TRƯỜNG ĐẠI HỌC VĂN LANG

**KHOA CÔNG NGHỆ THÔNG TIN**

A red and black shield with white logo

Description automatically generated

**BÁO CÁO ĐỒ ÁN 1**

NGÀNH: CÔNG NGHỆ THÔNG TIN

***Đề tài:***

**BOOKING CALENDAR**

**PETSHOP**

**SVTH: Nguyễn Nhật Huy**

**MSSV: 2274802010318**

**GVHD: ThS Huỳnh Thái Học**

TP. Hồ Chí Minh – năm 2024

**1 / Giao diện**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**2/ Chức năng**

**2.1 / Nhập thông tin dịch vụ**

**A white rectangle with black lines

Description automatically generated**

Nhập thông tin dịch vụ petshop bao gồm :

* Name
* Service

A close up of a box

Description automatically generated

**2.2 / Calendar booking**

**A screenshot of a computer

Description automatically generated**

Nhấn “Book” vào ngày mình cần đặt lịch , dữ liệu sẽ được đưa về database

Ngoài ra bạn có thể chuyển tháng theo ý muốn :

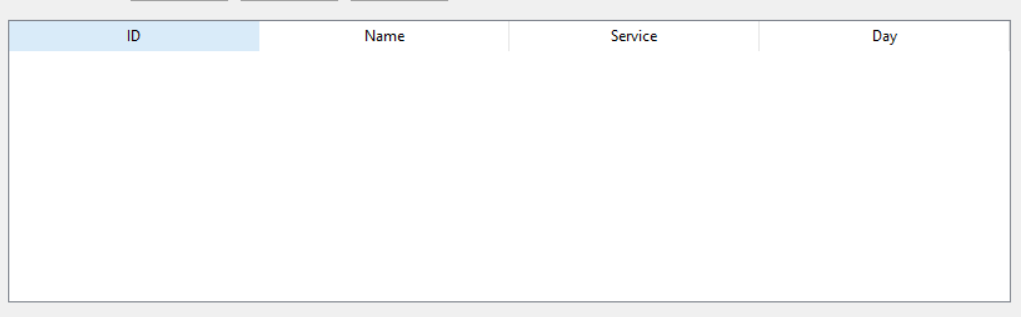
A close up of a text

Description automatically generated

A close up of a text

Description automatically generated

**2.3 / Load data**



Khi nhập đầy đủ thông tin vào chọn Book ở Calendar booking , thông tin về dịch vụ sẽ được show ra dưới table

A screenshot of a computer

Description automatically generated

**2.4 / Delete data**

**A white rectangular sign with black text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

Click vào dịch vụ bạn cần xóa

Sau đó nhấn delete để xóa dịch vụ được chỉ định

A screenshot of a computer

Description automatically generated

**2.5 / Close**



Nhấn Close để đóng cửa sổ

**3 / Mã chương trình**

import tkinter as tk

from tkinter import ttk, messagebox

import psycopg2

import calendar

from datetime import datetime

class DatabaseApp:

    def \_\_init\_\_(self, root):

        self.root = root

        self.root.title("DatabaseApp")

        self.db\_name = tk.StringVar(value='dbtest')

        self.user = tk.StringVar(value='postgres')

        self.password = tk.StringVar(value='123')

        self.host = tk.StringVar(value='localhost')

        self.port = tk.StringVar(value='5432')

        self.create\_widgets()

    def create\_widgets(self):

        connection\_frame = tk.Frame(self.root)

        connection\_frame.pack(pady=10)

        tk.Label(connection\_frame, text="DB Name:").grid(row=0, column=0, padx=5, pady=5)

        tk.Entry(connection\_frame, textvariable=self.db\_name).grid(row=0, column=1, padx=5, pady=5)

        tk.Label(connection\_frame, text="User:").grid(row=1, column=0, padx=5, pady=5)

        tk.Entry(connection\_frame, textvariable=self.user).grid(row=1, column=1, padx=5, pady=5)

