

TRƯỜNG ĐẠI HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO 2
LẬP TRÌNH PYTHON NÂNG CAO

Chủ đề:

App Tính Lương

HỌ VÀ TÊN: LƯƠNG NHẬT QUANG
MSSV: 2274802010710

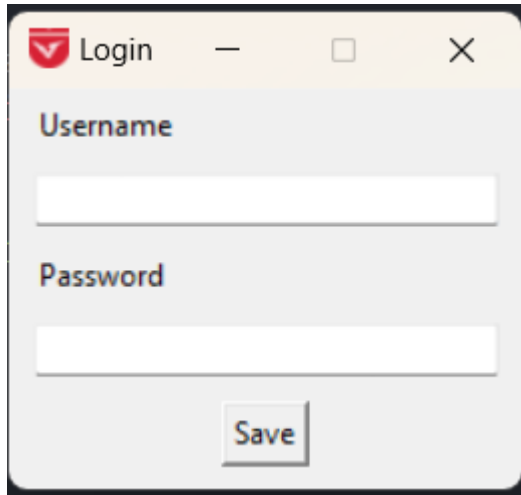
TP. Hồ Chí Minh – 7/2024

Contents

No table of contents entries found.

1. Giao diện

a. Giao diện Login

A screenshot of a login dialog box. The dialog has a title bar with a red icon containing a white checkmark, the text "Login", and standard window controls (minimize, maximize, close). The main area is light gray and contains two white input fields. The first field is labeled "Username" and the second is labeled "Password". Below the input fields is a button labeled "Save".

b. Giao diện mặc định (Nhân viên)

Tính lương

File Help

Nhan Vien Quan Ly

Luong Nhan Vien

Ho Ten:

Luong Co Ban:

So Ngay Lam Viec:

Tro Cap:

Save

Timkiem

Nhan vien: Search

Clear

c. Giao diện mặc định (Quản lý)

Tinh luong

File Help

Nhan Vien Quan Ly

Luong Quan Ly

Ho Ten:

Luong Co Ban

He So Chuc Vu

Thuong

Save

Ket Qua

Quan ly: Search

Clear

d. Tab File, Tab Help

Tinh luong

File Help

Exit Quan Ly

Luong Quan Ly

Ho Ten:

Luong Co Ban

He So Chuc Vu

Thuong

Save

Ket Qua

Quan ly: Search

Clear

Tinh luong

File Help

Nhập About

Luong Quan Ly

Ho Ten:

Luong Co Ban

He So Chuc Vu

Thuong

Save

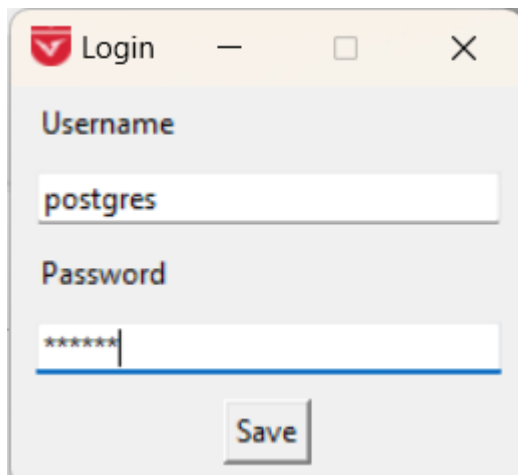
Ket Qua

Quan ly: Search

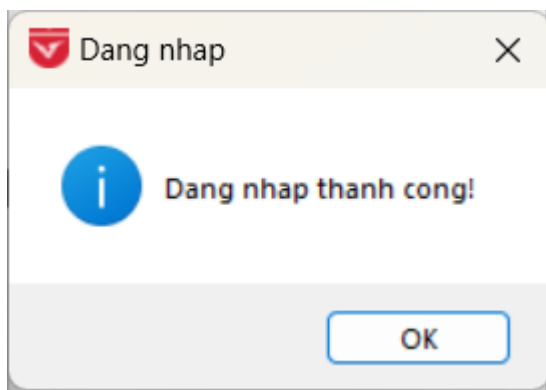
Clear

2. Chức năng

- Login, đồng thời connect vào database khi đăng nhập thành công

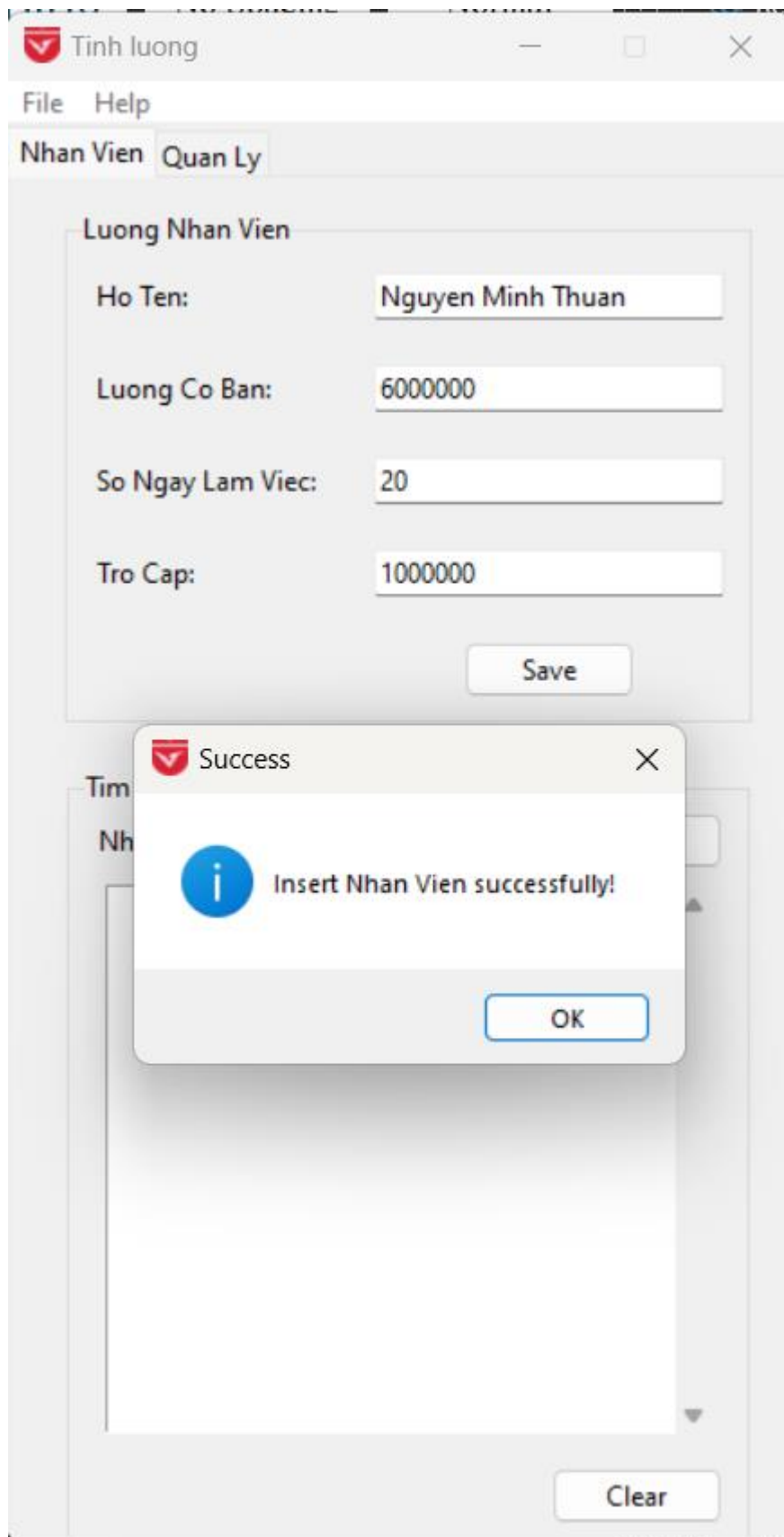


A screenshot of a 'Login' dialog box. The title bar contains a red checkmark icon, the text 'Login', and standard window controls (minimize, maximize, close). The dialog has two input fields: 'Username' with the text 'postgres' and 'Password' with masked characters '*****'. A 'Save' button is located at the bottom right.



A screenshot of a 'Dang nhap' (Login) dialog box. The title bar contains a red checkmark icon, the text 'Dang nhap', and a close button. The main area features a blue information icon and the text 'Dang nhap thanh cong!' (Login successful!). An 'OK' button is at the bottom right.

