

TRƯỜNG ĐẠI HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO PROJECT 1
LẬP TRÌNH PYTHON NÂNG CAO

Chủ đề:

BÁO CÁO PROJECT

SVTH: Trần Văn Hiếu – 2274802010262
LỚP: 241_71ITSE31003_0202

GVHD: Huỳnh Thái Học

Mục Lục

1	Project 1	4
1.1	Giao diện	4
1.2	Chức năng	5
1.2.1	Giải phương trình bậc 1	5
1.2.2	Chuyển đổi đơn vị độ dài km,cm,mm.....	6
1.2.3	Canvas.....	6
1.3	Mã chương trình.....	7
1.3.1	Giải phương trình bậc 1	8
1.3.2	Chuyển đổi đơn vị độ dài	9
1.3.3	Canvas.....	10
2	Project 2	10
2.1	Giao diện	10
2.2	Chức năng	12
2.2.1	Thêm hàng dữ liệu mới.....	12
2.2.2	Load dữ liệu	14
2.2.3	Chọn 1 hay nhiều hàng để xóa.....	15
2.2.4	Xóa dữ liệu được nhập trong mssv	15
2.2.5	Xóa nội dung trong ô nhập.....	17
2.2.6	Cập nhật dữ liệu	18
2.2.7	Xuất file excel.....	19
2.2.8	Thoát GUI	22
2.2.9	Hiển thị bản quyền	22
2.3	Mã chương trình.....	25
3	Project 3	33
3.1	Giao diện	33
3.1.1	Danh sách sinh viên	33
3.1.2	Thêm sinh viên.....	33
3.1.3	Sửa sinh viên.....	34

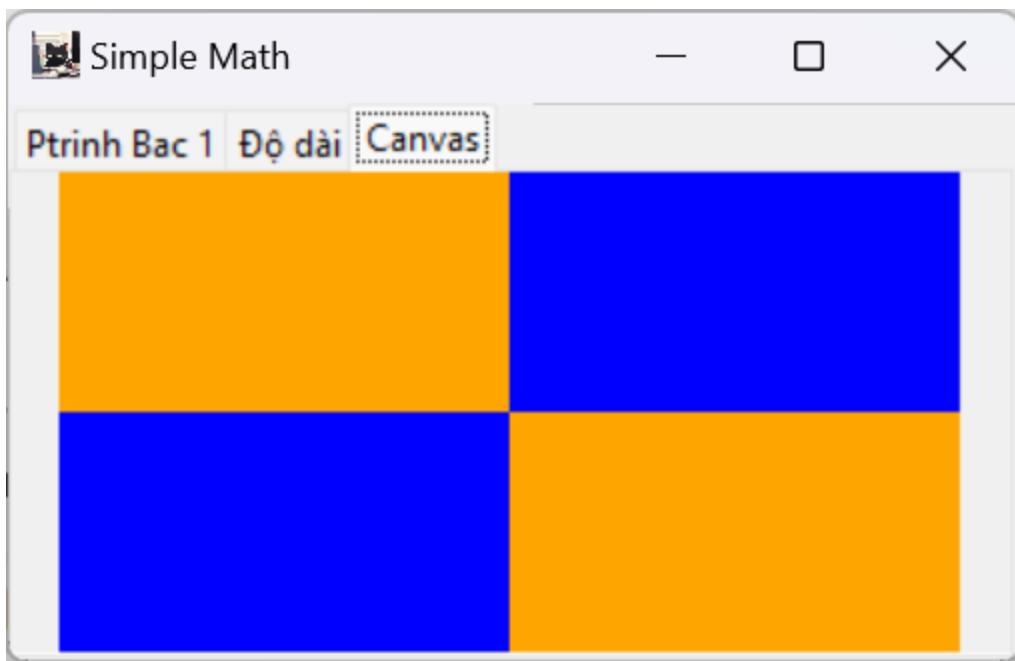
3.1.4	Xóa sinh viên	35
3.2	Mã nguồn	36
4	Github	40

1 Project 1

1.1 Giao diện

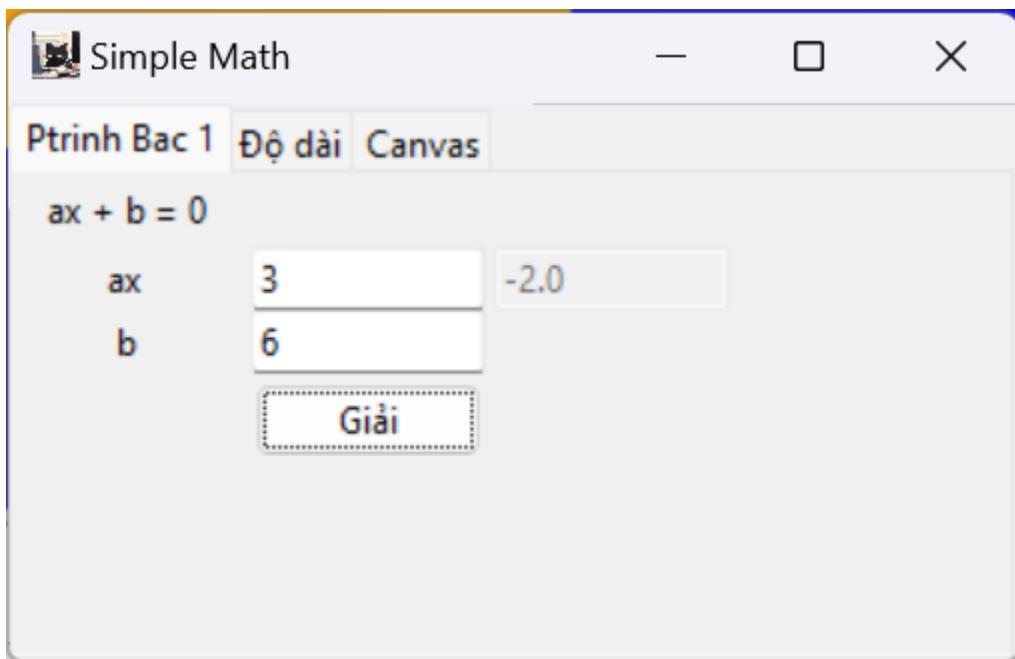
The screenshot shows a software window titled "Simple Math". The menu bar includes "Ptrinh Bac 1", "Độ dài", and "Canvas". The main area displays the equation $ax + b = 0$. There are two input fields: one for a (containing 0) and one for b (containing 0). A "Giải" (Solve) button is present below the inputs.

The screenshot shows a software window titled "Simple Math". The menu bar includes "Ptrinh Bac 1", "Độ dài", and "Canvas". The main area displays the text "Convert Between Kilometers, Centimeters, and Millimeters". It has two input fields: "From:" with an empty input box and a dropdown menu set to "Kilometers (km)", and "To:" with an input box containing "0.0" and a dropdown menu also set to "Kilometers (km)". A "Convert" button is located below the input fields.