        tk.Label(connection\_frame, text="Password:").grid(row=2, column=0, padx=5, pady=5)

        tk.Entry(connection\_frame, textvariable=self.password, show="\*").grid(row=2, column=1, padx=5, pady=5)

        tk.Label(connection\_frame, text="Host:").grid(row=3, column=0, padx=5, pady=5)

        tk.Entry(connection\_frame, textvariable=self.host).grid(row=3, column=1, padx=5, pady=5)

        tk.Label(connection\_frame, text="Port:").grid(row=4, column=0, padx=5, pady=5)

        tk.Entry(connection\_frame, textvariable=self.port).grid(row=4, column=1, padx=5, pady=5)

        tk.Button(connection\_frame, text="Connect", command=self.connect\_db).grid(row=5, columnspan=2, pady=10)

    def connect\_db(self):

        try:

            self.conn = psycopg2.connect(

                dbname=self.db\_name.get(),

                user=self.user.get(),

                password=self.password.get(),

                host=self.host.get(),

                port=self.port.get()

            )

            self.cursor = self.conn.cursor()

            messagebox.showinfo("Success", "Connected to the database successfully!")

            self.open\_main\_window()

        except Exception as e:

            messagebox.showerror("Connection Error", f"Failed to connect to the database: {e}")

    def open\_main\_window(self):

        self.root.withdraw()  # Đóng cửa sổ kết nối

        self.main\_window = tk.Toplevel(self.root)

        self.main\_window.title("Booking Calendar")

        BookingCalendarApp(self.main\_window, self.cursor, self.conn)  # Khởi tạo ứng dụng chính

class BookingCalendarApp:

    def \_\_init\_\_(self, root, cursor, conn):

        self.root = root

        self.cursor = cursor

        self.conn = conn

        # Biến toàn cục cho lịch và ngày đã đặt

        self.current\_year = datetime.now().year

        self.current\_month = datetime.now().month

        # Thiết lập giao diện chính

        self.setup\_ui()

        # Cập nhật lịch

        self.update\_calendar()

    def setup\_ui(self):

        # Khung chính để chứa tất cả các thành phần

        self.main\_frame = ttk.Frame(self.root)

        self.main\_frame.pack(fill=tk.BOTH, expand=True, padx=20, pady=20)

        # Khung cho việc nhập tên và dịch vụ

        self.entry\_frame = ttk.Frame(self.main\_frame)

        self.entry\_frame.pack(fill=tk.X, pady=10)

        self.name\_label = ttk.Label(self.entry\_frame, text="Enter name:")

        self.name\_label.grid(row=0, column=0, padx=10)

        self.name\_entry = ttk.Entry(self.entry\_frame)

        self.name\_entry.grid(row=0, column=1, padx=10, sticky=tk.W)

        self.service\_label = ttk.Label(self.entry\_frame, text="Service:")

        self.service\_label.grid(row=1, column=0, padx=10)

        self.service\_entry = ttk.Entry(self.entry\_frame)

        self.service\_entry.grid(row=1, column=1, padx=10, sticky=tk.W)

        # Khung chỉnh tháng

        self.month\_frame = ttk.Frame(self.main\_frame)

        self.month\_frame.pack(pady=10)

        self.prev\_month\_button = ttk.Button(self.month\_frame, text="<", command=lambda: self.change\_month(-1))

        self.prev\_month\_button.pack(side="left", padx=10)

        self.month\_label = ttk.Label(self.month\_frame, text=f"Month: {calendar.month\_name[self.current\_month]}")

        self.month\_label.pack(side="left", padx=10)

        self.next\_month\_button = ttk.Button(self.month\_frame, text=">", command=lambda: self.change\_month(1))

        self.next\_month\_button.pack(side="left", padx=10)

        # Khung lịch

        self.calendar\_frame = ttk.Frame(self.main\_frame)

        self.calendar\_frame.pack(pady=10)

