

Mes codes

Base.html

```
{% extends "base.html" %}

{% block content %}

<div class="auth-container" style="max-width: 500px; margin: 2rem auto; padding: 2rem; background: #161616; border-radius: 8px;">

    <h2 style="text-align: center; margin-bottom: 1.5rem;">Connexion</h2>

    {% if messages %}

        {% for message in messages %}

            <div class="alert" style="color: #e74c3c; margin-bottom: 1rem;">

                {{ message }}

            </div>

        {% endfor %}

    {% endif %}

    <form method="post" action="{% url 'login' %}{% if request.GET.next %}?next={{ request.GET.next }}{% endif %}" style="display: flex; flex-direction: column; gap: 1rem;">

        {% csrf_token %}

        {% if request.GET.next %}

            <input type="hidden" name="next" value="{{ request.GET.next }}">

        {% endif %}

        <div class="form-group">

            <label for="{{ form.username.id_for_label }}">Nom d'utilisateur</label>

            {{ form.username }}

        </div>

        <div class="form-group">

            <label for="{{ form.password.id_for_label }}">Mot de passe</label>

            {{ form.password }}

        </div>

        <button type="submit" class="btn" style="width: 100%; padding: 12px; background: #3498db; color: #000; border: none; border-radius: 5px; cursor: pointer; margin-top: 1rem;">

            Se connecter

        </button>

    </form>

</div>
```

```

</form>

<p style="text-align: center; margin-top: 1.5rem;">
    Pas encore de compte?
    <a href="{% url 'register' %}" style="color: #3498db; text-decoration: none;">S'inscrire</a>
</p>
</div>
{% endblock %}

```

## Categorie.html

```

{% extends "base.html" %}

{% block content %}

<section class="categories">
    <div class="container">
        <h2 class="section-title">Nos catégories</h2>

        <div class="category-grid">
            {% for category in categories %}
                <div class="category">
                    
                    <div class="category-content">
                        <h3 class="category-title">{{ category.name }}</h3>
                        <a href="{% url 'category_products' category.id %}" class="category-link">
                            Voir les produits
                            <svg width="18" height="12" viewBox="0 0 18 12" fill="none">
                                <path d="M17.5303 6.53033C17.8232 6.23744 17.8232 5.76256 17.5303 5.46967L12.7574
0.696699C12.4645 0.403806 11.9896 0.403806 11.6967 0.696699C11.4038 0.989592 11.4038 1.46447 11.6967
1.75736L15.9393 6L11.6967 10.2426C11.4038 10.5355 11.4038 11.0104 11.6967 11.3033C11.9896 11.5962 12.4645
11.5962 12.7574 11.3033L17.5303 6.53033ZM0 6.75H17V5.25H0V6.75Z" fill="#3498db"/>
                            </svg>
                        </a>
                    </div>
                </div>
            {% endfor %}
        </div>
    </div>
</section>
{% endblock %}

```

## Checkout.html

```
{% extends "base.html" %}

{% block content %}

<div class="checkout-container">

    <h1 class="page-title">Finaliser la commande</h1>

    <div class="checkout-content">
        <div class="cart-summary">
            <h2>Résumé de votre panier</h2>
            <div class="cart-items" id="checkout-items">
                <!-- Cart items will be loaded dynamically -->
            </div>
        </div>

        <div class="cart-totals">
            <div class="subtotal">
                <span>Sous-total:</span>
                <span id="subtotal">0 FCFA</span>
            </div>
            <div class="shipping">
                <span>Frais de livraison:</span>
                <span>2000 FCFA</span>
            </div>
            <div class="total">
                <span>Total:</span>
                <span id="total">2000 FCFA</span>
            </div>
        </div>
    </div>

    <div class="checkout-steps">
        <div class="step active" id="step-login">
            <h3>1. Compte</h3>
            <div class="step-content">
                {% if user.is_authenticated %}
                    <div class="user-info">
```

```

<p>Connecté en tant que: <strong>{{ user.get_full_name }}</strong></p>
<p>Email: {{ user.email }}</p>
</div>
<button class="btn btn-primary" onclick="showStep('step-shipping')">Continuer</button>
{% else %}
<div class="login-options">
<div class="tab-headers">
<div class="tab-header active" onclick="showTab('login')">Connexion</div>
<div class="tab-header" onclick="showTab('register')">Inscription</div>
</div>

<div class="tab-content active" id="login">
<form action="{% url 'login' %}" method="post" class="login-form">
{% csrf_token %}
<div class="form-group">
<label for="login-email">Email</label>
<input type="email" id="login-email" name="email" required>
</div>
<div class="form-group">
<label for="login-password">Mot de passe</label>
<input type="password" id="login-password" name="password" required>
</div>
<button type="submit" class="btn btn-primary">Se connecter</button>
</form>
<p class="mt-3">OU</p>
<button class="btn btn-secondary" onclick="proceedAsGuest()">Continuer en tant qu'invité</button>
</div>

<div class="tab-content" id="register">
<form action="{% url 'register' %}" method="post" class="register-form" enctype="multipart/form-data">
{% csrf_token %}
<div class="form-row">
<div class="form-group">
<label for="register-first-name">Prénom</label>
<input type="text" id="register-first-name" name="first_name" required>
</div>
<div class="form-group">

