TRƯỜNG ĐẠI HỌC HỌC VĂN LANG

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN MÔN HỌC

Lập trình Python nâng cao

NGÀNH: CÔNG NGHỆ THÔNG TIN

GVHD: Huỳnh Thái Học

SVTH: Trần Ngọc Long – 2274802010504

Tp. Hồ Chí Minh – năm 2024

1. Nhập thư viện cần thiết

```
import tkinter as tk from tkinter import ttk
from tkinter import messagebox
```

tkinter: Thư viện chính để xây dựng giao diện người dùng (GUI) trong Python.

ttk: Thư viện con của tkinter cung cấp các widget (thành phần giao diện) đẹp hơn và hiện đại hơn. **messagebox:** Thư viện cung cấp các hộp thoại thông báo cho người dùng

2. Khởi tạo cửa sổ chính

```
win = tk.Tk() win.title("Giao diện tính toán")
win.geometry("230x70") win.resizable(False,
False)
```

tk.Tk(): Tạo một đối tượng cửa sổ chính. **title():**

Thiết lập tiêu đề cho cửa sổ. **geometry():** Đặt kích

thước cửa sổ là 230x70 pixel.

resizable(): Đặt giá trị False cho cả chiều rộng và chiều cao để người dùng không thể thay đổi kích thước cửa sổ.

3. Tạo nhãn và ô nhập cho các giá trị a và b

```
# Các nhãn & ô nhập # Ô nhập số A  ttk.Label(win,
text="Số a: ").grid(column=0, row=0, sticky='w')
so_a = tk.StringVar() so_a_entered =
ttk.Entry(win, width=12, textvariable=so_a)
so_a_entered.grid(column=1, row=0)


# Ô nhập số B ttk.Label(win, text="Số b:
").grid(column=0, row=1, sticky='w') so_b =
tk.StringVar()  so_b_entered = ttk.Entry(win, width=12,
textvariable=so_b) so_b_entered.grid(column=1, row=1)
```

ttk.Label: Tạo nhãn với văn bản "Số a". **grid():** Đặt nhãn vào vị trí cột 0, hàng 0 trong lưới, với sticky='w' để căn trái. **tk.StringVar():** Tạo biến để lưu giá trị nhập vào từ ô nhập. **ttk.Entry:** Tạo ô nhập với chiều rộng là 12 ký tự.

textvariable=so_a: Liên kết ô nhập với biến so_a.

4. Tạo nhãn hiển thị kết quả

```
# Hiển thị kết quả ket_qua_label = ttk.Label(win, text="Kết quả:
")
ket_qua_label.grid(column=0, row=2, sticky='w', columnspan=2)
```

Tạo nhãn để hiển thị kết quả và đặt nó ở hàng 2, chiếm 2 cột (columnspan=2).

5. Hàm kiểm tra xem giá trị có phải là số hay không

```
# Hàm kiểm tra kí tự input có phải là số
không def is_number(gia_tri): try:

float(gia_tri)
return True
except
ValueError:
return False
```

Hàm này nhận vào một chuỗi (gia_tri) và cố gắng chuyển đổi nó thành số thực (float).

Nếu thành công, trả về True; nếu không, trả về False. Điều này giúp đảm bảo người dùng chỉ nhập giá trị số.

6. Các hàm phép tính và button

```
# Các hàm phép tính  # Phép cộng def cong():  if
(is_number(so_a.get()) and is_number(so_b.get())):

ket_qua_label.configure(text="Kết quả: " + str((float(so_a.get())
float(so_b.get())))) else:

                                                                                +



messagebox.showerror("Lỗi nhập liệu", "Vui lòng chỉ nhập giá trị
số")   button_cong = ttk.Button(win, text="+", width=3, command=cong)
button_cong.grid(column=2, row=0)


# Phép trừ def tru(): if (is_number(so_a.get()) and
is_number(so_b.get())):  ket_qua_label.configure(text="Kết quả: "
+ str((float(so_a.get())


                                                                                -
```

```
float(so_b.get()))))) else:
messagebox.showerror("Lỗi nhập liệu", "Vui lòng chỉ nhập giá trị
số")   button_tru = ttk.Button(win, text="-", width=3, command=tru)
button_tru.grid(column=3, row=0)


# Phép nhân def nhan(): if (is_number(so_a.get()) and
is_number(so_b.get())):  ket_qua_label.configure(text="Kết quả: "
+ str((float(so_a.get())
                                                                    *


float(so_b.get()))))) else:
messagebox.showerror("Lỗi nhập liệu", "Vui lòng chỉ nhập giá trị
số")   button_nhan = ttk.Button(win, text="*", width=3, command=nhan)
button_nhan.grid(column=2, row=1)


# Phép chia def chia(): if (is_number(so_a.get())
and is_number(so_b.get())):
ket_qua_label.configure(text="Kết quả: " + str((float(so_a.get())
float(so_b.get()))))) else:
messagebox.showerror("Lỗi nhập liệu", "Vui lòng chỉ nhập giá trị
số")   button_chia = ttk.Button(win, text="/", width=3, command=chia)
button_chia.grid(column=3, row=1)




                                                                    /
```

Mỗi phép toán đều có các nút bấm cho phép cộng, trừ, nhân, chia, và liên kết chúng với các hàm tương ứng đã định nghĩa và một hàm riêng, kiểm tra giá trị đầu vào và hiển thị kết quả.

Lấy giá trị từ so_a và so_b, kiểm tra xem chúng có phải là số không. Nếu đúng, tính toán cộng trừ nhân chai và cập nhật nhãn kết quả; nếu không, hiển thị thông báo lỗi.

7. Cấu hình cột để tránh ô nhập bị đẩy

```
# Ghi chú để tránh ô nhập bị đẩy win.grid_columnconfigure(0, weight=1)
win.grid_columnconfigure(1, weight=1) win.grid_columnconfigure(2,
weight=0)
win.grid_columnconfigure(3, weight=0)
```

Cấu hình cột trong lưới:

- Cột 0 và 1 (có chứa nhãn và ô nhập) được thiết lập với weight=1, cho phép chúng tự động chiếm không gian có sẵn.
- Cột 2 và 3 (chứa nút bấm) được thiết lập với weight=0, giúp giữ cho chúng ở kích thước cố định mà không ảnh hưởng đến các cột khác.

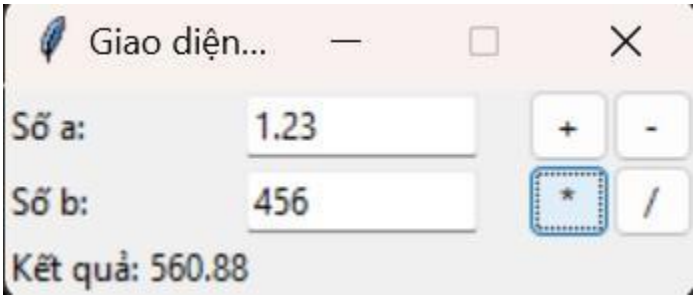
8. Chạy ứng dụng

```
win.mainloop()
```

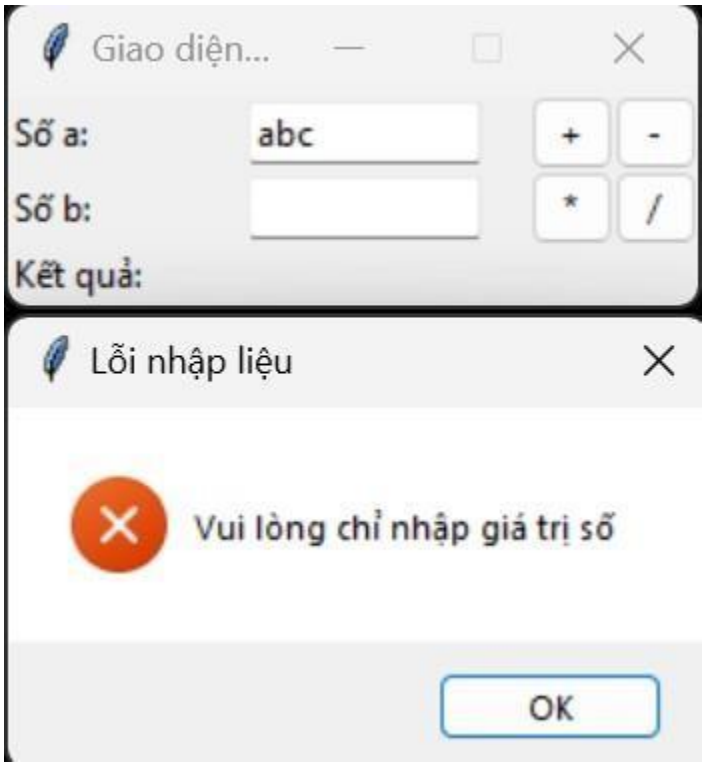
mainloop(): Bắt đầu vòng lặp chính của ứng dụng, cho phép người dùng tương tác với giao diện cho đến khi đóng cửa sổ.