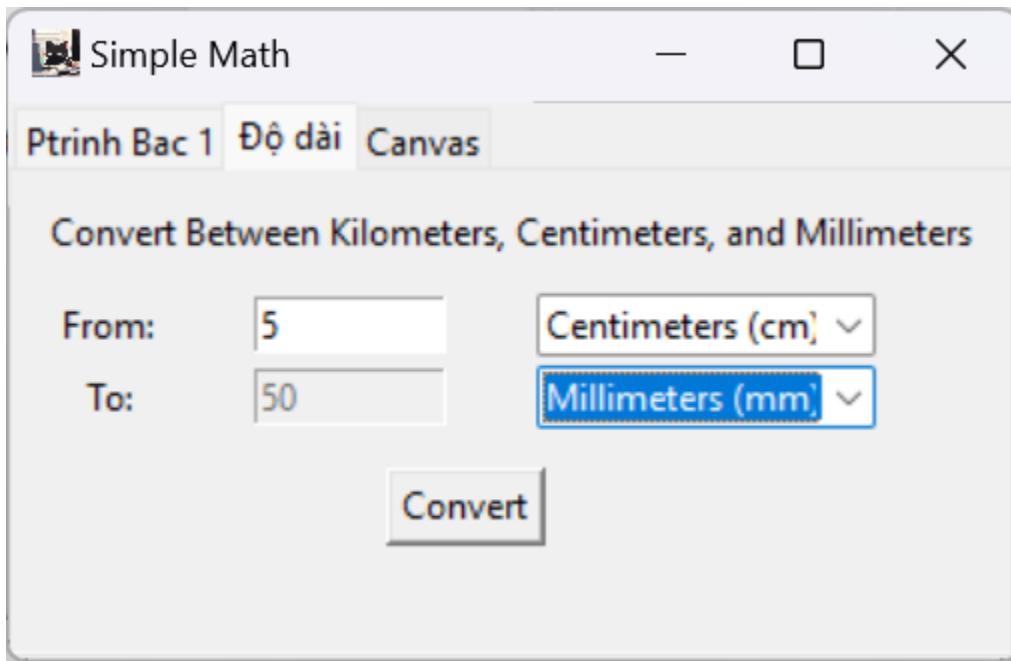


1.2 Chức năng

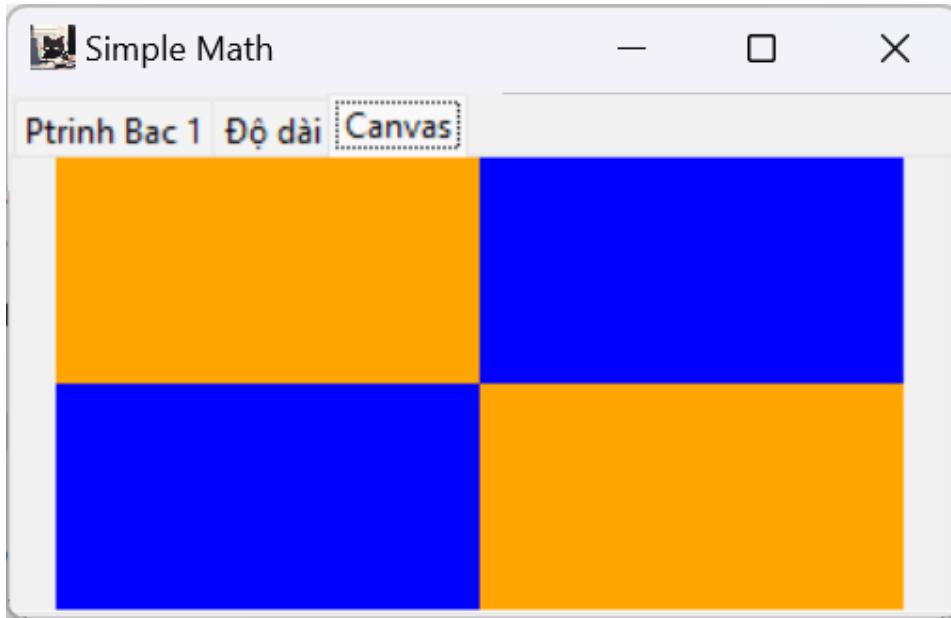
1.2.1 Giải phương trình bậc 1



1.2.2 Chuyển đổi đơn vị độ dài km, cm, mm



1.2.3 Canvas



1.3 Mã chương trình

1.3.1 Giải phương trình bậc 1

```
1 tabControl.add(tab1, text='Phương trình bậc 1') # Add the tab
2
3 tabControl.add(tab2, text='Độ dài')
4 tabControl.add(tab3, text="Canvas")
5 tabControl.pack(expand=1, fill='both') # Pack to make visible
6
7
8 main_label = ttk.Label(tab1, text="ax + b = 0")
9 main_label.grid(column=0, row=0 , padx=10 , pady=3)
10
11
12 a_label = ttk.Label(tab1, text="ax")
13 a_label.grid(column=0, row=2)
14 a = tk.IntVar()
15 a_entered = ttk.Entry(tab1, width=12, textvariable=a)
16 a_entered.grid(column=1, row=2 , padx=3)
17
18 b_label = ttk.Label(tab1, text="b")
19 b_label.grid(column=0, row=3)
20 b = tk.IntVar()
21 b_entered = ttk.Entry(tab1, width=12, textvariable=b)
22 b_entered.grid(column=1, row=3)
23
24 c = tk.IntVar()
25 c.set(0)
26
27 c_entered = ttk.Entry(tab1, width=12, textvariable=c, state="disabled")
28 c_entered.grid(column=2, row=2)
29 # Click Event Handle Func
30
31
32 def click_me():
33     try:
34         if a.get() != 0:
35             c.set(-b.get() / a.get())
36         else:
37             mbox.showerror("Error", "a phải khác 0")
38
39     # Lỗi này là do sai giá trị
40     except tk.TclError:
41         mbox.showerror("Input Error", "Nhập số vào a và b")
42     except:
43         mbox.showerror("Something wrong", "Check again")
44
45
46 # Adding a button
47 action = ttk.Button(tab1, text="Giải", command=click_me)
48 action.grid(column=0, row=4, columnspan=3,pady=3)
49
```