```

```

        <label for="register-last-name">Nom</label>
        <input type="text" id="register-last-name" name="last_name" required>
    </div>
</div>
<div class="form-group">
    <label for="register-phone">Numéro de téléphone</label>
    <input type="tel" id="register-phone" name="phone" required>
</div>
<div class="form-group">
    <label for="register-email">Email</label>
    <input type="email" id="register-email" name="email" required>
</div>
<div class="form-group">
    <label for="register-password">Mot de passe</label>
    <input type="password" id="register-password" name="password" required>
</div>
<div class="form-group">
    <label for="register-password-confirm">Confirmer le mot de passe</label>
    <input type="password" id="register-password-confirm" name="password_confirm" required>
</div>
<div class="form-group">
    <label for="register-profile-pic">Photo de profil (facultatif)</label>
    <input type="file" id="register-profile-pic" name="profile_pic">
</div>
    <button type="submit" class="btn btn-primary">S'inscrire</button>
</form>
</div>
</div>
{% endif %}
</div>
</div>

<div class="step" id="step-shipping">
    <h3>2. Adresse de livraison</h3>
    <div class="step-content">
        <form id="shipping-form">
            <div class="form-row">

```

```
<div class="form-group">
    <label for="shipping-first-name">Prénom</label>
    <input type="text" id="shipping-first-name" name="shipping-first-name" required>
</div>

<div class="form-group">
    <label for="shipping-last-name">Nom</label>
    <input type="text" id="shipping-last-name" name="shipping-last-name" required>
</div>
</div>

<div class="form-group">
    <label for="shipping-address">Adresse</label>
    <input type="text" id="shipping-address" name="shipping-address" required>
</div>

<div class="form-row">
    <div class="form-group">
        <label for="shipping-city">Ville</label>
        <input type="text" id="shipping-city" name="shipping-city" required>
    </div>
    <div class="form-group">
        <label for="shipping-postal">Code postal</label>
        <input type="text" id="shipping-postal" name="shipping-postal" required>
    </div>
</div>

<div class="form-group">
    <label for="shipping-phone">Téléphone</label>
    <input type="tel" id="shipping-phone" name="shipping-phone" required>
</div>

<div class="form-buttons">
    <button type="button" class="btn btn-outline" onclick="showStep('step-login')>Retour</button>
    <button type="button" class="btn btn-primary" onclick="saveShipping()">Continuer</button>
</div>
</form>
</div>

</div>

<div class="step" id="step-payment">
    <h3>3. Mode de paiement</h3>
```

```
<div class="step-content">

    <div class="payment-methods">
        <div class="payment-method">
            <input type="radio" id="payment-orange" name="payment-method" value="orange" checked>
            <label for="payment-orange">Orange Money</label>
            <div class="payment-details" id="orange-details">
                <button class="btn btn-orange" onclick="initiateOrangePayment()">
                    Payer avec Orange Money
                </button>
            <div id="orange-payment-status"></div>
        </div>
    </div>

    <div class="payment-method">
        <input type="radio" id="payment-wave" name="payment-method" value="wave">
        <label for="payment-wave">Wave</label>
        <div class="payment-details" id="wave-details">
            <button class="btn btn-wave" onclick="initiateWavePayment()">
                Payer avec Wave
            </button>
        <div id="wave-payment-status"></div>
        </div>
    </div>

    <div class="payment-method">
        <input type="radio" id="payment-cash" name="payment-method" value="cash">
        <label for="payment-cash">Paiement à la livraison</label>
        <div class="payment-details" id="cash-details">
            <p>Vous paierez en espèces à la livraison de votre commande.</p>
        </div>
    </div>
</div>

<div class="form-buttons">
    <button type="button" class="btn btn-outline" onclick="showStep('step-shipping')">Retour</button>
    <button type="button" class="btn btn-primary" id="place-order-btn" style="display:none;">Confirmer la commande</button>
</div>
```

```
</div>

</div>

</div>

<div class="step" id="step-confirmation">
    <h3>4. Confirmation</h3>
    <div class="step-content">
        <div class="confirmation-message">
            <div class="confirmation-icon">✓</div>
            <h2>Merci pour votre commande!</h2>
            <p>Votre commande a été reçue et sera traitée dans les plus brefs délais.</p>
            <p>Numéro de commande: <strong id="order-number">ORD-12345</strong></p>
            <p>Un email de confirmation a été envoyé à votre adresse email.</p>
        </div>
        <div class="form-buttons">
            <a href="{% url 'home' %}" class="btn btn-outline">Retour à l'accueil</a>
            <a href="{% url 'home' %}" class="btn btn-primary">Continuer vos achats</a>
        </div>
    </div>
</div>
</div>
</div>
```

```
{% block js %}

<style>
    .checkout-container {
        max-width: 1000px;
        margin: 40px auto;
    }