        # Treeview để hiển thị các ngày đã đặt

        self.tree = ttk.Treeview(self.main\_frame, columns=('ID', 'Name', 'Service', 'Day'), show='headings')

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

        self.tree.heading('Name', text='Name')

        self.tree.heading('Service', text='Service')

        self.tree.heading('Day', text='Day')

        self.tree.pack(fill=tk.BOTH, expand=True)

        # Khung nút hành động

        self.action\_buttons\_frame = ttk.Frame(self.main\_frame)

        self.action\_buttons\_frame.pack(pady=10)

        # Nút Xóa

        self.btn\_delete = ttk.Button(self.action\_buttons\_frame, text="Delete", command=self.delete\_booking)

        self.btn\_delete.pack(pady=10)

        # Nút Đóng ở dưới cùng

        self.btn\_close = ttk.Button(self.action\_buttons\_frame, text="Close", command=self.root.quit)

        self.btn\_close.pack(pady=10)

    def update\_calendar(self):

        # Xóa hiển thị lịch hiện tại

        for widget in self.calendar\_frame.winfo\_children():

            widget.destroy()

        # Tạo lịch cho tháng hiện tại

        days\_in\_month = calendar.monthrange(self.current\_year, self.current\_month)[1]

        for day in range(1, days\_in\_month + 1):

            day\_frame = ttk.Frame(self.calendar\_frame, relief="solid", borderwidth=1)

            day\_frame.grid(row=(day - 1) // 7, column=(day - 1) % 7, padx=5, pady=5)

            day\_label = ttk.Label(day\_frame, text=f"{day}/{self.current\_month}")

            day\_label.pack()

            book\_btn = ttk.Button(day\_frame, text="Book", command=lambda d=day: self.book\_day(d))

            book\_btn.pack()

    def book\_day(self, day):

        name = self.name\_entry.get()

        service = self.service\_entry.get()

        if not name or not service:

            messagebox.showerror("Error", "Please enter your name and service.")

            return

        # Chèn đặt chỗ vào database

        self.cursor.execute(

            "INSERT INTO bookdays (name, service, day) VALUES (%s, %s, %s)",

            (name, service, f"{day}/{self.current\_month}/{self.current\_year}")

        )

        self.conn.commit()

        # Cập nhật Treeview với dữ liệu từ database

        self.load\_bookings()

    def load\_bookings(self):

        # Tải đặt chỗ từ database và hiển thị trong Treeview

        self.tree.delete(\*self.tree.get\_children())

        self.cursor.execute("SELECT \* FROM bookdays")

        bookings = self.cursor.fetchall()

        for booking in bookings:

            self.tree.insert('', 'end', values=booking)

    def delete\_booking(self):

        selected = self.tree.selection()

        if selected:

            for item in selected:

                item\_values = self.tree.item(item, 'values')

                # Xóa đặt chỗ khỏi database

                self.cursor.execute(

                    "DELETE FROM bookdays WHERE id = %s",

                    (item\_values[0],)  # Giả sử cột đầu tiên là ID

                )

                self.conn.commit()

                self.tree.delete(item)  # Xóa khỏi Treeview

            messagebox.showinfo("Success", "Selected bookings deleted successfully.")

        else:

            messagebox.showwarning("Warning", "No booking selected.")

    def change\_month(self, delta):

        # Thay đổi tháng và cập nhật lịch

        self.current\_month += delta

        if self.current\_month < 1:

            self.current\_month = 12

            self.current\_year -= 1

        elif self.current\_month > 12:

            self.current\_month = 1

            self.current\_year += 1

        self.month\_label.config(text=f"Month: {calendar.month\_name[self.current\_month]}")

        self.update\_calendar()

if \_\_name\_\_ == "\_\_main\_\_":

    root = tk.Tk()

    app = DatabaseApp(root)

    root.mainloop()

**4 / Github link**

https://github.com/Perry110/pythonnc.git