

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



**BÁO CÁO ĐỒ ÁN**  
***Chuyên đề Python(COS525)***

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

Giảng viên hướng dẫn: Th.S Nguyễn Ngọc Minh

Sinh viên thực hiện: Trần Triệu Phú DTH235734  
Đoàn Minh Trí DTH235794

An Giang, tháng 11 năm 2025

# NỘI DUNG

## 1. Đặt vấn đề

### Tính cần thiết của đề tài Mục tiêu tổng quát

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ, việc ứng dụng công nghệ thông tin vào quản lý nhân sự là nhu cầu thiết yếu của các cơ quan, tổ chức và doanh nghiệp.

Hiện nay, nhiều đơn vị vẫn quản lý thông tin nhân viên bằng các phương pháp thủ công hoặc qua bảng tính Excel, gây khó khăn trong việc tra cứu, thống kê và đảm bảo tính chính xác của dữ liệu.

Vì vậy, việc xây dựng phần mềm quản lý nhân sự bằng ngôn ngữ Python là rất cần thiết, giúp tự động hóa các quy trình quản lý, giảm thiểu sai sót, tiết kiệm thời gian và nâng cao hiệu quả công việc. Ứng dụng “Quản lý nhân sự” có thể được sử dụng tại các doanh nghiệp nhỏ, trung tâm hoặc trường học để quản lý thông tin nhân viên một cách khoa học và hiệu quả. Ngoài ra đề tài giúp sinh viên củng cố và vận dụng các kiến thức đã học về lập trình Python, lập trình hướng đối tượng, làm việc với cơ sở dữ liệu và thiết kế giao diện bằng thư viện Tkinter, Python, MySQL.

### Mục tiêu tổng quát Mục tiêu cụ thể

Xây dựng một ứng dụng quản lý nhân sự bằng ngôn ngữ Python với giao diện thân thiện, dễ sử dụng, hỗ trợ người dùng thực hiện các nghiệp vụ quản lý nhân viên một cách nhanh chóng, chính xác và thuận tiện. Ứng dụng cho phép thực hiện các thao tác thêm, sửa, xóa, tìm kiếm và thống kê thông tin nhân viên, đáp ứng nhu cầu quản lý tại các doanh nghiệp vừa và nhỏ.

### Mục tiêu cụ thể

1. Thiết kế giao diện người dùng (GUI) bằng thư viện Tkinter, giúp người dùng có thể thao tác trực tiếp với chương trình một cách dễ dàng và trực quan.
2. Xây dựng cơ sở dữ liệu bằng SQLite3 để lưu trữ thông tin nhân viên một cách khoa học, an toàn và bền vững.
3. Cài đặt các chức năng chính của hệ thống, bao gồm:
  - Thêm mới nhân viên.
  - Sửa thông tin nhân viên.
  - Xóa nhân viên khỏi danh sách.
  - Tìm kiếm nhân viên theo các tiêu chí (mã, tên, phòng ban, chức vụ, ...).
  - Thống kê nhân viên theo nhiều tiêu chí khác nhau (tổng số, lương trung bình, theo phòng ban,...).
4. Đảm bảo tính ổn định, linh hoạt và dễ mở rộng của chương trình, có thể phát triển thêm các chức năng nâng cao như: đăng nhập, phân quyền người dùng, xuất báo cáo hoặc in danh sách nhân viên.

## 2. Tổng Quan cơ sở lý thuyết

### 2.1 Đặt vấn đề

Trong bất kỳ tổ chức, cơ quan hay doanh nghiệp nào, nhân sự luôn là yếu tố quan trọng quyết định sự phát triển bền vững. Tuy nhiên, khi số lượng nhân viên tăng lên, việc quản lý thông tin nhân viên bằng sổ sách hoặc file Excel trở nên rườm rà, tốn thời gian và dễ xảy ra sai sót. Việc tra cứu, chỉnh sửa, hoặc thống kê dữ liệu nhân viên gặp nhiều khó khăn, đặc biệt khi dữ liệu lớn hoặc cần cập nhật thường xuyên. Thực tế cho thấy, nhiều doanh nghiệp vừa và nhỏ hiện nay chưa có điều kiện áp dụng các hệ thống quản lý nhân sự chuyên nghiệp do chi phí phần mềm cao, yêu cầu cài đặt phức tạp, hoặc không phù hợp với quy mô tổ chức. Điều này dẫn đến việc hiệu suất làm việc giảm, khó kiểm soát thông tin nhân viên, và mất nhiều thời gian cho công tác hành chính.

Từ những vấn đề nêu trên, nhóm em nhận thấy việc xây dựng một phần mềm quản lý nhân sự nhỏ gọn, dễ sử dụng, thân thiện với người dùng là rất cần thiết. Đề tài “Phần mềm quản lý nhân sự” được thực hiện với mục tiêu:

- Giúp lưu trữ và quản lý thông tin nhân viên một cách khoa học, chính xác.
- Hỗ trợ người quản lý dễ dàng thực hiện các thao tác thêm, sửa, xóa, tìm kiếm nhân viên.
- Ứng dụng công nghệ lập trình Python để xây dựng hệ thống quản lý dữ liệu hiệu quả, dễ mở rộng và bảo trì.

### 2.2 Lịch sử giải quyết vấn đề:

Tại doanh nghiệp DKC, việc quản lý nhân sự vẫn đang sử dụng phương pháp thủ công là chủ yếu. Việc cập nhật hồ sơ, tìm kiếm thông tin của một nhân viên là rất khó khăn, mất nhiều thời gian và không cho được thông tin chính xác và đầy đủ nhất. Số giấy tờ bổ sung hàng năm trong hồ sơ Quản lý nhân sự ngày càng nhiều. Việc làm các báo thống kê theo yêu cầu mang tính điều tra chuyên đề thì càng gặp khó khăn hơn rất nhiều và thường không thỏa mãn được yêu cầu đặt ra.

Thực tế cho thấy việc Quản lý nhân sự bằng sổ sách có những ưu điểm và khuyết điểm sau:

**Ưu điểm:** Quản lý bằng sổ sách nên nhà quản lý không đòi hỏi có chuyên môn cao, vẫn quản lý được các nhân sự của mình.

**Nhược điểm:** Việc lưu trữ trên giấy tờ rất cồng kềnh, tốn kém mà dễ bị thất lạc, hư tổn. Việc cập nhật kịp thời các nhân viên khi chuyển công tác hay sang các phòng ban khác được thực hiện tốn thời gian và không chính xác. Tốn kém nhiều nhân lực, tiền bạc và thời gian đồng thời cũng làm giảm năng suất lao động của từng cá nhân trong doanh nghiệp.

- Do vậy, yêu cầu cần phải xây dựng một hệ thống Quản lý nhân sự nhằm khắc phục những nhược điểm trên.

\* Đặc biệt là 15 năm gần đây cũng có các công ty phát triển phần mềm quản lý nhân sự nhưng chưa được phát triển rộng rãi, phù hợp với đa số người dùng VD: Phần mềm MISA HRM. Đơn vị phát triển: Công ty Cổ phần MISA – Việt Nam. Thời gian ra mắt: Khoảng từ năm 2010, liên tục được nâng cấp và mở rộng tính năng cho đến nay. Địa điểm triển khai: Được sử dụng rộng rãi tại nhiều doanh nghiệp, cơ quan hành chính và tổ chức giáo dục trong cả nước. Kết quả đạt được: Hỗ trợ tốt cho công tác quản lý nhân sự, chấm công, tính lương, quản lý hợp đồng, đánh giá nhân viên và lập báo cáo tự động. Giao diện thân thiện, dễ sử dụng, có khả năng tích hợp với các phần mềm kế toán MISA khác. Hạn chế còn tồn tại: Là phần mềm thương mại, chi phí bản quyền cao; yêu cầu cấu hình máy tính tương đối lớn; khó triển khai đối với doanh nghiệp nhỏ hoặc người dùng cá nhân chỉ cần hệ thống đơn giản.

### **2.3 Phạm vi của đề tài:**

- Quản lý sơ yếu lý lịch với đầy đủ thông tin như: Họ tên, bí danh, ngày sinh, nơi sinh, dân tộc, tôn giáo, số thẻ công ty, số CMND, quê quán, hộ khẩu thường trú, nơi ở hiện tại, điện thoại liên hệ ...
- Quản lý thành viên và trình độ cán bộ với các thông tin như: Thông tin bản thân và người trong gia đình(ba mẹ, anh chị em), trình độ văn hóa và trình độ chuyên môn ngoại ngữ...
- Quản lý kinh nghiệm và uy tín công tác với thông tin như: Phẩm chất đạo đức, uy tín, trách nhiệm, tính kỷ luật và chuyên môn...
- Quản lý các thông tin về quá trình là đoàn thanh viên Thanh Niên CSHCM, phục vụ trong đội, tình trạng sức khỏe là thương binh...
- Quản lý diễn biến trình độ đào tạo như là trình độ học vấn, trình độ ngoại ngữ...
- Quản lý các diễn biến công tác: Quá trình trước khi tuyển dụng, quá trình điều động, bổ nhiệm hoặc chuyển chuyển giữa các đơn vị, bộ phận, theo dõi quá trình thực hiện kí kết hợp đồng lao động, quá trình nghỉ phép (tai nạn, nghỉ thai sản, ...), quá trình được cử đi đào tạo, quá trình đi công tác trong và ngoài nước, quá trình khen thưởng và kỉ luật
- Quản lý chấm công là quản lý ngày công làm việc, phân ca làm việc, hỗ trợ chấm công theo ngày, tháng...nghỉ phép, nghỉ thai sản, tăng ca...Hỗ trợ chấm công bằng thẻ từ hoặc vân tay, tự động cập nhập ngày phép trong năm.

### **2.4 Phương pháp nghiên cứu / Hướng giải quyết vấn đề**

- Phương pháp lý thuyết: Tìm hiểu các tài liệu về quản lý nhân sự, hệ thống thông tin quản lý, mô hình CSDL quan hệ (ERD), DFD.
- Phương pháp thực nghiệm: Phân tích yêu cầu, thiết kế và lập trình thử nghiệm phần mềm bằng công nghệ web (Tkinter,HTML, CSS, JS, PHP/MySQL hoặc framework tương đương).
- Phương pháp khảo sát: Thu thập ý kiến người dùng (nhân viên, quản lý nhân sự) về nhu cầu sử dụng phần mềm, từ đó điều chỉnh chức năng phù hợp.
- Phương pháp đánh giá: Kiểm thử chương trình, đối chiếu kết quả thực tế với yêu cầu ban đầu để hoàn thiện hệ thống.