Giao diện

Tính toán bình thường:



Nhập sai định dạng:



Chức năng

Cộng trừ nhân chia như một chiếc máy tính bình thường, có thể tính toán các giá trị thập phân và sẽ báo lỗi nếu nhập input sai định dạng

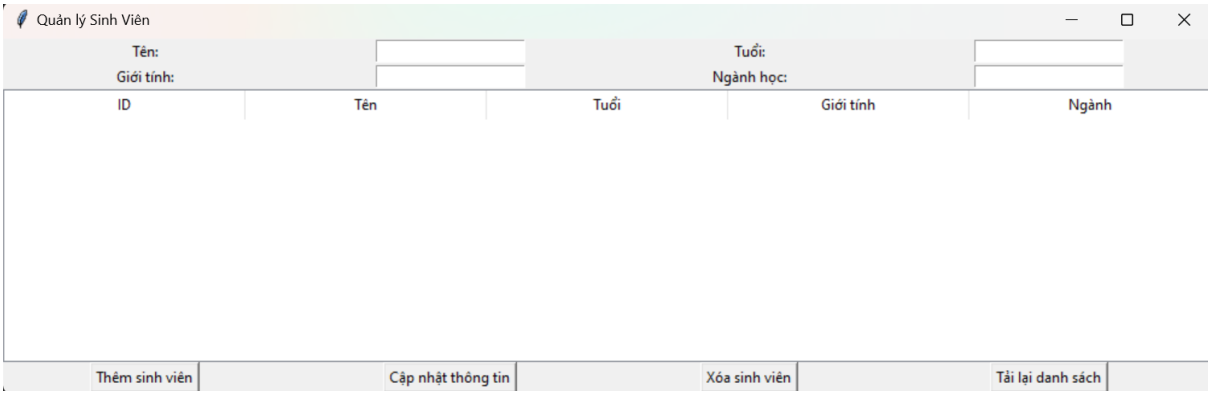
```python
import tkinter as tk
from tkinter import ttk
import psycopg2

cua_so = tk.Tk()
cua_so.title("Quản lý Sinh Viên")

tk.Label(cua_so, text="Tên:").grid(row=0, column=0)
nhap_ten = tk.Entry(cua_so)
nhap_ten.grid(row=0, column=1)

tk.Label(cua_so, text="Tuổi:").grid(row=0, column=2)
nhap_tuoi = tk.Entry(cua_so)
nhap_tuoi.grid(row=0, column=3)

tk.Label(cua_so, text="Giới tính:").grid(row=1, column=0)
nhap_gioi_tinh = tk.Entry(cua_so)
nhap_gioi_tinh.grid(row=1, column=1)

tk.Label(cua_so, text="Ngành học:").grid(row=1, column=2)
nhap_nganh_hoc = tk.Entry(cua_so)
nhap_nganh_hoc.grid(row=1, column=3)

cot = ('ID', 'Tên', 'Tuổi', 'Giới tính', 'Ngành')
bang = ttk.Treeview(cua_so, columns=cot, show='headings')

for cot_ten in cot:
    bang.heading(cot_ten, text=cot_ten)

bang.grid(row=2, column=0, columnspan=4)

nut_them = tk.Button(cua_so, text="Thêm sinh viên")
nut_them.grid(row=3, column=0)

nut_cap_nhat = tk.Button(cua_so, text="Cập nhật thông tin")
nut_cap_nhat.grid(row=3, column=1)
```

```python
nut_xoa = tk.Button(cua_so, text="Xóa sinh viên")
nut_xoa.grid(row=3, column=2)

nut_tai_lai = tk.Button(cua_so, text="Tải lại danh sách")
nut_tai_lai.grid(row=3, column=3)

def ket_noi_csdl():
    try:
        ket_noi = psycopg2.connect(
            dbname="postgres",
            user="postgres",
            password="1968",
            host="localhost",
            port="5432"
        )
        return ket_noi
    except Exception as e:
        print(f"Không thể kết nối cơ sở dữ liệu: {e}")
        return None

def them_sinh_vien():
    ket_noi = ket_noi_csdl()
    con_tro = ket_noi.cursor()

    ten = nhap_ten.get()
    tuoi = int(nhap_tuoi.get())
    gioi_tinh = nhap_gioi_tinh.get()
    nganh = nhap_nganh_hoc.get()

    cau_lenh = "INSERT INTO students (name, age, gender, major) VALUES (%s, %s, %s, %s)"
    con_tro.execute(cau_lenh, (ten, tuoi, gioi_tinh, nganh))

    ket_noi.commit()
    ket_noi.close()

nut_them.config(command=them_sinh_vien)
```

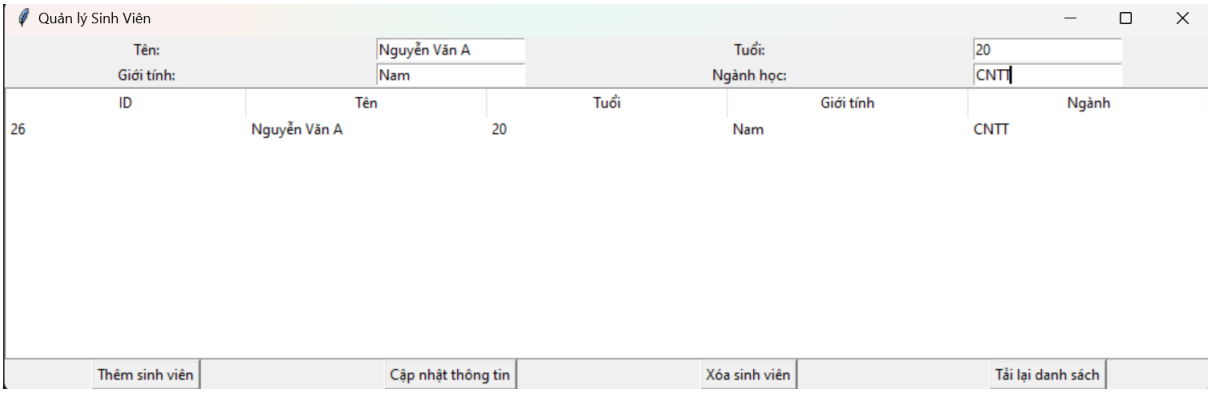
```python
def cap_nhat_sinh_vien():
    ket_noi = ket_noi_csdl()
    con_tro = ket_noi.cursor()

    sinh_vien_chon = bang.selection()[0]
    sinh_vien_id = bang.item(sinh_vien_chon, 'values')[0]

    ten = nhap_ten.get()
    tuoi = int(nhap_tuoi.get())
    gioi_tinh = nhap_gioi_tinh.get()
    nganh = nhap_nganh_hoc.get()

    cau_lenh = "UPDATE students SET name=%s, age=%s, gender=%s, major=%s WHERE id=%s"
    con_tro.execute(cau_lenh, (ten, tuoi, gioi_tinh, nganh, sinh_vien_id))

    ket_noi.commit()
    ket_noi.close()

nut_cap_nhat.config(command=cap_nhat_sinh_vien)

def xoa_sinh_vien():
    ket_noi = ket_noi_csdl()
    con_tro = ket_noi.cursor()

    sinh_vien_chon = bang.selection()[0]
    sinh_vien_id = bang.item(sinh_vien_chon, 'values')[0]

    cau_lenh = "DELETE FROM students WHERE id=%s"
    con_tro.execute(cau_lenh, (sinh_vien_id,))

    ket_noi.commit()
    ket_noi.close()

nut_xoa.config(command=xoa_sinh_vien)
```

```python
def tai_lai_danh_sach():
    for item in bang.get_children():
        bang.delete(item)

    ket_noi = ket_noi_csdl()
    con_tro = ket_noi.cursor()

    con_tro.execute("SELECT * FROM students")
    hang = con_tro.fetchall()

    for du_lieu in hang:
        bang.insert('', tk.END, values=du_lieu)

    ket_noi.close()

nut_tai_lai.config(command=tai_lai_danh_sach)

cua_so.mainloop()
```