1.3.2 Chuyển đổi đơn vị độ dài

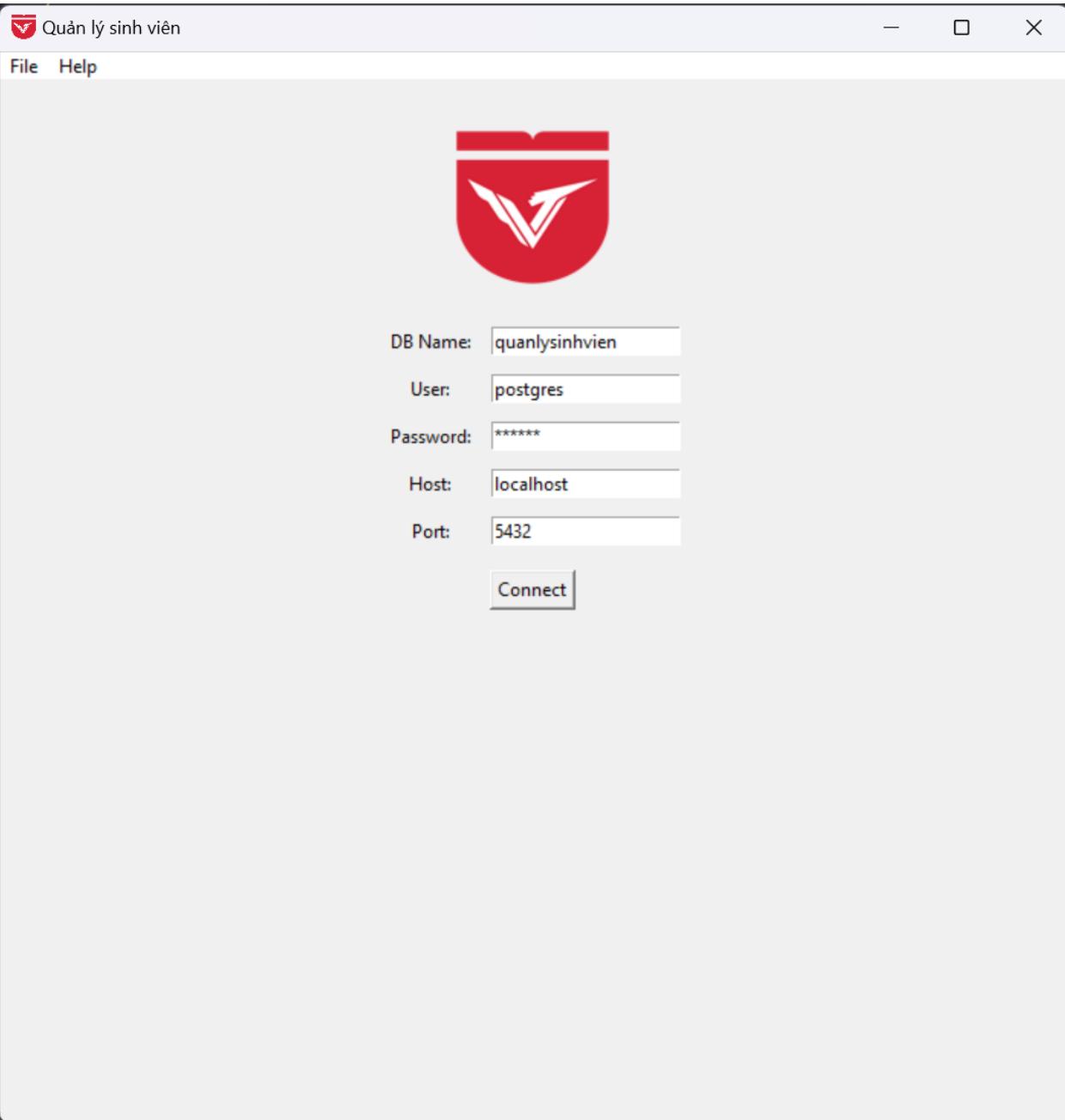
```
● ○ ●
1 def convert():
2     try:
3         input_value = float(entry.get())
4         from_unit = from_combo.get()
5         to_unit = to_combo.get()
6         conversion_factors = {
7             'Kilometers (km)': {
8                 'Kilometers (km)': 1,
9                 'Centimeters (cm)': 100000,
10                'Millimeters (mm)': 1000000
11            },
12            'Centimeters (cm)': {
13                'Kilometers (km)': 1/100000,
14                'Centimeters (cm)': 1,
15                'Millimeters (mm)': 10},
16            'Millimeters (mm)': {
17                'Kilometers (km)': 1/1000000,
18                'Centimeters (cm)': 1/10,
19                'Millimeters (mm)': 1
20            }
21        }
22
23        # Tính kết quả dựa trên hệ số chuyển đổi
24        result = input_value * conversion_factors[from_unit][to_unit]
25
26
27        re.set(format(result, '.10f').rstrip('0').rstrip('.'))
28        # Lỗi này là do sai giá trị
29    except tk.TclError:
30        mbox.showerror("Input Error", "Nhập số vào a và b")
31    except ValueError:
32        mbox.showerror("Lỗi nhập liệu", "Nhập số")
33    except:
34        mbox.showerror("Something wrong", "Check again")
35
36
37 title_label = tk.Label(
38     tab2, text="Convert Between Kilometers, Centimeters, and Millimeters")
39 title_label.grid(row=0, column=0, columnspan=4, padx=10, pady=10)
40
41 entry_label = tk.Label(tab2, text="From:")
42 entry_label.grid(row=1, column=0)
43
44 entry = tk.Entry(tab2, width=10)
45 entry.grid(row=1, column=1)
46
47 from_combo = ttk.Combobox(tab2, width=15, state='readonly')
48 from_combo['value'] = (
49     'Kilometers (km)', 'Centimeters (cm)', 'Millimeters (mm)')
50
51 from_combo.current(0)
52 from_combo.grid(row=1, column=2)
53
54 to_label = tk.Label(tab2, text="To:")
55 to_label.grid(row=2, column=0)
56
57 re = tk.DoubleVar()
58 entry_to = tk.Entry(tab2, width=10, state="disabled", textvariable=re)
59 entry_to.grid(row=2, column=1)
60
61 to_combo = ttk.Combobox(tab2, width=15, state='readonly')
62 to_combo['value'] = ('Kilometers (km)', 'Centimeters (cm)', 'Millimeters (mm)')
63 to_combo.current(0)
64 to_combo.grid(row=2, column=2, pady=3)
65
66 convert_button = tk.Button(tab2, text="Convert", command=convert)
67 convert_button.grid(row=4, column=0, columnspan=3, pady=10)
68
```

1.3.3 Canvas

```
1  tab3_frame = tk.Frame(tab3, bg='blue')
2  tab3_frame.pack()
3
4  for orange_color in range(2):
5      canvas = tk.Canvas(tab3_frame, width=150, height=80, highlightthickness=0, bg='orange')
6      canvas.grid(row=orange_color, column=orange_color)
7
8 win.mainloop()
9
```

2 Project 2

2.1 Giao diện



Quản lý sinh viên

File Help

Quản lý sinh viên VLU

MSSV	Họ	Tên
2274802010262	Trần Văn	Hiếu
2274802010612	Nguyễn Liên	Nhi
2274802010971	Bùi Kế Tôn	Tường

Entry Frame

MSSV :

Họ :

Tên :

Button Frame

2.2 Chức năng

2.2.1 Thêm hàng dữ liệu mới

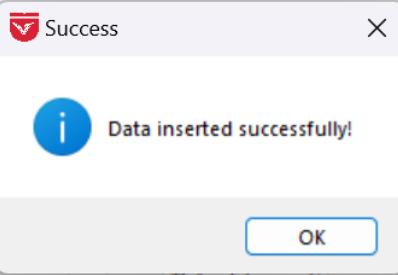
QLSV

Quản lý sinh viên

File Help

Quản lý sinh viên VLU

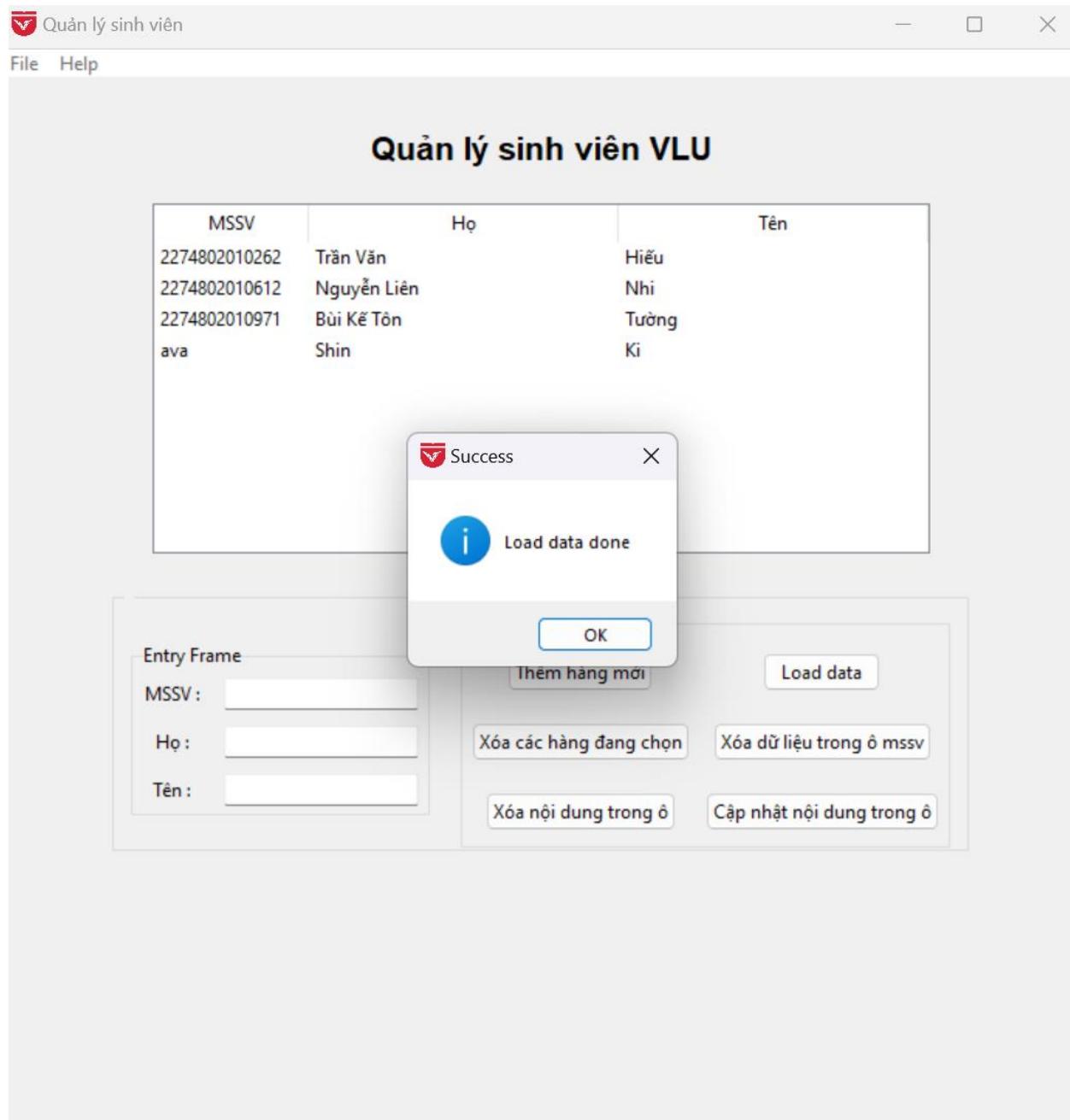
MSSV	Họ	Tên
2274802010262	Trần Văn	Hiếu
2274802010612	Nguyễn Liên	Nhi
2274802010971	Bùi Kế Tôn	Tường