## 2.5 Cơ sở lý thuyết về việc quản lý nhân sự.

1. Lý thuyết quản lý nhân sự (HRM): Cung cấp nền tảng về các thông tin cần quản lý như hồ sơ nhân viên, quá trình công tác, chấm công, khen thưởng – kỷ luật và các nghiệp vụ quản lý cơ bản. Đây là cơ sở để xác định chức năng của phần mềm.
2. Hệ thống thông tin quản lý (MIS): Giải thích vai trò của phần mềm trong việc thu thập, lưu trữ, xử lý và cung cấp thông tin nhân sự. Các mô hình như DFD và ERD được sử dụng để phân tích luồng thông tin và cấu trúc dữ liệu.
3. Cơ sở dữ liệu quan hệ: Sử dụng các khái niệm bảng, khóa chính, khóa ngoại, chuẩn hóa dữ liệu và truy vấn SQL. SQLite3 hoặc MySQL được áp dụng để lưu trữ dữ liệu nhân viên một cách khoa học và ổn định.
4. Lập trình hướng đối tượng (OOP): Các khái niệm lớp, đối tượng, kế thừa và đóng gói được dùng để xây dựng hệ thống có cấu trúc rõ ràng, dễ mở rộng và bảo trì.
5. Thiết kế giao diện người dùng (GUI): Dựa trên các nguyên tắc thiết kế giao diện trực quan và xử lý sự kiện. Tkinter được dùng để xây dựng giao diện thuận tiện cho thao tác thêm – sửa – xóa – tìm kiếm dữ liệu.
6. Tham khảo phần mềm HRM hiện có: Kế thừa quy trình quản lý hồ sơ nhân sự, chấm công, thống kê của các hệ thống như MISA HRM; đồng thời xây dựng phiên bản gọn nhẹ phù hợp với doanh nghiệp nhỏ và người dùng phổ thông.
7. Phần lý thuyết mới của đề tài: Xây dựng mô hình dữ liệu tối ưu, thuật toán tìm kiếm và thống kê linh hoạt, cùng giao diện thân thiện và khả năng mở rộng trong tương lai.

## 3. Kết Quả Đạt Được và Demo Code

### A. Demo Code (Minh họa Kỹ thuật)

Phần demo tập trung vào các đoạn mã minh họa việc vận dụng Python để giải quyết các vấn đề thực tế:

## 1. Kết nối và Đăng nhập An toàn (database.py & login\_view.py)

```
# database.py
import mysql.connector
from mysql.connector import Error

def get_connection():
    try:
        conn = mysql.connector.connect(
            host="localhost",
            user="root",
            password="",
            database="nhansu"
        )
        return conn
    except Error as e:
        print("❌ Lỗi kết nối CSDL:", e)
        return None

def dang_nhap(tendn, matkhou):
    conn = get_connection()
    if conn is None:
        return None

    try:
        cursor = conn.cursor(dictionary=True)
        query = "SELECT * FROM taikhoan WHERE TenDangNhap = %s AND MatKhou = %s"
        cursor.execute(query, (tendn, matkhou))
        user = cursor.fetchone()
        return user
    except Error as e:
        print("❌ Lỗi khi đăng nhập:", e)
        return None
    finally:
        cursor.close()
        conn.close()
```

## 2. Vận hành CRUD qua Tkinter (nhanvien.py)

```
# nhanvien.py
from tkinter import *
from tkinter import messagebox, ttk
from database import execute_query, fetch_all

def them_nv():
    ma = txt_ma.get().strip()
    ten = txt_ten.get().strip()
    gt = cbo_gt.get()
    ns = txt_ns.get().strip()
    cv = cbo_cv.get()

    if not ma or not ten or not ns:
        messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập đầy đủ thông tin")
        return

    query = """
    INSERT INTO nhanvien (MaNV, HoTen, GioiTinh, NgaySinh, ChucVu)
    VALUES (%s, %s, %s, %s, %s)
    """

    if execute_query(query, (ma, ten, gt, ns, cv)):
        messagebox.showinfo("Thành công", "Đã thêm nhân viên mới.")
        load_data()

btn_them = Button(frame_btn, text="Thêm", command=them_nv)
btn_them.pack(side="left")
```

## 3. Hiển thị Dữ liệu dạng Bảng (ttk.Treeview)

```
def load_data():
    tree.delete(*tree.get_children())

    query = """
    SELECT MaNV, HoTen, GioiTinh, NgaySinh, ChucVu
    FROM nhanvien
    """

    rows = fetch_all(query)
    for r in rows:
        tree.insert("", "end",
                    values=(r['MaNV'], r['HoTen'], r['GioiTinh'], r['NgaySinh'], r['ChucVu']))

load_data()
```

```

from database import fetch_all
from tkinter import ttk

tree = ttk.Treeview(win,
    columns=("MaNV", "HoTen", "GioiTinh", "NgaySinh", "ChucVu"),
    show="headings"
)

tree.heading("MaNV", text="Mã NV")
tree.heading("HoTen", text="Họ Tên")
tree.heading("GioiTinh", text="Giới Tính")
tree.heading("NgaySinh", text="Ngày Sinh")
tree.heading("ChucVu", text="Chức Vụ")

```

### B. Giao diện quản lý nhân sự



HRM System Login

## HUMAN RESOURCE MANAGEMENT

Tên đăng nhập  
admin


Mật khẩu  
.....

**ĐĂNG NHẬP**



HỆ THỐNG QUẢN LÝ NHÂN SỰ PRO

HRM SYSTEM



Admin

Chức vụ: Admin

> NHÂN VIÊN

PHÒNG BAN

CHỨC VỤ

HỢP ĐỒNG

LƯƠNG & THƯỞNG

ĐĂNG XUẤT

NHÂN VIÊN

Thông tin nhân viên

Mã NV (\*):

Họ tên (\*):

Ngày sinh:

2025-11-30

Phòng ban:

Chức vụ:

User/Pass (\*):

Làm mới

Thêm

Sửa


Xóa

Mã NV	Họ Tên	Phòng Ban	Chức Vụ	Tài khoản	Ngày Sinh
NV01	Nguyễn Văn Admin	Ban Giám Đốc	Tổng Giám Đốc	admin	1990-01-01
NV02	Trần Thị Thư Kỳ	Phòng Nhân Sự	Trưởng Phòng	thuky	1995-05-20
NV03	Lê Văn Code	Phòng IT - Kỹ Thuật	Nhân Viên Chính Thức	coder	1998-12-10
NV04	Phạm Thị Loan	Phòng Kế Toán	Trưởng Phòng	ketoan	1992-08-15
NV05	Hoàng Văn Sale	Phòng Kinh Doanh	Nhân Viên Chính Thức	sale01	1996-03-30
NV06	Đỗ Thị Hạnh	Phòng Nhân Sự	Nhân Viên Chính Thức	hanhdt	2000-01-01
NV07	Vũ Văn Hùng	Phòng IT - Kỹ Thuật	Phó Phòng	hungw	1993-07-22
NV08	Ngô Lan Anh	Phòng Kinh Doanh	Nhân Viên Chính Thức	lananh	1997-11-11
NV09	Bùi Đức Thắng	Phòng IT - Kỹ Thuật	Thực Tập Sinh	thangbd	1999-09-09
NV10	Lý Thị Mơ	Phòng Nhân Sự	Thực Tập Sinh	molt	2001-02-14
NV11	Đoàn Minh Trí	Phòng IT - Kỹ Thuật	Trưởng Phòng	tri	2005-01-21
NV12	Trần Triệu Phú	Phòng Kế Toán	Trưởng Phòng	phu	2005-04-07

Activate Windows  
Go to Settings to activate Windows.

HỆ THỐNG QUẢN LÝ NHÂN SỰ PRO

HRM SYSTEM



Admin

Chức vụ: Admin

> PHÒNG BAN

CHỨC VỤ

HỢP ĐỒNG

LƯƠNG & THƯỞNG

ĐĂNG XUẤT

PHÒNG BAN

QUẢN LÝ PHÒNG BAN

Mã PB (\*):

Tên PB (\*):

Địa chỉ:

SĐT:

THÊM

SỬA

XÓA

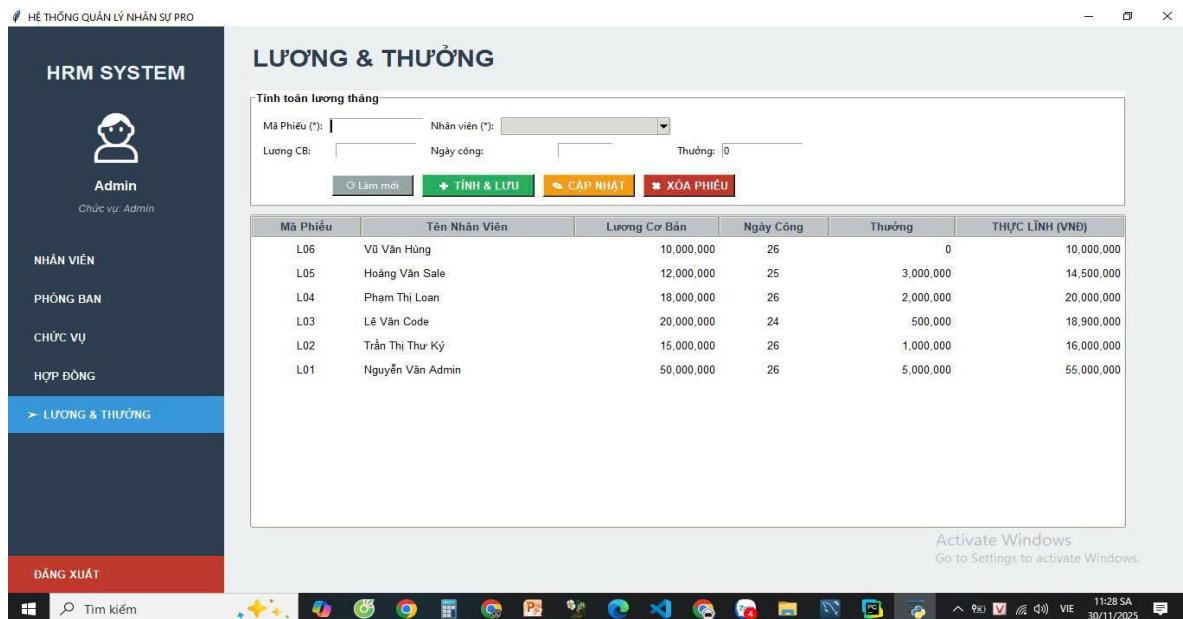
LÀM MỚI

Mã	Tên Phòng	Địa chỉ	SĐT
PB01	Ban Giám Đốc	Tầng 5	0901111111
PB02	Phòng Nhân Sự	Tầng 1	902222222
PB03	Phòng Kế Toán	Tầng 2	0903333333
PB04	Phòng IT - Kỹ Thuật	Tầng 3	0904444444
PB05	Phòng Kinh Doanh	Tầng 4	0905555555

Activate Windows  
Go to Settings to activate Windows.

8





## C. Các bước xây dựng ứng dụng

### 1. Cài đặt các Thư viện Cần thiết

Bước đầu tiên là thiết lập môi trường Python để có thể giao tiếp với MySQL và xây dựng giao diện đồ họa cho hệ thống quản lý nhân sự.