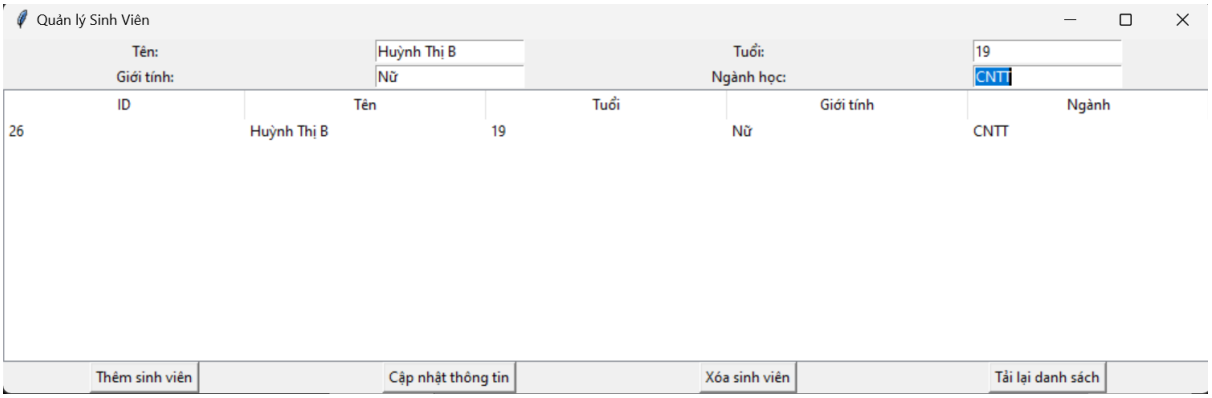
Kết quả:



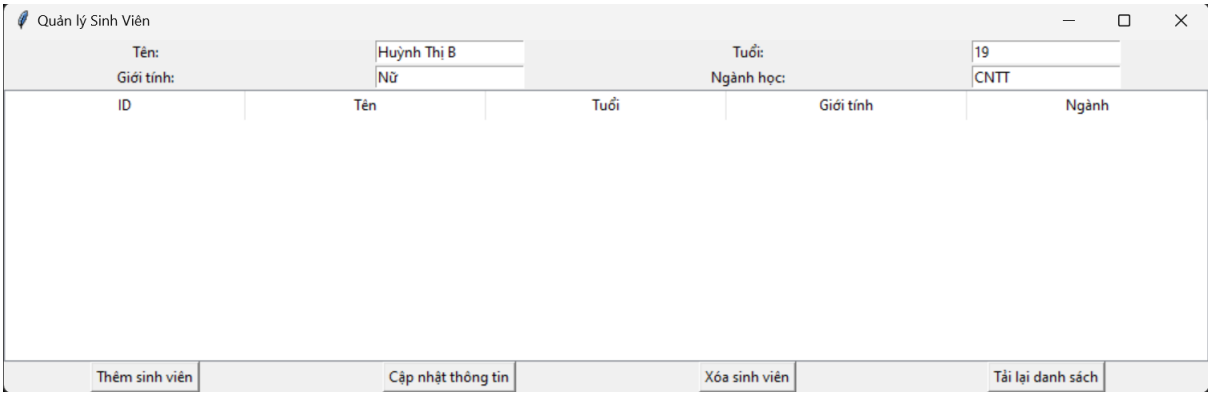
Thêm sinh viên:



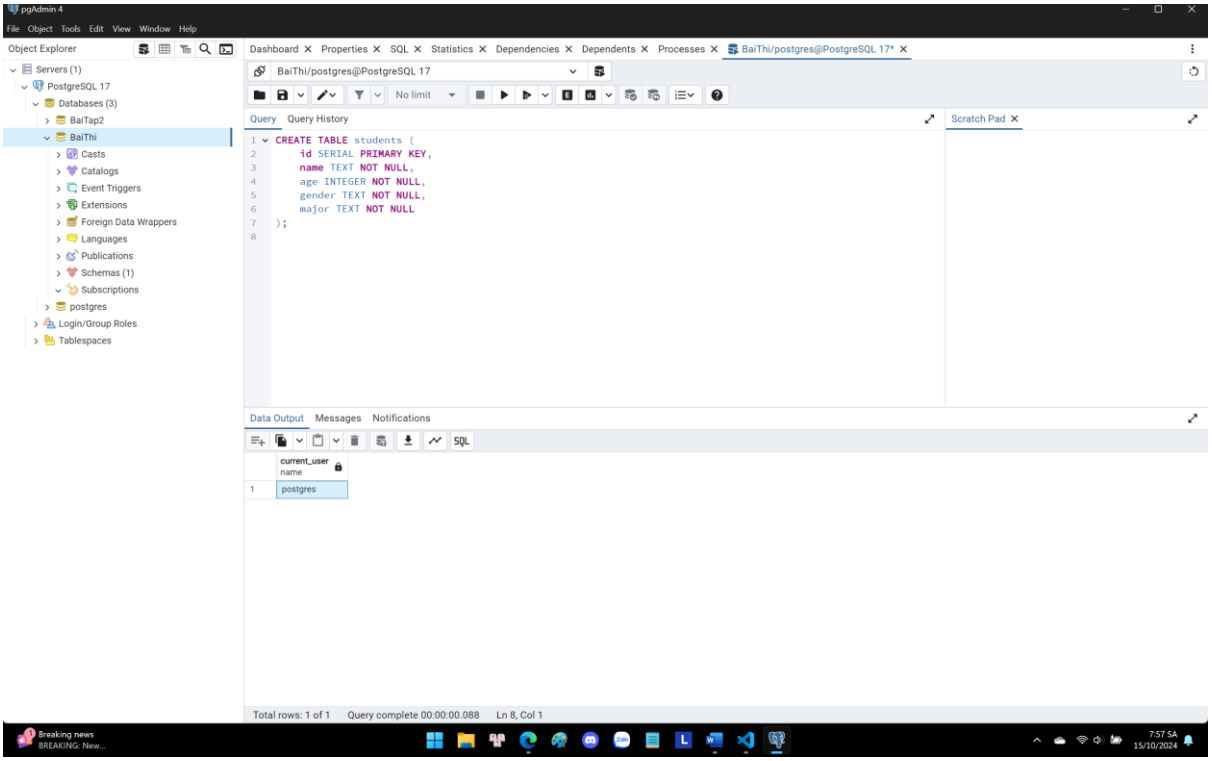
Cập nhập thông tin:



Xóa sinh viên:



Code PostgreSQL


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
Servers (1)
PostgreSQL 17
Databases (3)
BaiTap2
BaiThi
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
Subscriptions
postgres
Login/Group Roles
Tablespaces
Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
BaiThi/postgres@PostgreSQL 17*
BaiThi/postgres@PostgreSQL 17
No limit
Query
Query History
Scratch Pad
CREATE TABLE students (
id SERIAL PRIMARY KEY,
name TEXT NOT NULL,
age INTEGER NOT NULL,
gender TEXT NOT NULL,
major TEXT NOT NULL
);
Data Output
Messages
Notifications
current_user
name
postgres
Total rows: 1 of 1
Query complete 00:00:00.088
Ln 8, Col 1
Breaking news
BREAKING: New...
7:57 SA
15/10/2024