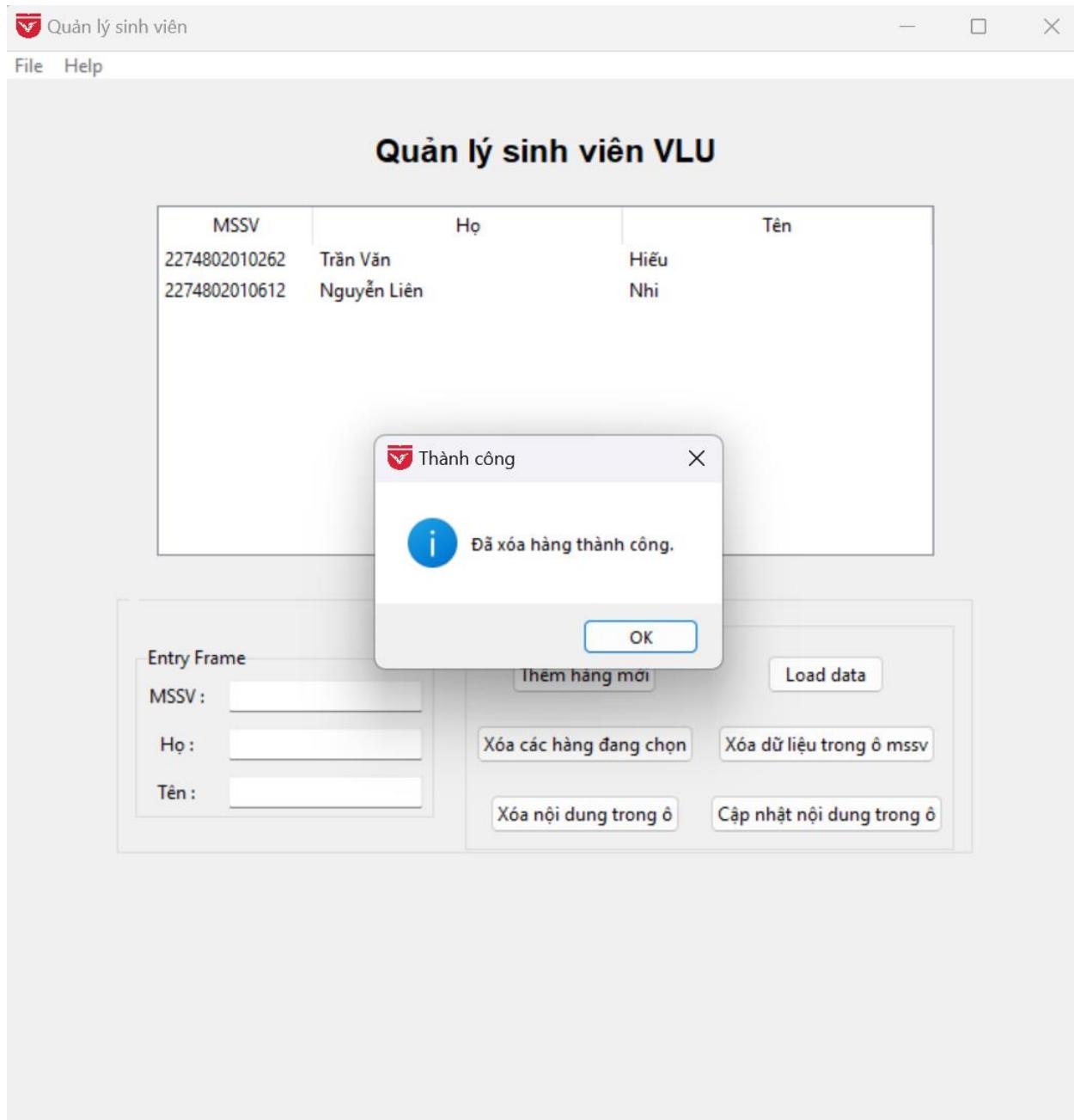
Entry Frame

MSSV : <input type="text" value="ava"/>	<input type="button" value="Thêm hàng mới"/>	<input type="button" value="Load data"/>
Họ : <input type="text" value="Shin"/>	<input type="button" value="Xóa các hàng đang chọn"/>	<input type="button" value="Xóa dữ liệu trong ô mssv"/>
Tên : <input type="text" value="Ki"/>	<input type="button" value="Xóa nội dung trong ô"/>	<input type="button" value="Cập nhật nội dung trong ô"/>

2.2.2 Load dữ liệu



2.2.3 Chọn 1 hay nhiều hàng để xóa



2.2.4 Xóa dữ liệu được nhập trong mssv

Quản lý sinh viên

File Help

Quản lý sinh viên VLU

MSSV	Họ	Tên
2274802010262	Trần Văn	Hiếu
2274802010612	Nguyễn Liên	Nhi
a	a	a

Done

Đã xóa thành công

OK

Entry Frame

MSSV :

Họ :

Tên :

Thêm hàng mới

Xóa các hàng đang chọn

Xóa dữ liệu trong ô mssv

Xóa nội dung trong ô

Cập nhật nội dung trong ô

Load data

2.2.5 Xóa nội dung trong ô nhập

Quản lý sinh viên

File Help

Quản lý sinh viên VLU

MSSV	Họ	Tên
2274802010262	Trần Văn	Hiếu
2274802010612	Nguyễn Liên	Nhi

Entry Frame

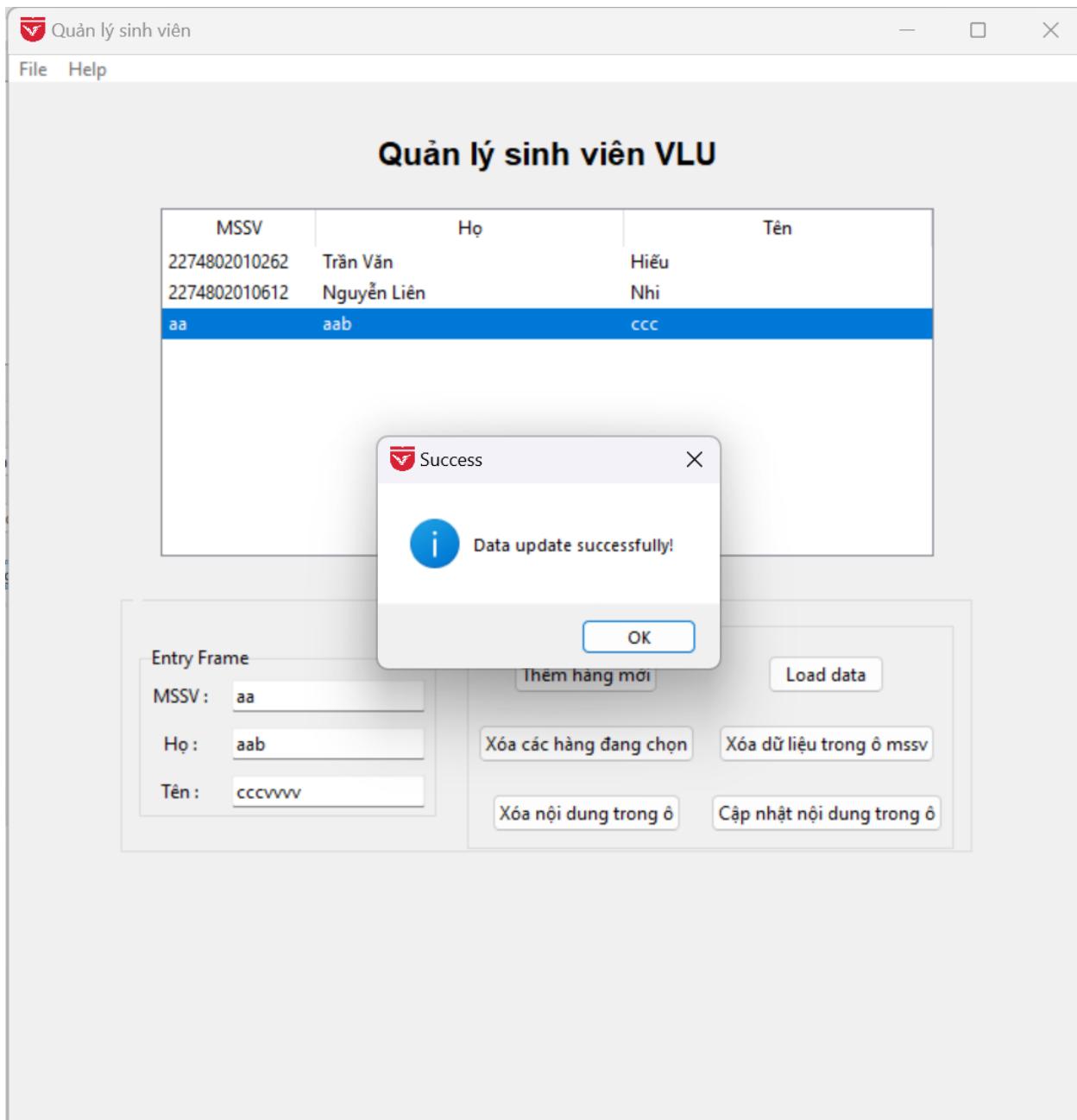
MSSV :

