

TRƯỜNG ĐẠI HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN CUỐI KỲ
MÔN HỌC LẬP TRÌNH PYTHON NÂNG CAO

Bài tập 02:

Quản lý sinh viên

SVTH: Trần Ngọc Phước – 2274802010699

LỚP: 241_71ITSE31003_02

GVHD: Huỳnh Thái Học

TP. Hồ Chí Minh – 10/2024

MỤC LỤC

1. Giới thiệu về psycopg2.....	3
2. Giao diện ứng dụng Quản lý sinh viên.	4
3. Chức năng.....	6
3.1 Load Data từ cơ sở dữ liệu:.....	6
3.2 Lọc sinh viên dựa vào lớp học:	6
3.3 Thêm sinh viên mới vào cơ sở dữ liệu:	6
4. Mã chương trình.	7
5. Đường link github.	10
--- HẾT---	10

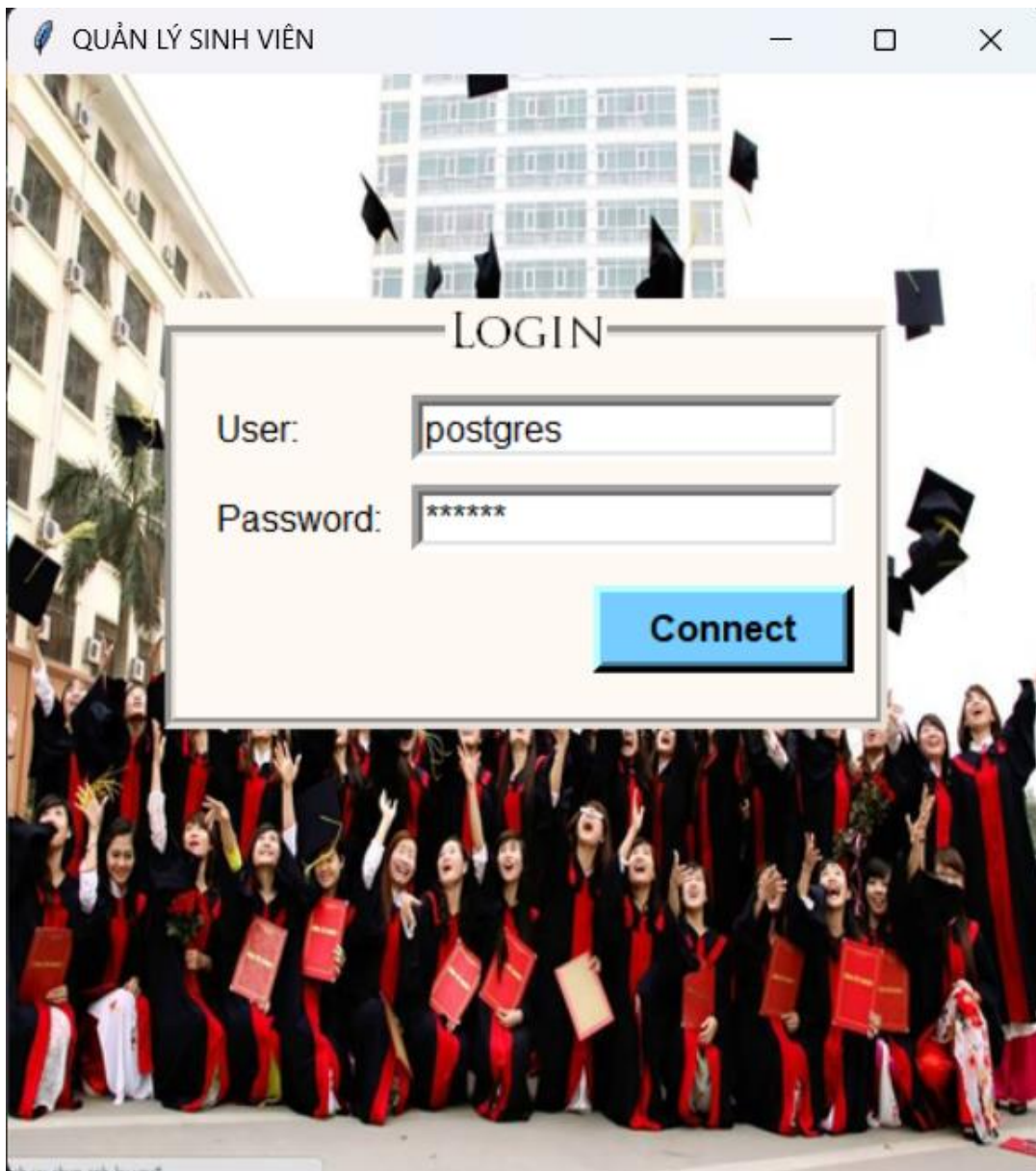
1. Giới thiệu về psycopg2.