Templates:

Cart.html

```
<!DOCTYPE html>
<html lang="vi">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Giỏ Hàng</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='cart_style.css') }}">
    <script>
        function calculateTotal() {
            let total = 0;
            const cartItems = document.querySelectorAll('.cart-item');

            cartItems.forEach(item => {
                const priceText = item.querySelector('.item-price').innerText.replace(' VNĐ', '').replace(/\./g, '').trim();
                const price = parseFloat(priceText);
                const quantity = parseInt(item.querySelector('.quantity-input').value);
                const itemTotal = price * quantity;
                item.querySelector('.item-total').innerText = itemTotal.toLocaleString('vi-VN') + ' VNĐ';
                total += itemTotal;
            });

            document.getElementById('total-amount').innerText = total.toLocaleString('vi-VN') + ' VNĐ';
        }

        function confirmDelete() {
            return confirm("Bạn có chắc chắn muốn xóa sản phẩm này khỏi giỏ hàng?");
        }
    </script>
</head>
<body>
    <div class="container">
        <h2>Giỏ Hàng</h2>
        <form method="POST" onsubmit="calculateTotal()">
            <div class="cart-items">
                {% for item in cart %}
                <div class="cart-item">
                    <h3>{{ item.name }}</h3> <!-- Hiển thị tên sản phẩm -->
```

```
                    <span class="item-price">Giá: {{ item.price | number_format(0, ',', '.') }} VNĐ</span>
                    <input type="number" class="quantity-input" name="quantity_{{ item.id }}" value="{{ item.get('quantity', 1) }}" min="
                    <span class="item-total">{{ (item.price * item.get('quantity', 1)) | number_format(0, ',', '.') }} VNĐ</span>
                    <a href="{{ url_for('remove_from_cart', product_id=item.id) }}" class="remove-item" onclick="return confirmDelete();"
                </div>
                {% endfor %}
            </div>
            <div class="total-container">
                Tổng cộng: <span id="total-amount">{{ total | number_format(0, ',', '.') }} VNĐ</span>
            </div>
            <div class="button-container">
                <button type="submit" onclick="calculateTotal()">Cập nhật giỏ hàng</button>
                <a href="{{ url_for('checkout') }}" class="back-link">Thanh toán</a>
            </div>
        </form>
        <a href="/" class="back-link">Quay lại trang sản phẩm</a>
    </div>
</body>
</html>
```

Checkout.html

```html
<!DOCTYPE html>
<html lang="vi">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Thanh Toán</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='checkout_style.css') }}">
</head>
<body>
    <div class="container">
        <h2>Thông Tin Thanh Toán</h2>
        <form method="POST" action="{{ url_for('checkout') }}">
            <!-- Thông tin cá nhân -->
            <label for="name">Tên:</label>
            <input type="text" id="name" name="name" required>

            <label for="phone">Điện thoại:</label>
            <input type="text" id="phone" name="phone" required>

            <label for="email">Email:</label>
            <input type="email" id="email" name="email" required>

            <!-- Thông tin địa chỉ -->
            <label for="address">Địa chỉ:</label>
            <input type="text" id="address" name="address" required>

            <label for="district">Quận/Huyện:</label>
            <input type="text" id="district" name="district" required>

            <label for="ward">Xã/Phường:</label>
            <input type="text" id="ward" name="ward" required>

            <label for="city">Tỉnh/Thành phố:</label>
            <input type="text" id="city" name="city" required>

            <!-- Phương thức thanh toán -->
            <label for="payment_method">Phương thức thanh toán:</label>
```

```
        <select id="payment_method" name="payment_method" required>
            <option value="credit_card">Thẻ tín dụng</option>
            <option value="bank_transfer">Chuyển khoản</option>
            <option value="e_wallet">Ví điện tử</option>
        </select>

        <!-- Mã giảm giá -->
        <label for="coupon">Mã giảm giá:</label>
        <input type="text" id="coupon" name="coupon" placeholder="Nhập mã giảm giá (nếu có)">

        <!-- Ghi chú -->
        <label for="notes">Ghi chú:</label>
        <textarea id="notes" name="notes" placeholder="Thêm ghi chú cho đơn hàng"></textarea>

        <!-- Thời gian giao hàng -->
        <label for="delivery_time">Thời gian giao hàng mong muốn:</label>
        <input type="text" id="delivery_time" name="delivery_time" placeholder="VD: Sáng, Chiều, Tối">

        <!-- Tùy chọn lưu thông tin -->
        <label>
            <input type="checkbox" name="save_info"> Lưu thông tin cho lần mua tiếp theo
        </label>

        <!-- Chi tiết đơn hàng -->
        <div class="order-summary">
            <h3>Chi tiết đơn hàng</h3>
            <ul>
                <li>Sản phẩm A - 1,000,000đ</li>
                <li>Sản phẩm B - 500,000đ</li>
                <li>Phí vận chuyển: 30,000đ</li>
                <li><strong>Tổng cộng: 1,530,000đ</strong></li>
            </ul>
        </div>
```

```
            <!-- Nút xác nhận -->
            <button type="submit">Xác nhận thanh toán</button>
        </form>
    </div>
</body>
</html>
```