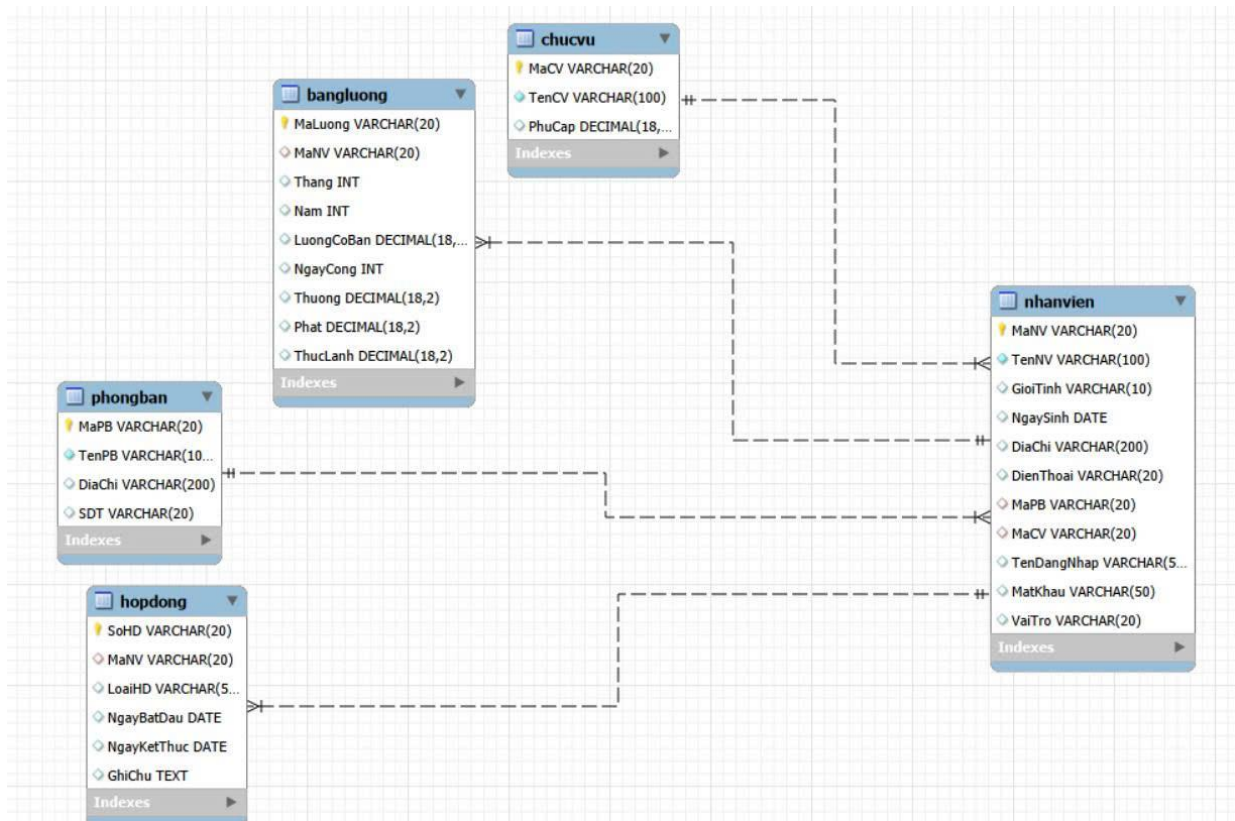
Python: Đảm bảo máy tính đã cài đặt Python 3.x.

MySQL Server: Cài đặt và khởi động dịch vụ MySQL để có thể lưu trữ thông tin nhân viên như hồ sơ cá nhân, phòng ban, chức vụ và hợp đồng lao động.

Thư viện Python (cài qua Terminal): mysql-connector-python: Thư viện tiêu chuẩn dùng để kết nối chương trình Python với MySQL Server, phục vụ các thao tác thêm, sửa, xóa, truy vấn dữ liệu nhân sự. VD: **pip install mysql-connector-python**

Tkcalendar: Thư viện mở rộng của Tkinter, giúp chọn ngày một cách trực quan, chẳng hạn như ngày sinh nhân viên, ngày ký hợp đồng, ngày vào làm. VD: **pip install tkcalendar**

### 2. Thiết kế cơ sở dữ liệu MySQL



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

Ứng dụng sử dụng **Tkinter** để tạo các cửa sổ chức năng, ô nhập liệu, nút thao tác và bảng Treeview nhằm quản lý toàn bộ dữ liệu nhân sự theo mô hình cơ sở dữ liệu đã thiết kế.

#### Các chức năng chính của giao diện gồm:

- **Quản lý Nhân viên:** Thêm, sửa, xóa và tìm kiếm nhân viên. Quản lý thông tin: mã NV, họ tên, giới tính, ngày sinh, địa chỉ, SĐT, phòng ban, chức vụ và tài khoản đăng nhập.
- **Quản lý Phòng ban:** Thêm, sửa, xóa phòng ban. Theo dõi thông tin phòng ban và danh sách nhân viên thuộc phòng ban đó.
- **Quản lý Chức vụ:** Quản lý tên chức vụ và phụ cấp. Dùng để phân công vị trí và phục vụ tính lương.
- **Quản lý Hợp đồng lao động:** Thêm và chỉnh sửa hợp đồng theo từng nhân viên. Theo dõi loại hợp đồng, ngày bắt đầu – kết thúc và ghi chú.
- **Quản lý Bảng lương:** Lưu và hiển thị lương theo tháng/năm: lương cơ bản, ngày công, thưởng, phạt và thực lãnh. Liên kết trực tiếp với chức vụ và thông tin nhân viên.
- **Thoát hệ thống:** Xác nhận trước khi thoát để tránh thao tác nhầm.

#### 4. Viết mã Python kết nối cơ sở dữ liệu:

```
import mysql.connector  
conn = mysql.connector.connect(  
    host="localhost",  
    user="root",  
    password="123456",  
    database="TenDatabase"  
)  
print("Kết nối MySQL thành công!")
```

#### 5. Xây dựng giao diện Tkinter

**Thêm dữ liệu:** Khi người dùng nhập đầy đủ thông tin vào form (Entry + Combobox) và nhấn nút “**Thêm**”:

- Mã Python sẽ gọi hàm INSERT INTO tương ứng.
- Nếu thành công → hiện thông báo messagebox.showinfo.
- Gọi load data() để cập nhật lại bảng Treeview, giúp hiển thị ngay nhân viên/chức vụ/phòng ban mới được thêm.

**Xem dữ liệu:** Khi người dùng mở một module (ví dụ: nhanvien.py, chucvu.py, phongban.py):

- Hệ thống chạy truy vấn SELECT \* FROM [tên bảng].
- Các dòng dữ liệu được tải vào bảng **ttk.Treeview** và hiển thị đầy đủ trên giao diện.

**Tìm kiếm:** Người dùng nhập từ khóa vào ô Tìm kiếm (theo Mã hoặc Tên) rồi nhấn nút “**Tìm**”:

- Python lấy từ khóa và truy vấn SELECT \* FROM [tên bảng] WHERE [Trường] LIKE %tukhoa%
- Bảng Treeview sẽ được làm mới và chỉ hiển thị các kết quả phù hợp.

**Xóa dữ liệu:** Khi người dùng chọn một dòng trong Treeview và nhấn “**Xóa**”:

- Hộp thoại xác nhận messagebox.askyesno sẽ xuất hiện.
- Nếu người dùng chọn **Yes**, lệnh **DELETE** sẽ được thực thi.
- Gọi lại **load\_data()** để cập nhật Treeview và loại bỏ dòng vừa bị xóa khỏi danh sách.

#### Tương thích với giao diện

- Màn hình chính có các nút: Nhân viên – Chức vụ – Phòng ban – Tài khoản – Thoát.

- Mỗi nút mở 1 module riêng (Toplevel).
- Các module đều có:
  - Form nhập liệu bên trái
  - Các nút chức năng (Thêm – Sửa – Xóa – Tìm – Thoát)
  - Bảng Treeview bên phải
- Toàn bộ cơ chế thêm – xem – tìm – xóa đều đồng nhất cho tất cả bảng trong quản lý nhân sự.

### **C. HƯỚNG DẪN CÀI ĐẶT & CODE ỨNG DỤNG**

Ứng dụng sẽ có:

- Giao diện đăng nhập cho nhân viên và quản lý (quản lý mới có thể chỉnh sửa thông tin).
- Giao diện lựa chọn mục cần xem - Nhập thông tin sản phẩm, loại, khách hàng, nhân viên
- Bảng danh sách sản phẩm, loại, khách hàng, nhân viên, hóa đơn
- Chức năng CRUD (Thêm, Sửa, Xóa, Lưu, Tìm kiếm).
- Dữ liệu lưu trực tiếp vào **MySQL Database** (không mất khi tắt ứng dụng).
- Giao diện Tkinter thiết kế chuẩn như hình minh họa ở trên

#### **1. Cài đặt môi trường**

Trước tiên cài đặt thư viện cần thiết:

```
pip install mysql-connector-python tkcalendar
```

Giải thích:

mysql-connector-python: dùng để kết nối Python với MySQL.

tkcalendar: để chọn ngày sinh bằng DateEntry.

#### **2. Chuẩn bị CSDLMySQL**

Mở MySQL (WorkBench, phpMyAdmin hoặc terminal) rồi tạo database và bảng

```

1  -- 1. Xóa database cũ đi để tránh lỗi trùng lặp
2  • DROP DATABASE IF EXISTS hr_management;
3
4  -- 2. Tạo lại database mới tính
5  • CREATE DATABASE hr_management CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
6  • USE hr_management;
7
8  -- 3. Tạo lại các bảng
9  • CREATE TABLE phongban (
10     MaPB VARCHAR(20) PRIMARY KEY,
11     TenPB VARCHAR(100) NOT NULL,
12     DiaChi VARCHAR(200),
13     SDT VARCHAR(20)
14 );
15
16 • CREATE TABLE chucvu (
17     MaCV VARCHAR(20) PRIMARY KEY,
18     TenCV VARCHAR(100) NOT NULL,
19     PhuCap DECIMAL(18,2) DEFAULT 0
20 );
21
22 • CREATE TABLE nhanvien (
23     MaNV VARCHAR(20) PRIMARY KEY,
24     TenNV VARCHAR(100) NOT NULL,
25
26     -- 22 • CREATE TABLE nhanvien (
27         MaNV VARCHAR(20) PRIMARY KEY,
28         TenNV VARCHAR(100) NOT NULL,
29         GioiTinh VARCHAR(10),
30         NgaySinh DATE,
31         DiaChi VARCHAR(200),
32         DienThoai VARCHAR(20),
33         MaPB VARCHAR(20),
34         MaCV VARCHAR(20),
35         TenDangNhap VARCHAR(50) UNIQUE, -- Cột này quan trọng để đăng nhập
36         MatKhau VARCHAR(50),
37         VaiTro VARCHAR(20) DEFAULT 'User',
38         FOREIGN KEY (MaPB) REFERENCES phongban(MaPB) ON DELETE SET NULL,
39         FOREIGN KEY (MaCV) REFERENCES chucvu(MaCV) ON DELETE SET NULL
40     );
41
42 • CREATE TABLE hopdong (
43     SoHD VARCHAR(20) PRIMARY KEY,
44     MaNV VARCHAR(20),
45     LoaiHD VARCHAR(50),
46     NgayBatDau DATE,
47     NgayKetThuc DATE,
48     GhiChu TEXT,
49     FOREIGN KEY (MaNV) REFERENCES nhanvien(MaNV) ON DELETE CASCADE