- b. Tính lương nhân viên và lưu vào csdl khi nhấn Save



- c. Tìm kiếm thông tin của nhân viên khi nhấn Search (“*” là tìm tất cả, “name” để tìm từng cá nhân)

Tính lương

File Help

Nhan VienQuan Ly

Luong Nhan Vien

Ho Ten:

Nguyen Minh Thuan

Luong Co Ban:

6000000

So Ngay Lam Viec:

20

Tro Cap:

1000000

Save

Timkiem

Nhan vien:

*

Search

1 Quang 0.0 0.0 0.0 0.0

2 Luong Nhat Quang 6000000.0

22.0 1000000.0 11400000.0

3 Nguyen Minh Thuan 6000000.0

20.0 1000000.0 11000000.0

Clear

Tính lương

File Help

Nhan Vien Quan Ly

Luong Nhan Vien

Ho Ten:

Luong Co Ban:

So Ngay Lam Viec:

Tro Cap:

Save

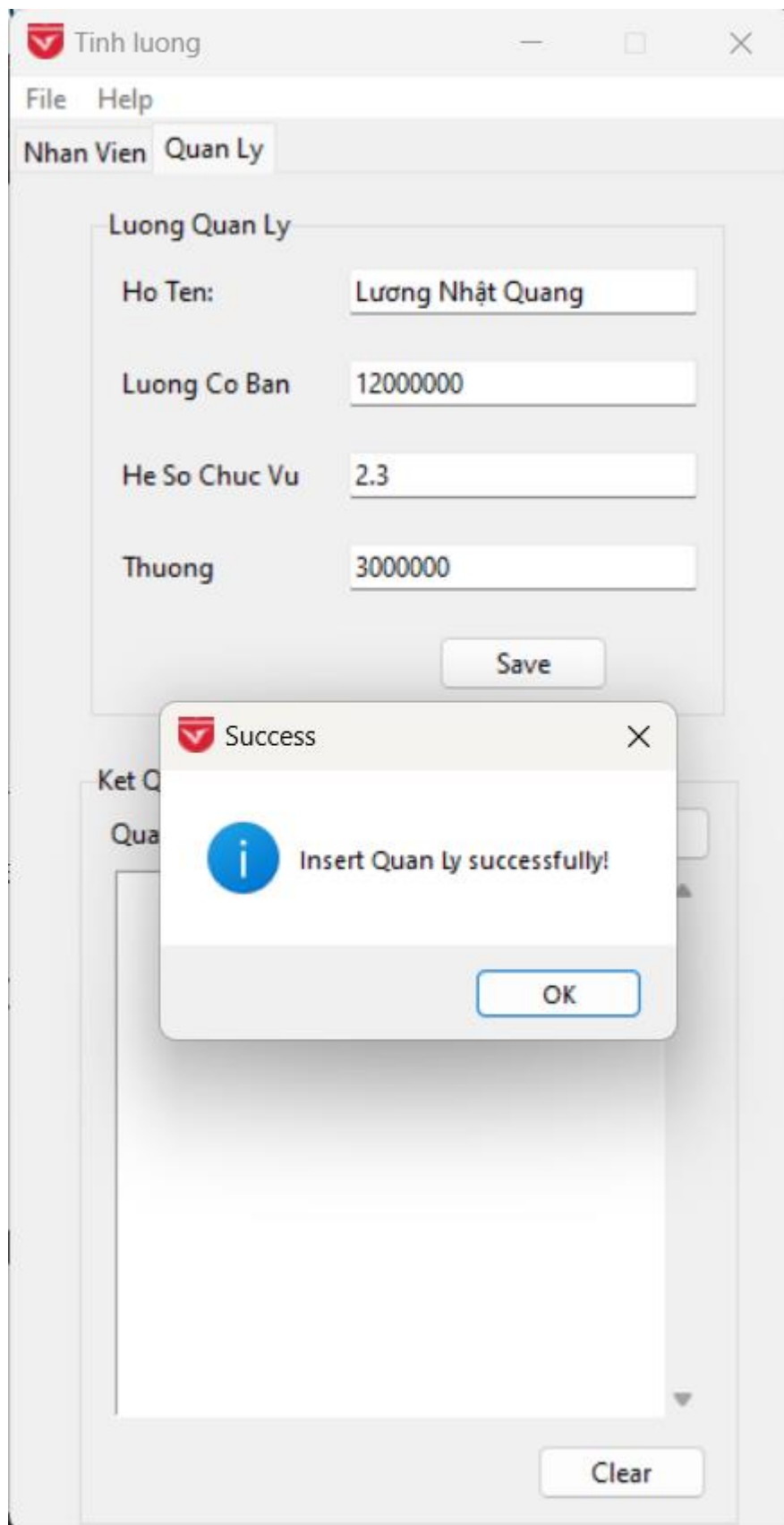
Timkiem

Nhan vien:

2 Luong Nhat Quang 6000000.0
22.0 1000000.0 11400000.0

Clear

d. Tính lương quản lý và lưu vào csdl khi nhấn Save



- e. Tìm kiếm thông tin của quản lý khi nhấn Search (“*” là tìm tất cả, “name” để tìm từng cá nhân)

Tính lương

File Help

Nhan Vien Quan Ly

Luong Quan Ly

Ho Ten: Lương Nhật Quang

Luong Co Ban 12000000

He So Chuc Vu 2.3

Thuong 3000000

Save

Ket Qua

Quan ly: * Search

```
1 QUang 0.0 0.0 0.0 0.0
2 Lương Nhật Quang 12000000.0
2.3 3000000.0
30599999.999999996
```

Clear

Tính lương

File Help

Nhan Viên Quan Ly

Luong Quan Ly

Ho Ten: Lương Nhật Quang

Luong Co Ban 12000000

He So Chuc Vu 2.3

Thuong 3000000

Save

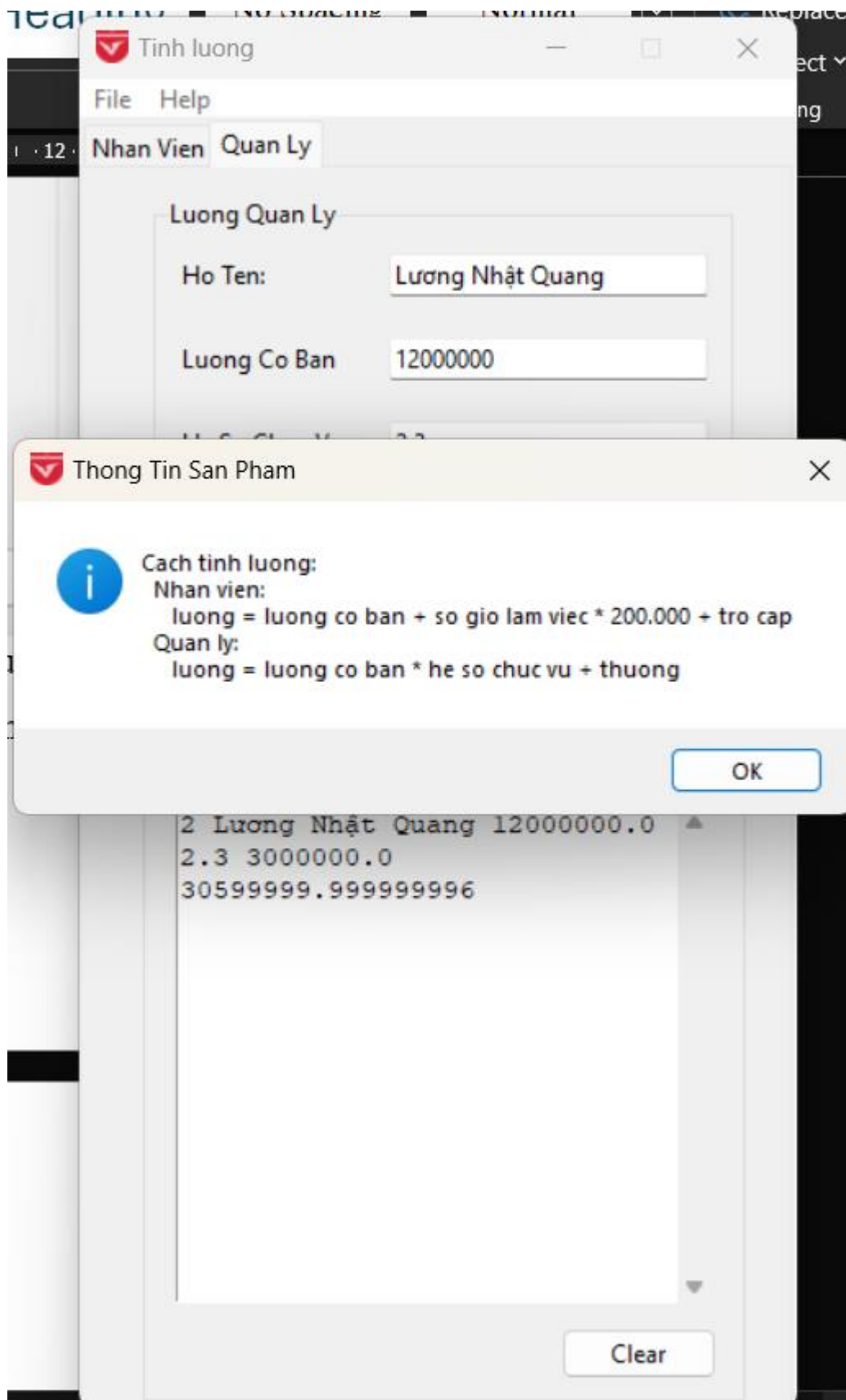
Ket Qua

Quan ly: ơng Nhật Quang Search

2 Lương Nhật Quang 12000000.0
2.3 3000000.0
30599999.999999996

Clear

- f. Thông báo khi nhấn vào Help -> About sẽ cung cấp cho người dùng cách lương của từng bộ phận được tính như nào