psycopg2 là một thư viện Python nổi bật được sử dụng để kết nối và tương tác với cơ sở dữ liệu PostgreSQL. Nó cung cấp các API mạnh mẽ, tuân thủ chuẩn DB-API 2.0 của Python, giúp các lập trình viên thực hiện các thao tác cơ sở dữ liệu như kết nối, truy vấn, và quản lý dữ liệu một cách hiệu quả.

Với khả năng hỗ trợ các tính năng nâng cao của PostgreSQL như xử lý giao dịch, kiểu dữ liệu tùy chỉnh (như ARRAY, JSON, UUID), và hstore, psycopg2 trở thành lựa chọn lý tưởng cho các ứng dụng yêu cầu hiệu suất cao và quản lý dữ liệu an toàn. Bên cạnh đó, thư viện này cũng hỗ trợ các mô hình lập trình đa luồng và bất đồng bộ, giúp tối ưu hóa hiệu năng trong các ứng dụng phức tạp.

2. Giao diện ứng dụng Quản lý sinh viên.

Giao diện để login vào ứng dụng



Giao diện khi login thành công vào ứng dụng

QUẢN LÝ SINH VIÊN

TRUY CẬP DỮ LIỆU

Nhập tên bảng:

LỌC SINH VIÊN

Nhập lớp:

BẢNG THÔNG TIN SINH VIÊN


STT	HỌ VÀ TÊN	MSSV	LỚP
1	Trần Ngọc phước	2274802010699	CNTT20
2	Trần Ngọc Phước	123456789	CNTT20
3	Trần Ngọc Phước	13456789	CNTT20
4	Trần Ngọc Phước	1315456789	CNTT22
5	Trần Ngọc Phước	13515456789	CNTT22
6	Trần Nhật Nam	151511	CNTT23
7	Vũ Thành Đạt	156518489	CNTT23
8	Vũ Thành Đạt	1658165616	CNTT23
9	Nguyễn Văn Bảo	16581656165	CNTT23
10	Lê Bảo Bình	215115515	CNTT24

THÊM SINH VIÊN

HỌ VÀ TÊN:

MSSV:

LỚP:



Giao diện khi Search thông tin sinh viên dựa vào mã lớp học

QUẢN LÝ SINH VIÊN

TRUY CẬP DỮ LIỆU

Nhập tên bảng:

LỌC SINH VIÊN

Nhập lớp:

BẢNG THÔNG TIN SINH VIÊN

STT	HỌ VÀ TÊN	MSSV	LỚP
1	Trần Nhật Nam	151511	CNTT23
2	Vũ Thành Đạt	156518489	CNTT23
3	Vũ Thành Đạt	1658165616	CNTT23
4	Nguyễn Văn Bảo	16581656165	CNTT23

THÊM SINH VIÊN

HỌ VÀ TÊN:

MSSV:

LỚP:



3. Chức năng.

Ứng dụng Quản lý sinh viên được xây dựng bằng Python và thư viện psycopg2 có các chức năng chính sau:

3.1 Load Data từ cơ sở dữ liệu:

- Người dùng có thể nhập tên danh sách sinh viên và ô “Nhập tên bảng”
- Phần này người dùng cung cấp dữ liệu đầu vào là “danhsach” và nhấn nút “Load Data”, ứng dụng sẽ tải toàn bộ danh sách sinh viên có trong cơ sở dữ liệu và hiển thị vào “BẢNG THÔNG TIN SINH VIÊN”

3.2 Lọc sinh viên dựa vào lớp học:

- Người dùng nhập lớp cần tìm vào ô “Nhập lớp”
- Phần này người dùng cung cấp dữ liệu đầu vào là “mã lớp” và nhấn nút “Seacrh”, ứng dụng sẽ hiển thị toàn bộ danh sách sinh viên của lớp học đó vào “BẢNG THÔNG TIN SINH VIÊN”.