```

```
.page-title {
    text-align: center;
    margin-bottom: 30px;
}
```

```
.checkout-content {  
    display: flex;  
    gap: 30px;  
}  
  
.cart-summary {  
    flex: 1;  
    background-color: white;  
    padding: 20px;  
    border-radius: 10px;  
    box-shadow: 0 4px 15px rgba(0,0,0,0.1);  
}  
  
.cart-summary h2 {  
    margin-bottom: 20px;  
    padding-bottom: 10px;  
    border-bottom: 1px solid #eee;  
}  
  
.cart-items {  
    margin-bottom: 20px;  
}  
  
.cart-item {  
    display: flex;  
    justify-content: space-between;  
    margin-bottom: 15px;  
    padding-bottom: 15px;  
    border-bottom: 1px solid #eee;  
}  
  
.cart-item-details {  
    flex: 3;  
}  
  
.cart-item-title {  
    font-weight: bold;  
}
```

```
}
```

```
.cart-item-price {
```

```
    flex: 1;
```

```
    text-align: right;
```

```
}
```

```
.cart-totals {
```

```
    margin-top: 20px;
```

```
    padding-top: 20px;
```

```
    border-top: 1px solid #eee;
```

```
}
```

```
.cart-totals > div {
```

```
    display: flex;
```

```
    justify-content: space-between;
```

```
    margin-bottom: 10px;
```

```
}
```

```
.total {
```

```
    font-weight: bold;
```

```
    font-size: 1.2em;
```

```
    padding-top: 10px;
```

```
    margin-top: 10px;
```

```
    border-top: 1px solid #eee;
```

```
}
```

```
.checkout-steps {
```

```
    flex: 2;
```

```
}
```

```
.step {
```

```
    background-color: white;
```

```
    border-radius: 10px;
```

```
    box-shadow: 0 4px 15px rgba(0,0,0,0.1);
```

```
    margin-bottom: 20px;
```

```
    overflow: hidden;
```

```
}
```

```
.step h3 {  
  padding: 15px 20px;  
  margin: 0;  
  background-color: #f8f8f8;  
  border-bottom: 1px solid #eee;  
}  
  
}
```

```
.step.active h3 {  
  background-color: #3366cc;  
  color: white;  
}  
  
}
```

```
.step-content {  
  padding: 20px;  
  display: none;  
}  
  
}
```

```
.step.active .step-content {  
  display: block;  
}  
  
}
```

```
.form-group {  
  margin-bottom: 15px;  
}  
  
}
```

```
.form-row {  
  display: flex;  
  gap: 15px;  
}  
  
}
```

```
.form-row .form-group {  
  flex: 1;  
}  
  
}
```

```
label {
```

```
        display: block;  
        margin-bottom: 5px;  
    }  
  
    input[type="text"],  
    input[type="email"],  
    input[type="password"],  
    input[type="tel"],  
    select {
```

```
        width: 100%;  
        padding: 10px;  
        border: 1px solid #ddd;  
        border-radius: 5px;  
    }
```

```
.form-buttons {  
    display: flex;  
    justify-content: space-between;  
    margin-top: 20px;  
    padding-top: 20px;  
    border-top: 1px solid #eee;  
}
```

```
.btn-outline {  
    background-color: transparent;  
    border: 1px solid #3366cc;  
    color: #3366cc;  
}
```

```
.btn-outline:hover {  
    background-color: #f0f5ff;  
}
```

```
.user-info {  
    background-color: #f8f8f8;  
    padding: 15px;  
    border-radius: 5px;
```

```
margin-bottom: 20px;  
}
```

```
.login-options {  
    border: 1px solid #eee;  
    border-radius: 5px;  
    overflow: hidden;  
}
```

```
.tab-headers {  
    display: flex;  
    background-color: #f8f8f8;  
}
```

```
.tab-header {  
    padding: 15px;  
    flex: 1;  
    text-align: center;  
    cursor: pointer;  
    transition: background-color 0.3s;  
}
```

```
.tab-header.active {  
    background-color: white;  
    font-weight: bold;  
}
```

```
.tab-content {  