Payment.html

```html
<!DOCTYPE html>
<html lang="vi">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Chọn Phương Thức Thanh Toán</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='checkout_style.css') }}">
    <style>
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            background-color: #f8f9fa; /* Màu nền nhẹ nhàng */
            color: #343a40; /* Màu chữ tối */
        }

        .container {
            max-width: 800px; /* Chiều rộng tối đa */
            margin: auto; /* Căn giữa */
            padding: 20px;
            background: white; /* Nền trắng */
            border-radius: 10px; /* Bo góc */
            box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1); /* Đổ bóng */
        }

        h2, h3 {
            text-align: center; /* Căn giữa tiêu đề */
            color: #28a745; /* Màu xanh lá cây cho tiêu đề */
        }

        .cart-items {
            margin: 20px 0;
        }

        .cart-item {
            border-bottom: 1px solid #ced4da; /* Viền dưới */
            padding: 10px 0; /* Khoảng cách trên/dưới */
        }
```

```css
.item-price, .item-quantity, .item-total {
    display: block; /* Hiển thị dưới dạng khối */
}

.total-container {
    font-weight: bold; /* Chữ đậm cho tổng tiền */
    text-align: right; /* Căn phải */
    margin-top: 20px; /* Khoảng cách trên tổng tiền */
}

.payment-methods {
    margin-top: 20px; /* Khoảng cách trên phương thức thanh toán */
}

.payment-option {
    background-color: #f8f9fa; /* Nền nhẹ cho các lựa chọn */
    border: 1px solid #ced4da; /* Viền xám */
    border-radius: 5px; /* Bo góc */
    padding: 15px; /* Khoảng cách bên trong */
    margin: 10px 0; /* Khoảng cách giữa các lựa chọn */
    cursor: pointer; /* Con trỏ chuột khi hover */
    transition: background-color 0.2s; /* Hiệu ứng chuyển đổi */
}

.payment-option:hover {
    background-color: #e2e6ea; /* Màu nền khi hover */
}

.selected {
    background-color: #28a745; /* Màu nền cho phần đã chọn */
    color: white; /* Màu chữ trắng */
}
```

```css
.selected h4 {
    color: white; /* Đảm bảo tiêu đề cũng trắng */
}

.bank-info {
    background-color: rgba(255, 255, 255, 0.9); /* Nền trắng trong suốt cho thông tin ngân hàng */
    color: #343a40; /* Màu chữ tối cho thông tin ngân hàng */
    border-radius: 5px; /* Bo góc */
    padding: 10px; /* Khoảng cách bên trong */
    margin-top: 10px; /* Khoảng cách trên */
}

.button-container {
    display: flex; /* Sử dụng flexbox để căn giữa */
    justify-content: space-between; /* Tách ra hai bên */
    margin-top: 20px; /* Khoảng cách trên nút */
}

button {
    background-color: #28a745; /* Màu nền cho nút */
    color: white; /* Màu chữ trắng */
    padding: 12px 20px; /* Khoảng cách bên trong nút */
    border: none; /* Không có viền */
    border-radius: 5px; /* Bo góc */
    cursor: pointer; /* Con trỏ chuột khi hover */
    transition: background-color 0.3s, transform 0.2s; /* Hiệu ứng chuyển đổi */
    font-size: 1em; /* Kích thước chữ */
    width: 48%; /* Chiếm 48% chiều rộng */
}

button:hover {
    background-color: #218838; /* Màu nền khi hover */
    transform: translateY(-2px); /* Hiệu ứng nhấc lên khi hover */
}
```

```css
/* Popup Modal */
.modal {
    display: none; /* Ẩn ban đầu */
    position: fixed; /* Vị trí cố định */
    z-index: 1; /* Đặt lên trên cùng */
    left: 0;
    top: 0;
    width: 100%; /* Chiếm toàn bộ chiều rộng */
    height: 100%; /* Chiếm toàn bộ chiều cao */
    overflow: auto; /* Cuộn nếu cần */
    background-color: rgba(0, 0, 0, 0.5); /* Nền tối với độ trong suốt */
}

.modal-content {
    background-color: #fefefe; /* Nền trắng */
    margin: 15% auto; /* Căn giữa */
    padding: 20px; /* Khoảng cách bên trong */
    border: 1px solid #888; /* Viền xám */
    width: 80%; /* Chiếm 80% chiều rộng */
    border-radius: 10px; /* Bo góc */
}

.close {
    color: #aaa; /* Màu xám cho nút đóng */
    float: right; /* Căn phải */
    font-size: 28px; /* Kích thước chữ */
    font-weight: bold; /* Chữ đậm */
}

.close:hover,
.close:focus {
    color: black; /* Màu đen khi hover hoặc focus */
    text-decoration: none; /* Không gạch chân */
    cursor: pointer; /* Con trỏ chuột khi hover */
```

```
    }
</style>
</head>
<body>
    <div class="container">
        <h2>Thông Tin Giỏ Hàng</h2>
        {% if cart %}
            <div class="cart-items">
                {% for item in cart %}
                <div class="cart-item">
                    <h3>{{ item.name }}</h3>
                    <span class="item-price">Giá: {{ item.price | number_format(0, ',', '.') }} VNĐ</span>
                    <span class="item-quantity">Số lượng: {{ item.quantity }}</span>
                    <span class="item-total">Tổng: {{ (item.price * item.quantity) | number_format(0, ',', '.') }} VNĐ</span>
                </div>
                {% endfor %}
            </div>
            <div class="total-container">
                Tổng cộng: <span id="total-amount">{{ total | number_format(0, ',', '.') }} VNĐ</span>
            </div>
        {% else %}
            <p>Giỏ hàng của bạn hiện đang trống.</p>
        {% endif %}

        <h3>Chọn Phương Thức Thanh Toán</h3>
        <div class="payment-methods">
            <div class="payment-option" onclick="selectPaymentMethod(this)">
                <h4>Thanh toán chuyển khoản</h4>
                <div class="bank-info" style="display: none;">
                    <p><strong>Chủ tài khoản:</strong> Trần Ngọc Long</p>
                    <p><strong>Số tài khoản:</strong> 21565507</p>
                    <p><strong>Ngân hàng:</strong> ABC</p>
```

```
                    <p><strong>Địa chỉ:</strong> UBND xã Gia Tân 3, 10T Ấp Phúc Nhạc, Gia Tân 3, Thống Nhất, Đồng Nai</p>
                </div>
            </div>
            <div class="payment-option" onclick="selectPaymentMethod(this)">
                <h4>Thanh toán khi nhận hàng (COD)</h4>
            </div>
        </div>

        <div class="button-container">
            <button onclick="window.history.back()">Quay lại</button>
            <button onclick="confirmOrder()">Xác nhận</button>
        </div>
    </div>

    <!-- Popup Modal -->
    <div id="orderModal" class="modal">
        <div class="modal-content">
            <span class="close" onclick="closeModal()">&times;</span>
            <h2>Đặt hàng thành công!</h2>
            <p>Cảm ơn bạn đã đặt hàng. Chúng tôi sẽ liên hệ với bạn sớm nhất có thể.</p>
        </div>
    </div>

    <script>
        function toggleBankInfo(element) {
            const bankInfo = element.querySelector('.bank-info');
            bankInfo.style.display = bankInfo.style.display === 'none' ? 'block' : 'none';
        }

        function selectPaymentMethod(element) {
            const options = document.querySelectorAll('.payment-option');
            options.forEach(option => {
```

```
            options.forEach(option => {
                option.classList.remove('selected'); // Xóa lớp đã chọn cho tất cả
            });
            element.classList.add('selected'); // Thêm lớp đã chọn cho phần đã chọn
            toggleBankInfo(element); // Hiển thị thông tin ngân hàng nếu phương thức chuyển khoản được chọn
        }

        function confirmOrder() {
            // Hiện modal thông báo đặt hàng thành công
            document.getElementById('orderModal').style.display = 'block';
        }

        function closeModal() {
            document.getElementById('orderModal').style.display = 'none';
            // Chuyển hướng về trang chủ sau 10 giây
            setTimeout(() => {
                window.location.href = "{{ url_for('index') }}";
            }, 2000); // Thay đổi thời gian nếu cần
        }

        // Đóng modal khi nhấn bên ngoài
        window.onclick = function(event) {
            const modal = document.getElementById('orderModal');
            if (event.target == modal) {
                closeModal();
            }
        }
    </script>
</body>
</html>
```

```python
from flask import Flask, render_template, request, redirect, url_for, session
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime
import locale

# Khởi tạo ứng dụng Flask
app = Flask(__name__,
            static_folder='static',
            template_folder='templates')

app.secret_key = 'your_secret_key'  # Bảo mật session

# Cấu hình kết nối đến PostgreSQL
app.config['SQLALCHEMY_DATABASE_URI'] = 'postgresql://postgres:ngoclong@localhost:5432/postgres'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)

# Đặt định dạng ngôn ngữ cho VN
locale.setlocale(locale.LC_ALL, 'vi_VN.UTF-8')

# Định nghĩa model cho sản phẩm
class Product(db.Model):
    __tablename__ = 'products'
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    price = db.Column(db.Integer, nullable=False)

# Định nghĩa model cho đơn hàng
class Order(db.Model):
    __tablename__ = 'orders'
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    phone = db.Column(db.String(20), nullable=False)
    email = db.Column(db.String(100))
    address = db.Column(db.String(200), nullable=False)
    district = db.Column(db.String(100))
    ward = db.Column(db.String(100))
```