3. Mã chương trình

a. class **TinhLuong**:

a. def `__init__`

```

class TinhLuong:
    def __init__(self, root, conn):
        self.root = root
        self.root.title("Tinh luong")
        self.root.resizable(False,False)

        self.conn = conn
        self.cur = self.conn.cursor()

        # Kích thước cửa sổ
        window_width = 350
        window_height = 650

        # Lấy kích thước màn hình
        screen_width = self.root.winfo_screenwidth()
        screen_height = self.root.winfo_screenheight()

        # Tính toán tọa độ x và y để đặt cửa sổ ở trung tâm
        center_x = int(screen_width/2 - window_width/2)
        center_y = int(screen_height/2 - window_height/2)

        # Đặt kích thước và vị trí cửa sổ
        self.root.geometry(f'{window_width}x{window_height}+{center_x}+{center_y}')

        #Đặt logo co form
        self.root.iconbitmap('2274802010710_LuongNhatQuang_BaoCao2/IconDHVL.ico')

        #menu bar
        menu_bar = Menu(self.root)
        self.root.config(menu=menu_bar)
        file_menu = Menu(menu_bar, tearoff=0)
        file_menu.add_command(label="Exit", command=exit)
        menu_bar.add_cascade(label="File", menu=file_menu)

        help_menu = Menu(menu_bar, tearoff=0)
        help_menu.add_command(label="About", command=self.ThongTinSanPham)
        menu_bar.add_cascade(label="Help", menu=help_menu)

        tabControl = ttk.Notebook(self.root)
        tabControl.pack(expand=1, fill="both")

        self.NhanVien(tabControl)
        self.QuanLy(tabControl)

```

b. def NhanVien

```

def NhanVien(self, tabControl):
    #tab Nhan Vien
    tabNhanVien = ttk.Frame(tabControl)
    tabControl.add(tabNhanVien, text="Nhan Vien")

    #frame Nhan Vien
    frame_NhanVien = ttk.LabelFrame(tabNhanVien, text="Luong Nhan Vien")
    frame_NhanVien.grid(column=0, row=0, padx=25, pady=15)

    #label Tinh
    lbl_name = ttk.Label(frame_NhanVien, text="Ho Ten: ")
    lbl_name.grid(column=0, row=0, padx=10, pady=10, sticky="W")

    lbl_lcb = ttk.Label(frame_NhanVien, text="Luong Co Ban: ")
    lbl_lcb.grid(column=0, row=1, padx=10, pady=10, sticky="W")

    lbl_snlv = ttk.Label(frame_NhanVien, text="So Ngay Lam Viec: ")
    lbl_snlv.grid(column=0, row=2, padx=10, pady=10, sticky="W")

    lbl_trocap = ttk.Label(frame_NhanVien, text="Tro Cap: ")
    lbl_trocap.grid(column=0, row=3, padx=10, pady=10, sticky="W")

    #textbox
    self.name_nhanvien = tk.StringVar()
    txt_name = ttk.Entry(frame_NhanVien, width=25, textvariable=self.name_nhanvien)
    txt_name.grid(column=1, row=0, padx=10, pady=10)
    txt_name.focus()

    self.lcb_nhanvien = tk.DoubleVar()
    txt_lcb = ttk.Entry(frame_NhanVien, width=25, textvariable=self.lcb_nhanvien)
    txt_lcb.grid(column=1, row=1, padx=10, pady=10)

    self.snlv = tk.DoubleVar()
    txt_sglv = ttk.Entry(frame_NhanVien, width=25, textvariable=self.snlv)
    txt_sglv.grid(column=1, row=2, padx=10, pady=10)

    self.trocap = tk.DoubleVar()
    txt_trocap = ttk.Entry(frame_NhanVien, width=25, textvariable=self.trocap)
    txt_trocap.grid(column=1, row=3, padx=10, pady=10)

