TRƯỜNG ĐẠI HỌC HỌC VĂN LANG KHOA CÔNG NGHỆ THÔNG TIN





BÁO CÁO ĐÒ ÁN MÔN HỌC

LẬP TRÌNH PYTHON NÂNG CAO

NGÀNH: CÔNG NGHỆ THÔNG TIN

Tên Đồ án:

TẠO HỆ THỐNG QUẨN LÝ SINH VIÊN (PYTHON GUI)

SVTH: Vũ Hoàng Phi Hùng

MSSV: 2274802010296

GVHD: Tổng Hùng Anh

Tp. Hồ Chí Minh – năm 2024

LÒI CẢM ƠN

Viết một báo cáo đồ án môn học là một trong những việc khó nhất mà chúng em phải hoàn thành trong quá trình học một môn học. Trong quá trình thực hiện đề tài chúng em đã gặp rất nhiều khó khăn và bỡ ngỡ. Nếu không có những sự giúp đỡ và lời động viên chân thành của nhiều người có lẽ chúng em khó có thể hoàn thành tốt tiểu luận này. Đầu tiên chúng em xin gửi lời biết ơn chân thành đến thầy NV A, dẫn chúng em truc tiếp hướng hoàn thành tiểu luân người Những ý kiến đóng góp của thầy là vô cùng hữu ích, nó giúp chúng em nhân ra các khuyết điểm của đồ án. Cảm on thầy và các bạn trường Đại học Văn Lang là những người đã cùng nhóm em sát cánh và trải nghiệm để hoàn thành đồ án môn học. Nhóm thực hiện báo cáo

MỤC LỤC

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT – HỆ THÔNG QUẨN LÝ SINH VIÊN	4
1.1 Lý do chọn đề tài	4
1.2. Mục tiêu của đề tài	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT – HỆ THỐNG QUẨN LÝ SINH VIÊN	4
2.1 Khái niệm về hệ thống quản lý sinh viên	4
2.2 Các tính năng của hệ thống quản lý sinh viên	4
2.3 Các công nghệ sử dụng trong dự án	7
CHƯƠNG 3: SOURCE CODE	7
3.1 Các gói thư viện	7
3.2 Trang Đăng nhập và tính năng	7
3.3 Khởi tạo lớp và thiết lập giao diện	8
3.4 Các trường nhập liệu	g
3.5 Bảng hiển thị thông tin sinh viên	S
3.6 Các nút Action	10
3.7 Chức năng tìm kiếm	10
3.8 Thanh Menu	10
3.9 Tính Năng save thông tin	11
3.10 Tính năng Delete sinh viên	11
3.11 Tính năng tìm Search sinh viên	12
3.12 Tính năng show all sinh viên	12
3.13 Liên Kết với Database	12
CHƯƠNG 4 NHẬN XÉT	13
CHLIONC 5 LÉT I HÂN	13

CHƯƠNG 1. CƠ SỞ LÝ THUYẾT – HỆ THỐNG QUẢN LÝ SINH VIÊN

1.1 Lý do chọn đề tài

- Quản lý sinh viên là một trong những công việc quan trọng của các cơ sở giáo dục. Với sự phát triển của công nghệ thông tin, việc áp dụng các hệ thống quản lý sinh viên điện tử giúp việc quản lý trở nên dễ dàng, nhanh chóng và hiệu quả hơn. Vì vậy, tôi quyết định thực hiện đồ án "Hệ thống quản lý sinh viên" nhằm ứng dụng công nghệ vào việc quản lý thông tin sinh viên tại một trường học hoặc cơ sở giáo dục.

1.2. Mục tiêu của đề tài

- Mục tiêu chính của đồ án là xây dựng một hệ thống phần mềm quản lý sinh viên giúp lưu trữ và quản lý thông tin sinh viên và các thông tin liên quan khác. Hệ thống sẽ cung cấp các chức năng cơ bản như thêm, sửa, xóa thông tin sinh viên

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT – HỆ THỐNG QUẢN LÝ SINH VIÊN

