```
import tkinter as tk
from tkinter import messagebox
import random

def get_product_price(product, supermarket):
    prices = {
        'arroz': {'Super Muffato': 5.99, 'Amigão': 6.29, 'Viscardi': 5.79},
        'feijão': {'Super Muffato': 4.99, 'Amigão': 5.19, 'Viscardi': 4.89},
        'macarrão': {'Super Muffato': 2.49, 'Amigão': 2.29, 'Viscardi': 2.39},
        'óleo de soja': {'Super Muffato': 3.99, 'Amigão': 3.89, 'Viscardi': 4.09},
        'sal': {'Super Muffato': 1.29, 'Amigão': 1.19, 'Viscardi': 1.39},
        'açúcar': {'Super Muffato': 2.99, 'Amigão': 2.79, 'Viscardi': 2.89},
        'café': {'Super Muffato': 8.49, 'Amigão': 8.29, 'Viscardi': 8.59},
        'leite': {'Super Muffato': 3.49, 'Amigão': 3.39, 'Viscardi': 3.59},
        'queijo muçarela': {'Super Muffato': 15.99, 'Amigão': 15.49, 'Viscardi': 16.29},
        'presunto': {'Super Muffato': 12.99, 'Amigão': 12.49, 'Viscardi': 13.29},
        'pão de forma': {'Super Muffato': 5.49, 'Amigão': 5.29, 'Viscardi': 5.59},
        'manteiga': {'Super Muffato': 7.99, 'Amigão': 7.79, 'Viscardi': 8.09},
        'ovos': {'Super Muffato': 8.99, 'Amigão': 8.49, 'Viscardi': 9.29},
        'arroz integral': {'Super Muffato': 6.99, 'Amigão': 7.29, 'Viscardi': 6.79},
        'feijão preto': {'Super Muffato': 5.99, 'Amigão': 6.29, 'Viscardi': 5.79},
        'macarrão integral': {'Super Muffato': 2.99, 'Amigão': 2.69, 'Viscardi': 2.89},
        'azeite de oliva': {'Super Muffato': 9.99, 'Amigão': 9.79, 'Viscardi': 10.29},
        'vinagre': {'Super Muffato': 1.79, 'Amigão': 1.69, 'Viscardi': 1.89},
        'farinha de trigo': {'Super Muffato': 3.49, 'Amigão': 3.29, 'Viscardi': 3.59},
        'fermento em pó': {'Super Muffato': 2.29, 'Amigão': 2.19, 'Viscardi': 2.39},
        'chocolate em pó': {'Super Muffato': 4.49, 'Amigão': 4.29, 'Viscardi': 4.59},
        'achocolatado em pó': {'Super Muffato': 6.49, 'Amigão': 6.29, 'Viscardi': 6.59},
        'geléia': {'Super Muffato': 5.99, 'Amigão': 5.79, 'Viscardi': 6.09},
        'biscoito de maisena': {'Super Muffato': 3.49, 'Amigão': 3.29, 'Viscardi': 3.59},
        'biscoito recheado': {'Super Muffato': 2.99, 'Amigão': 2.79, 'Viscardi': 2.89},
        'chá': {'Super Muffato': 4.99, 'Amigão': 4.79, 'Viscardi': 5.09},
        'suco de laranja': {'Super Muffato': 4.49, 'Amigão': 4.29, 'Viscardi': 4.59},
        'refrigerante': {'Super Muffato': 6.99, 'Amigão': 6.89, 'Viscardi': 7.09},
        'cerveja': {'Super Muffato': 3.99, 'Amigão': 3.89, 'Viscardi': 4.09},
        'vinho': {'Super Muffato': 19.99, 'Amigão': 18.99, 'Viscardi': 20.49},
        'água mineral': {'Super Muffato': 2.49, 'Amigão': 2.29, 'Viscardi': 2.59},
        'sabão em pó': {'Super Muffato': 9.99, 'Amigão': 9.79, 'Viscardi': 10.29},
        'detergente': {'Super Muffato': 2.99, 'Amigão': 2.79, 'Viscardi': 2.89},
        'papel higiênico': {'Super Muffato': 6.99, 'Amigão': 6.79, 'Viscardi': 7.09},
        'shampoo': {'Super Muffato': 12.99, 'Amigão': 12.49, 'Viscardi': 13.29},
        'condicionador': {'Super Muffato': 10.99, 'Amigão': 10.49, 'Viscardi': 11.29},
        'sabonete': {'Super Muffato': 1.99, 'Amigão': 1.89, 'Viscardi': 2.09},
        'creme dental': {'Super Muffato': 3.99, 'Amigão': 3.79, 'Viscardi': 4.09},
        'pão francês': {'Super Muffato': 3.49, 'Amigão': 3.29, 'Viscardi': 3.59},
        'batata': {'Super Muffato': 2.49, 'Amigão': 2.29, 'Viscardi': 2.59},
        'cebola': {'Super Muffato': 3.99, 'Amigão': 3.79, 'Viscardi': 4.09},
        'alho': {'Super Muffato': 5.99, 'Amigão': 5.49, 'Viscardi': 6.29},
        'tomate': {'Super Muffato': 4.99, 'Amigão': 4.79, 'Viscardi': 5.09},
        'cenoura': {'Super Muffato': 2.99, 'Amigão': 2.79, 'Viscardi': 2.89},
