

```

# -*- coding: utf-8 -*-
"""
Created on Fri Jan 5 20:08:20 2024

@author: DELL
"""

import tkinter as tk
from connexion_db import connect_to_db
import mysql.connector

def clear_entries(*entries):
    for entry in entries:
        entry.delete(0, tk.END)

def ajouter_client(database):
    def add_client(nom, adresse, email, telephone):
        if database:
            try:
                cursor = database.cursor()
                cursor.execute("SELECT MAX(id_client) + 1 FROM client")
                next_id = cursor.fetchone()[0]
                query = "INSERT INTO client (id_client, nom_client, adresse, adresse_email, num_"
                values = (next_id, nom, adresse, email, telephone)
                cursor.execute(query, values)
                database.commit()
                print("Client ajouté avec succès.")
                clear_entries(entry_nom, entry_adr, entry_email, entry_telephone)
            except mysql.connector.Error as err:
                print(f"Erreur lors de l'ajout du client : {err}")
            finally:
                cursor.close()
                database.close()
        else:
            print("La connexion à la base de données a échoué.")

    def ajouter_clt():
        nom = entry_nom.get()
        adresse = entry_adr.get()
        email = entry_email.get()
        telephone = entry_telephone.get()
        add_client(nom, adresse, email, telephone)

    root = tk.Tk()
    root.title("Ajouter un client")

    label_nom = tk.Label(root, text="Nom:")
    label_nom.grid(row=0, column=0)
    entry_nom = tk.Entry(root)
    entry_nom.grid(row=0, column=1)

    label_adresse = tk.Label(root, text="adresse:")
    label_adresse.grid(row=1, column=0)
    entry_adr = tk.Entry(root)
    entry_adr.grid(row=1, column=1)

    label_email = tk.Label(root, text="Email:")
    label_email.grid(row=2, column=0)
    entry_email = tk.Entry(root)
    entry_email.grid(row=2, column=1)

```

```

label_telephone = tk.Label(root, text="Téléphone:")
label_telephone.grid(row=3, column=0)
entry_telephone = tk.Entry(root)
entry_telephone.grid(row=3, column=1)

btn_ajouter = tk.Button(root, text="Ajouter Client", command=ajouter_clt)
btn_ajouter.grid(row=4, columnspan=2)

root.mainloop()

```

```

def chercher_client(database):
    def rechercher():
        client_recherche = entry_client.get()
        try:
            cursor = database.cursor()
            query = f"SELECT CONCAT('id: ', id_client, ', nom: ', nom_client, ', adresse: ', adr"
            cursor.execute(query)
            result = cursor.fetchall()

            text_area.delete(1.0, tk.END)

            for row in result:
                text_area.insert(tk.END, row[0] + "\n")
        except mysql.connector.Error as err:
            print(f"Erreur lors de la recherche de produit : {err}")

```

```

root = tk.Tk()
root.title("Recherche de client")

label = tk.Label(root, text="Nom du Client à rechercher:")
label.pack()

entry_client = tk.Entry(root)
entry_client.pack()

search_button = tk.Button(root, text="Rechercher", command=rechercher)
search_button.pack()

text_area = tk.Text(root, height=10, width=80)
text_area.pack()

root.mainloop()

```

```

def lister_client(database):
    try:
        cursor = database.cursor()
        cursor.execute("SELECT CONCAT('id: ', id_client, ', nom: ', nom_client, ', adresse: ', a"
        result = cursor.fetchall()
        root = tk.Tk()
        root.title("Liste des clients")

        text_area = tk.Text(root, height=10, width=80)
        text_area.pack()

        for row in result:
            text_area.insert(tk.END, row[0] + "\n")

        root.mainloop()
    except mysql.connector.Error as err:
        print(f"Erreur lors de la récupération des clients : {err}")

```

```

def supprimer_client(database):
    def delete_clt():
        client_name = entry_nom_client.get()
        try:
            cursor = database.cursor()
            query = "DELETE FROM client WHERE nom_client = %s"
            cursor.execute(query, (client_name,))
            database.commit()
            print("Produit supprimé avec succès.")
            clear_entries(entry_nom_client)
        except mysql.connector.Error as err:
            print(f"Erreur lors de la suppression du client : {err}")

    root = tk.Tk()
    root.title("Suppression de client")

    label_nom = tk.Label(root, text="Nom du client à supprimer:")
    label_nom.pack()

    entry_nom_client = tk.Entry(root)
    entry_nom_client.pack()

    btn_supprimer = tk.Button(root, text="Supprimer", command=delete_clt)
    btn_supprimer.pack()

    root.mainloop()

def modifier_client(database):
    def rechercher_client():
        nom_client = entry_nom_client.get()
        try:
            cursor = database.cursor()
            query = "SELECT * FROM client WHERE nom_client = %s"
            cursor.execute(query, (nom_client,))
            result = cursor.fetchone()

            if result:
                root_modif = tk.Tk()
                root_modif.title("Modifier client")

                labels = ["Nom du client:", "Adresse:", "Email:", "Téléphone:"]
                entries = []
                for i, label_text in enumerate(labels):
                    label = tk.Label(root_modif, text=label_text)
                    label.grid(row=i, column=0)
                    entry = tk.Entry(root_modif)
                    entry.insert(tk.END, result[i+1])
                    entry.grid(row=i, column=1)
                    entries.append(entry)

                def modifier():
                    new_values = [entry.get() for entry in entries]
                    update_client(database, result[0], new_values)
                    clear_entries(*entries)
                clear_entries(entry_nom_client)
                btn_modifier = tk.Button(root_modif, text="Modifier", command=modifier)
                btn_modifier.grid(row=len(labels), columnspan=2)

                root_modif.mainloop()
            else:

```

```

        print("Client non trouvé.")
    except mysql.connector.Error as err:
        print(f"Erreur lors de la recherche du client : {err}")
def update_client(database, id_client, new_values):
    try:
        cursor = database.cursor()
        query = "UPDATE client SET nom_client = %s, adresse = %s, adresse_email = %s, num_tel = %s"
        cursor.execute(query, (*new_values, id_client))
        database.commit()
        print("client mis à jour avec succès !")
    except mysql.connector.Error as err:
        print(f"Erreur lors de la mise à jour du client : {err}")

root = tk.Tk()
root.title("Modifier client")

label_nom_client = tk.Label(root, text="Nom du client à Modifier:")
label_nom_client.pack()

entry_nom_client = tk.Entry(root)
entry_nom_client.pack()

btn_rechercher = tk.Button(root, text="Rechercher", command=rechercher_client)
btn_rechercher.pack()

root.mainloop()

```