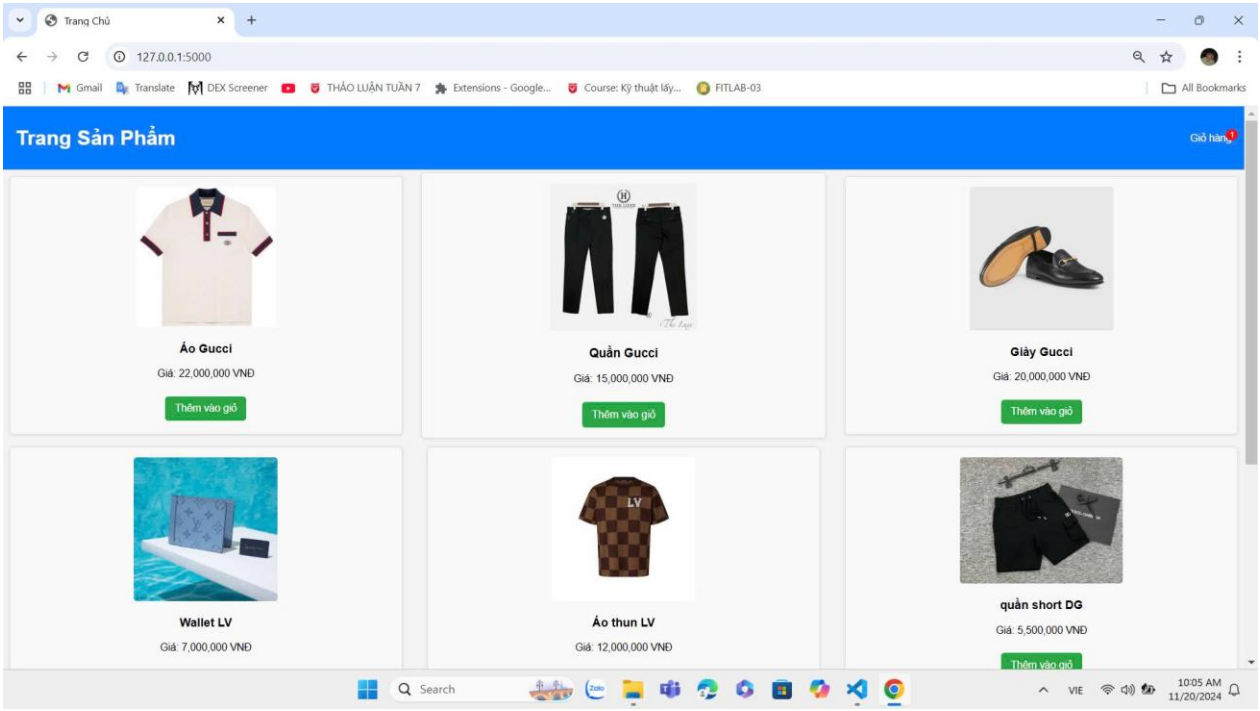
```
    city = db.Column(db.String(100))
    total_amount = db.Column(db.Integer, nullable=False)
    created_at = db.Column(db.DateTime, default=datetime.utcnow)

# Định nghĩa model cho mục đơn hàng
class OrderItem(db.Model):
    __tablename__ = 'order_items'
    id = db.Column(db.Integer, primary_key=True)
    order_id = db.Column(db.Integer, db.ForeignKey('orders.id'), nullable=False)
    product_id = db.Column(db.Integer, db.ForeignKey('products.id'), nullable=False)
    quantity = db.Column(db.Integer, nullable=False)
    price = db.Column(db.Integer, nullable=False)

# Hàm định dạng số cho template Jinja2
def number_format(value, decimal_places=0, decimal_point=',', thousands_sep='.'):
    if value is None:
        return ''
    formatted_value = f"{value:,.{decimal_places}f}".replace(',', 'X').replace('.', decimal_point).replace('X', thousands_sep)
    return formatted_value

# Đăng ký hàm định dạng với Jinja2
app.jinja_env.filters['number_format'] = number_format

# Route trang chủ
@app.route('/')
def index():
    products = Product.query.all()
    cart_items = session.get('cart', [])
    cart_count = sum(item.get('quantity', 0) for item in cart_items)
    return render_template('index.html', products=products, cart_count=cart_count)

# Thêm sản phẩm vào giỏ hàng
@app.route('/add_to_cart/<int:product_id>')
def add_to_cart(product_id):
    product = Product.query.get_or_404(product_id)
```

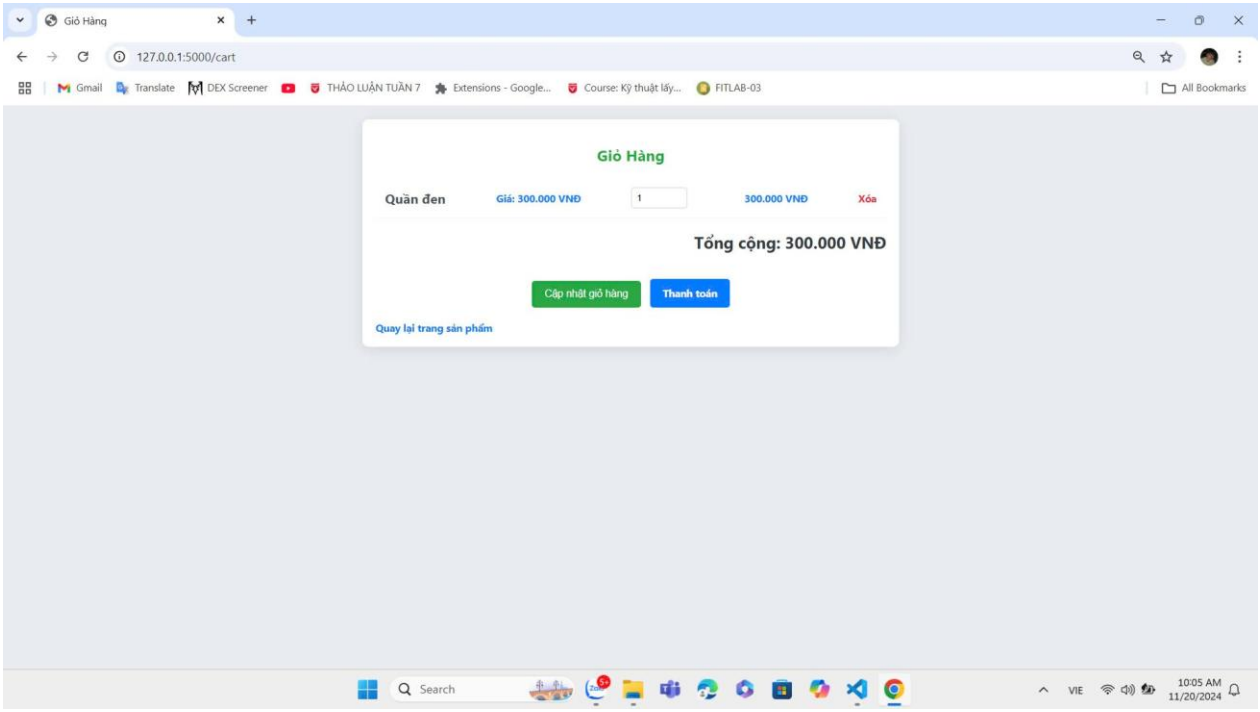
```
    if 'cart' not in session:
        session['cart'] = []
    existing_product = next((item for item in session['cart'] if item['id'] == product_id), None)
    if existing_product:
        existing_product['quantity'] += 1
    else:
        session['cart'].append({
            'id': product.id,
            'name': product.name,
            'price': product.price,
            'quantity': 1
        })
    session.modified = True
    return redirect(url_for('index'))

# Route giỏ hàng
@app.route('/cart', methods=['GET', 'POST'])
def cart():
    if request.method == 'POST':
        for item in session.get('cart', []):
            quantity = request.form.get(f'quantity_{item["id"]}', 1)
            item['quantity'] = int(quantity) if quantity.isdigit() and int(quantity) > 0 else 1
    cart_items = session.get('cart', [])
    total = sum(item['price'] * item.get('quantity', 1) for item in cart_items)
    return render_template('cart.html', cart=cart_items, total=total)

# Xóa sản phẩm khỏi giỏ hàng
@app.route('/remove_from_cart/<int:product_id>')
def remove_from_cart(product_id):
    if 'cart' in session:
        session['cart'] = [item for item in session['cart'] if item['id'] != product_id]
        session.modified = True
    return redirect(url_for('cart'))

# Route thanh toán
```