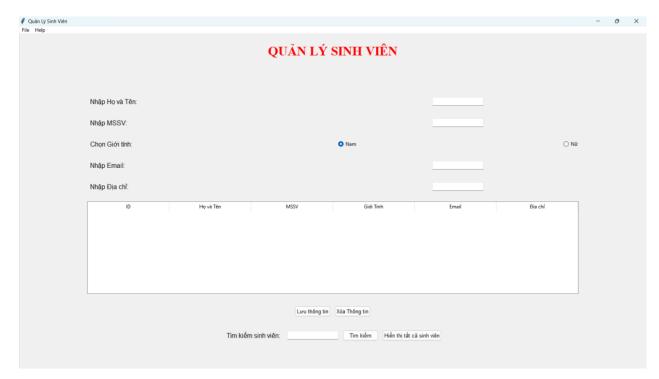
2.1 Khái niệm về hệ thống quản lý sinh viên

Hệ thống quản lý sinh viên là một phần mềm giúp quản lý toàn bộ các thông tin liên quan đến sinh viên như: thông tin cá nhân, v.v. Mục đích của hệ thống này là giúp giảm thiểu công sức và thời gian cho công việc quản lý.

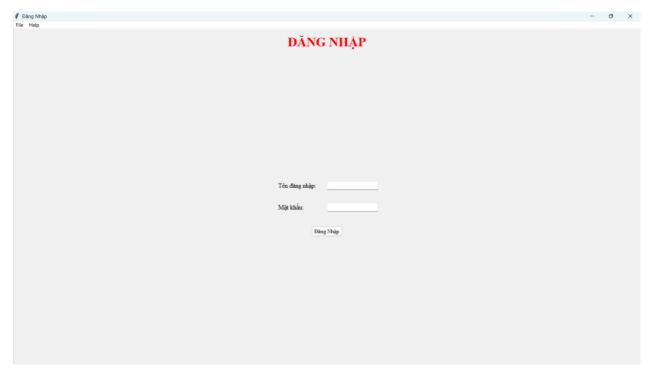
2.2 Các tính năng của hệ thống quản lý sinh viên

Các tính năng chính của hệ thống bao gồm:

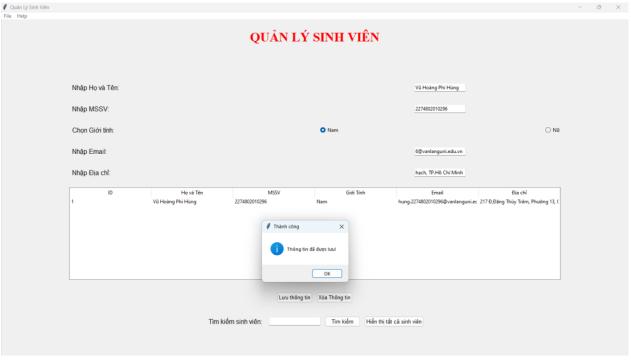
• Quản lý thông tin sinh viên: Thêm, sửa, xóa thông tin sinh viên như tên, mã sinh viên, ngày sinh, địa chỉ, số điện thoại, email.

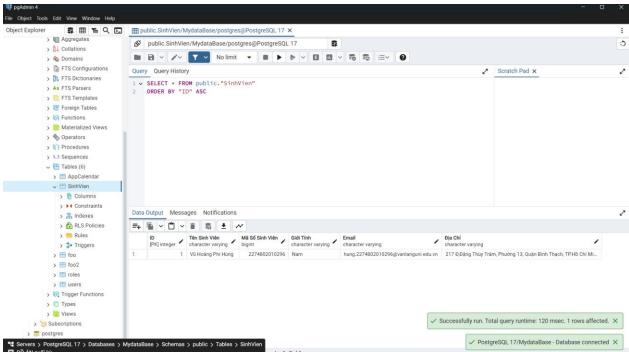


• Đăng Nhập vào hệ thống quản lý sinh viên



• Liên Kết với Database (lưu thông tin lên SQL)





2.3 Các công nghệ sử dụng trong dự án

- Ngôn ngữ lập trình: Python
- Cơ sở dữ liệu: Pgadmin v4
- Giao diện người dùng: Tkinter (cho ứng dụng desktop)

CHUONG 3: SOURCE CODE

3.1 Các gói thư viện

```
PoAnCuoiKy.py > State Login > DoanCuoiKy.py > Login > Login > DoancuoiKy.py > Login > Login > Login > DoancuoiKy.py > Login > Lo
```