Họ :

Tên :

Button Frame

2.2.6 Cập nhật dữ liệu



2.2.7 Xuất file excel

MSSV	Họ	Tên
2274802010262	Trần Văn	Hiếu
2274802010612	Nguyễn Liên	Nhi
aa	aab	cccvvvv

Entry Frame

MSSV :

Họ :

Tên :

Button Frame

Thêm hàng mới

Load data

Xóa các hàng đang chọn

Xóa dữ liệu trong ô mssv

Xóa nội dung trong ô

Cập nhật nội dung trong ô

Quản lý sinh viên

File Help

Quản lý sinh viên VLU

MSSV	Họ	Tên
2274802010262	Trần Văn	Hiếu
2274802010612	Nguyễn Liên	Nhi
aa	aab	cccvvvv

Xuất file

Dữ liệu đã được xuất ra file Excel thành công!

OK

Entry Frame

MSSV :

Họ :

Tên :

Thêm hàng mới

Xóa các hàng đang chọn

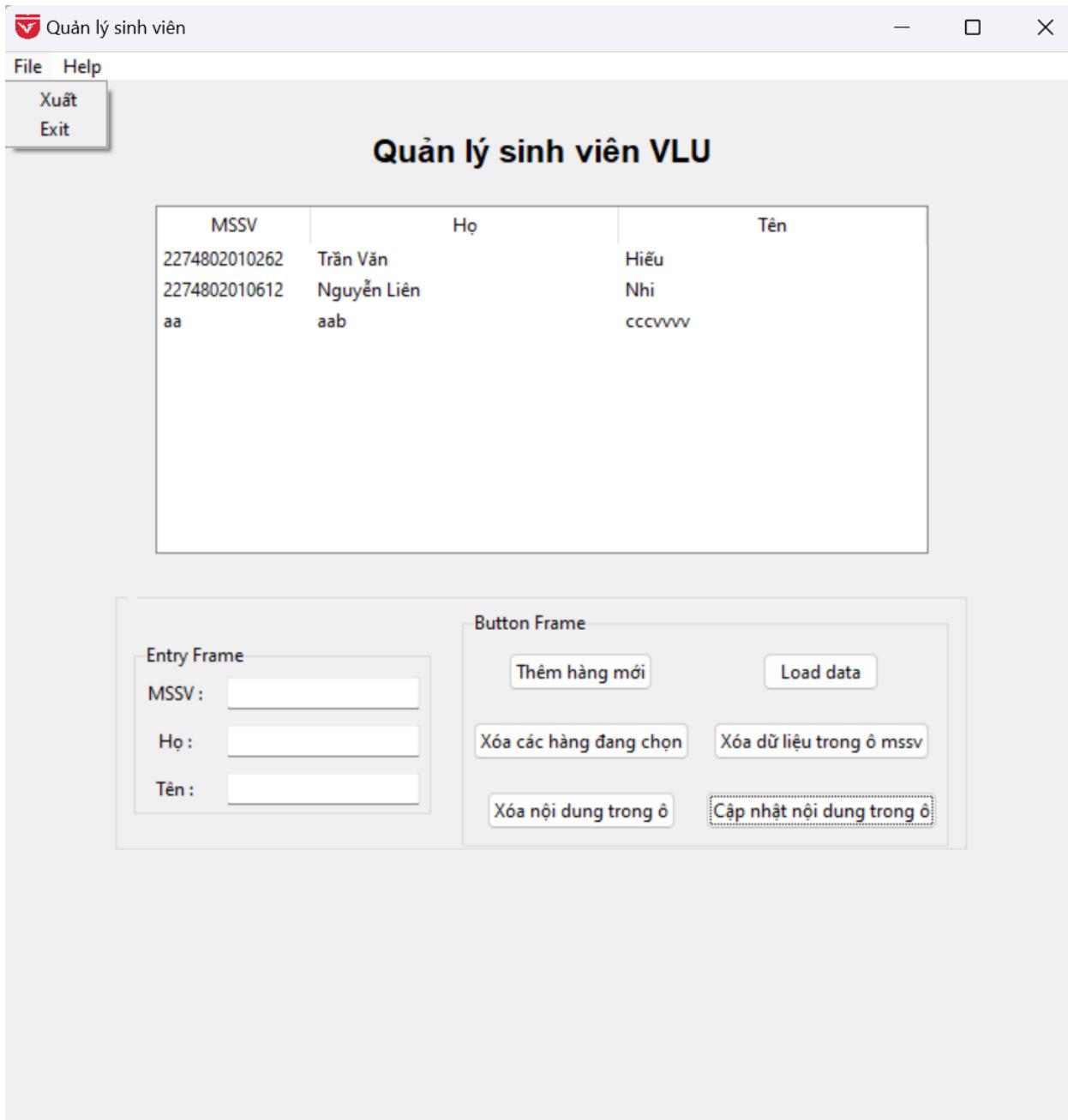
Xóa dữ liệu trong ô mssv

Xóa nội dung trong ô

Cập nhật nội dung trong ô

Load data

2.2.8 Thoát GUI



2.2.9 Hiển thị bản quyền

Quản lý sinh viên

File Help

About

Quản lý sinh viên VLU

MSSV	Họ	Tên
2274802010262	Trần Văn	Hiếu
2274802010612	Nguyễn Liên	Nhi
aa	aab	cccvvvv

Entry Frame

MSSV :

Họ :

Tên :

Button Frame

Quản lý sinh viên

File Help

Quản lý sinh viên VLU

MSSV	Họ	Tên
2274802010262	Trần Văn	Hiếu
2274802010612	Nguyễn Liên	Nhi
aa	aab	cccvvvv

About

This is a GUI made by Hieu Tran version 1.0
Release in 10/13/2024

OK

Entry Frame

MSSV :

Họ :

Tên :