        'banana': {'Super Muffato': 2.49, 'Amigão': 2.29, 'Viscardi': 2.59},
        'maçã': {'Super Muffato': 4.99, 'Amigão': 4.79, 'Viscardi': 5.09},
        'laranja': {'Super Muffato': 3.49, 'Amigão': 3.29, 'Viscardi': 3.59},
        'melancia': {'Super Muffato': 2.99, 'Amigão': 2.79, 'Viscardi': 2.89},
        'abacaxi': {'Super Muffato': 5.99, 'Amigão': 5.49, 'Viscardi': 6.29},
        'morango': {'Super Muffato': 7.99, 'Amigão': 7.49, 'Viscardi': 8.29},
        'uva': {'Super Muffato': 8.99, 'Amigão': 8.49, 'Viscardi': 9.29},
        'pêssego': {'Super Muffato': 6.99, 'Amigão': 6.49, 'Viscardi': 7.29},
        'melão': {'Super Muffato': 4.99, 'Amigão': 4.79, 'Viscardi': 5.09},
        'couve': {'Super Muffato': 2.49, 'Amigão': 2.29, 'Viscardi': 2.59},
    }
    return prices.get(product, {}).get(supermarket)

def find_best_price(product):
    best_price = float('inf')
    best_supermarket = None
    all_prices = {}

    for supermarket in ["Super Muffato", "Amigão", "Viscardi"]:
        price = get_product_price(product, supermarket)
        all_prices[supermarket] = price
        if price is not None and price < best_price:
            best_price = price
            best_supermarket = supermarket

    if best_supermarket is None:
        return None, None, all_prices
    else:
        return best_supermarket, best_price, all_prices

def add_to_cart(product, price):
    cart.append((product, price))
    messagebox.showinfo("Adicionado", f"{product} adicionado ao carrinho.")

def search_product():
    product = entry_product.get()
    best_supermarket, best_price, all_prices = find_best_price(product)

    if best_supermarket is None:
        messagebox.showerror("Erro", "Produto não encontrado.")
        return

    results_text = f"Melhor preço para {product}: {best_supermarket} por R$
{best_price:.2f}\n\nComparação de preços:\n"
    for supermarket, price in all_prices.items():
        if price is not None:
            results_text += f"  {supermarket}: R$ {price:.2f}\n"
        else:
            results_text += f"  {supermarket}: Não encontrado\n"

    result_label.config(text=results_text)
    add_to_cart_button.config(command=lambda: add_to_cart(product, best_price))

def view_cart():
    clear_screen()
    total_cost = sum(price for _, price in cart)
    cart_text = "Carrinho:\n"
    for product, price in cart:
        cart_text += f"  {product}: R$ {price:.2f}\n"
    cart_text += f"\nTotal: R$ {total_cost:.2f}"

    def checkout():
        clear_screen()
        tk.Label(app, text="Formas de Pagamento:", font=("Arial", 14), bg=navy_blue,
fg=white).pack(pady=10)
        payment_method = tk.StringVar(value="")

        tk.Radiobutton(app, text="Cartão de Crédito", variable=payment_method,
value="Cartão de Crédito", font=("Arial", 12), bg=navy_blue, fg=white, command=lambda:
show_payment_form(payment_method)).pack()
        tk.Radiobutton(app, text="Pix", variable=payment_method, value="Pix", font=("Arial",
12), bg=navy_blue, fg=white, command=lambda:
show_payment_form(payment_method)).pack()

        payment_details_frame = tk.Frame(app, bg=navy_blue)
        payment_details_frame.pack(pady=10)

        tk.Label(app, text="Endereço para Entrega:", font=("Arial", 14), bg=navy_blue,
fg=white).pack(pady=10)
        address_entry = tk.Entry(app, font=("Arial", 12), width=30)
        address_entry.pack(pady=10)

        def confirm_payment():
            address = address_entry.get()
            if address:
                messagebox.showinfo("Confirmação", f"Pagamento confirmado!\nEndereço:
{address}\nEstimativa de entrega: 38 minutos.")
                app.destroy()
            else:
                messagebox.showerror("Erro", "Por favor, insira um endereço válido.")

