanDangKiem, NgayHetHanBaoHiem,
            TrangThai, MaNhanVienHienTai
        ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
        """
        cur.execute(
            sql,
            (
                bienso, loaixe, hangsx, dongxe, namsx_int,
                vin, ngay_dk, ngay_bh, trangthai_value, manv,
            ),
        )
        conn.commit()
        messagebox.showinfo("Thành công", "Đã thêm xe mới thành công")
        return True
    except pyodbc.IntegrityError:
        conn.rollback()
        messagebox.showerror("Lỗi Trùng lặp", f"Biển số xe '{bienso}'"
đã tồn tại.")
        return False
    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể thêm
xe:\n{str(e)}")
        return False
    finally:
        if conn: conn.close()

def luu_xe_da_sua(widgets):
    """(UPDATE) Lưu thay đổi sau khi sửa."""
    bienso = widgets["entry_bienso"].get()
    if not bienso:
        messagebox.showwarning("Thiếu dữ liệu", "Không có Biển số xe
để cập nhật")
        return False

    # Lấy dữ liệu từ form
    loaixe = widgets["entry_loaixe"].get()
    hangsx = widgets["entry_hangsx"].get()

```

```

dongxe = widgets["entry_dongxe"].get()
namsx = widgets["entry_namsx"].get()
vin = widgets["entry_vin"].get()
ngay_dk = widgets["date_dangkiem"].get_date().strftime("%Y-%m-%d")
ngay_bh = widgets["date_baokiem"].get_date().strftime("%Y-%m-%d")
trangthai_text = widgets["cbb_trangthai_var"].get()

driver_name_to_id = widgets["driver_name_to_id"]
selected_name = widgets["cbb_manv_var"].get()
manv = driver_name_to_id.get(selected_name, None)

trangthai_value = 1
if trangthai_text == "Bảo trì": trangthai_value = 0
elif trangthai_text == "Hồng": trangthai_value = 2

if not loaxe:
    messagebox.showwarning("Thiếu dữ liệu", "Loại xe không được
rõng")
    return False

try:
    namsx_int = int(namsx) if namsx else None
except ValueError:
    messagebox.showwarning("Sai định dạng", "Năm sản xuất phải là
số")
    return False

conn = utils.connect_db()
if conn is None: return False

try:
    cur = conn.cursor()
    sql = """
        UPDATE Xe SET
            LoaiXe = ?, HangSanXuat = ?, DongXe = ?, NamSanXuat = ?,
            SoKhungVIN = ?, NgayHetHanDangKiem = ?, NgayHetHanBaoHiem
            = ?,
            TrangThai = ?, MaNhanVienHienTai = ?
        WHERE BienSoXe = ?
    """
    cur.execute(
        sql,
        (
            loaxe, hangsx, dongxe, namsx_int, vin, ngay_dk,
            ngay_bh, trangthai_value, manv,
            bienso, # Where clause
        ),
    )
    conn.commit()
    messagebox.showinfo("Thành công", "Đã cập nhật thông tin xe")

```

```

        return True
    except Exception as e:
        conn.rollback()
        messagebox.showerror("Lỗi SQL", f"Không thể cập nhật
xe:\n{str(e)}")
        return False
    finally:
        if conn: conn.close()

def save_data(widgets):
    """(NÚT LƯU) Tự động kiểm tra xem nên Thêm mới (INSERT) hay Cập
nhật (UPDATE)."""
    success = False

    if widgets["current_mode"] == "EDIT":
        success = luu_xe_da_sua(widgets)
    elif widgets["current_mode"] == "ADD":
        success = them_xe(widgets)
    else:
        messagebox.showwarning("Chưa Sửa/Thêm", "Vui lòng nhấn 'Thêm'
hoặc 'Sửa' trước khi Lưu.")
        return

    # Nếu Lưu thành công (dù là Thêm hay Sửa), tải lại dữ liệu
    if success:
        load_data(widgets)

def xoa_xe_vinhvien(widgets):
    """(DELETE) Xóa vĩnh viễn xe và MỌI DỮ LIỆU LIÊN QUAN."""
    selected = widgets["tree"].selection()
    if not selected:
        messagebox.showwarning("Chưa chọn", "Hãy chọn một xe trong
danh sách để xóa")
        return

    bienso = widgets["entry_bienso"].get()
    if not bienso:
        messagebox.showwarning("Lỗi", "Không tìm thấy biển số xe. Vui
lòng chọn lại.")
        return

    msg_xacnhan = (
        f"BẠN CÓ CHẮC CHẮN MUỐN XÓA VĨNH VIỄN XE '{bienso}'?\n\n"
        "CẢNH BÁO: Thao tác này KHÔNG THỂ hoàn tác.\n"
        "Tất cả Lịch sử chuyến đi, Lịch sử nhiên liệu, và Lịch sử bảo
trì của xe này sẽ bị XÓA SẠCH."
    )

    if not messagebox.askyesno("XÁC NHẬN XÓA VĨNH VIỄN", msg_xacnhan):
        return

```