Thêm hàng mới

Xóa các hàng đang chọn

Xóa dữ liệu trong ô mssv

Xóa nội dung trong ô

Cập nhật nội dung trong ô

Load data

2.3 Mã chương trình

```
1 import tkinter as tk
2 from tkinter import Menu, ttk, messagebox, filedialog
3 import psycopg2
4 from psycopg2 import sql
5 import pandas as pd
6 from PIL import Image, ImageTk
7
8
9
10 class TableApp():
11     def __init__(self, root):
12         self.root = root
13         self.root.iconbitmap('logo.ico')
14         self.root.title("Kết nối database")
15         # File menu
16         self.menu_bar = Menu(self.root)
17         file_menu = Menu(self.menu_bar, tearoff=0)
18         self.menu_bar.add_cascade(Label="File", menu=file_menu)
19         file_menu.add_command(Label="Xuất", command=self.export_to_excel)
20         file_menu.add_command(Label="Exit", command=self.quit_app)
21         # Help_menu
22         help_menu = Menu(self.menu_bar, tearoff=0)
23         self.menu_bar.add_cascade(Label="Help", menu=help_menu)
24         help_menu.add_command(Label="About", command=self.msg_box_info)
25         self.root.config(menu=self.menu_bar)
26         self.root.title("Quản lý sinh viên")
27         self.db_name = tk.StringVar(value='quanlysinhvien')
28         self.user = tk.StringVar(value='postgres')
29         self.password = tk.StringVar(value='123456')
30         self.host = tk.StringVar(value='localhost')
31         self.port = tk.StringVar(value='5432')
32         self.table_name = tk.StringVar(value='sinhvien')
33         original_image = Image.open("logovlu.png")
34         resized_image = original_image.resize((100, 100))
35         self.image = ImageTk.PhotoImage(resized_image)
36
37         width = 700
38         height = 700
39         self.root.geometry(f"{height}x{width}")
40         screen_width = self.root.winfo_screenwidth()
41         screen_height = self.root.winfo_screenheight()
42         x = (screen_width - width) // 2
43         y = (screen_height - height) // 2
44         self.root.geometry(f'{width}x{height}+{x}+{y}')
45
46         self.widgets_connect()
47
```

```
1
2 def widgets_connect(self):
3
4     self.connection_frame = tk.Frame(self.root)
5     self.connection_frame.pack(pady=10)
6
7     image_label = tk.Label(self.connection_frame, image=self.image)
8     image_label.grid(row=0, column=0, columnspan=3, pady=20)
9
10    tk.Label(self.connection_frame, text="DB Name:").grid(
11        row=1, column=0, padx=5, pady=5)
12    tk.Entry(self.connection_frame, textvariable=self.db_name).grid(
13        row=1, column=1, padx=5, pady=5)
14
15    tk.Label(self.connection_frame, text="User:").grid(
16        row=2, column=0, padx=5, pady=5)
17    tk.Entry(self.connection_frame, textvariable=self.user).grid(
18        row=2, column=1, padx=5, pady=5)
19
20    tk.Label(self.connection_frame, text="Password:").grid(
21        row=3, column=0, padx=5, pady=5)
22    tk.Entry(self.connection_frame, textvariable=self.password,
23             show="*").grid(row=3, column=1, padx=5, pady=5)
24
25    tk.Label(self.connection_frame, text="Host:").grid(
26        row=4, column=0, padx=5, pady=5)
27    tk.Entry(self.connection_frame, textvariable=self.host).grid(
28        row=4, column=1, padx=5, pady=5)
29
30    tk.Label(self.connection_frame, text="Port:").grid(
31        row=5, column=0, padx=5, pady=5)
32    tk.Entry(self.connection_frame, textvariable=self.port).grid(
33        row=5, column=1, padx=5, pady=5)
34
35    tk.Button(self.connection_frame, text="Connect",
36              command=self.connect_to_manage).grid(row=6, columnspan=2, pady=10)
37
```

```
1 def widgets_manage(self):
2
3     # Khung chứa bảng
4     self.table_frame = tk.Frame(self.root)
5     self.table_frame.pack(pady=10)
6     tk.Label(self.table_frame, text="Quản lý sinh viên VLU", border=2, font=(
7         "Helvetica", 16, "bold")).grid(column=0, row=0, columnspan=3, pady=20)
8
9     # Tạo bảng Treeview
10    self.data_table = ttk.Treeview(self.table_frame, columns=(
11        "MSSV", "Họ", "Tên"), show="headings", height=10)
12    self.data_table.heading("MSSV", text="MSSV")
13    self.data_table.heading("Họ", text="Họ")
14    self.data_table.heading("Tên", text="Tên")
15
16    self.data_table.column("MSSV", width=100)
17    self.data_table.column("Họ", width=200)
18    self.data_table.column("Tên", width=200)
19    self.data_table.grid(column=0, row=2)
20    self.data_table.bind("<ButtonRelease-1>", self.on_tree_select)
21    self.load_data()
22
23    # Khung chứa các ô nhập liệu
24    form_frame = ttk.LabelFrame(self.root, text="")
25    form_frame.pack(pady=10)
26
27    # Nhập và ô nhập MSSV
28    input_frame = ttk.LabelFrame(form_frame, text="Entry Frame")
29    input_frame.grid(padx=10, row=0, column=0)
30    ttk.Label(input_frame, text="MSSV : ").grid(
31        row=0, column=0, padx=5, pady=5)
32    self.mssv = tk.StringVar()
33    ttk.Entry(input_frame, textvariable=self.mssv).grid(
34        row=0, column=1, padx=5, pady=5)
35
36    # Nhập và ô nhập Họ Tên
37    ttk.Label(input_frame, text="Họ : ").grid(
38        row=1, column=0, padx=5, pady=5)
39    self.ho = tk.StringVar()
40    ttk.Entry(input_frame, textvariable=self.ho).grid(
41        row=1, column=1, padx=5, pady=5)
42
43    ttk.Label(input_frame, text="Tên : ").grid(
44        row=2, column=0, padx=5, pady=5)
45    self.ten = tk.StringVar()
46    ttk.Entry(input_frame, textvariable=self.ten).grid(
47        row=2, column=1, padx=5, pady=5)
48
49    # Frame button
50    button_frame = ttk.LabelFrame(form_frame, text="Button Frame")
51    button_frame.grid(padx=10, row=0, column=1)
52    # Nút để thêm hàng mới
53    add_button = ttk.Button(
54        button_frame, text="Thêm hàng mới", command=self.add_data_button)
55    add_button.grid(column=0, row=0, padx=5, pady=10)
56
57    # Nút để Load data
58    load_button = ttk.Button(
59        button_frame, text="Load data", command=self.load_data_button)
60    load_button.grid(column=1, row=0, padx=5, pady=10)
61    # Nút để xóa data đang chọn
62    delete_button = ttk.Button(
63        button_frame, text="Xóa các hàng đang chọn", command=self.delete_selected_row)
64    delete_button.grid(column=0, row=1, padx=5, pady=10)
65
66    # Nút để xóa data trong ô
67    delete_input_button = ttk.Button(
68        button_frame, text="Xóa dữ liệu trong ô mssv", command=self.delete_data_button)
69    delete_input_button.grid(column=1, row=1, padx=5, pady=10)
70
71    # Nút clear input
72    clear_button = ttk.Button(
73        button_frame, text="Xóa nội dung trong ô", command=self.clear_inputs)
74    clear_button.grid(column=0, row=2, padx=5, pady=10)
75    # Nút update data
76    update_button = ttk.Button(
77        button_frame, text="Cập nhật nội dung trong ô", command=self.update_data_button)
78    update_button.grid(column=1, row=2, padx=5, pady=10)
79
```

```
1
2     def on_tree_select(self, event):
3         """Xử lý khi người dùng chọn hàng trong Treeview"""
4         selected_items = self.data_table.selection()
5         if selected_items:
6             item = selected_items[0]
7             values = self.data_table.item(item, 'values')
8             mssv = values[0]
9             ho = values[1]
10            ten = values[2]
11            self.mssv.set(mssv)
12            self.ho.set(ho)
13            self.ten.set(ten)
14
15    def insert_data_button(self):
16        mssv = self.mssv.get().strip()
17        ho = self.ho.get().strip()
18        ten = self.ten.get().strip()
19
20        if self.check_exist(mssv):
21            self.insert_data()
22            self.add_table(mssv, ho, ten)
23            self.clear_inputs()
24            self.load_data()
25        else:
26            messagebox.showerror("Error", "Mã sinh viên không tồn tại")
27
28    def add_data_button(self):
29        """Thêm sinh viên mới vào cơ sở dữ liệu và bảng Treeview"""
30        mssv = self.mssv.get().strip()
31        ho = self.ho.get().strip()
32        ten = self.ten.get().strip()
33
34        if self.validate_input(mssv, ho, ten):
35            if self.check_exist(mssv):