3.3 Thêm sinh viên mới vào cơ sở dữ liệu:

- Người dùng nhập tên và lớp vào hai ô nhập liệu tương ứng.
- Phần này người dùng cung cấp dữ liệu đầu vào là “tên sinh viên mới”, “MSSV” và “Mã lớp học” khi nhấn nút “Insert Data”, ứng dụng sẽ thêm sinh viên mới đó vào mã lớp học tương ứng.

4. Mã chương trình.

Về giao diện:

- Toàn bộ giao diện được sắp xếp gọn gàng, sử dụng các phương pháp quản lý bố cục grid() và LabelFrame để tạo các khối nội dung phân tách rõ ràng, gồm ba phần chính: giao diện login, giao diện hệ thống chứa các chức năng chính.

```
28 def create_login_screen(self):
29     font_title = font.Font(family='Trajan Pro', size=16,)
30     font_text = font.Font(family='Eccentric Std', size=12)
31     font_bold = font.Font(family='Eccentric Std', size=12, 'bold')
32
33     # Mờ ảnh nền bằng Pillow
34     image = Image.open("imgs/dai-hoc-tot-o-vn-2.png") # Đường dẫn đến file ảnh
35     image = image.resize((500, 500), Image.ANTIALIAS) # Điều chỉnh kích thước ảnh
36     bg_image = ImageTk.PhotoImage(image)
37
38     # Tạo một Label để chứa ảnh nền
39     self.background_label = tk.Label(self.root, image=bg_image)
40     self.background_label.place(x=0, y=0, relwidth=1, relheight=1)
41     # Giữ tham chiếu đến ảnh nền để tránh bị garbage collected
42     self.background_label.image = bg_image
43
44     self.connection_frame = tk.LabelFrame(self.root, text="Login", font=font_title, bd=5, bg="#FEF9F2", pady=10, padx=10, labelanchor='n')
45     self.connection_frame.grid(row=0, column=0, padx=70, pady=(100,200))
46
47     tk.Label(self.connection_frame, text="User:", font=font_text, bg="#FEF9F2", ).grid(row=0, column=0, padx=5, pady=5, sticky='w')
48     tk.Entry(self.connection_frame, textvariable=self.user, font=font_text, bd=5).grid(row=0, column=1, padx=5, pady=5)
49
50     tk.Label(self.connection_frame, text="Password:", font=font_text, bg="#FEF9F2", ).grid(row=1, column=0, padx=5, pady=5, sticky='w')
51     tk.Entry(self.connection_frame, textvariable=self.password, show="*", font=font_text, bd=5).grid(row=1, column=1, padx=5, pady=5)
52
53     tk.Button(self.connection_frame, text="Connect", font=font_bold, command=self.connect_db, bd=5, width=10, bg='#77CDDF').grid(row=2, column=1,
54
```

```
74 def create_main_screen(self):
75     font_title = font.Font(family='Trajan Pro', size=16,)
76     font_text = font.Font(family='Eccentric Std', size=12)
77     font_bold = font.Font(family='Eccentric Std', size=12, 'bold')
78
79     # Đặt màu nền là màu #384B70 cho giao diện chính
80     self.root['background'] = '#384B70'
81
82     label_title = tk.Label(self.root, text="QUẢN LÝ SINH VIÊN", font=font_title, fg="white", bg='#384B70')
83     label_title.grid(row=0, column=0, columnspan=2, pady=(10,0))
84
85     # Query section
86     query_frame = tk.LabelFrame(self.root, text="Truy cập dữ liệu", width=200, font=font_title, fg="white", bg='#384B70', bd=5)
87     query_frame.grid(row=1, column=0, padx=10, pady=10, sticky='w')
88
89     tk.Label(query_frame, text="Nhập tên bảng:", font=font_text, bg='#384B70', fg='white').grid(row=0, column=0, padx=5, pady=5)
90     tk.Entry(query_frame, textvariable=self.table_name, font=font_text, bd=5).grid(row=0, column=1, padx=5, pady=5)
91     tk.Button(query_frame, text="Load Data", command=self.load_data, font=font_bold, bg="#D4BDAC", bd=2).grid(row=0, column=2, columnspan=1)
92
93     label_titledb = tk.Label(self.root, text="BẢNG THÔNG TIN SINH VIÊN", font=font_title, fg="#FFBD73", bg='#384B70')
94     label_titledb.grid(row=2, column=0, padx=10, sticky='w')
95
96     # Data display table
97     style = ttk.Style()
98     style.configure("Treeview.Heading", background="#384B70", foreground="#9A7E6F", font=font_bold)
```



```

100 self.tree = ttk.Treeview(self.root, columns=('STT', 'HỌ VÀ TÊN', 'MSSV', 'LỚP'), show='headings', height=10)
101
102 self.tree.column('STT', width=50, anchor='center')
103 self.tree.column('HỌ VÀ TÊN', width=180, anchor='w')
104 self.tree.column('MSSV', width=100, anchor='w')
105 self.tree.column('LỚP', width=100, anchor='center')
106
107
108 # Set heading name
109 self.tree.heading('STT', text='STT')
110 self.tree.heading('HỌ VÀ TÊN', text='HỌ VÀ TÊN')
111 self.tree.heading('MSSV', text='MSSV')
112 self.tree.heading('LỚP', text='LỚP')
113
114
115 self.tree.grid(row=3, column=0, padx=5, pady=5, )
116
117 # Insert section
118 insert_frame = tk.LabelFrame(self.root, text="Thêm Sinh Viên", font=font_title, fg="white", bg='#384B70', labelanchor='n', bd=5)
119 insert_frame.grid(row=4, column=0, columnspan=1, padx=10, pady=10)
120
121 self.column1 = tk.StringVar() # Họ và tên
122 self.column2 = tk.StringVar() # MSSV
123 self.column3 = tk.StringVar() # Lớp
124