```



```

47
48 • CREATE TABLE bangluong (
49     MaLuong VARCHAR(20) PRIMARY KEY,
50     MaNV VARCHAR(20),
51     Thang INT,
52     Nam INT,
53     LuongCoBan DECIMAL(18,2),
54     NgayCong INT,
55     Thuong DECIMAL(18,2),
56     Phat DECIMAL(18,2),
57     ThucLanh DECIMAL(18,2),
58     FOREIGN KEY (MaNV) REFERENCES nhanvien(MaNV) ON DELETE CASCADE
59 );
60
61 -- 4. Thêm tài khoản Admin (QUAN TRỌNG)
62 • INSERT INTO phongban VALUES ('PB00', 'Ban Giám Đốc', 'Tầng 1', '0000');
63 • INSERT INTO chucvu VALUES ('CV00', 'Admin', 0);
64 • INSERT INTO nhanvien (MaNV, TenNV, MaPB, MaCV, TenDangNhap, MatKhau, VaiTro)
65   VALUES ('ADMIN', 'Quản Trị Viên', 'PB00', 'CV00', 'admin', 'admin', 'Admin');
66 • USE hr_management;

61 -- 4. Thêm tài khoản Admin (QUAN TRỌNG)
62 • INSERT INTO phongban VALUES ('PB00', 'Ban Giám Đốc', 'Tầng 1', '0000');
63 • INSERT INTO chucvu VALUES ('CV00', 'Admin', 0);
64 • INSERT INTO nhanvien (MaNV, TenNV, MaPB, MaCV, TenDangNhap, MatKhau, VaiTro)
65   VALUES ('ADMIN', 'Quản Trị Viên', 'PB00', 'CV00', 'admin', 'admin', 'Admin');
66 • USE hr_management;
67
68 -- 1. Xóa dữ liệu cũ (để tránh lỗi trùng lặp nếu chạy nhiều lần)
69 • SET FOREIGN_KEY_CHECKS = 0;
70 • TRUNCATE TABLE bangluong;
71 • TRUNCATE TABLE hopdong;
72 • TRUNCATE TABLE nhanvien;
73 • TRUNCATE TABLE chucvu;
74 • TRUNCATE TABLE phongban;
75 • SET FOREIGN_KEY_CHECKS = 1;
76
77 -- 2. Thêm Phòng Ban
78 • INSERT INTO phongban (MaPB, TenPB, DiaChi, SDT) VALUES
79   ('PB01', 'Ban Giám Đốc', 'Tầng 5', '0901111111'),
80   ('PB02', 'Phòng Nhân Sự', 'Tầng 2', '0902222222'),
81   ('PB03', 'Phòng Kế Toán', 'Tầng 2', '0903333333'),
82   ('PB04', 'Phòng IT - Kỹ Thuật', 'Tầng 3', '0904444444'),
83   ('PB05', 'Phòng Kinh Doanh', 'Tầng 4', '0905555555');

```



```

85 -- 3. Thêm Chức Vụ
86 • INSERT INTO chucvu (MaCV, TenCV, PhuCap) VALUES
87 ('CV01', 'Tổng Giám Đốc', 10000000),
88 ('CV02', 'Trưởng Phòng', 5000000),
89 ('CV03', 'Phó Phòng', 3000000),
90 ('CV04', 'Nhân Viên Chính Thức', 1000000),
91 ('CV05', 'Thực Tập Sinh', 500000);
92
93 -- 4. Thêm Nhân Viên (Lưu ý: Admin mật khẩu là admin, User mật khẩu là 123)
94 • INSERT INTO nhanvien (MaNV, TenNV, GioiTinh, NgaySinh, DiaChi, DienThoai, MaPB, MaCV, TenDangNhap, MatKhau, VaiTro) VALUES
95 ('NV01', 'Nguyễn Văn Admin', 'Nam', '1990-01-01', 'Hà Nội', '0988888888', 'PB01', 'CV01', 'admin', 'admin', 'Admin'),
96 ('NV02', 'Trần Thị Thư Ký', 'Nữ', '1995-05-20', 'Đà Nẵng', '0977777777', 'PB02', 'CV02', 'thuky', '123', 'User'),
97 ('NV03', 'Lê Văn Code', 'Nam', '1998-12-10', 'TP.HCM', '0966666666', 'PB04', 'CV04', 'coder', '123', 'User'),
98 ('NV04', 'Phạm Thị Kế Toán', 'Nữ', '1992-06-15', 'Hải Phòng', '0955555555', 'PB03', 'CV02', 'ketuan', '123', 'User'),
99 ('NV05', 'Hoàng Văn Sale', 'Nam', '1996-03-30', 'Cần Thơ', '0944444444', 'PB05', 'CV04', 'sale01', '123', 'User'),
100 ('NV06', 'Đỗ Thị Hạnh', 'Nữ', '2000-01-01', 'Hà Nội', '0933333333', 'PB02', 'CV04', 'hanhdt', '123', 'User'),
101 ('NV07', 'Vũ Văn Hùng', 'Nam', '1993-07-22', 'Nghệ An', '0922222222', 'PB04', 'CV03', 'hungvv', '123', 'User'),
102 ('NV08', 'Ngô Lan Anh', 'Nữ', '1997-11-11', 'Huế', '0911111111', 'PB05', 'CV04', 'lananh', '123', 'User'),
103 ('NV09', 'Bùi Đức Thắng', 'Nam', '1999-09-09', 'Thanh Hóa', '0900000000', 'PB04', 'CV05', 'thangbd', '123', 'User'),
104 ('NV10', 'Lý Thị Mơ', 'Nữ', '2001-02-14', 'Sapa', '0899999999', 'PB02', 'CV05', 'molt', '123', 'User');

106 -- 5. Thêm Hợp Đồng
107 • INSERT INTO hopdong (SoHD, MaNV, LoaiHD, NgayBatDau, NgayKetThuc, GhiChu) VALUES
108 ('HD01', 'NV01', 'Vô thời hạn', '2020-01-01', NULL, 'Hợp đồng sắp'),
109 ('HD02', 'NV02', '3 Năm', '2022-01-01', '2025-01-01', 'Hợp đồng chính thức'),
110 ('HD03', 'NV03', '1 Năm', '2023-06-01', '2024-06-01', 'Hợp đồng thử việc xong'),
111 ('HD04', 'NV04', '3 Năm', '2021-05-05', '2024-05-05', 'Tái ký lần 1');
112
113 -- 6. Thêm Bảng Lương (Tháng 11/2024)
114 • INSERT INTO bangluong (MaLuong, MaNV, Thang, Nam, LuongCoBan, NgayCong, Thuong, Phat, ThucLanh) VALUES
115 ('L01', 'NV01', 11, 2024, 50000000, 26, 5000000, 0, 55000000),
116 ('L02', 'NV02', 11, 2024, 15000000, 26, 1000000, 0, 16000000),
117 ('L03', 'NV03', 11, 2024, 20000000, 24, 500000, 200000, 18900000),
118 ('L04', 'NV04', 11, 2024, 18000000, 26, 2000000, 0, 20000000),
119 ('L05', 'NV05', 11, 2024, 12000000, 25, 3000000, 0, 14500000);
120
121 • UPDATE nhanvien SET TenNV = 'Quản Trị Viên' WHERE MaNV = 'ADMIN';

```

## Bảng lương

```
bangluong.py X phu.py
C: > Users > Admin > Downloads > QLNSS > QLNSS > bangluong.py > ...

bangluong.py X phu.py
C: > Users > Admin > Downloads > QLNSS > QLNSS > bangluong.py > ...

6 class BangLuongPage(tk.Frame):
7     def __init__(self, parent, controller):
32         self.e_nc.grid(row=1, column=3, padx=5, pady=5)
33
34         tk.Label(frm, text="Thưởng:", bg="white").grid(row=1, column=4, sticky="w")
35         self.e_thuong = tk.Entry(frm, width=15);
36         self.e_thuong.insert(0, "0")
37         self.e_thuong.grid(row=1, column=5, padx=5, pady=5)
38
39         # Dòng 3: Các nút chức năng (Buttons)
40         btn_box = tk.Frame(frm, bg="white")
41         btn_box.grid(row=2, columnspan=6, pady=(15, 0))
42
43         # Nút làm mới
44         tk.Button(btn_box, text="Ơ làm mới", command=self.clear,
45                 bg="#95a5a6", fg="white", width=12).pack(side="left", padx=5)
46
47         # Nút Thêm (Tính & Lưu)
48         tk.Button(btn_box, text="+ TÍNH & LƯU", command=self.them,
49                 bg="#27ae60", fg="white", font=("Arial", 10, "bold"), width=15).pack(side="left", padx=5)
50
51         # Nút Sửa
52         tk.Button(btn_box, text="CẬP NHẬT", command=self.sua,
53                 bg="#f39c12", fg="white", font=("Arial", 10, "bold"), width=12).pack(side="left", padx=5)
54
55         # Nút Xóa
56         tk.Button(btn_box, text="X XÓA PHIẾU", command=self.xoa,
57                 bg="#e0392b", fg="white", font=("Arial", 10, "bold"), width=12).pack(side="left", padx=5)
58
59         # --- BẢNG DỮ LIỆU (TREEVIEW) ---
60         self.tree = ttk.Treeview(self, columns=("Ma", "Ten", "LCB", "NC", "Thuong", "Thuc"), show="headings", height=12)
61
```

```
C: > Users > Admin > Downloads > QLNSS > QLNSS > bangluong.py > ...

6 class BangLuongPage(tk.Frame):
7     def __init__(self, parent, controller):
61
62         self.tree.heading("Ma", text="Mã Phiếu");
63         self.tree.column("Ma", width=80, anchor="center")
64         self.tree.heading("Ten", text="Tên Nhân Viên");
65         self.tree.column("Ten", width=200)
66         self.tree.heading("LCB", text="Lương Cơ Bản");
67         self.tree.column("LCB", anchor="e", width=120)
68         self.tree.heading("NC", text="Ngày Công");
69         self.tree.column("NC", anchor="center", width=80)
70         self.tree.heading("Thuong", text="Thưởng");
71         self.tree.column("Thuong", anchor="e", width=100)
72         self.tree.heading("Thuc", text="THỰC LĨNH (VNĐ)");
73         self.tree.column("Thuc", anchor="e", width=150)
74
75         self.tree.pack(fill="both", expand=True)
76
77         # Sự kiện click vào bảng
78         self.tree.bind("<<TreeviewSelect>>", self.get_row)
79
80         self.nv_map = {}
81
82         def load_data(self):
83             # Load danh sách nhân viên vào Combobox
84             nvs = fetch_all("SELECT MaNV, TenNV FROM nhanvien")
85             self.nv_map = {f'{r["TenNV"]} ({r["MaNV"]})': r["MaNV"] for r in nvs}
86             self.cb_nv['values'] = list(self.nv_map.keys())
87
88             # Load dữ liệu bảng lương
89             for i in self.tree.get_children(): self.tree.delete(i)
90
```

```

bangluong.py X  phu.py
C: > Users > Admin > Downloads > QLNSS > QLNSS > bangluong.py > ...
6  class BangLuongPage(tk.Frame):
82  def load_data(self):
90
91      sql = """SELECT b.MaLuong, n.TenNV, b.LuongCoBan, b.NgayCong, b.Thuong, b.ThucLanh, n.MaNV
92      FROM bangluong b
93      JOIN nhanvien n ON b.MaNV = n.MaNV
94      ORDER BY b.MaLuong DESC""" # Sắp xếp mới nhất lên đầu
95
96      for r in fetch_all(sql):
97          # Format số tiền có dấu phẩy ngăn cách
98          lcb_fmt = f"{r['LuongCoBan']:,.0f}"
99          thuong_fmt = f"{r['Thuong']:,.0f}"
100          thuc_fmt = f"{r['ThucLanh']:,.0f}"
101
102          self.tree.insert("", "end",
103                          values=(r['MaLuong'], r['TenNV'], lcb_fmt, r['NgayCong'], thuong_fmt, thuc_fmt, r['MaNV']))
104
105  def get_row(self, event):
106      try:
107          sel = self.tree.selection()
108          if not sel: return
109          vals = self.tree.item(sel[0])['values']
110          # vals: 0=Ma, 1=Ten, 2=LCB, 3=NC, 4=Thuong, 5=Thuc, 6=MaNV (ẩn)
111
112          self.clear()
113          self.e_ma.insert(0, vals[0])
114          self.e_ma.config(state='disabled') # Khóa mã phiếu lại để không sửa lung tung
115
116          # Map lại tên nhân viên vào combobox
117          target_ma_nv = str(vals[6]) # Lấy mã NV từ cột ẩn
118          for k, v in self.nv_map.items():
119              if v == target_ma_nv:

```

```

bangluong.py X  phu.py
C: > Users > Admin > Downloads > QLNSS > QLNSS > bangluong.py > ...
6  class BangLuongPage(tk.Frame):
105  def get_row(self, event):
119
120      if v == target_ma_nv:
121          self.cb_nv.set(k)
122          break
123
124      # Xóa dấu phẩy tiền tệ trước khi đưa vào ô nhập (VD: "1,000,000" -> "1000000")
125      self.e_lcb.insert(0, str(vals[2]).replace(",", ""))
126      self.e_nc.insert(0, vals[3])
127      self.e_thuong.insert(0, str(vals[4]).replace(",", ""))
128
129  except IndexError:
130      pass
131
132  def clear(self):
133      self.e_ma.config(state='normal') # Mở khóa để nhập mới
134      self.e_ma.delete(0, tk.END)
135      self.cb_nv.set('')
136      self.e_lcb.delete(0, tk.END)
137      self.e_nc.delete(0, tk.END)
138      self.e_thuong.delete(0, tk.END)
139      self.e_thuong.insert(0, "0")
140      self.e_ma.focus()
141
142  def get_and_validate_input(self):
143      """Hàm phụ trợ: Lấy dữ liệu và kiểm tra lỗi, tự động tính Thực Lĩnh"""
144      ma_phieu = self.e_ma.get().strip()
145      nhan_vien = self.cb_nv.get()
146
147      if not ma_phieu or not nhan_vien:
148          messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập Mã phiếu và chọn Nhân viên!")
149      return None

```

```

bangluong.py x  phu.py
C:\> Users > Admin > Downloads > QLNSS > QLNSS > bangluong.py > ...
6 class BangLuongPage(tk.Frame):
141 def get_and_validate_input(self):
148     return None
149
150     try:
151         lcb = float(self.e_lcb.get())
152         nc = int(self.e_nc.get())
153         thuong = float(self.e_thuong.get())
154
155         if lcb < 0 or nc < 0 or thuong < 0:
156             messagebox.showerror("Lỗi logic", "Số tiền và ngày công không được âm!")
157             return None
158
159         # CÔNG THỨC TÍNH LƯƠNG
160         thuc_lanh = (lcb / 26 * nc) + thuong
161
162         ma_nv = self.nv_map.get(nhan_vien)
163         return (ma_phieu, ma_nv, lcb, nc, thuong, thuc_lanh)
164
165     except ValueError:
166         messagebox.showerror("Lỗi nhập liệu", "Lương, Ngày công, Thưởng phải là số hợp lệ!")
167         return None
168
169 def them(self):
170     data = self.get_and_validate_input()
171     if not data: return
172
173     # Kiểm tra trùng mã phiếu
174     check = fetch_all("SELECT * FROM bangluong WHERE MaLuong=%s", (data[0],))
175     if check:
176         messagebox.showerror("Trùng lặp", f"Mã phiếu '{data[0]}' đã tồn tại! Vui lòng chọn mã khác.")
177         return

```

```

bangluong.py x  phu.py
C:\> Users > Admin > Downloads > QLNSS > QLNSS > bangluong.py > ...
6 class BangLuongPage(tk.Frame):
169 def them(self):
178     # Thực hiện Thêm
179     sql = "INSERT INTO bangluong(MaLuong, MaNV, LuongCoBan, NgayCong, Thuong, ThucLanh) VALUES(%s, %s, %s, %s, %s, %s)"
180     if execute_query(sql, data):
181         self.load_data()
182         self.clear()
183         messagebox.showinfo("Thành công", f"Đã lưu phiếu lương.\nThực lĩnh: {data[5]:.0f} VNĐ")
184
185     def sua(self):
186         # Kiểm tra xem có đang chọn dòng nào không hoặc mã có bị trống không
187         if self.e_ma.cget('state') == 'normal':
188             messagebox.showwarning("Chú ý", "Bạn đang ở chế độ Thêm mới. Hãy chọn một dòng trong bảng để Sửa.")
189             return
190
191         data = self.get_and_validate_input() # data[0] là Mã Phiếu
192         if not data: return
193
194         # Thực hiện Update
195         sql = """UPDATE bangluong
196             SET MaNV=%s, LuongCoBan=%s, NgayCong=%s, Thuong=%s, ThucLanh=%s
197             WHERE MaLuong=%s"""
198         # Đảo thứ tự params cho khớp với SQL: MaNV, LCB, NC, Thuong, ThucLanh, MaLuong (ở cuối)
199         params = (data[1], data[2], data[3], data[4], data[5], data[0])
200
201         if execute_query(sql, params):
202             self.load_data()
203             self.clear()
204             messagebox.showinfo("Thành công", f"Đã cập nhật phiếu {data[0]}. \nThực lĩnh mới: {data[5]:.0f} VNĐ")
205
206     def xoa(self):
207

```

## Bảng chức vụ

```
chucvu.py X phu.py •
C:\Users\Admin>Downloads>QLNSS>QLNSS> chucvu.py
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 from database import fetch_all, execute_query
4
5
6 class ChucVuPage(tk.Frame):
7     def __init__(self, parent, controller):
8         tk.Frame.__init__(self, parent, bg="#ecf0f1")
9         self.controller = controller
10
11         # --- FORM NHẬP LIỆU ---
12         frm = tk.LabelFrame(self, text="Quản lý Chức vụ", bg="white", padx=15, pady=15)
13         frm.pack(fill="x", pady=(0, 10))
14
15         tk.Label(frm, text="Mã CV (*):", bg="white").grid(row=0, column=0, sticky="w")
16         self.txt_ma = tk.Entry(frm, width=15);
17         self.txt_ma.grid(row=0, column=1, padx=10, pady=5)
18
19         tk.Label(frm, text="Tên CV (*):", bg="white").grid(row=0, column=2, sticky="w")
20         self.txt_ten = tk.Entry(frm, width=25);
21         self.txt_ten.grid(row=0, column=3, padx=10, pady=5)
22
23         tk.Label(frm, text="Phụ Cấp (VNĐ):", bg="white").grid(row=0, column=4, sticky="w")
24         self.txt_pc = tk.Entry(frm, width=15);
25         self.txt_pc.insert(0, "0");
26         self.txt_pc.grid(row=0, column=5, padx=10, pady=5)
27
28         # Buttons
29         btn_box = tk.Frame(frm, bg="white")
30         btn_box.grid(row=1, colspan=6, pady=10)
31
32         tk.Button(btn_box, text="Làm mới", bg="#95a5a6", fg="white", width=10, command=self.clear).pack(side="left",
33
chucvu.py X phu.py •
C:\Users\Admin>Downloads>QLNSS>QLNSS> chucvu.py > ...
6 class ChucVuPage(tk.Frame):
7     def them(self):
8         if not self.validate(): return
9
10         # Check trùng
11         if fetch_all("SELECT * FROM chucvu WHERE MaCV=%s", (self.txt_ma.get(),)):
12             messagebox.showerror("Lỗi", "Mã chức vụ đã tồn tại!")
13             return
14
15         if execute_query("INSERT INTO chucvu VALUES(%s,%s,%s)",
16                         (self.txt_ma.get(), self.txt_ten.get(), self.txt_pc.get())):
17             self.load_data();
18             self.clear();
19             messagebox.showinfo("OK", "Đã thêm chức vụ")
20
21     def sua(self):
22         if not self.txt_ma.get(): return
23         if not self.validate(): return
24
25         if execute_query("UPDATE chucvu SET TenCV=%s, PhuCap=%s WHERE MaCV=%s",
26                         (self.txt_ten.get(), self.txt_pc.get(), self.txt_ma.get())):
27             self.load_data();
28             self.clear();
29             messagebox.showinfo("OK", "Đã cập nhật")
30
31     def xoa(self):
32         sel = self.tree.selection()
33         if sel and messagebox.askyesno("Xác nhận", "Xóa chức vụ này?"):
34             if execute_query("DELETE FROM chucvu WHERE MaCV=%s", (self.tree.item(sel[0])['values'][0],)):
35                 self.load_data();
36                 self.clear()
```

```
chucvu.py x phu.py •
C:\Users\Admin>Downloads>QLNSS>QLNSS> chucvu.py > ...
6 class ChucVuPage(tk.Frame):
57 def get_row(self, event):
63     self.clear()
64     self.txt_ma.insert(0, item[0])
65     self.txt_ma.config(state='disabled') # Không sửa mã
66     self.txt_ten.insert(0, item[1])
67     # Xóa dấu phẩy khi đưa lại vào ô nhập (ví dụ: 1,000,000 -> 1000000)
68     pc_str = str(item[2]).replace(",", "")
69     self.txt_pc.insert(0, pc_str)
70 except:
71     pass
72
73 def clear(self):
74     self.txt_ma.config(state='normal')
75     self.txt_ma.delete(0, tk.END)
76     self.txt_ten.delete(0, tk.END)
77     self.txt_pc.delete(0, tk.END)
78     self.txt_pc.insert(0, "0")
79     self.txt_ma.focus()
80
81 def validate(self):
82     if not self.txt_ma.get().strip() or not self.txt_ten.get().strip():
83         messagebox.showwarning("Thiếu dữ liệu", "Vui lòng nhập Mã và Tên chức vụ!")
84         return False
85     try:
86         float(self.txt_pc.get())
87     except ValueError:
88         messagebox.showerror("Lỗi", "Phụ cấp phải là số!")
89         return False
90     return True
91
92 def them(self):
93
chucvu.py x phu.py •
C:\Users\Admin>Downloads>QLNSS>QLNSS> chucvu.py > ...
6 class ChucVuPage(tk.Frame):
7 def __init__(self, parent, controller):
33
34     tk.Button(btn_box, text="Thêm", bg="#2980b9", fg="white", width=10, command=self.them).pack(side="left", padx=5)
35     tk.Button(btn_box, text="Sửa", bg="#f39c12", fg="white", width=10, command=self.sua).pack(side="left", padx=5)
36     tk.Button(btn_box, text="Xóa", bg="#c0392b", fg="white", width=10, command=self.xoa).pack(side="left", padx=5)
37
38     # --- TABLE ---
39     self.tree = ttk.Treeview(self, columns=("Ma", "Ten", "PC"), show="headings")
40     self.tree.heading("Ma", text="Mã CV");
41     self.tree.column("Ma", width=100)
42     self.tree.heading("Ten", text="Tên Chức Vụ");
43     self.tree.column("Ten", width=250)
44     self.tree.heading("PC", text="Phụ cấp (VNĐ)");
45     self.tree.column("PC", anchor="e")
46     self.tree.pack(fill="both", expand=True)
47
48     # SỰ KIẾN CLICK
49     self.tree.bind("<<TreeviewSelect>>", self.get_row)
50
51 def load_data(self):
52     for i in self.tree.get_children(): self.tree.delete(i)
53     for r in fetch_all("SELECT * FROM chucvu"):
54         # Format tiền có dấu phẩy
55         self.tree.insert("", "end", values=(r['Macv'], r['Tencv'], f"{r['Phucap']:,}"))
56
57 def get_row(self, event):
58     try:
59         sel = self.tree.selection()
60         if not sel: return
61         item = self.tree.item(sel[0])['values']
62
```