36                messagebox.showerror("Error", "Mã sinh viên đã tồn tại")
37            else:
38                self.insert_data()
39                self.add_table(mssv, ho, ten)
40                self.clear_inputs()
41                self.load_data()
42
43    def update_data_button(self):
44        mssv = self.mssv.get().strip()
45        if self.check_exist(mssv):
46            self.update_data()
47            self.clear_inputs()
48            self.load_data()
49        else:
50            messagebox.showerror("Error", "Mã sinh viên không tồn tại")
51
52    def delete_data_button(self):
53        mssv = self.mssv.get().strip()
54        if self.check_exist(mssv):
55            self.delete_data(mssv)
56            self.clear_inputs()
57            messagebox.showinfo("Done", "Đã xóa thành công")
58            self.load_data()
59
60        else:
61            messagebox.showerror("Error", "Mã sinh viên không tồn tại")
62
```

```
1
2 def load_data_button(self):
3     if self.load_data():
4         messagebox.showinfo("Success", "Load data done")
5
6 def validate_input(self, mssv, ho, ten):
7     """Kiểm tra đầu vào không được để trống."""
8     if not mssv or not ho or not ten:
9         messagebox.showwarning(
10             "Lỗi nhập liệu", "MSSV và Họ Tên không được để trống.")
11     return False
12     return True
13
14 def add_table(self, mssv, ho, ten):
15     """Thêm hàng mới vào bảng Treeview."""
16     new_row = (mssv, ho, ten)
17     self.data_table.insert("", tk.END, values=new_row)
18
19 def clear_inputs(self):
20     """Xóa các ô nhập liệu sau khi thêm dữ liệu thành công."""
21     self.mssv.set("")
22     self.ho.set("")
23     self.ten.set("")
24
25 # ! Còn Return True False
26 def connect_db(self):
27     """Kết nối tới cơ sở dữ liệu PostgreSQL"""
28     try:
29         self.conn = psycopg2.connect(
30             dbname=self.db_name.get(),
31             user=self.user.get(),
32             password=self.password.get(),
33             host=self.host.get(),
34             port=self.port.get()
35         )
36         self.cur = self.conn.cursor()
37         messagebox.showinfo(
38             "Success", "Connected to the database successfully!")
39         return True
40     except Exception as e:
41         messagebox.showerror(
42             "Error", f"Error connecting to the database: {e}")
43         return False
44
45 def insert_data_tabletree(self, rows):
46     # Đầu * Là giải nén tuple thành các phần tử riêng lẻ
47     # Xóa toàn bộ dữ liệu hiện tại
48     self.data_table.delete(*self.data_table.get_children())
49     for row in rows:
50         # các hàng được chèn không có quan hệ cha-con
51         self.data_table.insert("", tk.END, values=row)
52
53 def load_data(self):
54     """Tải dữ liệu từ cơ sở dữ liệu và hiển thị trong bảng Treeview"""
55     try:
56         query = sql.SQL(
57             "SELECT * FROM {}").format(sql.Identifier(self.table_name.get()))
58         self.cur.execute(query)
59         rows = self.cur.fetchall()
60         self.insert_data_tabletree(rows)
61         return True
62     except Exception as e:
63         self.conn.rollback()
64         messagebox.showerror("Error", f"Error loading data: {e}")
65         return False
66
```

```
1 def insert_data(self):
2     """Chèn dữ liệu sinh viên vào cơ sở dữ liệu"""
3     try:
4         insert_query = sql.SQL("INSERT INTO {} (mssv, ho,ten) VALUES (%s, %s, %s)").format(
5             sql.Identifier(self.table_name.get()))
6         data_to_insert = (self.mssv.get(), self.ho.get(), self.ten.get())
7         self.cur.execute(insert_query, data_to_insert)
8         self.conn.commit()
9         messagebox.showinfo("Success", "Data inserted successfully!")
10    except Exception as e:
11        self.conn.rollback()
12        messagebox.showerror("Error", f"Error inserting data: {e}")
13
14 def update_data(self):
15     try:
16         update_query = sql.SQL("UPDATE {} SET ho = %s, ten = %s WHERE mssv = %s").format(
17             sql.Identifier(self.table_name.get()))
18         data_to_update = (self.ho.get(), self.ten.get(), self.mssv.get())
19         self.cur.execute(update_query, data_to_update)
20         self.conn.commit()
21         messagebox.showinfo("Success", "Data update successfully!")
22    except Exception as e:
23        self.conn.rollback()
24        messagebox.showerror("Error", f"Error inserting data: {e}")
25
26 def delete_selected_row(self):
27     """Xóa hàng đang được chọn trong Treeview"""
28     try:
29         # Lấy tất cả các hàng đang được chọn
30         selected_items = self.data_table.selection()
31         if not selected_items:
32             messagebox.showwarning("Lỗi", "Vui lòng chọn hàng để xóa.")
33             return
34
35         for item in selected_items:
36             values = self.data_table.item(item, 'values')
37             mssv = values[0]
38             self.delete_data(mssv)
39             if item in self.data_table.get_children():
40                 self.data_table.delete(item)
41             else:
42                 messagebox.showwarning(
43                     "Lỗi", f"Item {item} không tồn tại trong bảng.")
44             self.clear_inputs()
45             messagebox.showinfo("Thành công", "Đã xóa hàng thành công.")
46    except Exception as e:
47        messagebox.showerror("Error", f"Error deleting selected row: {e}")
48
49 def delete_data_input(self):
50     try:
51         self.delete_data(self.mssv.get())
52         self.load_data()
53         self.clear_inputs()
54         messagebox.showinfo("Success", "Đã xóa thành công")
55    except Exception as e:
56        messagebox.showerror("Error", f"{e}")
57
58 def delete_data(self, mssv):
59     """Xóa dữ liệu sinh viên khỏi cơ sở dữ liệu theo MSSV"""
60     try:
61         delete_query = sql.SQL("DELETE FROM {} WHERE mssv = %s").format(
62             sql.Identifier(self.table_name.get()))
63         self.cur.execute(delete_query, (mssv,))
64         self.conn.commit()
65    except Exception as e:
66        self.conn.rollback()
67        messagebox.showerror(
68            "Error", f"Error deleting data from database: {e}")
69
70
```

```
1  def connect_to_manage(self):
2      check = self.connect_db()
3      if check:
4          self.connection_frame.pack_forget()
5          self.widgets_manage()
6
7  def msg_box_info(self):
8      messagebox.showinfo(
9          "About", "This is a GUI made by Hieu Tran version 1.0\n Release in 10/13/2024\n")
10
11 def quit_app(self):
12     self.root.destroy()
13
14 def check_exist(self, mssv):
15     children = self.data_table.get_children()
16     for child in children:
17         values = self.data_table.item(child, 'values')
18         if values[0] == mssv:
19             return True
20     return False
21
22 def export_to_excel(self):
23     try:
24         rows = []
25         for item in self.data_table.get_children():
26             row_data = self.data_table.item(item)['values']
27             rows.append(row_data)
28
29         # Tạo DataFrame
30         df = pd.DataFrame(rows, columns=["MSSV", "Họ", "Tên"])
31         df.index = df.index + 1
32
33         # Mở hộp thoại Lưu tệp để chọn đường dẫn Lưu
34         file_path = filedialog.asksaveasfilename(defaultextension=".xlsx",
35                                         filetypes=[
36                                             ("Excel files", "*.xlsx"), ("All files", "*.*")],
37                                         title="Chọn nơi lưu file")
38
39         if file_path: # Nếu người dùng không hủy
40             # Xuất ra file Excel
41             df.to_excel(file_path, index=True)
42             messagebox.showinfo(
43                 "Xuất file", "Đữ liệu đã được xuất ra file Excel thành công!")
44         else:
45             messagebox.showwarning("Hủy", "Bạn đã hủy lưu file.")
46
47     except Exception as e:
48         messagebox.showwarning("Warning", f"Có lỗi xảy ra: {str(e)}")
49
50 if __name__ == "__main__":
51     root = tk.Tk()
52     TableApp(root)
53     root.mainloop()
54
```

