

TRƯỜNG ĐẠI HỌC HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN MÔN HỌC
LẬP TRÌNH PYTHON NÂNG CAO
NGÀNH: CÔNG NGHỆ THÔNG TIN

Tên Đồ án:

**TẠO HỆ THỐNG QUẢN LÝ SINH VIÊN
(PYTHON GUI)**

SVTH: Vũ Hoàng Phi Hùng
MSSV: 2274802010296
GVHD: Tống Hùng Anh

Tp. Hồ Chí Minh – năm 2024

LỜI 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, người trực tiếp hướng dẫn chúng em hoàn thành tiểu luận này. Những ý kiến đóng góp của thầy là vô cùng hữu ích, nó giúp chúng em nhận ra các khuyết điểm của đồ án. Cảm ơn 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 THỐNG QUẢN 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 QUẢN 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	9
3.5 Bảng hiển thị thông tin sinh viên	9
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
CHƯƠNG 5 KẾT LUẬN	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.

Quản Lý Sinh Viên

File Help

QUẢN LÝ SINH VIÊN

Nhập Họ và Tên:

Nhập MSSV:

Chọn Giới tính: ☒ Nam ☐ Nữ

Nhập Email:

Nhập Địa chỉ:

ID	Họ và Tên	MSSV	Giới Tính	Email	Địa chỉ
1	Vũ Hoàng Phi Hùng	2274802010296	Nam	hung.2274802010296@vanlanguni.edu.vn	217 Đ.Đặng Thùy Trâm, Phường 13, Quận Bình Thạnh, TP.Hồ Chí Minh

Thành công

Thông tin đã được lưu!

OK

Tìm kiếm sinh viên:

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- Aggregates
- Collations
- Domains
- FTS Configurations
- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (6)
- AppCalendar
- SinhVien**
 - Columns
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
- foo
- foo2
- roles
- users
- Trigger Functions
- Types
- Views
- Subscriptions
- postgres

public.SinhVien/MydataBase/postgres@PostgreSQL 17

Query Query History

```
1 SELECT * FROM public."SinhVien"
2 ORDER BY "ID" ASC
```

Data Output Messages Notifications

ID	Tên Sinh Viên	Mã Số Sinh Viên	Giới Tính	Email	Địa Chỉ
1	Vũ Hoàng Phi Hùng	2274802010296	Nam	hung.2274802010296@vanlanguni.edu.vn	217 Đ.Đặng Thùy Trâm, Phường 13, Quận Bình Thạnh, TP.Hồ Chí Minh

Successfully run. Total query runtime: 120 msec. 1 rows affected.

PostgreSQL 17/MydataBase - Database connected

Servers > PostgreSQL 17 > Databases > MydataBase > Schemas > public > Tables > SinhVien

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)

CHƯƠNG 3: SOURCE CODE

3.1 Các gói thư viện

```
DoAnCuoiKy.py > Login > __init__
1  import tkinter as tk
2  from tkinter import ttk
3  from tkinter import messagebox, Menu
4  from my_db2 import D_T_B2
```

3.2 Trang Đăng nhập và tính năng

```
DoAnCuoiKy.py > Login > __init__
6  class Login:
7      def __init__(self, master):
8          self.master = master
9          self.master.title("Đăng Nhập")
10         self.master.geometry("500x500")
11         self.master.configure(bg="#f0f0f0")
12
13         self.title_label = tk.Label(self.master, text="ĐĂNG NHẬP", font=("Times New Roman", 24, "bold"), fg="red", bg="#f0f0f0")
14         self.title_label.pack(pady=10)
15
16         style = ttk.Style()
17         style.configure("TLabel", font=("Times New Roman", 12), padding=5)
18         style.configure("TButton", font=("Times New Roman", 10))
19
20         frame = ttk.Frame(self.master, padding="10")
21         frame.pack(expand=True)
22
23         self.label_user = ttk.Label(frame, text="Tên đăng nhập:")
24         self.label_user.grid(row=0, column=0, padx=10, pady=10, sticky="w")
25         self.entry_user = ttk.Entry(frame)
26         self.entry_user.grid(row=0, column=1, padx=10, pady=10)
27
28         self.label_pass = ttk.Label(frame, text="Mật khẩu:")
29         self.label_pass.grid(row=1, column=0, padx=10, pady=10, sticky="w")
30         self.entry_pass = ttk.Entry(frame, show="*")
31         self.entry_pass.grid(row=1, column=1, padx=10, pady=10)
32
33         self.button_login = ttk.Button(frame, text="Đăng Nhập", command=self.login)
34         self.button_login.grid(row=2, column=0, columnspan=2, pady=20)
```

```

56     def login(self):
57         username = self.entry_user.get()
58         password = self.entry_pass.get()
59
60         if username == "admin" and password == "password":
61             self.master.destroy()
62             main_app()
63         else:
64             messagebox.showerror("Lỗi", "Tên đăng nhập hoặc mật khẩu không đúng!")
65