```

conn = utils.connect_db()
if conn is None: return

try:
    cur = conn.cursor()
    # Xóa dữ liệu ở các bảng con trước (bảng có khóa ngoại)
    cur.execute("DELETE FROM NhatKyNhienLieu WHERE BienSoXe=?",
(bienso,))
    cur.execute("DELETE FROM LichSuBaoTri WHERE BienSoXe=?",
(bienso,))
    cur.execute("DELETE FROM ChuyenDi WHERE BienSoXe=?",
(bienso,))

    # Xóa dữ liệu ở bảng cha (bảng Xe)
    cur.execute("DELETE FROM Xe WHERE BienSoXe=?",
(bienso,))

    conn.commit()
    messagebox.showinfo(
        "Thành công", f"Đã xóa vĩnh viễn xe '{bienso}' và tất cả
dữ liệu liên quan."
    )
    load_data(widgets)
except Exception as e:
    conn.rollback()
    messagebox.showerror("Lỗi SQL", f"Không thể xóa xe:\n{n{str(e)}}")
finally:
    if conn: conn.close()

```

### - Hàm tạo trang

```

def create_page(master):
    """
    Hàm này được main.py gọi.
    Nó tạo ra toàn bộ nội dung trang và đặt vào 'master' (là
    main_frame).
    """

    page_frame = ttk.Frame(master, style="TFrame")
    utils.setup_theme(page_frame)

    lbl_title = ttk.Label(page_frame, text="QUẢN LÝ XE",
style="Title TLabel")
    lbl_title.pack(pady=15)

    frame_info = ttk.Frame(page_frame, style="TFrame")

```

```

frame_info.pack(pady=5, padx=20, fill="x")

# --- Cột 1 ---
ttk.Label(frame_info, text="Biển số xe:").grid(row=0, column=0,
padx=5, pady=8, sticky="w")
entry_bienso = ttk.Entry(frame_info, width=30)
entry_bienso.grid(row=0, column=1, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Loại xe:").grid(row=1, column=0,
padx=5, pady=8, sticky="w")
entry_loaixe = ttk.Entry(frame_info, width=30)
entry_loaixe.grid(row=1, column=1, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Hãng sản xuất:").grid(row=2, column=0,
padx=5, pady=8, sticky="w")
entry_hangsx = ttk.Entry(frame_info, width=30)
entry_hangsx.grid(row=2, column=1, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Đòng xe:").grid(row=3, column=0,
padx=5, pady=8, sticky="w")
entry_dongxe = ttk.Entry(frame_info, width=30)
entry_dongxe.grid(row=3, column=1, padx=5, pady=8, sticky="w")

# ===== NÂNG CẤP =====
ttk.Label(frame_info, text="Tên người lái:").grid(row=4, column=0,
padx=5, pady=8, sticky="w")

# Tải dữ liệu tài xế bằng hàm helper
driver_id_to_name, driver_name_to_id, driver_display_list =
_load_driver_data()

cbb_manv_var = tk.StringVar()
cbb_manv = ttk.Combobox(
    frame_info,
    textvariable=cbb_manv_var,
    values=driver_display_list, # Dùng danh sách tên
    width=28,
    state="readonly"
)
cbb_manv.grid(row=4, column=1, padx=5, pady=8, sticky="w")
# ===== KẾT THÚC NÂNG CẤP =====

# --- Cột 2 ---
ttk.Label(frame_info, text="Năm sản xuất:").grid(row=0, column=2,
padx=15, pady=8, sticky="w")

```

```

entry_namsx = ttk.Entry(frame_info, width=30)
entry_namsx.grid(row=0, column=3, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Số khung (VIN):").grid(row=1, column=2,
padx=15, pady=8, sticky="w")
entry_vin = ttk.Entry(frame_info, width=30)
entry_vin.grid(row=1, column=3, padx=5, pady=8, sticky="w")

date_entry_style_options = {
    "width": 28, "date_pattern": "yyyy-MM-dd",
    "background": utils.theme_colors["bg_entry"],
    "foreground": utils.theme_colors["text"],
    "disabledbackground": utils.theme_colors["disabled_bg"],
    "disabledforeground": utils.theme_colors["text_disabled"],
    "bordercolor": utils.theme_colors["bg_entry"],
    "headersbackground": utils.theme_colors["accent"],
    "headersforeground": utils.theme_colors["accent_text"],
    "selectbackground": utils.theme_colors["accent"],
    "selectforeground": utils.theme_colors["accent_text"],
}
}

ttk.Label(frame_info, text="Ngày hết hạn ĐK:").grid(row=2,
column=2, padx=15, pady=8, sticky="w")
date_dangkiem = DateEntry(frame_info, **date_entry_style_options)
date_dangkiem.grid(row=2, column=3, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Ngày hết hạn BH:").grid(row=3,
column=2, padx=15, pady=8, sticky="w")
date_baothi = DateEntry(frame_info, **date_entry_style_options)
date_baothi.grid(row=3, column=3, padx=5, pady=8, sticky="w")

ttk.Label(frame_info, text="Trạng thái:").grid(row=4, column=2,
padx=15, pady=8, sticky="w")
trangthai_options = ["Bảo trì", "Hoạt động", "Hỏng"]
cbb_trangthai_var = tk.StringVar()
cbb_trangthai = ttk.Combobox(
    frame_info, textvariable=cbb_trangthai_var,
    values=trangthai_options,
    width=28, state="readonly",
)
cbb_trangthai.grid(row=4, column=3, padx=5, pady=8, sticky="w")
cbb_trangthai.set("Hoạt động")

frame_info.columnconfigure(1, weight=1)

```

```

frame_info.columnconfigure(3, weight=1)