## Bảng phòng ban

```
phongban.py X  phu.py
C: > Users > Admin > Downloads > QLNSS > QLNSS > phongban.py > ...
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 from database import fetch_all, execute_query
4
5
6 class PhongBanPage(tk.Frame):
7     def __init__(self, parent, controller):
8         tk.Frame.__init__(self, parent, bg="#ecf0f1")
9         self.controller = controller
10
11         # Layout
12         container = tk.Frame(self, bg="#ecf0f1")
13         container.pack(fill="both", expand=True)
14
15         left = tk.Frame(container, bg="white", width=300, padx=10, pady=10)
16         left.pack(side="left", fill="y", padx=(0, 10))
17
18         right = tk.Frame(container, bg="white")
19         right.pack(side="right", fill="both", expand=True)
20
21         # Form
22         tk.Label(left, text="QUẢN LÝ PHÒNG BAN", font=("Arial", 12, "bold"), bg="white", fg="#2980b9").pack(pady=10)
23
24         tk.Label(left, text="Mã PB (*)", bg="white", anchor="w").pack(fill="x")
25         self.txt_ma = tk.Entry(left)
26         self.txt_ma.pack(fill="x", pady=5)
27
28         tk.Label(left, text="Tên PB (*)", bg="white", anchor="w").pack(fill="x")
29         self.txt_ten = tk.Entry(left)
30         self.txt_ten.pack(fill="x", pady=5)
31
32         tk.Label(left, text="Địa chỉ:", bg="white", anchor="w").pack(fill="x")
33
34         self.txt_dc = tk.Entry(left)
35         self.txt_dc.pack(fill="x", pady=5)
36
37         tk.Label(left, text="SDT:", bg="white", anchor="w").pack(fill="x")
38         self.txt_sdt = tk.Entry(left)
39         self.txt_sdt.pack(fill="x", pady=5)
40
41         tk.Button(left, text="THÊM", bg="#27ae60", fg="white", pady=5, command=self.them).pack(fill="x", pady=(20, 5))
42         tk.Button(left, text="SỬA", bg="#f39c12", fg="white", pady=5, command=self.sua).pack(fill="x", pady=5)
43         tk.Button(left, text="XÓA", bg="#c0392b", fg="white", pady=5, command=self.xoa).pack(fill="x", pady=5)
44         tk.Button(left, text="LÀM MỚI", bg="#95a5a6", fg="white", pady=5, command=self.clear).pack(fill="x", pady=5)
45
46         # Table
47         self.tree = ttk.Treeview(right, columns=("Ma", "Ten", "DC", "SDT"), show="headings")
48         self.tree.heading("Ma", text="Mã");
49         self.tree.heading("Ten", text="Tên Phòng");
50         self.tree.heading("DC", text="Địa chỉ");
51         self.tree.heading("SDT", text="SDT")
52         self.tree.pack(fill="both", expand=True)
53         self.tree.bind("<ButtonRelease-1>", self.on_click)
54
55     def load_data(self):
56         for i in self.tree.get_children(): self.tree.delete(i)
57         for r in fetch_all("SELECT * FROM phongban"):
58             self.tree.insert("", "end", values=(r['MaPB'], r['TenPB'], r['DiaChi'], r['SDT']))
59
60     def clear(self):
61         self.txt_ma.config(state='normal')
62         self.txt_ma.delete(0, tk.END)
63         self.txt_ten.delete(0, tk.END)
```

```

phongban.py X  phu.py
C: > Users > Admin > Downloads > QLNSS > QLNSS > phongban.py > ...
6 class PhongBanPage(tk.Frame):
59 def clear(self):
60     self.txt_dc.delete(0, tk.END)
61     self.txt_sdt.delete(0, tk.END)
62
63     self.txt_dc.delete(0, tk.END)
64     self.txt_sdt.delete(0, tk.END)
65
66 def them(self):
67     ma = self.txt_ma.get().strip()
68     ten = self.txt_ten.get().strip()
69
70     if not ma or not ten:
71         messagebox.showwarning("Thiếu dữ liệu", "Mã PB và Tên PB không được để trống!")
72         return
73
74     # Check trùng
75     if fetch_all("SELECT * FROM phongban WHERE MaPB=%s", (ma,)):
76         messagebox.showerror("Trùng lặp", f"Mã phòng ban '{ma}' đã tồn tại!")
77         return
78
79     if execute_query("INSERT INTO phongban VALUES(%s,%s,%s,%s)",
80                     (ma, ten, self.txt_dc.get(), self.txt_sdt.get())):
81         self.load_data();
82         self.clear();
83         messagebox.showinfo("OK", "Đã thêm phòng ban")
84
85 def sua(self):
86     ma = self.txt_ma.get().strip()
87     if not ma: return
88
89     if execute_query("UPDATE phongban SET TenPB=%s, DiaChi=%s, SDT=%s WHERE MaPB=%s",
90                     (self.txt_ten.get(), self.txt_dc.get(), self.txt_sdt.get(), ma)):
91         self.load_data();
92         self.clear();

```

```

92         self.clear();
93         messagebox.showinfo("OK", "Đã cập nhật")
94
95 def xoa(self):
96     sel = self.tree.selection()
97     if not sel: return
98     ma = self.tree.item(sel[0])['values'][0]
99
100     if messagebox.askyesno("Cảnh báo", f"Xóa phòng ban {ma} sẽ xóa luôn nhân viên thuộc phòng này. Tiếp tục?"):
101         execute_query("DELETE FROM phongban WHERE MaPB=%s", (ma,))
102         self.load_data();
103         self.clear()
104
105 def on_click(self, e):
106     sel = self.tree.selection()
107     if sel:
108         r = self.tree.item(sel[0])['values']
109         self.clear()
110         self.txt_ma.insert(0, r[0])
111         self.txt_ma.config(state='disabled') # Khóa mã không cho sửa
112         self.txt_ten.insert(0, r[1])
113         self.txt_dc.insert(0, r[2])
114         self.txt_sdt.insert(0, r[3])

```



## Bảng Database

```
1  > import mysql.connector
2      from mysql.connector import Error
3
4  > def get_connection(): 3 usages
5  >     try:
6          connection = mysql.connector.connect(
7              host='localhost',
8              user='root',
9              password='1234', # <--- ĐIỂN PASS MYSQL CỦA BẠN
10             database='hr_management',
11             charset='utf8mb4'
12         )
13         return connection
14     except Error as e:
15         print("Lỗi kết nối:", e)
16         return None
```

```
17
18 def fetch_all(query, params=None): 18 usages
19     conn = get_connection()
20     if not conn: return []
21     try:
22         cur = conn.cursor(dictionary=True)
23         cur.execute(query, params or ())
24         return cur.fetchall()
25     except Error as e:
26         print(f"Lỗi truy vấn: {e}")
27         return []
28     finally:
29         if conn.is_connected(): conn.close()
```

## Bảng main.py

```
31 def execute_query(query, params=None): 20 usages
32     conn = get_connection()
33     if not conn: return False
34     try:
35         cur = conn.cursor()
36         cur.execute(query, params or ())
37         conn.commit()
38         return True
39     except Error as e:
40         print(f"Lỗi thực thi: {e}")
41         return False
42     finally:
43         if conn.is_connected(): conn.close()
44
45 def dang_nhap(ten, mk): 2 usages
46     conn = get_connection()
47     if not conn: return None
48     try:
49         cur = conn.cursor(dictionary=True)
50         cur.execute(operation="SELECT * FROM nhanvien WHERE TenDangNhap=%s AND MatKhau=%s", params=(ten, mk))
51         return cur.fetchone()
52     finally:
53         if conn.is_connected(): conn.close()
```

```
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 # Import các class giao diện từ các file con
4 from nhanvien import NhanVienPage
5 from phongban import PhongBanPage
6 from chucvu import ChucVuPage
7 from hopdong import HopDongPage
8 from bangluong import BangLuongPage
9
10
11 class HRMApp: 1 usage
12     def __init__(self, user_data):
13         self.user = user_data
14         self.root = tk.Tk()
15         self.root.title("HỆ THỐNG QUẢN LÝ NHÂN SỰ PRO")
16         self.root.geometry("1200x700")
17         self.root.state("zoomed") # Full màn hình
18
19         # --- STYLE ---
20         style = ttk.Style()
21         style.theme_use("clam")
22         style.configure(style="Treeview", rowheight=30, font=("Arial", 10))
23         style.configure(style="Treeview.Hheading", font=("Arial", 10, "bold"), background="#bdc3c7", foreground="#2c3e50")
24
25         # --- SIDEBAR (Menu trái) ---
26         self.sidebar = tk.Frame(self.root, bg="#2c3e50", width=250)
27         self.sidebar.pack(side="left", fill="y")
28         self.sidebar.pack_propagate(False)
29
30         # Avatar & Info
31         tk.Label(self.sidebar, text="HRM SYSTEM", font=("Helvetica", 18, "bold"), bg="#2c3e50", fg="#ecf0f1").pack(
32             pady=(40, 20))
33         tk.Label(self.sidebar, text="☺", font=("Arial", 50), bg="#2c3e50", fg="white").pack(pady=(0, 10))
34
35         # Hiển thị Tên (Lấy từ DB)
36         # --- SỬA LẠI ĐOẠN NÀY ---
37         # Xử lý tên: Bỏ chữ "Nguyễn Văn" nếu có
```

```

40 # Nếu lỡ xóa hết thì để mặc định là Admin
41 if not ten_hien_thi:
42     ten_hien_thi = "Admin"
43
44 tk.Label(self.sidebar, text=ten_hien_thi, font=("Arial", 12, "bold"),
45         bg="#2c3e50", fg="white", wraplength=230).pack(padx=10)
46 # -----
47
48 # Hiển thị Vai trò
49 role = self.user.get('VaiTro', 'NhanVien')
50 tk.Label(self.sidebar, text=f"Chức vụ: {role}", font=("Arial", 9, "italic"),
51         bg="#2c3e50", fg="#bdc3c7").pack(pady=(5, 30))
52
53 # MENU BUTTONS
54 self.btn_dict = {}
55 menus = [
56     ("NHÂN VIÊN", NhanVienPage),
57     ("PHÒNG BAN", PhongBanPage),
58     ("CHỨC VỤ", ChucVuPage),
59     ("HỢP ĐỒNG", HopDongPage),
60     ("LƯƠNG & THUỐC", BangLuongPage)
61 ]
62
63 for text, page_class in menus:
64     btn = tk.Button(self.sidebar, text=f" {text}", font=("Arial", 10, "bold"),
65         bg="#2c3e50", fg="white", bd=0, anchor="w", padx=20, pady=12,
66         activebackground="#34495e", activeforeground="white",
67         command=lambda p=page_class, t=text: self.show_frame(p, t))
68     btn.pack(fill="x", pady=1)
69     self.btn_dict[text] = btn
70
71 # Logout
72 tk.Button(self.sidebar, text=" ĐĂNG XUẤT", font=("Arial", 10, "bold"), bg="#c0392b", fg="white", bd=0,
73         anchor="w", padx=20, pady=12,
74         command=self.logout).pack(side="bottom", fill="x")
75
76 # --- MAIN CONTENT (Bên phải) ---
77 self.main_area = tk.Frame(self.root, bg="#ecf0f1")
78 self.main_area.pack(side="right", fill="both", expand=True)
79
80 self.lbl_title = tk.Label(self.main_area, text="Trang chủ", font=("Arial", 22, "bold"), bg="#ecf0f1",
81         fg="#2c3e50")
82 self.lbl_title.pack(anchor="w", padx=30, pady=20)
83
84 # Container chứa các trang
85 self.container = tk.Frame(self.main_area, bg="#ecf0f1")
86 self.container.pack(fill="both", expand=True, padx=30, pady=(0, 30))
87
88 self.frames = {}
89 for F in (NhanVienPage, PhongBanPage, ChucVuPage, HopDongPage, BangLuongPage):
90     page_name = F.__name__
91     frame = F(parent=self.container, controller=self)
92     self.frames[page_name] = frame
93     frame.grid(row=0, column=0, sticky="nsew")
94
95 # Mặc định mở trang Nhân Viên
96 self.show_frame(NhanVienPage, title="NHÂN VIÊN")
97 self.root.mainloop()
98