3.2 Trang Đăng nhập và tính năng

```
🕏 DoAnCuoiKy.py > ધ Login > 🛇 _init_
              self.master = master
              self.master.title("Đăng Nhập")
              self.master.geometry("500x500")
              self.master.configure(bg="#f0f0f0")
              self.title_label = tk.Label(self.master, text="ĐĂNG NHẬP", font=("Times New Roman", 24, "bold"), fg="red", bg="#f0f0f0")
              self.title_label.pack(pady=10)
              style.configure("Tlabel", font=("Times New Roman", 12), padding=5) style.configure("TButton", font=("Times New Roman", 10))
              frame = ttk.Frame(self.master, padding="10")
              frame.pack(expand=True)
              self.label_user = ttk.Label(frame, text="Tên đăng nhập:")
              self.label_user.grid(row=0, column=0, padx=10, pady=10, sticky="W")
              self.entry_user = ttk.Entry(frame)
              self.entry_user.grid(row=0, column=1, padx=10, pady=10)
              self.label_pass = ttk.Label(frame, text="Mật khẩu:")
              self.label_pass.grid(row=1, column=0, padx=10, pady=10, sticky="W")
              self.entry_pass = ttk.Entry(frame, show=""
              self.entry_pass.grid(row=1, column=1, padx=10, pady=10)
              self.button_login = ttk.Button(frame, text="Đăng Nhập", command=self.login)
              self.button_login.grid(row=2, column=0, columnspan=2, pady=20)
```

```
def login(self):
    username = self.entry_user.get()
    password = self.entry_pass.get()

if username == "admin" and password == "password":
    self.master.destroy()
    main_app()
else:
    messagebox.showerror("Lỗi", "Tên đăng nhập hoặc mật khẩu không đúng!")
```

3.3 Khởi tạo lớp và thiết lập giao diện

```
class Sinhvien:

def __init__(self, master):

self.master = master

self.master.geometry("600x700")

self.master.geometry("600x700")

self.master.configure(bg="#f0f0f0")

self.title_label = tk.Label(self.master, text="QUÂN LÝ SINH VIÊN", font=("Times New Roman", 24, "bold"), fg="red", bg="#f0f0f0")

self.title_label.pack(pady=20)

# Khởi tạo đối tượng cơ sở dữ liệu

self.db = D_T_B2()

# Cài đặt các thành phần giao diện

style = ttk.style()

style.configure("TLabel", font=("Arial", 12), padding=5)

style.configure("TButton", font=("Arial", 10))

main_frame = ttk.frame(self.master, padding="10")

main_frame.pack(expand=True)
```

3.4 Các trường nhập liệu

```
DoAnCuoiKy.py > 4 Sinhvien > 0 _init_
      class Sinhvien:
          def __init__(self, master):
              # Ho và Tên
              self.ten sv = ttk.Label(main frame, text='Nhập Họ và Tên:')
              self.ten_sv.grid(row=0, column=0, padx=10, pady=10, sticky="W")
              self.ten entry = ttk.Entry(main frame)
              self.ten_entry.grid(row=0, column=1, padx=10, pady=10)
              # MSSV
              self.mssv = ttk.Label(main_frame, text='Nhập MSSV:')
              self.mssv.grid(row=1, column=0, padx=10, pady=10, sticky="W")
              self.mssv_entry = ttk.Entry(main_frame)
              self.mssv_entry.grid(row=1, column=1, padx=10, pady=10)
              # Giới tính
              self.gender_label = ttk.Label(main_frame, text='Chon Giới tính:')
              self.gender label.grid(row=2, column=0, padx=10, pady=10, sticky="W")
              self.gender = tk.StringVar()
104
              self.gender.set("Nam")
              self.radio_male = ttk.Radiobutton(main_frame, text="Nam", variable=self.gender, value="Nam")
              self.radio_male.grid(row=2, column=1, padx=10, pady=5, sticky="W")
              self.radio female = ttk.Radiobutton(main frame, text="No", variable=self.gender, value="No")
              self.radio_female.grid(row=2, column=1, padx=10, pady=5, sticky="E")
              # Fmail
              self.email_label = ttk.Label(main_frame, text='Nhập Email:')
              self.email_label.grid(row=3, column=0, padx=10, pady=10, sticky="W")
              self.email_entry = ttk.Entry(main_frame)
              self.email entry.grid(row=3, column=1, padx=10, pady=10)
              # Địa chỉ
              self.address label = ttk.Label(main frame, text='Nhập Địa chí:')
              self.address_label.grid(row=4, column=0, padx=10, pady=10, sticky="W")
              self.address_entry = ttk.Entry(main_frame)
              self.address_entry.grid(row=4, column=1, padx=10, pady=10)
```