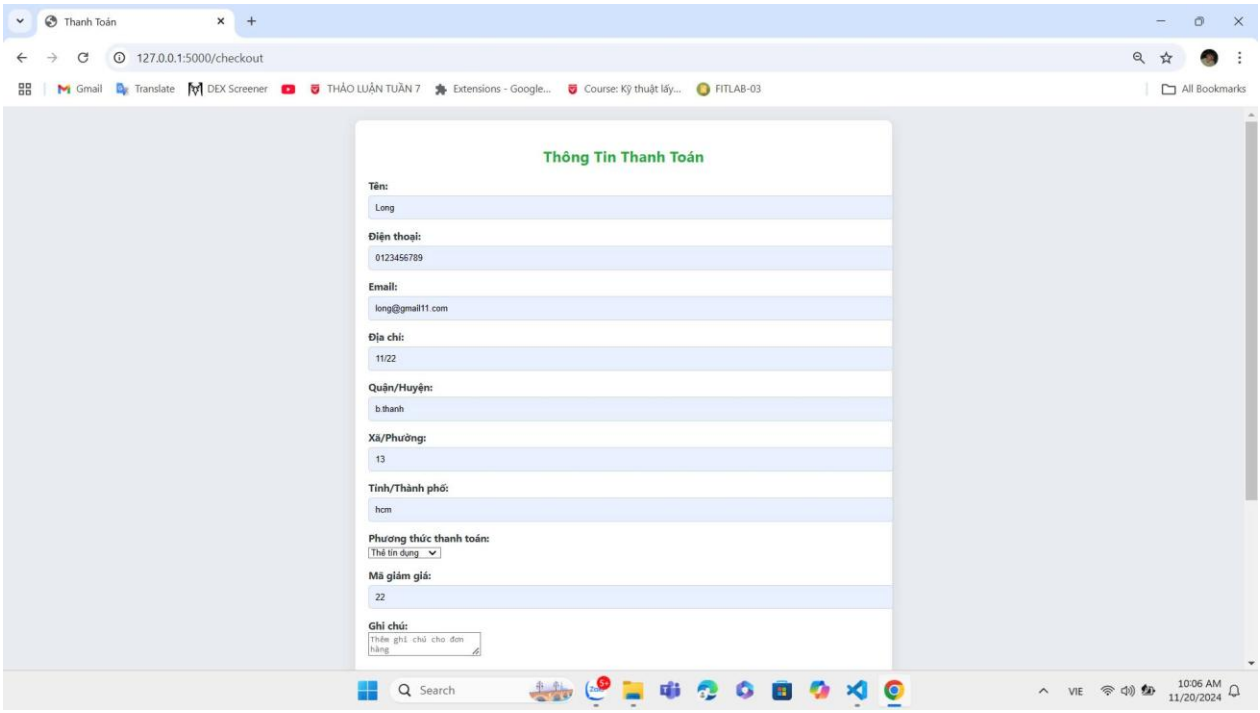
```python
@app.route('/checkout', methods=['GET', 'POST'])
def checkout():
    if request.method == 'POST':
        cart_items = session.get('cart', [])
        total = sum(item['price'] * item.get('quantity', 1) for item in cart_items)
        order = Order(
            name=request.form.get('name'),
            phone=request.form.get('phone'),
            email=request.form.get('email'),
            address=request.form.get('address'),
            district=request.form.get('district'),
            ward=request.form.get('ward'),
            city=request.form.get('city'),
            total_amount=total
        )
        db.session.add(order)
        db.session.flush()
        for item in cart_items:
            order_item = OrderItem(
                order_id=order.id,
                product_id=item['id'],
                quantity=item['quantity'],
                price=item['price']
            )
            db.session.add(order_item)
        db.session.commit()
        session['cart'] = []
        return redirect(url_for('payment'))
    return render_template('checkout.html')

# Route trang thanh toán
@app.route('/payment')
def payment():
    cart_items = session.get('cart', [])
    total = sum(item['price'] * item.get('quantity', 1) for item in cart_items)
    return render_template('payment.html', cart=cart_items, total=total)
```

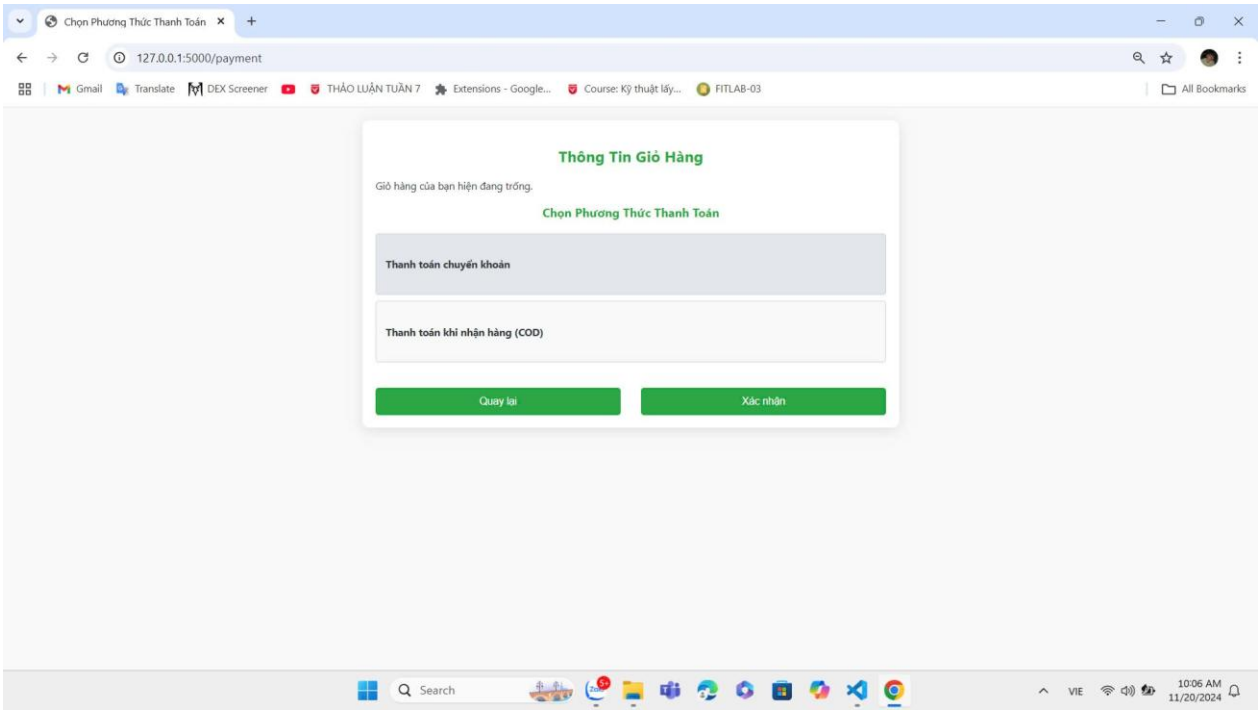
```python
# Khởi tạo database và thêm sản phẩm mẫu
def init_db():
    with app.app_context():
        db.create_all()
        if not Product.query.first():
            products = [
                Product(name="Áo Gucci", price=22000000),
                Product(name="Quần Gucci", price=15000000),
                Product(name="Giày Gucci", price=20000000),
                Product(name="Áo thun Gucci", price=12000000),
                Product(name="quần short Gucci", price=5500000),
                Product(name="wallet Gucci", price=7000000),
            ]
            for product in products:
                db.session.add(product)
            db.session.commit()
            print("Đã khởi tạo database và thêm dữ liệu mẫu!")
        else:
            print("Database đã có dữ liệu!")
# Chạy ứng dụng
if __name__ == '__main__':
    init_db()
    app.run(debug=True)
```