    padding: 20px;  
    display: none;  
}
```

```
.tab-content.active {  
    display: block;  
}
```

```
.payment-method {
```

```
margin-bottom: 20px;  
padding: 15px;  
border: 1px solid #eee;  
border-radius: 5px;  
}
```

```
.payment-method label {  
font-weight: bold;  
display: inline;  
margin-left: 10px;  
}
```

```
.payment-details {  
margin-top: 15px;  
padding-top: 15px;  
border-top: 1px solid #eee;  
}
```

```
.confirmation-message {  
text-align: center;  
padding: 30px 0;  
margin-bottom: 20px;  
}
```

```
.confirmation-icon {  
width: 80px;  
height: 80px;  
background-color: #28a745;  
border-radius: 50%;  
color: white;  
display: flex;  
align-items: center;  
justify-content: center;  
font-size: 40px;  
margin: 0 auto 20px;  
}
```

```
.btn-orange {  
    background-color: #FF7900;  
    color: white;  
}
```

```
.btn-orange:hover {  
    background-color: #e06d00;  
}
```

```
.btn-wave {  
    background-color: #1D5ECD;  
    color: white;  
}
```

```
.btn-wave:hover {  
    background-color: #174ba3;  
}
```

```
.payment-status {  
    margin-top: 15px;  
    padding: 10px;  
    border-radius: 5px;  
}
```

```
.payment-success {  
    background-color: #d4edda;  
    color: #155724;  
}
```

```
.payment-error {  
    background-color: #f8d7da;  
    color: #721c24;  
}
```

```
.payment-processing {  
    background-color: #fff3cd;  
    color: #856404;
```

```
}

@media (max-width: 768px) {

    .checkout-content {
        flex-direction: column;
    }

    .form-row {
        flex-direction: column;
        gap: 0;
    }

    .form-buttons {
        flex-direction: column;
        gap: 10px;
    }

    .form-buttons .btn {
        width: 100%;
    }
}

</style>

<script type="text/javascript">

$(document).ready(function() {
    // Charger les produits du panier
    loadCartItems();

    // Configurer les onglets de connexion/inscription
    $('.tab-header').click(function() {
        $('.tab-header').removeClass('active');
        $(this).addClass('active');

        var tabId = $(this).text().toLowerCase().trim();
        $('.tab-content').removeClass('active');
        $('#' + tabId).addClass('active');
    });
});
```

```

// Configurer les boutons radio de paiement
$('input[name="payment-method"]').change(function() {
    $('.payment-details').hide();
    $('#' + $(this).val() + '-details').show();
});

// Afficher par défaut les détails du paiement mobile
$('#orange-details').show();

});

function loadCartItems() {
    var Panier = JSON.parse(localStorage.getItem('Panier')) || {};
    var cartItems = $('#checkout-items');
    cartItems.empty();

    var subtotal = 0;

    for (var id in Panier) {
        var item = Panier[id];
        var itemTotal = item[2];
        subtotal += itemTotal;

        var cartItem = $('

</div>');
        cartItem.append('<div class="cart-item-details"><div class="cart-item-title">' + item[1] + '</div><div class="cart-item-quantity">' + item[0] + ' x ' + (itemTotal / item[0]) + ' FCFA</div></div>');
        cartItem.append('<div class="cart-item-price">' + itemTotal + ' FCFA</div>');

        cartItems.append(cartItem);
    }

    $('#subtotal').text(subtotal + ' FCFA');
    $('#total').text((subtotal + 2000) + ' FCFA');

    // Si le panier est vide, rediriger vers la page d'accueil
    if (Object.keys(Panier).length === 0) {
        alert('Votre panier est vide. Veuillez ajouter des produits avant de passer à la caisse.');
    }
}