3.5 Bảng hiển thị thông tin sinh viên

```
# Bảng hiến thị thông tin sinh viên

self.tree = ttk.Treeview(main_frame, columns=("ID", "Tên Sinh Viên", "Mã Số Sinh Viên", "Giới Tính", "Email", "Dịa chỉ"),

self.tree.heading("ID", text="ID")

self.tree.heading("Tên Sinh Viên", text="Họ và Tên")

self.tree.heading("Mã Số Sinh Viên", text="MSSV")

self.tree.heading("Giới Tính", text="Giới Tính")

self.tree.heading("Email", text="Giới Tính")

self.tree.heading("Dịa chỉ", text="Dịa chỉ")

self.tree.heading("Dịa chỉ", text="Dịa chỉ")

self.tree.grid(row=5, column=0, columnspan=2, padx=10, pady=10)
```

3.6 Các nút Action

```
# Button actions
button_frame = ttk.Frame(main_frame, padding="10")
button_frame.grid(row=6, column=0, columnspan=2)

action1 = ttk.Button(button_frame, text='Luu thông tin', command=self.save_infor)
action1.pack(side="left", padx=5, pady=10)

action2 = ttk.Button(button_frame, text='Xóa Thông tin', command=self.delete_infor)
action2.pack(side="left", padx=5, pady=10)
```

3.7 Chức năng tìm kiếm

```
# Chức năng tìm kiếm

search_frame = ttk.Frame(main_frame, padding="10")

search_frame.grid(row=7, column=0, columnspan=2)

self.search_label = ttk.Label(search_frame, text="Tîm kiếm sinh viên:")

self.search_label.pack(side="left", padx=5)

self.search_entry = ttk.Entry(search_frame)

self.search_entry.pack(side="left", padx=5)

self.search_button = ttk.Button(search_frame, text="Tîm kiếm", command=self.search_infor)

self.search_button.pack(side="left", padx=5)

self.show_all_button = ttk.Button(search_frame, text="Hiến thị tất cả sinh viên", command=self.show_all_students)

self.show_all_button.pack(side="left", padx=5)
```

3.8 Thanh Menu

```
# Tạo menu
self.menu_bar = Menu(self.master)
self.master.config(menu=self.menu_bar)

file_menu = Menu(self.menu_bar, tearoff=0)
self.menu_bar.add_cascade(label="File", menu=file_menu)
file_menu.add_command(label="New")
file_menu.add_command(label="Open")
file_menu.add_command(label="Exit", command=self.master.quit)

file_menu.add_command(label="Exit", command=self.master.quit)

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

def show_warning(self):
messagebox.showwarning("Python Message Warning Box",

"A Python GUI created using tkinter:\nWarning: There might be a bug in this code.")
```

3.9 Tính Năng save thông tin

```
def save_infor(self):

ho_va_ten = self.ten_entry.get()

ma_so_sv = self.mssv_entry.get()

gender = self.gender.get()

email = self.email_entry.get()

dia_chi = self.address_entry.get()

if ho_va_ten and ma_so_sv and gender and email and dia_chi:

id_value = self.db.save_infor(ho_va_ten, ma_so_sv, gender, email, dia_chi)

if id_value:

self.tree.insert("", "end", values=(id_value, ho_va_ten, ma_so_sv, gender, email, dia_chi))

messagebox.showinfo("Thanh cong", "Thong tin da duvo luv!")

else:

messagebox.showerror("Loi", "Khong the luv thong tin.")

else:

messagebox.showwarning("Loi", "Vui long dien day du thong tin!")
```

Database

```
def save_infor(self, ho_va_ten, ma_so_sv, gender, email, dia_chi):

try:

self.cur.execute(

'INSERT INTO "SinhVien" ("Tên Sinh Viên", "Mā Số Sinh Viên", "Giới Tính", "Email", "Địa Chi") VALUES (%s, %s, %s, %s)',

(ho_va_ten, ma_so_sv, gender, email, dia_chi))

self.cun.commit()

self.cur.execute('SELECT LASTVAL();')

return self.cur.fetchone()[0]

except Exception as e:

messagebox.showerror("Lỗi", f"Cổ lỗi xảy ra khi lưu thông tin: {e}")

return None
```

