

```

# -*- coding: utf-8 -*-
"""
Created on Wed Jan 10 14:44:24 2024

@author: DELL
"""

import tkinter as tk
from connexion_db import connect_to_db
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas
import os
import mysql.connector
from datetime import datetime

database = connect_to_db()

def create_command(nom_client, nom_produit, prix_unitaire, quantite, montant_total):
    maintenant = datetime.now()
    date_heure_formattees = maintenant.strftime("%Y-%m-%d_%H-%M-%S")
    pdf_filename = f"COM_{nom_client}_{date_heure_formattees}.pdf"
    pdf_path = os.path.abspath(pdf_filename)
    c = canvas.Canvas(pdf_filename, pagesize=letter)
    c.drawString(100, 750, "Commande")
    c.drawString(100, 730, f"Nom du client: {nom_client}")
    c.drawString(100, 710, f"Nom du produit: {nom_produit}")
    c.drawString(100, 690, f"Prix unitaire: {prix_unitaire}")
    c.drawString(100, 670, f"Quantité: {quantite}")
    c.drawString(100, 650, f"Montant total: {montant_total}")
    c.save()
    save_to_database(pdf_filename, pdf_path)
    return pdf_filename

def save_to_database(nom_fichier_pdf, chemin_pdf):
    try:
        cursor = database.cursor()
        query = "INSERT INTO commande (id_cmd, chemin) VALUES (%s, %s)"
        values = (nom_fichier_pdf, chemin_pdf)
        cursor.execute(query, values)
        database.commit()
        print("Informations de commande enregistrées dans la base de données.")
    except mysql.connector.Error as err:
        print(f"Erreur lors de l'enregistrement des informations de commande : {err}")

def check_stock(cursor, nom_produit, quantite):
    cursor.execute("SELECT quantite FROM produit WHERE nom_produit = %s", (nom_produit,))
    stock = cursor.fetchone()
    if stock and stock[0] >= quantite:
        return True
    return False

def update_stock(cursor, nom_produit, quantite):
    cursor.execute("UPDATE produit SET quantite = quantite - %s WHERE nom_produit = %s", (quantite, nom_produit,))

def vendre_produit(database):
    def vendre():
        nom_client = entry_nom_client.get()
        nom_produit = entry_nom_produit.get()

```

```

prix_unitaire = float(entry_prix_unitaire.get())
quantite = int(entry_quantite.get())
montant_total = prix_unitaire * quantite

entry_montant_total.config(state='normal')
entry_montant_total.delete(0, tk.END)
entry_montant_total.insert(0, montant_total)
entry_montant_total.config(state='readonly')

cursor = database.cursor()
product_available = check_stock(cursor, nom_produit, quantite)

if product_available:
    update_stock(cursor, nom_produit, quantite)
    database.commit()
    create_command(nom_client, nom_produit, prix_unitaire, quantite, montant_total)
    print("Commande effectuée avec succès.")
else:
    print("Le produit n'est pas en stock ou la quantité demandée est insuffisante.")

root = tk.Tk()
root.title("Passer commande")
root.configure(bg='#7D4FFE')

label_nom_client = tk.Label(root, text="Nom du client:", bg='#7D4FFE', fg="white")
label_nom_client.grid(row=0, column=0)
entry_nom_client = tk.Entry(root)
entry_nom_client.grid(row=0, column=1)

label_nom_produit = tk.Label(root, text="Nom du produit:", bg='#7D4FFE', fg="white")
label_nom_produit.grid(row=1, column=0)
entry_nom_produit = tk.Entry(root)
entry_nom_produit.grid(row=1, column=1)

label_prix_unitaire = tk.Label(root, text="Prix unitaire:", bg='#7D4FFE', fg="white")
label_prix_unitaire.grid(row=2, column=0)
entry_prix_unitaire = tk.Entry(root)
entry_prix_unitaire.grid(row=2, column=1)

label_quantite = tk.Label(root, text="Quantité:", bg='#7D4FFE', fg="white")
label_quantite.grid(row=3, column=0)
entry_quantite = tk.Entry(root)
entry_quantite.grid(row=3, column=1)

btn_vendre = tk.Button(root, text="commander", command=vendre)
btn_vendre.grid(row=4, columnspan=2)

entry_montant_total = tk.Entry(root, state='readonly')
entry_montant_total.grid(row=5, columnspan=2)
root.mainloop()

```