```



```

#btn Tinh
ttk.Button(frame_NhanVien, text="Save", command=self.insert_DB_NhanVien).grid(column=1, row=4, pady=10)

#frame Tinh Luong Nhan Vien
frame_LuongNhanVien = ttk.LabelFrame(tabNhanVien, text="Tim kiem")
frame_LuongNhanVien.grid(column=0, row=1, padx=20, pady=5)

#label luong
lbl_luong = ttk.Label(frame_LuongNhanVien, text="Nhan vien:")
lbl_luong.grid(column=0, row=0, padx=10, pady=2, sticky="W")

#input Search
self.search_nhanvien = tk.StringVar()
txt_search = ttk.Entry(frame_LuongNhanVien, width=16, textvariable=self.search_nhanvien)
txt_search.grid(column=1, row=0, padx=10, pady=2)

#btn Search
ttk.Button(frame_LuongNhanVien, text="Search", command=self.Search_LuongNhanVien).grid(column=2, row=0, sticky="E", padx=10, pady=2)

#scrolledtext
scr_w = 31
scr_h = 15
self.scroll_NhanVien = scrolledtext.ScrolledText(frame_LuongNhanVien, width = scr_w, height = scr_h, wrap = tk.WORD)
self.scroll_NhanVien.grid(column=0, row=1, columnspan=3, padx=10, pady=5)

#btn Clear
ttk.Button(frame_LuongNhanVien, text="Clear", command=self.Clear_LuongNhanVien).grid(column=2, row=2, sticky="E", padx=10, pady=10)

```

c. def QuanLy

```

def QuanLy(self, tabControl):
    #tab Quan Ly
    tabQuanLy = ttk.Frame(tabControl)
    tabControl.add(tabQuanLy, text="Quan Ly")

    #frame Nhan Vien
    frame_QuanLy = ttk.LabelFrame(tabQuanLy, text="Luong Quan Ly")
    frame_QuanLy.grid(column=0, row=0, padx=30, pady=15)

    #label Tinh
    lbl_name = ttk.Label(frame_QuanLy, text="Ho Ten: ")
    lbl_name.grid(column=0, row=0, padx=10, pady=10, sticky="W")

    lbl_lcb = ttk.Label(frame_QuanLy, text="Luong Co Ban")
    lbl_lcb.grid(column=0, row=1, padx=10, pady=10, sticky="W")

    lbl_snlv = ttk.Label(frame_QuanLy, text="He So Chuc Vu")
    lbl_snlv.grid(column=0, row=2, padx=10, pady=10, sticky="W")

    lbl_trocap = ttk.Label(frame_QuanLy, text="Thuong")
    lbl_trocap.grid(column=0, row=3, padx=10, pady=10, sticky="W")

    #textbox
    self.name_quanly = tk.StringVar()
    txt_name = ttk.Entry(frame_QuanLy, width=25, textvariable=self.name_quanly)
    txt_name.grid(column=1, row=0, padx=10, pady=10)
    txt_name.focus()

    self.lcb_quanly = tk.DoubleVar()
    txt_lcb = ttk.Entry(frame_QuanLy, width=25, textvariable=self.lcb_quanly)
    txt_lcb.grid(column=1, row=1, padx=10, pady=10)

    self.hscv = tk.DoubleVar()
    txt_hscv = ttk.Entry(frame_QuanLy, width=25, textvariable=self.hscv)
    txt_hscv.grid(column=1, row=2, padx=10, pady=10)

    self.thuong = tk.DoubleVar()
    txt_thuong = ttk.Entry(frame_QuanLy, width=25, textvariable=self.thuong)
    txt_thuong.grid(column=1, row=3, padx=10, pady=10)

    #btn Tinh
    ttk.Button(frame_QuanLy, text="Save", command=self.insert_DB_QuanLy).grid(column=1, row=4, pady=10)

    #frame Tinh Luong Nhan Vien
    frame_LuongQuanLy = ttk.LabelFrame(tabQuanLy, text="Ket Qua")
    frame_LuongQuanLy.grid(column=0, row=1, padx=30, pady=5)

    #label luong
    lbl_luong = ttk.Label(frame_LuongQuanLy, text="Quan ly:")
    lbl_luong.grid(column=0, row=0, padx=10, pady=2, sticky="W")

    #input search
    self.search_quanly = tk.StringVar()
    txt_search = ttk.Entry(frame_LuongQuanLy, width=16, textvariable=self.search_quanly)
    txt_search.grid(column=1, row=0, padx=10, pady=2)

    #btn Search
    ttk.Button(frame_LuongQuanLy, text="Search", command=self.Search_LuongQuanLy).grid(column=2, row=0, sticky="E", padx=10, pady=2)

    #scrolledtext
    scr_w = 30
    scr_h = 15
    self.scroll_QuanLy = scrolledtext.ScrolledText(frame_LuongQuanLy, width = scr_w, height = scr_h, wrap = tk.WORD)
    self.scroll_QuanLy.grid(column=0, row=1, columnspan=3, padx=10, pady=2)

    #btn Clear
    ttk.Button(frame_LuongQuanLy, text="Clear", command=self.Clear_LuongQuanLy).grid(column=2, row=2, sticky="E", padx=12, pady=10)