```

```

        window.location.href = "{% url 'home' %}";
    }

}

function showTab(tabId) {
    $('.tab-header').removeClass('active');
    $('.tab-header').each(function() {
        if ($(this).text().toLowerCase().trim() === tabId) {
            $(this).addClass('active');
        }
    });
}

$('.tab-content').removeClass('active');
$('#' + tabId).addClass('active');

}

function showStep(stepId) {
    $('.step').removeClass('active');
    $('#' + stepId).addClass('active');

    // Scroll to the active step
    $('html, body').animate({
        scrollTop: $('#' + stepId).offset().top - 20
    }, 300);
}

function proceedAsGuest() {
    showStep('step-shipping');
}

function saveShipping() {
    // Validation simple
    var form = document.getElementById('shipping-form');
    if (form.checkValidity()) {
        // Enregistrer les données d'expédition dans sessionStorage
        var shippingData = {
            firstName: $('#shipping-first-name').val(),

```

```

lastName: $('#shipping-last-name').val(),
address: $('#shipping-address').val(),
city: $('#shipping-city').val(),
postalCode: $('#shipping-postal').val(),
phone: $('#shipping-phone').val()

};

sessionStorage.setItem('shippingData', JSON.stringify(shippingData));
showStep('step-payment');

} else {
form.reportValidity();
}

}

// Configuration des APIs

const paymentConfig = {

orange: {
merchantCode: "VOTRE_CODE_MARCHAND_ORANGE",
apiKey: "VOTRE_CLE_API_ORANGE",
callbackUrl: "https://votreboutique.com/payment-callback/orange"
},
wave: {
merchantId: "VOTRE_ID_MARCHAND_WAVE",
apiKey: "VOTRE_CLE_API_WAVE",
callbackUrl: "https://votreboutique.com/payment-callback/wave"
}
};

// Initialisation du paiement Orange Money

function initiateOrangePayment() {

const Panier = JSON.parse(localStorage.getItem('Panier')) || {};
const total = calculateTotal(Panier) + 2000; // Ajout des frais de livraison

$('#orange-payment-status').html('<div class="payment-status payment-processing">Initialisation du paiement...</div>');

placeOrder();
}

```

```

}

// Initialisation du paiement Wave

function initiateWavePayment() {

  const Panier = JSON.parse(localStorage.getItem('Panier')) || {};
  const total = calculateTotal(Panier) + 2000; // Ajout des frais de livraison

  $('#wave-payment-status').html('<div class="payment-status payment-processing">Initialisation du paiement...</div>');

  // En production, remplacer par l'appel réel à placeOrder()
  placeOrder();
}

// Fonction pour calculer le total

function calculateTotal(Panier) {

  let subtotal = 0;
  for (const id in Panier) {
    subtotal += Panier[id][2];
  }
  return subtotal;
}

// Fonctions de simulation (à remplacer par des appels API réels)

function simulateOrangePayment(amount) {

  return new Promise((resolve, reject) => {
    // Simuler un succès 80% du temps
    if (Math.random() > 0.2) {
      resolve({
        status: 'SUCCESS',
        transactionId: 'OM' + Date.now(),
        amount: amount,
        timestamp: new Date().toISOString()
      });
    } else {
      reject(new Error('Échec de la transaction Orange Money'));
    }
  });
}

```

```

}

function simulateWavePayment(amount) {
  return new Promise((resolve, reject) => {
    // Simuler un succès 80% du temps
    if (Math.random() > 0.2) {
      resolve({
        status: 'SUCCESS',
        transactionId: 'WV' + Date.now(),
        amount: amount,
        timestamp: new Date().toISOString()
      });
    } else {
      reject(new Error('Échec de la transaction Wave'));
    }
  });
}

function placeOrder() {
  const paymentMethod = $('input[name="payment-method"]:checked').val();

  // Récupérer les données d'expédition stockées
  const shippingData = JSON.parse(sessionStorage.getItem('shippingData') || '{}');

  const formData = {
    'items': localStorage.getItem('Panier'),
    'total': $('#total').text().replace(' FCFA', ''),
    'address': shippingData.address || '',
    'ville': shippingData.city || '',
    'zipcode': shippingData.postalCode || '',
    'payment_method': paymentMethod,
    'csrfmiddlewaretoken': '{{ csrf_token }}'
  };

  $.ajax({
    url: "{% url 'checkout' %}",
    method: "POST",
    data: formData,
  });
}

```

```

success: function(response) {
    if (response.payment_url) {
        window.location.href = response.payment_url;
    } else {
        localStorage.removeItem('Panier');
        window.location.href = "{% url 'confirmation' %}";
    }
},
error: function(xhr) {
    alert("Erreur : " + (xhr.responseJSON ? xhr.responseJSON.error : "Erreur de paiement"));
}
});

}

// Initialisation
$(document).ready(function() {
// Configurer les boutons radio de paiement
$('input[name="payment-method"]').change(function() {
    $('.payment-details').hide();
    $('#' + $(this).val() + '-details').show();

// Afficher le bouton pour l'option de paiement en espèces
if ($(this).val() === 'cash') {
    $('#place-order-btn').show();
} else {
    $('#place-order-btn').hide();
}
});

}

// Afficher par défaut les détails du paiement Orange
$('#orange-details').show();

// Lier le bouton de commande
$('#place-order-btn').click(placeOrder);
});

</script>
{% endblock %}

```

## Confirmation.html

```
{% extends "base.html" %}

{% block content %}

<div class="row">

    <div class="col-md-12">

        <h3 class="text-dark text-center bg-warning">Salut Mr/Mrs {{ name }}</h3>

        <h4 class="text-muted">Merci de nous faire confiance Mr/Mrs {{ name }}. Nous avons recu votre commande vous seriez livrez dans un delai de 1 jour.</h4>

    </div>

</div>

{% endblock %}
```