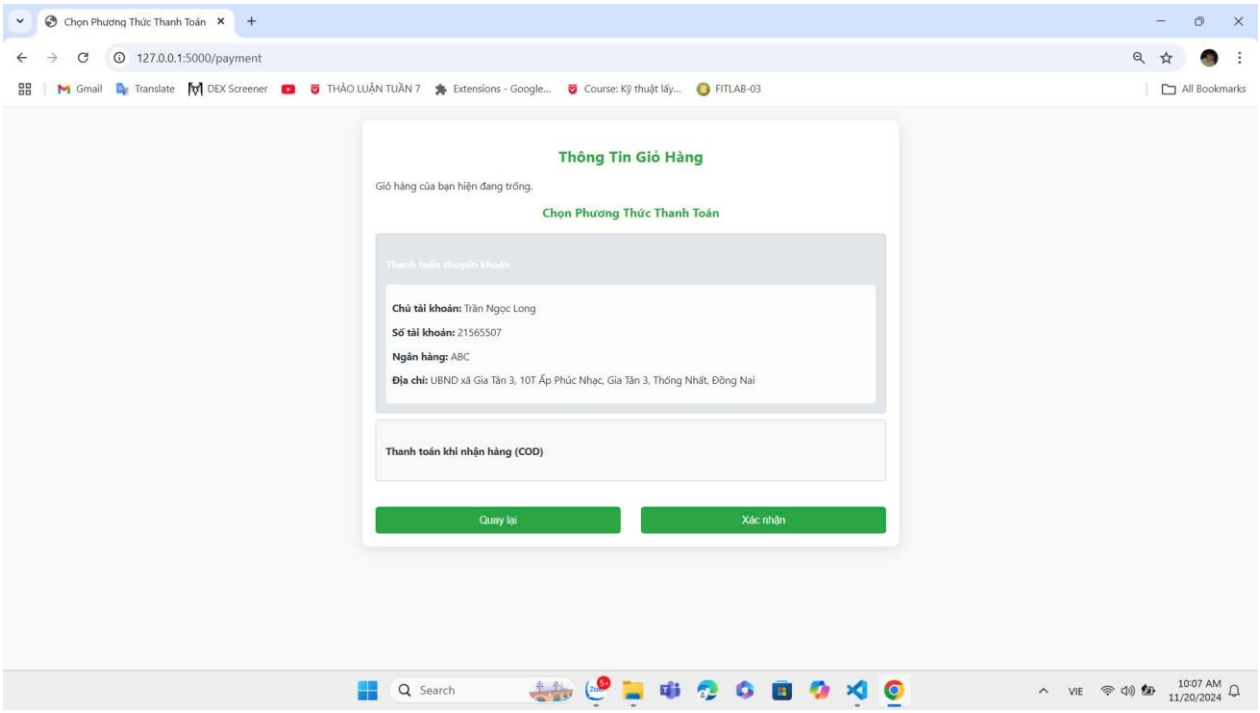
Kết quả

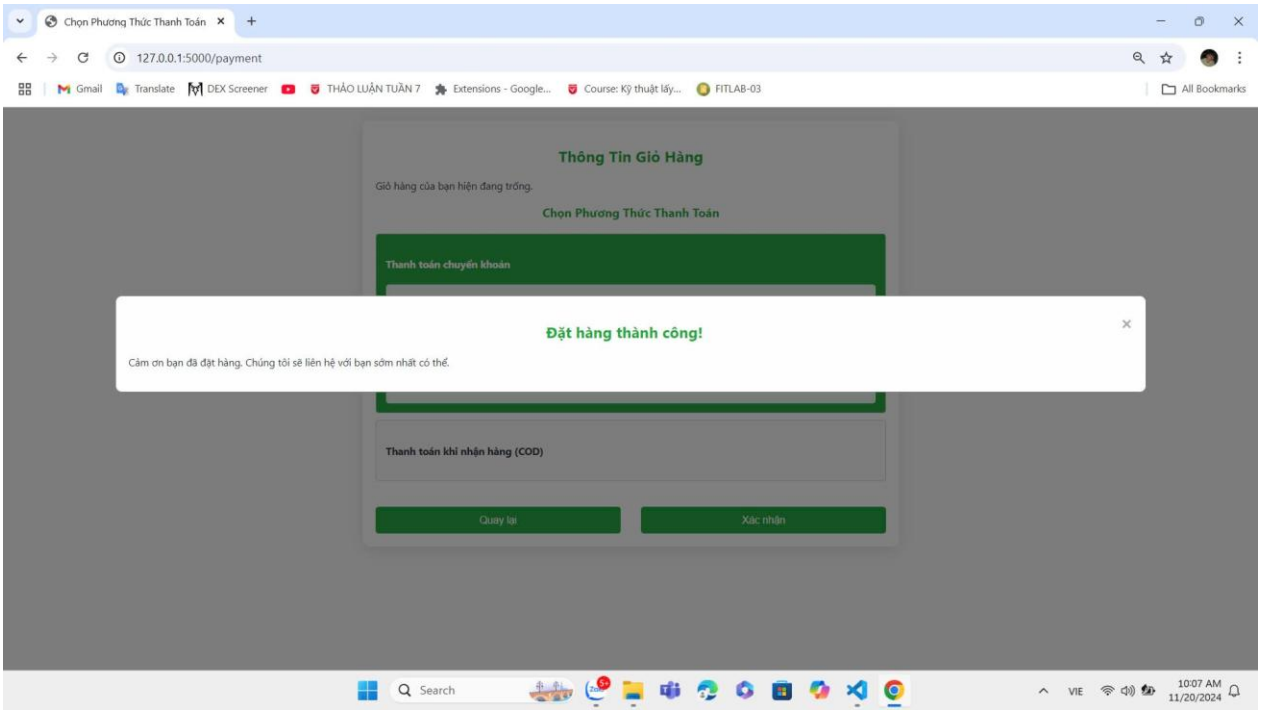

Trang Chủ
127.0.0.1:5000
Trang Sản Phẩm
Giỏ hàng
Áo Gucci
Giá: 22,000,000 VNĐ
Thêm vào giỏ
Quần Gucci
Giá: 15,000,000 VNĐ
Thêm vào giỏ
Giày Gucci
Giá: 20,000,000 VNĐ
Thêm vào giỏ
Wallet LV
Giá: 7,000,000 VNĐ
Áo thun LV
Giá: 12,000,000 VNĐ
quần short DG
Giá: 5,500,000 VNĐ
Thêm vào giỏ


Giỏ Hàng
127.0.0.1:5000/cart
Giỏ Hàng
Quần đen
Giá: 300.000 VNĐ
1
300.000 VNĐ
Xóa
Tổng cộng: 300.000 VNĐ
Cập nhật giỏ hàng
Thanh toán
Quay lại trang sản phẩm


Thanh Toán
127.0.0.1:5000/checkout
Gmail
Translate
DEX Screener
THẢO LUẬN TUẦN 7
Extensions - Google...
Course: Kỹ thuật lấy...
FITLAB-03
All Bookmarks
Thông Tin Thanh Toán
Tên:
Long
Điện thoại:
0123456789
Email:
long@gmail11.com
Địa chỉ:
11/22
Quận/Huyện:
b.thanh
Xã/Phường:
13
Tỉnh/Thành phố:
hcm
Phương thức thanh toán:
Thẻ tín dụng
Mã giảm giá:
22
Ghi chú:
Thêm ghi chú cho đơn hàng
Search
VIE
10:06 AM
11/20/2024


Chọn Phương Thức Thanh Toán
127.0.0.1:5000/payment
Gmail
Translate
DEX Screener
THẢO LUẬN TUẦN 7
Extensions - Google...
Course: Kỹ thuật lấy...
FITLAB-03
All Bookmarks
Thông Tin Giỏ Hàng
Giỏ hàng của bạn hiện đang trống.
Chọn Phương Thức Thanh Toán
Thanh toán chuyển khoản
Thanh toán khi nhận hàng (COD)
Quay lại
Xác nhận
Search
VIE
10:06 AM
11/20/2024


Chọn Phương Thức Thanh Toán
127.0.0.1:5000/payment
Thông Tin Giỏ Hàng
Giỏ hàng của bạn hiện đang trống.
Chọn Phương Thức Thanh Toán
Thanh toán chuyển khoản
Chủ tài khoản: Trần Ngọc Long
Số tài khoản: 21565507
Ngân hàng: ABC
Địa chỉ: UBND xã Gia Tân 3, 10T Ấp Phúc Nhạc, Gia Tân 3, Thống Nhất, Đồng Nai
Thanh toán khi nhận hàng (COD)
Quay lại
Xác nhận


Chọn Phương Thức Thanh Toán
127.0.0.1:5000/payment
Thông Tin Giỏ Hàng
Giỏ hàng của bạn hiện đang trống.
Chọn Phương Thức Thanh Toán
Thanh toán chuyển khoản
Đặt hàng thành công!
Cảm ơn bạn đã đặt hàng. Chúng tôi sẽ liên hệ với bạn sớm nhất có thể.
Thanh toán khi nhận hàng (COD)
Quay lại
Xác nhận