```

d. def LuongNhanVien

```

def LuongNhanVien(self):
    try:
        lcb = self.lcb_nhanvien.get()
        snlv = self.snlv.get()
        trocap = self.trocap.get()
        return lcb + snlv * 200000 + trocap
    except Exception as ex:
        messagebox.showerror("Input Error", "Lỗi dữ liệu đầu vào!\nVui lòng xem và nhập lại")

```

e. def LuongQuanLy

```

def LuongQuanLy(self):
    try:
        lcb = self.lcb_quanly.get()
        hscv = self.hscv.get()
        thuong = self.thuong.get()
        return lcb * hscv + thuong
    except Exception as ex:
        messagebox.showerror("Input Error", "Lỗi dữ liệu đầu vào!\nVui lòng xem và nhập lại")

```

f. def ThongTinSanPham

```

def ThongTinSanPham(self):
    messagebox.showinfo("Thong Tin San Pham", "Cach tinh luong:\n Nhan vien: \n      luong = luong co ban + so gio lam viec * 200.000 + tro cap\n Quan ly: \n      luong

```

g. def Clear_LuongNhanVien

```

def Clear_LuongNhanVien(self):
    self.scroll_NhanVien.delete(1.0, tk.END)

```

h. def Clear_LuongQuanLy

```

def Clear_LuongQuanLy(self):
    self.scroll_QuanLy.delete(1.0, tk.END)

```

i. def insert_DB_NhanVien

```

def insert_DB_NhanVien(self):
    try:
        if (self.lcb_nhanvien.get() >= 0 and self.snlv.get() >= 0 and self.trocap.get() >= 0) and (self.name_nhanvien.get().strip()):
            insert_query = sql.SQL("INSERT INTO {} (hoten, luongcoban, songaylamviec, trocap, luong) VALUES (%s, %s, %s, %s, %s)").format(sql.Identifier("nhanvien"))
            data_to_insert = (self.name_nhanvien.get(), self.lcb_nhanvien.get(), self.snlv.get(), self.trocap.get(), self.LuongNhanVien())
            self.cur.execute(insert_query, data_to_insert)
            self.conn.commit()
            messagebox.showinfo("Success", "Insert Nhan Vien successfully!")
        else:
            messagebox.showwarning("Warning", "Please check your input.")
    except Exception as e:
        messagebox.showerror("Error", f"Error inserting data: {e}")

```

j. def insert_DB_QuanLy

```

def insert_DB_QuanLy(self):
    try:
        if (self.lcb_quanly.get() >= 0 and self.hscv.get() >= 0 and self.thuong.get() >= 0) and (self.name_quanly.get().strip()):
            insert_query = sql.SQL("INSERT INTO {} (hoten, luongcoban, hesochucvu, thuong, luong) VALUES (%s, %s, %s, %s, %s)".format(sql.Identifier("quanly")))
            data_to_insert = (self.name_quanly.get(), self.lcb_quanly.get(), self.hscv.get(), self.thuong.get(), self.LuongQuanLy())
            self.cur.execute(insert_query, data_to_insert)
            self.conn.commit()
            messagebox.showinfo("Success", "Insert Quan Ly successfully!")
        else:
            messagebox.showwarning("Warning", "Please check your input.")
    except Exception as e:
        messagebox.showerror("Error", f"Error inserting data: {e}")

```

k. def Search_LuongNhanVien

```

def Search_LuongNhanVien(self):
    try:
        timkiem = self.search_nhanvien.get()
        query_value = sql.SQL("SELECT * FROM {} WHERE hoten = %s").format(sql.Identifier("nhanvien"))
        self.cur.execute(query_value, (timkiem,))
        row_value = self.cur.fetchall()

        if timkiem == "*":
            query = sql.SQL("SELECT * FROM {}").format(sql.Identifier("nhanvien"))
            self.cur.execute(query)
            rows = self.cur.fetchall()
            self.scroll_NhanVien.delete(1.0, tk.END)
            for values in rows:
                for value in values:
                    self.scroll_NhanVien.insert(tk.END, f"{value} ")
                self.scroll_NhanVien.insert(tk.END, "\n")
        elif not row_value:
            messagebox.showinfo("Thong Bao", "Khong tim thay nhan vien \nhoac do nhap sai")
        else:
            self.scroll_NhanVien.delete(1.0, tk.END)
            for values in row_value:
                for value in values:
                    self.scroll_NhanVien.insert(tk.END, f"{value} ")

    except Exception as e:
        messagebox.showerror("Error", f"Error loading data: {e}")