```

3.3 Khởi tạo lớp và thiết lập giao diện

```

66 class Sinhvien:
67     def __init__(self, master):
68         self.master = master
69         self.master.title('Quản Lý Sinh Viên')
70         self.master.geometry("600x700")
71         self.master.configure(bg="#f0f0f0")
72
73         self.title_label = tk.Label(self.master, text="QUẢN LÝ SINH VIÊN", font=("Times New Roman", 24, "bold"), fg="red", bg="#f0f0f0")
74         self.title_label.pack(pady=20)
75
76         # Khởi tạo đối tượng cơ sở dữ liệu
77         self.db = D_T_B2()
78
79         # Cài đặt các thành phần giao diện
80         style = ttk.Style()
81         style.configure("TLabel", font=("Arial", 12), padding=5)
82         style.configure("TButton", font=("Arial", 10))
83
84         main_frame = ttk.Frame(self.master, padding="10")
85         main_frame.pack(expand=True)
86

```


3.4 Các trường nhập liệu

```
DoAnCuoiKy.py > Sinhvien > _init_
66 class Sinhvien:
67     def __init__(self, master):
87         # Họ và Tên
88         self.ten_sv = ttk.Label(main_frame, text='Nhập Họ và Tên:')
89         self.ten_sv.grid(row=0, column=0, padx=10, pady=10, sticky="W")
90         self.ten_entry = ttk.Entry(main_frame)
91         self.ten_entry.grid(row=0, column=1, padx=10, pady=10)
92
93         # MSSV
94         self.mssv = ttk.Label(main_frame, text='Nhập MSSV:')
95         self.mssv.grid(row=1, column=0, padx=10, pady=10, sticky="W")
96         self.mssv_entry = ttk.Entry(main_frame)
97         self.mssv_entry.grid(row=1, column=1, padx=10, pady=10)
98
99         # Giới tính
100        self.gender_label = ttk.Label(main_frame, text='Chọn Giới tính:')
101        self.gender_label.grid(row=2, column=0, padx=10, pady=10, sticky="W")
102
103        self.gender = tk.StringVar()
104        self.gender.set("Nam")
105
106        self.radio_male = ttk.Radiobutton(main_frame, text="Nam", variable=self.gender, value="Nam")
107        self.radio_male.grid(row=2, column=1, padx=10, pady=5, sticky="W")
108
109        self.radio_female = ttk.Radiobutton(main_frame, text="Nữ", variable=self.gender, value="Nữ")
110        self.radio_female.grid(row=2, column=1, padx=10, pady=5, sticky="E")
111
112        # Email
113        self.email_label = ttk.Label(main_frame, text='Nhập Email:')
114        self.email_label.grid(row=3, column=0, padx=10, pady=10, sticky="W")
115        self.email_entry = ttk.Entry(main_frame)
116        self.email_entry.grid(row=3, column=1, padx=10, pady=10)
117
118        # Địa chỉ
119        self.address_label = ttk.Label(main_frame, text='Nhập Địa chỉ:')
120        self.address_label.grid(row=4, column=0, padx=10, pady=10, sticky="W")
121        self.address_entry = ttk.Entry(main_frame)
122        self.address_entry.grid(row=4, column=1, padx=10, pady=10)
```