3.10 Tính năng Delete sinh viên

```
def delete_infor(self):
    selected_item = self.tree.selection()
    if selected_item:
        item = self.tree.item(selected_item)
        id_value = item['values'][0]
        self.db.delete_infor(id_value)
        self.tree.delete(selected_item)
        messagebox.showinfo("Thành công", "Thông tin đã được xóa!")
    else:
        messagebox.showwarning("Lỗi", "Chưa chọn thông tin nào để xóa.")
```

Database

```
def delete_infor(self, id_value):
    try:
        self.cur.execute('DELETE FROM "SinhVien" WHERE "ID" = %s', (id_value,))
        self.conn.commit()
        except Exception as e:
        messagebox.showerror("Lõi", f"Có lỗi xảy ra khi xóa thông tin: {e}")
```

3.11 Tính năng tìm Search sinh viên

Database

```
def search_infor(self, search_term):

try:

self.cur.execute(

'SELECT "ID", "Tên Sinh Viên", "Mã Số Sinh Viên", "Giới Tính", "Email", "Dịa Chi" FROM "SinhVien" '

"WHERE "Tên Sinh Viên" LIKE %S OR CAST("Nã Số Sinh Viên" AS TEXT) LIKE %S OR "Email" LIKE %S OR "Dịa Chi" LIKE %s',

(f'%{search_term}%', f'%{search_term}%', f'%{search_term}%'))

return self.cur.fetchall()

except Exception as e:

messagebox.showerror("Lỗi", f"Cổ lỗi xảy ra khi tìm kiếm: {e}")

return []
```

3.12 Tính năng show all sinh viên

```
def show_all_students(self):
results = self.db.show_all_students()
self.tree.delete(*self.tree.get_children())
for row in results:
self.tree.insert("", "end", values=row)
```

Database

```
def show_all_students(self):

try:

self.cur.execute('SELECT "ID", "Tên Sinh Viên", "Mã Số Sinh Viên", "Giới Tính", "Email", "Địa Chỉ" FROM "SinhVien"')

return self.cur.fetchall()

except Exception as e:

messagebox.showerror("Lỗi", f"Cố lỗi xảy ra khi tải danh sách: {e}")

return []
```

3.13 Liên Kết với Database

- import class D_T_B2 từ file my_db2.py

```
4 from my db2 import D T B2
```

```
# Khởi tạo đối tượng cơ sở dữ liệu self.db = D_T_B2()
```

Liên Kết với Database

CHƯƠNG 4 NHẬN XÉT

Ưu Điểm:

 Hệ thống giúp cho user quản lý thông tin sinh viên bao gồm, Họ và Tên, Mã số sinh viên, Email, Địa Chỉ, Giới tính có các tính năng như: Update, Delete, Save, Search, Show

Nhược Điểm:

 Hệ thống vẫn còn nhiều thiếu sót như: không có cập nhập điểm, tổng kết điểm không có quản lý lịch học cho sinh viên. Hệ thống vẫn chưa có đầy đủ và còn nhiều thiếu sót

CHƯƠNG 5 KẾT LUẬN

- LINK GITHUB

https://github.com/Junglou/Python-N-ng-Cao.git

TÀI LIỆU THAM KHẢO

- [1] Hoang S, Nguyen K, Huynh T, et al., "Chest X-ray Severity Score as a Putative Predictor of Clinical Outcome in Hospitalized Patients: An Experience From a Vietnamese Covid-19 Field Hospital", 10.7759/cureus.23323, 2022.
- [2] L. J. Muhammad, Ebrahem A. Algehyne, et al., "Supervised Machine Learning Models for Prediction of COVID-19 Infection using Epidemiology Dataset", Advances in Computational Approaches for Artificial Intelligence, Image Processing, IoT and Cloud Applications, 2020.
- [3] Zoabi, Y., Deri-Rozov, S. & Shomron, N, "Machine learning-based prediction of COVID-19 diagnosis based on symptoms", 2021.