```

```

125 tk.Label(insert_frame, text="HỌ VÀ TÊN:", font=font_text, bg='#384B70', fg='white').grid(row=0, column=0, padx=5, pady=5)
126 tk.Entry(insert_frame, textvariable=self.column1, width=33, font=font_text, bd=5).grid(row=0, column=1, padx=5, pady=5)
127
128 tk.Label(insert_frame, text="MSSV:", font=font_text, bg='#384B70', fg='white').grid(row=1, column=0, padx=5, pady=5, sticky='w')
129 tk.Entry(insert_frame, textvariable=self.column2, width=33, font=font_text, bd=5).grid(row=1, column=1, padx=5, pady=5)
130
131
132 tk.Label(insert_frame, text="LỚP:", font=font_text, bg='#384B70', fg='white').grid(row=2, column=0, padx=5, pady=5, sticky='w')
133 tk.Entry(insert_frame, textvariable=self.column3, width=33, font=font_text, bd=5).grid(row=2, column=1, padx=5, pady=5)
134
135 tk.Button(insert_frame, text="Insert Data", command=self.insert_data, width=15, font=font_bold, bg="#D4BDAC", bd=5).grid(row=3, column=1, padx=5, pady=5)
136
137 # Search section
138 search_frame = tk.LabelFrame(self.root, text="Lọc sinh viên", font=font_title, fg="white", bg='#384B70', bd=5)
139 search_frame.grid(row=1, column=1, sticky='w', padx=(0,10))
140
141 self.search_value = tk.StringVar()
142
143 tk.Label(search_frame, text="Nhập lớp:", font=font_text, bg='#384B70', fg='white').grid(row=0, column=0, padx=5, pady=5)
144 tk.Entry(search_frame, textvariable=self.search_value, font=font_text, bd=5).grid(row=0, column=1, padx=5, pady=5)
145
146 tk.Button(search_frame, text="Tìm Kiếm", command=self.search_data, font=font_bold, bg="#D4BDAC", bd=2).grid(row=0, column=2, padx=5, pady=5)
147