# ===== Frame nút =====
frame_btn = ttk.Frame(page_frame, style="TFrame")
frame_btn.pack(pady=10)

# ===== Bảng danh sách xe =====
lbl_ds = ttk.Label(page_frame, text="Danh sách xe",
style="Header.TLabel")
lbl_ds.pack(pady=(10, 5), padx=20, anchor="w")

frame_tree = ttk.Frame(page_frame, style="TFrame")
frame_tree.pack(pady=10, padx=20, fill="both", expand=True)

scrollbar_y = ttk.Scrollbar(frame_tree, orient=tk.VERTICAL,
style="Vertical.TScrollbar")
scrollbar_x = ttk.Scrollbar(frame_tree, orient=tk.HORIZONTAL,
style="Horizontal.TScrollbar")

columns = ("bienso", "loaixe", "hangsx", "namsx", "trangthai",
"ngay_dk")
tree = ttk.Treeview(
    frame_tree, columns=columns, show="headings", height=10,
    yscrollcommand=scrollbar_y.set, xscrollcommand=scrollbar_x.set,
)
scrollbar_y.config(command=tree.yview)
scrollbar_x.config(command=tree.xview)
scrollbar_y.pack(side=tk.RIGHT, fill=tk.Y)
scrollbar_x.pack(side=tk.BOTTOM, fill=tk.X)

tree.heading("bienso", text="Biển số xe")
tree.column("bienso", width=100, anchor="center")
tree.heading("loaixe", text="Loại xe")
tree.column("loaixe", width=150)
tree.heading("hangsx", text="Hãng sản xuất")
tree.column("hangsx", width=150)
tree.heading("namsx", text="Năm SX")
tree.column("namsx", width=80, anchor="center")
tree.heading("trangthai", text="Trạng thái")
tree.column("trangthai", width=100, anchor="center")
tree.heading("ngay_dk", text="Ngày hết hạn ĐK")
tree.column("ngay_dk", width=150, anchor="center")

```

```

tree.pack(fill="both", expand=True)

# 3. TẠO TỪ ĐIỂN 'widgets'
widgets = {
    "tree": tree,
    "entry_bienso": entry_bienso,
    "entry_loaixe": entry_loaixe,
    "entry_hangsx": entry_hangsx,
    "entry_dongxe": entry_dongxe,
    "entry_namsx": entry_namsx,
    "entry_vin": entry_vin,
    "date_dangkiem": date_dangkiem,
    "date_baohiem": date_baohiem,
    "cbb_trangthai": cbb_trangthai,
    "cbb_trangthai_var": cbb_trangthai_var,
    "current_mode": "VIEW",
}

# Thêm các widget và map mới
"cbb_manv": cbb_manv,
"cbb_manv_var": cbb_manv_var,
"driver_id_to_name": driver_id_to_name,
"driver_name_to_id": driver_name_to_id
}

# (Code tạo nút)
btn_them = ttk.Button(frame_btn, text="Thêm", width=8,
command=lambda: clear_input(widgets))
btn_them.grid(row=0, column=0, padx=10)
btn_luu = ttk.Button(frame_btn, text="Lưu", width=8,
command=lambda: save_data(widgets))
btn_luu.grid(row=0, column=1, padx=10)
btn_sua = ttk.Button(frame_btn, text="Sửa", width=8,
command=lambda: chon_xe_de_sua(widgets))
btn_sua.grid(row=0, column=2, padx=10)
btn_huy = ttk.Button(frame_btn, text="Hủy", width=8,
command=lambda: load_data(widgets))
btn_huy.grid(row=0, column=3, padx=10)
btn_xoa = ttk.Button(frame_btn, text="Xóa", width=8,
command=lambda: xoa_xe_vinhvien(widgets))
btn_xoa.grid(row=0, column=4, padx=10)

# 4. KẾT NỐI BINDING
tree.bind("<>TreeviewSelect>>", lambda event: on_item_select(event,
widgets))

# 5. TẢI DỮ LIỆU LẦN ĐẦU

```

```
load_data(widgets)

# 6. TRẢ VỀ FRAME CHÍNH
return page_frame
```

## **V. TÀI LIỆU THAM KHẢO**

YeuCauDoAn\_Python\_nnminh.pdf (Ths. Nguyễn Ngọc Minh (BM.02/09/2025)).

Giáo Trình LẬP TRÌNH PYTHON (Ths.Huỳnh Phước Hải (Chủ biên),  
Ths.Nguyễn Văn Hòa, 49/QĐ-NXB cấp ngày 13/4/2023)

<https://timoday.edu.vn/xay-dung-chuong-trinh-quan-ly-ban-hang-bang-c/>