```

```

99     def show_frame(self, page_class, title): 2 usages
100         self.lbl_title.config(text=title)
101
102         # Đổi màu nút active
103         for k, btn in self.btn_dict.items():
104             if k == title:
105                 btn.configure(bg="#3498db", text=f"> {k}")
106             else:
107                 btn.configure(bg="#2c3e50", text=f" {k}")
108
109         page_name = page_class.__name__
110         frame = self.frames[page_name]
111         frame.tkraise()
112         if hasattr(frame, "load_data"):
113             frame.load_data()
114
115     def logout(self): 1 usage
116         if messagebox.askyesno( title: "Đăng xuất", message: "Bạn muốn thoát?"):
117             self.root.destroy()
118             from login_view import LoginApp
119             LoginApp()

```

## Bảng login.view.py

```
1 import tkinter as tk
2 from tkinter import messagebox
3 from database import dang_nhap
4
5
6 class LoginApp: 3 usages
7     def __init__(self):
8         self.root = tk.Tk()
9         self.root.title("HRM System Login")
10        self.root.geometry("400x500")
11        self.root.configure(bg="#ecf0f1")
12
13        # Căn giữa màn hình
14        screen_width = self.root.winfo_screenwidth()
15        screen_height = self.root.winfo_screenheight()
16        x = (screen_width - 400) // 2
17        y = (screen_height - 500) // 2
18        self.root.geometry(f"400x500+{x}+{y}")
19
20        # Header
21        header = tk.Frame(self.root, bg="#2c3e50", height=150)
22        header.pack(fill="x")
23        tk.Label(header, text="HUMAN RESOURCE\nMANAGEMENT", font=("Helvetica", 18, "bold"),
24                bg="#2c3e50", fg="white").place(relx=0.5, rely=0.5, anchor="center")
25
26        # Form Frame
27        frm = tk.Frame(self.root, bg="white", padx=20, pady=20)
28        frm.place(relx=0.5, rely=0.4, anchor="n", width=350, height=250)
29
30        # Username
31        tk.Label(frm, text="Tên đăng nhập", bg="white", fg="#7f8c8d", font=("Arial", 10)).pack(anchor="w")
32        self.entry_user = tk.Entry(frm, font=("Arial", 12), bd=0, bg="#ecf0f1")
33        self.entry_user.pack(fill="x", pady=5, ipady=5)
34
35        # Password
36        tk.Label(frm, text="Mật khẩu", bg="white", fg="#7f8c8d", font=("Arial", 10)).pack(anchor="w", pady=(10, 0))
37        self.entry_pass = tk.Entry(frm, font=("Arial", 12), show="", bd=0, bg="#ecf0f1")
38        self.entry_pass.pack(fill="x", pady=5, ipady=5)
39
40        # Button
41        tk.Button(frm, text="ĐĂNG NHẬP", font=("Arial", 11, "bold"), bg="#2980b9", fg="white",
42                relief="flat", cursor="hand2", command=self.check_login).pack(fill="x", pady=20, ipady=5)
43
44        self.root.bind('<Return>', lambda e: self.check_login())
45        self.root.mainloop()
46
47    def check_login(self): 2 usages
48        u = self.entry_user.get()
49        p = self.entry_pass.get()
50        user = dang_nhap(u, p)
51        if user:
52            self.root.destroy()
53            import main
54            main.HRMApp(user)
55        else:
56            messagebox.showerror(title="Lỗi", message="Sai thông tin đăng nhập!")
57
58
59 if __name__ == "__main__":
60     LoginApp()
```



## Bảng nhân viên.py

```
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 from tkcalendar import DateEntry
4 from database import fetch_all, execute_query
5
6
7 class NhanVienPage(tk.Frame): 4 usages
8     def __init__(self, parent, controller):
9         tk.Frame.__init__(self, parent, bg="#ecf0f1")
10        self.controller = controller
11
12        # --- FORM NHẬP LIỆU ---
13        frm = tk.LabelFrame(self, text="Thông tin nhân viên", bg="white", padx=15, pady=15)
14        frm.pack(fill="x", pady=(0, 10))
15
16        frm.grid_columnconfigure(index=0, minsize=90)
17        frm.grid_columnconfigure(index=1, minsize=120)
18        frm.grid_columnconfigure(index=2, minsize=90)
19        frm.grid_columnconfigure(index=3, weight=1)
20        frm.grid_columnconfigure(index=4, minsize=90)
21        frm.grid_columnconfigure(index=5, minsize=120)
22        frm.grid_columnconfigure(index=6, minsize=120)
23
24        # Dòng 1
25        tk.Label(frm, text="Mã NV (*):", bg="white").grid(row=0, column=0, sticky="w")
26        self.e_ma = tk.Entry(frm, width=15)
27        self.e_ma.grid(row=0, column=1, padx=5, pady=5)
28
29        tk.Label(frm, text="Họ tên (*):", bg="white").grid(row=0, column=2, sticky="w")
30        self.e_ten = tk.Entry(frm, width=25)
31        self.e_ten.grid(row=0, column=3, padx=5, pady=5)
32
33        tk.Label(frm, text="Ngày sinh:", bg="white").grid(row=0, column=4, sticky="w")
34        self.e_ns = DateEntry(frm, date_pattern='yyyy-mm-dd', width=12)
35        self.e_ns.grid(row=0, column=5, padx=5, pady=5)
```

```

37         # Dòng 2
38         tk.Label(frm, text="Phòng ban:", bg="white").grid(row=1, column=0, sticky="w")
39         self.cb_pb = ttk.Combobox(frm, state="readonly", width=13)
40         self.cb_pb.grid(row=1, column=1, padx=5, pady=5)
41
42         tk.Label(frm, text="Chức vụ:", bg="white").grid(row=1, column=2, sticky="w")
43         self.cb_cv = ttk.Combobox(frm, state="readonly", width=23)
44         self.cb_cv.grid(row=1, column=3, padx=5, pady=5)
45
46         tk.Label(frm, text="User/Pass (*):", bg="white").grid(row=1, column=4, sticky="w")
47         self.e_user = tk.Entry(frm, width=12)
48         self.e_user.grid(row=1, column=5, padx=5, pady=5)
49         self.e_pass = tk.Entry(frm, width=12, show="*")
50         self.e_pass.grid(row=1, column=6, padx=5, pady=5)
51
52     # Nút chức năng
53     btn_box = tk.Frame(frm, bg="white")
54     btn_box.grid(row=2, columnspan=8, pady=10)
55
56     tk.Button(btn_box, text="Làm mới", bg="#95a5a6", fg="white", width=10, command=self.clear_form).pack(side="left", padx=5)
57     tk.Button(btn_box, text="Thêm", bg="#2980b9", fg="white", width=10, command=self.them).pack(side="left", padx=5)
58     tk.Button(btn_box, text="Sửa", bg="#f39c12", fg="white", width=10, command=self.sua).pack(side="left", padx=5)
59     tk.Button(btn_box, text="Xóa", bg="#c0392b", fg="white", width=10, command=self.xoa).pack(side="left", padx=5)
60
61     # --- BẢNG DỮ LIỆU ---
62     cols = ("Ma", "Ten", "PB", "CV", "User", "NgaySinh")
63     self.tree = ttk.Treeview(self, columns=cols, show="headings")
64
65     # Heading căn giữa
66     self.tree.heading("Ma", text="Mã NV", anchor="center")
67     self.tree.heading("Ten", text="Họ Tên", anchor="center")
68     self.tree.heading("PB", text="Phòng Ban", anchor="center")
69     self.tree.heading("CV", text="Chức Vụ", anchor="center")
70     self.tree.heading("User", text="Tài Khoản", anchor="center")
71     self.tree.heading("NgaySinh", text="Ngày Sinh", anchor="center")

```

```

52     # Nút chức năng
53     btn_box = tk.Frame(frm, bg="white")
54     btn_box.grid(row=2, columnspan=8, pady=10)
55
56     tk.Button(btn_box, text="Làm mới", bg="#95a5a6", fg="white", width=10, command=self.clear_form).pack(side="left", padx=5)
57     tk.Button(btn_box, text="Thêm", bg="#2980b9", fg="white", width=10, command=self.them).pack(side="left", padx=5)
58     tk.Button(btn_box, text="Sửa", bg="#f39c12", fg="white", width=10, command=self.sua).pack(side="left", padx=5)
59     tk.Button(btn_box, text="Xóa", bg="#c0392b", fg="white", width=10, command=self.xoa).pack(side="left", padx=5)
60
61     # --- BẢNG DỮ LIỆU ---
62     cols = ("Ma", "Ten", "PB", "CV", "User", "NgaySinh")
63     self.tree = ttk.Treeview(self, columns=cols, show="headings")
64
65     # Heading căn giữa
66     self.tree.heading("Ma", text="Mã NV", anchor="center")
67     self.tree.heading("Ten", text="Họ Tên", anchor="center")
68     self.tree.heading("PB", text="Phòng Ban", anchor="center")
69     self.tree.heading("CV", text="Chức Vụ", anchor="center")
70     self.tree.heading("User", text="Tài Khoản", anchor="center")
71     self.tree.heading("NgaySinh", text="Ngày Sinh", anchor="center")

```

```

73         # Căn dữ liệu từng cột sang trái (anchor="w")
74         for col in cols:
75             self.tree.column(col, anchor="w")
76
77         # Chiều rộng từng cột
78         self.tree.column("Ma", width=80)
79         self.tree.column("Ten", width=150)
80         self.tree.column("PB", width=120)
81         self.tree.column("CV", width=120)
82         self.tree.column("User", width=100)
83         self.tree.column("NgaySinh", width=100)
84
85         self.tree.pack(fill="both", expand=True)
86         self.tree.bind("<<TreeviewSelect>>", self.get_row)
87
88         self.pb_map = {}
89         self.cv_map = {}

```

```

91     # --- LOAD DỮ LIỆU ---
92     def load_data(self): 4 usages (1 dynamic)
93         pbs = fetch_all("SELECT MaPB, TenPB FROM phongban")
94         self.pb_map = {r['TenPB']: r['MaPB'] for r in pbs}
95         self.cb_pb['values'] = list(self.pb_map.keys())
96
97         cvs = fetch_all("SELECT MaCV, TenCV FROM chucvu")
98         self.cv_map = {r['TenCV']: r['MaCV'] for r in cvs}
99         self.cb_cv['values'] = list(self.cv_map.keys())
100
101         for i in self.tree.get_children():
102             self.tree.delete(i)
103         sql = """SELECT n.MaNV, n.TenNV, p.TenPB, c.TenCV, n.TenDangNhap, n.NgaySinh
104                FROM nhanvien n
105                LEFT JOIN phongban p ON n.MaPB=p.MaPB
106                LEFT JOIN chucvu c ON n.MaCV=c.MaCV"""
107         for r in fetch_all(sql):
108             self.tree.insert( parent: "", index: "end",
109                             values=(r['MaNV'], r['TenNV'], r['TenPB'], r['TenCV'], r['TenDangNhap'], r['NgaySinh']))