```

l. def Search_LuongQuanLy

```

def Search_LuongQuanLy(self):
    try:
        timkiem = self.search_quanly.get()
        query_value = sql.SQL("SELECT * FROM {} WHERE hoten = %s").format(sql.Identifier("quanly"))
        self.cur.execute(query_value, (timkiem,))
        row_value = self.cur.fetchall()

        if timkiem == "":
            query = sql.SQL("SELECT * FROM {}").format(sql.Identifier("quanly"))
            self.cur.execute(query)
            rows = self.cur.fetchall()
            self.scroll_QuanLy.delete(1.0, tk.END)
            for values in rows:
                for value in values:
                    self.scroll_QuanLy.insert(tk.END, f"{value} ")
                    self.scroll_QuanLy.insert(tk.END, "\n")
            elif not row_value:
                messagebox.showinfo("Thong Bao", "Khong tim thay quan ly \nhoac do nhap sai")
            elif row_value:
                self.scroll_QuanLy.delete(1.0, tk.END)
                for values in row_value:
                    for value in values:
                        self.scroll_QuanLy.insert(tk.END, f"{value} ")

    except Exception as e:
        messagebox.showerror("Error", f"Error loading data: {e}")

```

b. class Login:

i. def __init__

```

class Login:
    def __init__(self, win):
        self.win = win
        self.win.title("Login")
        self.win.resizable(False, False)

        # Kích thước cửa sổ
        window_width = 205
        window_height = 160

        # Lấy kích thước màn hình
        screen_width = self.win.winfo_screenwidth()
        screen_height = self.win.winfo_screenheight()

        # Tính toán tọa độ x và y để đặt cửa sổ ở trung tâm
        center_x = int(screen_width/2 - window_width/2)
        center_y = int(screen_height/2 - window_height/2)

        # Đặt kích thước và vị trí cửa sổ
        self.win.geometry(f'{window_width}x{window_height}+{center_x}+{center_y}')

        # Đặt logo có form
        self.win.iconbitmap('2274802010710_LuongNhatQuang_BaoCao2/IconDHVL.ico')

        self.database = "Management"
        self.host = "localhost"
        self.port = "5432"
        self.user = None
        self.password = None

        ttk.Label(self.win, text="Username").grid(column=0, row=0, padx=10, pady=5, sticky="W")

        self.username_entry = tk.StringVar()
        username_entry = ttk.Entry(self.win, width=30, textvariable=self.username_entry)
        username_entry.grid(column=0, row=1, padx=10, pady=5, sticky="W")

        ttk.Label(self.win, text="Password").grid(column=0, row=2, padx=10, pady=5, sticky="W")

        self.password_entry = tk.StringVar()
        password_entry = ttk.Entry(self.win, width=30, textvariable=self.password_entry, show="*")
        password_entry.grid(column=0, row=3, padx=10, pady=5, sticky="W")

        tk.Button(self.win, text="Save", command=self.save_and_switch).grid(column=0, row=4, pady=5)

```

```

if __name__ == "__main__":
    win = tk.Tk()
    Login(win)
    win.mainloop()

```

ii. def conn_DB

```
def conn_DB(self):
    try:
        self.conn = psycopg2.connect(database = self.database, user = self.user, password = self.password, host = self.host, port = self.port)
        messagebox.showinfo("Đăng nhập", "Đăng nhập thành công!")
        return True
    except Exception as e:
        messagebox.showerror("Lỗi đăng nhập!", "Đăng nhập không thành công")
        return False
```

iii. def save_and_switch

```
def save_and_switch(self):
    try:
        self.user = self.username_entry.get()
        self.password = self.password_entry.get()
        if self.conn_DB():
            self.win.destroy()
            root = tk.Tk()
            TinhLuong(root, self.conn)
            root.mainloop()
    except Exception as e:
        pass
```

4. Github

[GitHub - NhatQuangIT1301/pythonnc](https://github.com/NhatQuangIT1301/pythonnc)