## Detail.html

```
{% extends 'base.html' %}

{% block content %}

<div class="row">

    <div class="col-md-6">

        <div>

            

        </div>

    </div>

    <div class="col-md-6">

        <div class="text-warning display-3">

            {{ product.title }}

        </div>

        <div class="text-danger">

            Prix: FCFA {{ product.price }}

        </div>

        <div>

            {{ product.description }}

        </div>

    </div>

<!-- Nouveau code pour les favoris - Style intégré -->

{% if user.is_authenticated %}

<div style="margin: 20px 0;">
```

```

{% if product.is_favorite(user) %}

<form method="post" action="{% url 'remove_favorite' product.id %}" style="display: inline;">
    {% csrf_token %}

    <button type="submit" style="background: #ff6b6b; color: #000; border: none; padding: 8px 15px; border-radius: 5px; cursor: pointer; font-weight: 600;">

        ❤️ Retirer des favoris
    </button>
</form>

{% else %}

<form method="post" action="{% url 'add_favorite' product.id %}" style="display: inline;">
    {% csrf_token %}

    <button type="submit" style="background: #333; color: #fff; border: none; padding: 8px 15px; border-radius: 5px; cursor: pointer; font-weight: 600;">

        🌟 Ajouter aux favoris
    </button>
</form>

{% endif %}

</div>

{% endif %}

<a href="#"><button class="btn btn-success">Acheter</button></a>
</div>
</div>
{% endblock %}

```

## Favorites.html

```

{% extends 'base.html' %}

{% block content %}

<div class="favorites-container">

    <h2>Mes Favoris</h2>

    {% if favorites %}

        <div class="favorites-grid">

            {% for fav in favorites %}

                <div class="favorite-item">

                    <a href="{% url 'detail' fav.product.id %}">
                        
                        <h3>{{ fav.product.title }}</h3>
                
```

```
<p>{{ fav.product.price }} XOF</p>
</a>
<form method="post" action="{% url 'remove_favorite' fav.product.id %}">
    {% csrf_token %}
    <button type="submit" class="remove-btn">Retirer des favoris</button>
</form>
</div>
{% endfor %}
</div>
{% else %}
<p>Vous n'avez aucun produit dans vos favoris.</p>
{% endif %}
</div>

<style>
.favorites-container {
    padding: 2rem;
    max-width: 1200px;
    margin: 0 auto;
}

.favorites-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
    gap: 2rem;
    margin-top: 2rem;
}

.favorite-item {
    background: #161616;
    padding: 1rem;
    border-radius: 8px;
    transition: transform 0.3s;
}

.favorite-item:hover {
    transform: translateY(-5px);
}
```

```
}
```

```
.favorite-item img {  
    width: 100%;  
    height: 200px;  
    object-fit: cover;  
    border-radius: 5px;  
}
```

```
.remove-btn {  
    background: #ff6b6b;  
    color: #000;  
    border: none;  
    padding: 8px 15px;  
    border-radius: 5px;  
    cursor: pointer;  
    margin-top: 10px;  
    font-weight: 600;  
    transition: background 0.3s;  
}
```

```
.remove-btn:hover {  
    background: #ff3b3b;  
}  
</style>
```

```
{% endblock %}
```

## Index.html

```
{% extends "base.html" %}  
{% block content %}  
<section class="hero">  
    <div class="container">  
        <div class="hero-content">  
            <span class="hero-tag">Nouveautés 100% fait main</span>  
            <h1 class="hero-title">L'artisanat <span>ivoirien</span> dans toute sa splendeur</h1>  
            <p class="hero-desc">Pièces uniques créées avec passion par nos artisans. Des trésors culturels à porter au quotidien.</p>  
            <a href="#" class="btn">Découvrir la collection</a>
```

```
</div>

<div class="hero-image">
  
</div>
</div>
</section>

<section class="search-section">
  <div class="container">
    <div class="search-container">
      <form class="search-form">
        <input type="search" name="item-name" placeholder="Recherchez des articles..." class="search-input">
        <button type="submit" class="search-btn">Rechercher</button>
      </form>
    </div>
  </div>
</section>

<section class="products">
  <div class="container">
    <h2 class="section-title">Nos Produits</h2>
    <div class="product-grid">
      {# for product in product_object #}
      <div class="product">
        <div class="product-image">
          
        <span class="product-tag">Populaire</span>
        <div class="product-actions">
          <span class="product-action-btn view-btn" onclick="window.location.href='{{ product.id }}'">@</span>
          <span class="product-action-btn fav-btn">♥</span>
        </div>
      </div>
      <div class="product-info">
        <div class="product-category">Catégorie</div>
        <h3 class="product-title" id="aa{{ product.id }}">{{ product.title }}</h3>
        <div class="product-price">
          <span class="current-price" id="price{{ product.id }}" data-price="{{ product.price }}">
```

```

        {{ product.price }} FCFA
    </span>
</div>
<button id="{{ product.id }}" class="add-to-cart ted">Ajouter au panier</button>
</div>
</div>
{% endfor %}
</div>

<div class="pagination">
<ul>
    {% if product_object.has_previous %}
    <li><a href="?page={{ product_object.previous_page_number }}">Précédent</a></li>
    {% endif %}

    <li class="active"><a href="?page={{ product_object.number }}">{{ product_object.number }}</a></li>

    {% if product_object.has_next %}
    <li><a href="?page={{ product_object.next_page_number }}">Suivant</a></li>
    {% endif %}
</ul>
</div>
</div>
</section>
{% endblock %}

{% block js %}
<style>
/* Styles spécifiques à la page d'accueil */
.hero {
    height: 90vh;
    display: flex;
    align-items: center;
    position: relative;
    overflow: hidden;
    margin-bottom: 60px;
}