3.5 Bảng hiển thị thông tin sinh viên

```
124        # Bảng hiển thị thông tin sinh viên
125        self.tree = ttk.Treeview(main_frame, columns=("ID", "Tên Sinh Viên", "Mã Số Sinh Viên", "Giới Tính", "Email", "Địa chỉ"),
126                                show="headings")
127        self.tree.heading("ID", text="ID")
128        self.tree.heading("Tên Sinh Viên", text="Họ và Tên")
129        self.tree.heading("Mã Số Sinh Viên", text="MSSV")
130        self.tree.heading("Giới Tính", text="Giới Tính")
131        self.tree.heading("Email", text="Email")
132        self.tree.heading("Địa chỉ", text="Địa chỉ")
133        self.tree.grid(row=5, column=0, columnspan=2, padx=10, pady=10)
134
```

3.6 Các nút Action

```
135         # Button actions
136         button_frame = ttk.Frame(main_frame, padding="10")
137         button_frame.grid(row=6, column=0, columnspan=2)
138
139         action1 = ttk.Button(button_frame, text='Lưu thông tin', command=self.save_infor)
140         action1.pack(side="left", padx=5, pady=10)
141
142         action2 = ttk.Button(button_frame, text='Xóa Thông tin', command=self.delete_infor)
143         action2.pack(side="left", padx=5, pady=10)
```

3.7 Chức năng tìm kiếm

```
145         # Chức năng tìm kiếm
146         search_frame = ttk.Frame(main_frame, padding="10")
147         search_frame.grid(row=7, column=0, columnspan=2)
148
149         self.search_label = ttk.Label(search_frame, text="Tìm kiếm sinh viên:")
150         self.search_label.pack(side="left", padx=5)
151
152         self.search_entry = ttk.Entry(search_frame)
153         self.search_entry.pack(side="left", padx=5)
154
155         self.search_button = ttk.Button(search_frame, text="Tìm kiếm", command=self.search_infor)
156         self.search_button.pack(side="left", padx=5)
157
158         self.show_all_button = ttk.Button(search_frame, text="Hiển thị tất cả sinh viên", command=self.show_all_students)
159         self.show_all_button.pack(side="left", padx=5)
```

3.8 Thanh Menu

```
161         # Tạo menu
162         self.menu_bar = Menu(self.master)
163         self.master.config(menu=self.menu_bar)
164
165         file_menu = Menu(self.menu_bar, tearoff=0)
166         self.menu_bar.add_cascade(label="File", menu=file_menu)
167         file_menu.add_command(label="New")
168         file_menu.add_command(label="Open")
169         file_menu.add_separator()
170         file_menu.add_command(label="Exit", command=self.master.quit)
171
172         help_menu = Menu(self.menu_bar, tearoff=0)
173         self.menu_bar.add_cascade(label="Help", menu=help_menu)
174         help_menu.add_command(label="About", command=self.show_warning)
175
176         def show_warning(self):
177             messagebox.showwarning("Python Message Warning Box",
178                                     "A Python GUI created using tkinter:\nWarning: There might be a bug in this code.")
179
```

3.9 Tính Năng save thông tin

```
180     def save_infor(self):
181         ho_va_ten = self.ten_entry.get()
182         ma_so_sv = self.mssv_entry.get()
183         gender = self.gender.get()
184         email = self.email_entry.get()
185         dia_chi = self.address_entry.get()
186
187         if ho_va_ten and ma_so_sv and gender and email and dia_chi:
188             id_value = self.db.save_infor(ho_va_ten, ma_so_sv, gender, email, dia_chi)
189             if id_value:
190                 self.tree.insert("", "end", values=(id_value, ho_va_ten, ma_so_sv, gender, email, dia_chi))
191                 messagebox.showinfo("Thành công", "Thông tin đã được lưu!")
192             else:
193                 messagebox.showerror("Lỗi", "Không thể lưu thông tin.")
194         else:
195             messagebox.showwarning("Lỗi", "Vui lòng điền đầy đủ thông tin!")
```