```

```

148 # thêm ảnh vào root
149 image = Image.open("imgs/dai-hoc-tot-o-vn-2.png") # Đường dẫn đến ảnh của bạn
150 # Thay đổi kích thước ảnh nếu cần
151 image = image.resize((380, 420), Image.ANTIALIAS) # Điều chỉnh kích thước (nếu cần)
152 # Chuyển ảnh thành định dạng tkinter
153 photo = ImageTk.PhotoImage(image)
154 # Tạo Label và chèn ảnh vào
155 label = tk.Label(root, image=photo)
156 label.grid(row=3, column=1, rowspan=2, pady=10, sticky='n')
157 # Giữ tham chiếu tới ảnh để tránh bị garbage collected
158 label.image = photo
159
160

```


Về các chức năng chính:

- Kết nối vào database

```
55 def connect_db(self):
56     try:
57         self.conn = psycopg2.connect(
58             dbname=self.db_name.get(),
59             user=self.user.get(),
60             password=self.password.get(),
61             host=self.host.get(),
62             port=self.port.get()
63         )
64         self.cur = self.conn.cursor()
65         messagebox.showinfo("Success", "Connected to the database successfully!")
66         self.connection_frame.grid_forget() # Ẩn khung đăng nhập sau khi thành công
67         self.background_label.place_forget() # Ẩn hình nền
68         self.create_main_screen()
69     except Exception as e:
70         messagebox.showerror("Error", f"Error connecting to the database: {e}")
```

- Load Data

```
162 def load_data(self):
163     try:
164         query = sql.SQL("SELECT * FROM {}").format(sql.Identifier(self.table_name.get()))
165         self.database.cur.execute(query)
166         rows = self.database.cur.fetchall()
167         self.tree.delete(*self.tree.get_children()) # Clear previous data
168         for index, row in enumerate(rows, start=1):
169             self.tree.insert('', 'end', values=(index, row[1], row[2], row[3])) # Assumes columns are [mssv, name, class]
170     except Exception as e:
171         messagebox.showerror("Error", f"Không tìm thấy dữ liệu: {e}")
172
```

- Insert data

```
175 def insert_data(self):
176     name = self.column1.get()
177     mssv = self.column2.get()
178     student_class = self.column3.get()
179
180     if not name or not mssv or not student_class:
181         messagebox.showerror("Error", "Không được bỏ trống ô nhập liệu!")
182         return
183
184     try:
185         # Check if MSSV already exists
186         check_query = sql.SQL("SELECT 1 FROM {} WHERE mssv = %s").format(sql.Identifier(self.table_name.get()))
187         self.database.cur.execute(check_query, (mssv,))
188         if self.database.cur.fetchone():
189             messagebox.showerror("Error", "MSSV đã tồn tại!")
190             return
191
192         insert_query = sql.SQL("INSERT INTO {} (name, mssv, class) VALUES (%s, %s, %s)").format(sql.Identifier(self.table_name.get()))
193         self.database.cur.execute(insert_query, (name, mssv, student_class))
194         self.database.conn.commit()
195         messagebox.showinfo("Success", "Thêm sinh viên mới thành công!")
196     except Exception as e:
197         messagebox.showerror("Error", f"Error inserting data: {e}")
```

- Search data

```
201     def search_data(self):
202         try:
203             search_query = sql.SQL("SELECT * FROM {} WHERE class = %s").format(sql.Identifier(self.table_name.get()))
204             self.database.cur.execute(search_query, (self.search_value.get(),))
205             rows = self.database.cur.fetchall()
206
207             # Clear previous data from Treeview
208             for item in self.tree.get_children():
209                 self.tree.delete(item)
210
211             # Display search results
212             if rows:
213                 for idx, row in enumerate(rows, start=1):
214                     self.tree.insert("", "end", values=(idx, row[1], row[2], row[3])) # Assuming the order of columns is (mssv, name, c
215             else:
216                 messagebox.showinfo("No Data", "Không tìm thấy lớp này!")
217         except Exception as e:
218             messagebox.showerror("Error", f"Error searching data: {e}")
```

5. Đường link github.

<https://github.com/phuocsasc/PythonNC/tree/master/Baitap01>

--- HẾT ---