```

```
.hero::before {  
    content: "";  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
    background: url('/api/placeholder/1400/900') center/cover;  
    filter: grayscale(70%) brightness(0.4);  
    z-index: -1;  
}  
  
}
```

```
.hero-content {  
    max-width: 600px;  
    position: relative;  
    z-index: 2;  
}
```

```
.hero-tag {  
    display: inline-block;  
    background: #3498db;  
    color: #000;  
    padding: 8px 15px;  
    font-weight: 600;  
    letter-spacing: 2px;  
    text-transform: uppercase;  
    font-size: 14px;  
    margin-bottom: 20px;  
    transform: skew(-10deg);  
}
```

```
.hero-title {  
    font-size: 64px;  
    font-weight: 800;  
    line-height: 1.1;  
    margin-bottom: 25px;
```

```
    text-transform: uppercase;  
}  
  
 .hero-title span {
```

```
    color: #FF8200;  
    position: relative;  
    display: inline-block;  
}
```

```
.hero-title span::after {  
    content: " ";  
    position: absolute;  
    left: 0;  
    bottom: 5px;  
    width: 100%;  
    height: 8px;  
    background: #3498db;  
    opacity: 0.3;  
    z-index: -1;  
}
```

```
.hero-desc {  
    font-size: 18px;  
    line-height: 1.6;  
    margin-bottom: 40px;  
    color: #ccc;  
}
```

```
.btn {  
    display: inline-block;  
    padding: 15px 35px;  
    background: none;  
    border: 2px solid #3498db;  
    color: #fff;  
    font-weight: 600;  
    text-transform: uppercase;  
    letter-spacing: 2px;
```

```
cursor: pointer;
position: relative;
overflow: hidden;
z-index: 1;
text-decoration: none;
transition: color 0.3s ease;
}
```

```
.btn::before {
  content: "";
  position: absolute;
  top: 0;
  left: -100%;
  width: 100%;
  height: 100%;
  background: #3498db;
  z-index: -1;
  transition: left 0.3s ease;
}
```

```
.btn:hover {
  color: #000;
}
```

```
.btn:hover::before {
  left: 0;
}
```

```
.hero-image {
  position: absolute;
  top: 50%;
  right: 5%;
  transform: translateY(-50%);
  width: 50%;
  max-width: 600px;
  z-index: 1;
}
```

```
.hero-image img {  
    width: 100%;  
    filter: brightness(0.8) contrast(1.2);  
    transform: perspective(1000px) rotateY(-15deg);  
}
```

```
/* Search section */
```

```
.search-section {  
    margin-bottom: 60px;  
}
```

```
.search-container {  
    max-width: 800px;  
    margin: 0 auto;  
}
```

```
.search-form {  
    display: flex;  
    background: #161616;  
    border-radius: 10px;  
    overflow: hidden;  
    box-shadow: 0 10px 30px rgba(0,0,0,0.2);  
}
```

```
.search-input {  
    flex-grow: 1;  
    padding: 20px;  
    background: transparent;  
    border: none;  
    color: #fff;  
    font-size: 16px;  
    outline: none;  
}
```

```
.search-btn {  
    background: #3498db;
```

```
color: #000;  
border: none;  
padding: 0 30px;  
font-weight: 600;  
text-transform: uppercase;  
cursor: pointer;  
transition: background 0.3s;  
}  
  
}
```

```
.search-btn:hover {  
background: #2980b9;  
}
```

```
/* Products section */  
.products {  
padding: 60px 0;  
position: relative;  
}
```

```
.section-title {  
font-size: 42px;  
font-weight: 800;  
margin-bottom: 60px;  
text-align: center;  
position: relative;  
text-transform: uppercase;  
}
```

```
.section-title::after {  
content: " ";  
position: absolute;  
left: 50%;  
bottom: -15px;  
width: 80px;  
height: 4px;  
background: #3498db;  
transform: translateX(-50%);
```

```
}

.product-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(280px, 1fr));
    gap: 30px;
    margin-bottom: 60px;
}

.product {
    background: #161616;
    border-radius: 10px;
    overflow: hidden;
    box-shadow: 0 10px 30px rgba(0,0,0,0.2);
    transition: transform 0.4s;
}