3 Project 3

3.1 Giao diện

3.1.1 Danh sách sinh viên

Student Management

[Logout](#) [Add Student](#)

MSSV	Last Name	First Name	Photo	Actions
2274802010262	Trần Văn	Hiếu		Sửa Xóa
2274802010332	Hồ Minh	Vọng		Sửa Xóa

3.1.2 Thêm sinh viên

Add Student

MSSV

Last Name

First Name

Photo

3.1.3 Sửa sinh viên

Edit Student

MSSV

Last Name

First Name

Photo



3.1.4 Xóa sinh viên

Student Management

Logout Add Student

127.0.0.1:5050 cho biết
Bạn có chắc chắn muốn xóa sinh viên này không?

OK Huỷ

MSSV	Last Name	First Name	Photo	Actions
2274802010332	Hồ Minh	Vọng		Sửa Xóa
2274802010262	Trần Văn	Hiếu		Sửa Xóa

3.2 Mă nguồn

```
1 import psycopg2
2 from psycopg2 import sql
3 import base64
4 class Database:
5     def __init__(self, db_name, user, password, host, port, table_name):
6         self.db_name = db_name
7         self.user = user
8         self.password = password
9         self.host = host
10        self.port = port
11        self.table_name = table_name
12
13    def connect_db(self):
14        try:
15            self.conn = psycopg2.connect(
16                dbname=self.db_name,
17                user=self.user,
18                password=self.password,
19                host=self.host,
20                port=self.port
21            )
22            self.cur = self.conn.cursor()
23            return True
24        except Exception as e:
25            print(f"Connection error: {e}")
26            return False
27
28    def insert_data(self, mssv, ho, ten, image_path):
29        try:
30            with open(image_path, "rb") as image_file:
31                image_data = base64.b64encode(image_file.read()).decode('utf-8')
32            insert_query = sql.SQL("INSERT INTO {} (mssv, ho, ten, image) VALUES (%s, %s, %s, %s)").format(
33                sql.Identifier(self.table_name))
34            data_to_insert = (mssv, ho, ten, image_data)
35            self.cur.execute(insert_query, data_to_insert)
36            self.conn.commit()
37            return True
38        except Exception as e:
39            self.conn.rollback()
40            print(f"Insertion error: {e}")
41            return False
42
43    def load_data(self):
44        try:
45            query = sql.SQL("SELECT mssv, ho, ten, image FROM {}").format(sql.Identifier(self.table_name))
46            self.cur.execute(query)
47            rows = self.cur.fetchall()
48            return rows
49        except Exception as e:
50            print(f"Loading error: {e}")
51            return False
52
53    def get_student_by_mssv(self, mssv):
54        try:
55            query = sql.SQL("SELECT mssv, ho, ten, image FROM {} WHERE mssv = %s").format(
56                sql.Identifier(self.table_name)
57            )
58            self.cur.execute(query, (mssv,))
59            return self.cur.fetchone()
60        except Exception as e:
61            print(f"Error fetching student: {e}")
62            return None
63
64    def update_data(self, mssv, ho, ten, image_path=None):
65        try:
66            if image_path:
67                with open(image_path, "rb") as image_file:
68                    image_data = base64.b64encode(image_file.read()).decode('utf-8')
69                    update_query = sql.SQL("UPDATE {} SET ho = %s, ten = %s, image = %s WHERE mssv = %s").format(
70                        sql.Identifier(self.table_name)
71                    )
72                    data_to_update = (ho, ten, image_data, mssv)
73            else:
74                update_query = sql.SQL("UPDATE {} SET ho = %s, ten = %s WHERE mssv = %s").format(
75                    sql.Identifier(self.table_name)
76                )
77                data_to_update = (ho, ten, mssv)
78            self.cur.execute(update_query, data_to_update)
79            self.conn.commit()
80            return True
81        except Exception as e:
82            self.conn.rollback()
83            print(f"Update error: {e}")
84            return False
85
86
87    def delete_data(self, mssv):
88        try:
89            delete_query = sql.SQL("DELETE FROM {} WHERE mssv = %s").format(sql.Identifier(self.table_name))
90            self.cur.execute(delete_query, (mssv,))
91            self.conn.commit()
92            return True
93        except Exception as e:
94            self.conn.rollback()
95            print(f"Delete error: {e}")
96            return False
97
98
```

```
● ● ●
1  from flask import Flask, render_template, request, redirect, session, flash
2  from flask_bootstrap import Bootstrap
3  from model import Database
4  import os
5
6  app = Flask(__name__)
7  Bootstrap(app)
8  app.secret_key = "AFCAHNF&A@435" # For session management
9
10 # Configure database connection
11
12 db = Database("quanlysinhvien", "postgres", "123456", "localhost", "5432", "sinhvien")
13 db.connect_db()
14
15 # Routes
16 @app.route("/")
17 def index():
18     if "username" in session:
19         return redirect("/students")
20     return redirect("/login")
21
22 @app.route("/login", methods=["GET", "POST"])
23 def login():
24     if request.method == "POST":
25         username = request.form.get("username")
26         password = request.form.get("password")
27         # if db.validate_user(username, password):
28         #     session["username"] = username
29         #     return redirect("/students")
30         # else:
31         #     flash("Invalid username or password", "danger")
32         if username == "postgres" and password == "123456":
33             session["username"] = username
34             return redirect("/students")
35         else:
36             flash("Invalid username or password", "danger")
37     return render_template("login.html")
38
39 @app.route("/logout")
40 def logout():
41     session.pop("username", None)
42     return redirect("/login")
43
44 @app.route("/students", methods=["GET", "POST"])
45 def students():
46     if "username" not in session:
47         return redirect("/login")
48     if request.method == "POST":
49         mssv = request.form.get("mssv")
50         ho = request.form.get("ho")
51         ten = request.form.get("ten")
52         image = request.files["image"]
53         image_path = os.path.join("uploads", image.filename)
54         image.save(image_path)
55         if db.insert_data(mssv, ho, ten, image_path):
56             flash("Student added successfully", "success")
57         else:
58             flash("Error adding student", "danger")
59     students = db.load_data()
60     return render_template("students.html", students=students)
61
```

```
● ● ●
```

```
1  @app.route("/add_student", methods=["GET", "POST"])
2  def add_student():
3      if "username" not in session:
4          return redirect("/login")
5      if request.method == "POST":
6          mssv = request.form.get("mssv")
7          ho = request.form.get("ho")
8          ten = request.form.get("ten")
9          image = request.files["image"]
10         image_path = os.path.join("uploads", image.filename)
11         image.save(image_path)
12         if db.insert_data(mssv, ho, ten, image_path):
13             flash("Student added successfully", "success")
14         else:
15             flash("Error adding student", "danger")
16         return redirect("/students")
17     return render_template("add_student.html")
18
19
20 @app.route("/edit_student<string:mssv>", methods=["GET", "POST"])
21 def edit_student(mssv):
22     if "username" not in session:
23         return redirect("/login")
24     if request.method == "POST":
25         ho = request.form.get("ho")
26         ten = request.form.get("ten")
27         image = request.files.get("image")
28         image_path = None
29         if image:
30             image_path = os.path.join("uploads", image.filename)
31             image.save(image_path)
32         if db.update_data(mssv, ho, ten, image_path):
33             flash("Student updated successfully", "success")
34         else:
35             flash("Error updating student", "danger")
36         return redirect("/students")
37     student = db.get_student_by_mssv(mssv) # Retrieve student details
38     return render_template("edit_student.html", student=student)
39
40 @app.route("/delete_student<string:mssv>")
41 def delete_student(mssv):
42     if "username" not in session:
43         return redirect("/login")
44     if db.delete_data(mssv):
45         flash("Student deleted successfully", "success")
46     else:
47         flash("Error deleting student", "danger")
48     return redirect("/students")
49
50 if __name__ == "__main__":
51     if not os.path.exists("/uploads"):
52         os.mkdir("/uploads")
53     app.run(debug=True, port=5050)
54
```

4 Github

https://github.com/Nonope2/AdvancePython_Project.git