```



```

111 # --- CHON DUNG ---
112 def get_row(self, event): 1 usage
113     try:
114         sel = self.tree.selection()
115         if not sel: return
116         data = self.tree.item(sel[0], option="values")
117         self.clear_form()
118         self.e_ma.insert(index=0, data[0])
119         self.e_ma.config(state='disabled')
120         self.e_ten.insert(index=0, data[1])
121         if data[2]: self.cb_pb.set(data[2])
122         if data[3]: self.cb_cv.set(data[3])
123         self.e_user.insert(index=0, data[4])
124         try:
125             self.e_ns.set_date(data[5])
126         except:
127             pass
128     except:
129         pass
130

```

```

131 # --- CLEAR FORM ---
132 def clear_form(self): 5 usages
133     self.e_ma.config(state='normal')
134     self.e_ma.delete(first=0, tk.END)
135     self.e_ten.delete(first=0, tk.END)
136     self.e_user.delete(first=0, tk.END)
137     self.e_pass.delete(first=0, tk.END)
138     self.cb_pb.set('')
139     self.cb_cv.set('')
140     self.e_ma.focus()
141
142 # --- VALIDATE ---
143 def validate(self, is_add=True): 1 usage
144     if not self.e_ma.get().strip() or not self.e_ten.get().strip() or \
145         not self.e_user.get().strip() or (is_add and not self.e_pass.get().strip()):
146         messagebox.showwarning(title="Thiếu thông tin", message="Vui lòng nhập đầy đủ các trường có dấu (*)")
147         return False
148
149     if is_add:
150         chk = fetch_all(query="SELECT * FROM nhanvien WHERE MaNV=%s OR TenDangNhap=%s",
151             params=(self.e_ma.get(), self.e_user.get()))
152

```

```

152         if chk:
153             messagebox.showerror(title="Trùng lặp", message="Mã NV hoặc Tên đăng nhập đã tồn tại!")
154             return False
155         return True
156
157 # --- THÊM NHÂN VIÊN ---
158 def them(self): 1 usage
159     if not self.validate(is_add=True): return
160     try:
161         mpb = self.pb_map.get(self.cb_pb.get())
162         mcv = self.cv_map.get(self.cb_cv.get())
163         if execute_query(
164             query="INSERT INTO nhanvien(MaNV,TenNV,MaPB,MacV,TenDangNhap,MatKhau,NgaySinh) VALUES(%s,%s,%s,%s,%s,%s,%s)",
165             params=(self.e_ma.get(), self.e_ten.get(), mpb, mcv, self.e_user.get(), self.e_pass.get(), self.e_ns.get())
166         ):
167             self.load_data()
168             self.clear_form()
169             messagebox.showinfo(title="Thành công", message="Đã thêm nhân viên mới")
170     except Exception as e:
171         messagebox.showerror(title="Lỗi SQL", str(e))
172

```

```

173 # --- SỬA NHÂN VIÊN ---
174 def sua(self): 1 usage
175     if not self.e_ma.get(): return
176     try:
177         mpb = self.pb_map.get(self.cb_pb.get())
178         mcv = self.cv_map.get(self.cb_cv.get())
179         new_pass = self.e_pass.get().strip()
180         if new_pass:
181             sql = "UPDATE nhanvien SET TenNV=%s, MaPB=%s, MaCV=%s, NgaySinh=%s, MatKhau=%s WHERE MaNV=%s"
182             params = (self.e_ten.get(), mpb, mcv, self.e_ns.get(), new_pass, self.e_ma.get())
183         else:
184             sql = "UPDATE nhanvien SET TenNV=%s, MaPB=%s, MaCV=%s, NgaySinh=%s WHERE MaNV=%s"
185             params = (self.e_ten.get(), mpb, mcv, self.e_ns.get(), self.e_ma.get())
186         if execute_query(sql, params):
187             self.load_data()
188             self.clear_form()
189             messagebox.showinfo(title="Thành công", message="Đã cập nhật thông tin")
190     except Exception as e:
191         messagebox.showerror(title="Lỗi", str(e))

```

```

193 # --- XÓA NHÂN VIÊN ---
194 def xoa(self): 1 usage
195     sel = self.tree.selection()
196     if not sel:
197         messagebox.showwarning(title="Chú ý", message="Chưa chọn nhân viên cần xóa")
198         return
199     manv = self.tree.item(sel[0])['values'][0]
200     if manv == 'ADMIN':
201         messagebox.showerror(title="Cấm", message="Không thể xóa tài khoản Admin!")
202         return
203     if messagebox.askyesno(title="Xác nhận", message=f"Bạn có chắc muốn xóa nhân viên {manv}?"):
204         if execute_query(query="DELETE FROM nhanvien WHERE MaNV=%s", params=(manv,)):
205             self.load_data()
206             self.clear_form()

```

## Bảng hợp đồng:

```
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 from tkcalendar import DateEntry
4 from database import fetch_all, execute_query
5
6
7 class HopDongPage(tk.Frame): 3 usages
8     def __init__(self, parent, controller):
9         tk.Frame.__init__(self, parent, bg="#ecf0f1")
10        self.controller = controller
11
12        # --- FORM ---
13        frm = tk.LabelFrame(self, text="Hợp đồng lao động", bg="white", padx=10, pady=10)
14        frm.pack(fill="x", pady=(0, 10))
15
16        tk.Label(frm, text="Số HD (*):", bg="white").grid(row=0, column=0, sticky="w")
17        self.e_so = tk.Entry(frm);
18        self.e_so.grid(row=0, column=1, padx=5, pady=5)
19
20        tk.Label(frm, text="Nhân viên (*):", bg="white").grid(row=0, column=2, sticky="w")
21        self.cb_nv = ttk.Combobox(frm, state="readonly", width=25);
22        self.cb_nv.grid(row=0, column=3, padx=5, pady=5)
23
24        tk.Label(frm, text="Loại HD:", bg="white").grid(row=0, column=4, sticky="w")
25        self.cb_loai = ttk.Combobox(frm, values=["12 Tháng", "36 Tháng", "Vô thời hạn", "Thử việc"], state="readonly")
26        self.cb_loai.current(0)
27        self.cb_loai.grid(row=0, column=5, padx=5, pady=5)
28
29        tk.Label(frm, text="Ngày BD:", bg="white").grid(row=0, column=6, sticky="w")
30        self.e_ngay = DateEntry(frm, date_pattern='yyyy-mm-dd');
31        self.e_ngay.grid(row=0, column=7, padx=5, pady=5)
32
33        # Buttons
34        btn_box = tk.Frame(frm, bg="white")
35        btn_box.grid(row=1, columnspan=8, pady=10)
36        tk.Button(btn_box, text="Làm mới", command=self.clear, bg="#95a5a6", fg="white", width=10).pack(side="left",
37                                                                                                     padx=5)
38        tk.Button(btn_box, text="Ký HD (Thêm)", command=self.them, bg="#27ae60", fg="white", width=12).pack(side="left",
39                                                                                                     padx=5)
40        tk.Button(btn_box, text="Xóa", command=self.xoa, bg="#c0392b", fg="white", width=10).pack(side="left",
41                                                                                               padx=5)
42        tk.Button(btn_box, text="Hủy HD", command=self.xoa, bg="#c0392b", fg="white", width=10).pack(side="left",
43                                                                                               padx=5)
44
45        # --- TABLE ---
46        self.tree = ttk.Treeview(self, columns=("So", "Ten", "Loai", "Ngay"), show="headings")
47        self.tree.heading("So", text="Số HD");
48        self.tree.heading("Ten", text="Nhân viên");
49        self.tree.heading("Loai", text="Loại HD");
50        self.tree.heading("Ngay", text="Ngày Bắt đầu");
51        self.tree.pack(fill="both", expand=True)
52
53        self.tree.bind("<<TreeviewSelect>>", self.get_row)
54        self.nv_map = {}
55
56    def load_data(self): 4 usages (1 dynamic)
57        nvs = fetch_all("SELECT MaNV, TenNV FROM nhanvien")
58        # Format map: "Tên NV (Mã)" để dễ tìm
59        self.nv_map = {f'{r["TenNV"]}' (f'{r["MaNV"]}'): r["MaNV"] for r in nvs}
60        self.cb_nv['values'] = list(self.nv_map.keys())
61
62        for i in self.tree.get_children(): self.tree.delete(i)
63        sql = "SELECT h.SoHD, n.TenNV, h.LoaiHD, h.NgayBatDau, n.MaNV FROM hopdong h JOIN nhanvien n ON h.MaNV=n.MaNV"
```

```

64         for r in fetch_all(sql):
65             # Lưu MaNV ẩn vào tag hoặc dùng map để tìm lại, ở đây ta hiển thị TenNV
66             self.tree.insert( parent="", index="end", values=(r['SoHD'], r['TenNV'], r['LoaiHD'], r['NgayBatDau'], r['MaNV']))
67
68     def get_row(self, event): 1 usage
69         try:
70             sel = self.tree.selection()
71             if not sel: return
72             vals = self.tree.item(sel[0])['values']
73             # vals: 0=SoHD 1=TenNV 2=LoaiHD 3=Nngay 4=MaNV (cột ẩn nếu có ở đầu ta lấy từ query)
74             self.clear()
75             self.e_so.insert(index=0, vals[0])
76             self.e_so.config(state='disabled')
77
78             # Tự động chọn Combobox dựa vào Tên hoặc Mã
79             # Do ta lấy thêm cột MaNV ở vị trí index 4 (dù không hiện header nhưng values vẫn có nếu insert đủ)
80             target_ma = str(vals[4])
81             for k, v in self.nv_map.items():
82                 if v == target_ma:
83                     self.cb_nv.set(k)
84                     break
85
86             self.cb_loai.set(vals[2])
87             self.e_ngay.set_date(vals[3])
88         except:
89             pass
90

```

```

91     # --- LOAD DỮ LIỆU ---
92     def load_data(self): 4 usages (1 dynamic)
93         pbs = fetch_all("SELECT MaPB, TenPB FROM phongban")
94         self.pb_map = {r['TenPB']: r['MaPB'] for r in pbs}
95         self.cb_pb['values'] = list(self.pb_map.keys())
96
97         cvs = fetch_all("SELECT MaCV, TenCV FROM chucvu")
98         self.cv_map = {r['TenCV']: r['MaCV'] for r in cvs}
99         self.cb_cv['values'] = list(self.cv_map.keys())
100
101         for i in self.tree.get_children():
102             self.tree.delete(i)
103         sql = """SELECT n.MaNV, n.TenNV, p.TenPB, c.TenCV, n.TenDangNhap, n.NgaySinh
104             FROM nhanvien n
105             LEFT JOIN phongban p ON n.MaPB=p.MaPB
106             LEFT JOIN chucvu c ON n.MaCV=c.MaCV"""
107         for r in fetch_all(sql):
108             self.tree.insert( parent="", index="end",
109                             values=(r['MaNV'], r['TenNV'], r['TenPB'], r['TenCV'], r['TenDangNhap'], r['NgaySinh']))
110

```

```

111     def sua(self): 1 usage
112         if not self.e_so.get(): return
113         mnv = self.nv_map.get(self.cb_nv.get())
114         if execute_query( query="UPDATE hopdong SET MaNV=%s, LoaiHD=%s, NgayBatDau=%s WHERE SoHD=%s",
115                         params=(mnv, self.cb_loai.get(), self.e_ngay.get(), self.e_so.get())):
116             self.load_data();
117             self.clear();
118             messagebox.showinfo( title="OK", message="Đã cập nhật")
119
120     def xoa(self): 1 usage
121         sel = self.tree.selection()
122         if sel and messagebox.askyesno( title="Xóa", message="Hủy hợp đồng này?"):
123             execute_query( query="DELETE FROM hopdong WHERE SoHD=%s", params=(self.tree.item(sel[0])['values'][0],))
124             self.load_data();
125             self.clear()

```