        tk.Button(app, text="Confirmar Pagamento", font=("Arial", 14), bg=orange, fg=white,
command=confirm_payment).pack(pady=20)
```

```
    def show_payment_form(payment_method):
        clear_payment_details_frame()
        if payment_method.get() == "Cartão de Crédito":
            show_credit_card_form(payment_details_frame)
        elif payment_method.get() == "Pix":
            show_pix_instructions(payment_details_frame)

    def show_credit_card_form(frame):
        tk.Label(frame, text="Detalhes do Cartão de Crédito", font=("Arial", 14), bg=navy_blue,
fg=white).pack(pady=10)

        tk.Label(frame, text="Número do Cartão:", font=("Arial", 12), bg=navy_blue,
fg=white).pack()
        card_number_entry = tk.Entry(frame, font=("Arial", 12), width=30)
        card_number_entry.pack(pady=5)

        tk.Label(frame, text="Validade (MM/AA):", font=("Arial", 12), bg=navy_blue,
fg=white).pack()
        card_validity_entry = tk.Entry(frame, font=("Arial", 12), width=30)
        card_validity_entry.pack(pady=5)

        tk.Label(frame, text="CVC:", font=("Arial", 12), bg=navy_blue, fg=white).pack()
        card_cvc_entry = tk.Entry(frame, font=("Arial", 12), width=30)
        card_cvc_entry.pack(pady=5)

        tk.Button(frame, text="Voltar", font=("Arial", 12), bg=orange, fg=white,
command=checkout).pack(pady=20)

    def show_pix_instructions(frame):
        tk.Label(frame, text="Pagamento via Pix", font=("Arial", 14), bg=navy_blue,
fg=white).pack(pady=10)

        pix_code = str(random.randint(100000000000000, 999999999999999))
        tk.Label(frame, text=f"Código Pix: {pix_code}", font=("Arial", 12), bg=navy_blue,
fg=white).pack(pady=10)
        tk.Label(frame, text="Copie e cole o código no seu aplicativo do banco.", font=("Arial",
12), bg=navy_blue, fg=white).pack(pady=10)

        tk.Button(frame, text="Voltar", font=("Arial", 12), bg=orange, fg=white,
command=checkout).pack(pady=20)

    def clear_payment_details_frame():
        for widget in payment_details_frame.winfo_children():
            widget.pack_forget()

    messagebox.showinfo("Carrinho", cart_text)
    checkout()
```

```
def clear_screen():
    for widget in app.winfo_children():
        widget.pack_forget()

app = tk.Tk()
app.title("Melhor Preço")
app.geometry("375x667")

navy_blue = "#000080"
orange = "#FFA500"
white = "#FFFFFF"

app.configure(bg=navy_blue)

cart = []

frame = tk.Frame(app, bg=navy_blue)
frame.pack(pady=20)

title_label = tk.Label(frame, text="Melhor Preço", font=("Arial", 24), fg=white, bg=navy_blue)
title_label.pack(pady=20)

entry_label = tk.Label(frame, text="Digite o nome do produto:", font=("Arial", 14), fg=white,
bg=navy_blue)
entry_label.pack(pady=10)

entry_product = tk.Entry(frame, font=("Arial", 14), width=25)
entry_product.pack(pady=10)

search_button = tk.Button(frame, text="Procurar", font=("Arial", 14), bg=orange, fg=white,
command=search_product)
search_button.pack(pady=20)

result_label = tk.Label(frame, text="", font=("Arial", 12), fg=white, bg=navy_blue,
justify=tk.LEFT)
result_label.pack(pady=10)

add_to_cart_button = tk.Button(frame, text="Adicionar ao Carrinho", font=("Arial", 14),
bg=orange, fg=white)
add_to_cart_button.pack(pady=10)

view_cart_button = tk.Button(frame, text="Ver Carrinho", font=("Arial", 14), bg=orange,
fg=white, command=view_cart)
view_cart_button.pack(pady=20)

app.mainloop()
```