- Database

```
18     def save_infor(self, ho_va_ten, ma_so_sv, gender, email, dia_chi):
19         try:
20             self.cur.execute(
21                 'INSERT INTO "SinhVien" ("Tên Sinh Viên", "Mã Số Sinh Viên", "Giới Tính", "Email", "Địa Chỉ") VALUES (%s, %s, %s, %s, %s)',
22                 (ho_va_ten, ma_so_sv, gender, email, dia_chi))
23             self.conn.commit()
24             self.cur.execute('SELECT LASTVAL();')
25             return self.cur.fetchone()[0]
26         except Exception as e:
27             messagebox.showerror("Lỗi", f"Có lỗi xảy ra khi lưu thông tin: {e}")
28         return None
```

3.10 Tính năng Delete sinh viên

```
197     def delete_infor(self):
198         selected_item = self.tree.selection()
199         if selected_item:
200             item = self.tree.item(selected_item)
201             id_value = item['values'][0]
202             self.db.delete_infor(id_value)
203             self.tree.delete(selected_item)
204             messagebox.showinfo("Thành công", "Thông tin đã được xóa!")
205         else:
206             messagebox.showwarning("Lỗi", "Chưa chọn thông tin nào để xóa.")
207
```

- Database

```
30     def delete_infor(self, id_value):
31         try:
32             self.cur.execute('DELETE FROM "SinhVien" WHERE "ID" = %s', (id_value,))
33             self.conn.commit()
34         except Exception as e:
35             messagebox.showerror("Lỗi", f"Có lỗi xảy ra khi xóa thông tin: {e}")
36
```

3.11 Tính năng tìm Search sinh viên

```
208     def search_infor(self):
209         search_term = self.search_entry.get()
210         if search_term:
211             results = self.db.search_infor(search_term)
212             self.tree.delete(*self.tree.get_children())
213             for row in results:
214                 self.tree.insert("", "end", values=row)
215             if not results:
216                 messagebox.showinfo("Kết quả", "Không tìm thấy sinh viên!")
217         else:
218             messagebox.showwarning("Lỗi", "Vui lòng nhập từ khóa tìm kiếm!")
```

- Database

```
37     def search_infor(self, search_term):
38         try:
39             self.cur.execute(
40                 'SELECT "ID", "Tên Sinh Viên", "Mã Số Sinh Viên", "Giới Tính", "Email", "Địa Chỉ" FROM "SinhVien" '
41                 'WHERE "Tên Sinh Viên" LIKE %s OR CAST("Mã Số Sinh Viên" AS TEXT) LIKE %s OR "Email" LIKE %s OR "Địa Chỉ" LIKE %s',
42                 (f'{search_term}%', f'{search_term}%', f'{search_term}%', f'{search_term}%'))
43             return self.cur.fetchall()
44         except Exception as e:
45             messagebox.showerror("Lỗi", f"Có lỗi xảy ra khi tìm kiếm: {e}")
46         return []
```

3.12 Tính năng show all sinh viên

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

- Database

```
48     def show_all_students(self):
49         try:
50             self.cur.execute('SELECT "ID", "Tên Sinh Viên", "Mã Số Sinh Viên", "Giới Tính", "Email", "Địa Chỉ" FROM "SinhVien"')
51             return self.cur.fetchall()
52         except Exception as e:
53             messagebox.showerror("Lỗi", f"Có lỗi xảy ra khi tải danh sách: {e}")
54         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

```
my_db2.py > ...
1 import psycopg2
2 from tkinter import messagebox
3
4 class D_T_B2:
5     def __init__(self):
6         try:
7             self.conn = psycopg2.connect(
8                 host="localhost",
9                 database="MydataBase",
10                user="postgres",
11                password="17468977",
12                port="5432"
13            )
14            self.cur = self.conn.cursor()
15        except Exception as e:
16            messagebox.showerror("Lỗi kết nối", f"Có lỗi xảy ra khi kết nối đến cơ sở dữ liệu: {e}")
17
```

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, Ebrahim 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.