.product:hover {
    transform: translateY(-10px);
}

.product-image {
    position: relative;
    height: 280px;
    overflow: hidden;
}

.product-image img {
    width: 100%;
    height: 100%;
    object-fit: cover;
    transition: transform 0.5s;
}

.product:hover .product-image img {
    transform: scale(1.1);
}
```

```
.product-tag {  
    position: absolute;  
    top: 15px;  
    left: 15px;  
    background: #3498db;  
    color: #000;  
    padding: 5px 10px;  
    font-size: 12px;  
    font-weight: 600;  
    text-transform: uppercase;  
}  
  
}
```

```
.product-actions {  
    position: absolute;  
    right: 15px;  
    bottom: 15px;  
    display: flex;  
    flex-direction: column;  
    opacity: 0;  
    transform: translateX(20px);  
    transition: all 0.3s;  
}  
  
}
```

```
.product:hover .product-actions {  
    opacity: 1;  
    transform: translateX(0);  
}  
  
}
```

```
.product-action-btn {  
    background: #000;  
    color: #fff;  
    width: 40px;  
    height: 40px;  
    border-radius: 50%;  
    display: flex;  
    align-items: center;
```

```
justify-content: center;  
margin-bottom: 10px;  
cursor: pointer;  
transition: background 0.3s;  
}
```

```
.product-action-btn:hover {  
background: #3498db;  
color: #000;  
}
```

```
.product-info {  
padding: 25px;  
}
```

```
.product-category {  
color: #3498db;  
font-size: 14px;  
margin-bottom: 10px;  
text-transform: uppercase;  
letter-spacing: 1px;  
}
```

```
.product-title {  
font-size: 18px;  
font-weight: 600;  
margin-bottom: 10px;  
}
```

```
.product-price {  
display: flex;  
align-items: center;  
margin-bottom: 15px;  
}
```

```
.current-price {  
font-size: 20px;
```

```
    font-weight: 700;
    color: #3498db;
}

.old-price {
    font-size: 16px;
    text-decoration: line-through;
    color: #777;
    margin-left: 10px;
}

.add-to-cart {
    display: block;
    width: 100%;
    padding: 12px;
    background: #3498db;
    color: #000;
    border: none;
    border-radius: 5px;
    font-weight: 600;
    text-transform: uppercase;
    cursor: pointer;
    transition: background 0.3s;
}

.add-to-cart:hover {
    background: #2980b9;
}

/* Pagination */

.pagination {
    display: flex;
    justify-content: center;
}

.pagination ul {
    display: flex;
```

```
list-style: none;
padding: 0;
margin: 0;
}

.pagination li {
margin: 0 5px;
}

.pagination a {
display: block;
padding: 10px 15px;
background: #161616;
color: #fff;
text-decoration: none;
border-radius: 5px;
transition: background 0.3s;
}

.pagination li.active a {
background: #3498db;
color: #000;
}

.pagination a:hover {
background: #3498db;
color: #000;
}

/* Cart Popup Styles */
.cart-popup {
position: fixed;
top: 0;
right: -400px;
width: 400px;
height: 100vh;
background: #111;
```

```
z-index: 1000;  
box-shadow: -5px 0 30px rgba(0,0,0,0.3);  
transition: right 0.4s ease;  
overflow-y: auto;  
}
```

```
.cart-popup.active {  
    right: 0;  
}
```

```
.cart-header {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    padding: 20px;  
    border-bottom: 1px solid #222;  
}
```

```
.cart-title {  
    font-size: 24px;  
    font-weight: 700;  
}
```

```
.close-cart {  
    background: none;  
    border: none;  
    color: #fff;  
    font-size: 24px;  
    cursor: pointer;  
}
```

```
.cart-items {  
    padding: 20px;  
}
```

```
.cart-item {  
    display: flex;
```

```
margin-bottom: 20px;  
padding-bottom: 20px;  
border-bottom: 1px solid #222;  
}
```

```
.cart-item-img {  
width: 80px;  
height: 80px;  
overflow: hidden;  
border-radius: 5px;  
margin-right: 15px;  
}
```

```
.cart-item-img img {  
width: 100%;  
height: 100%;  
object-fit: cover;  
}
```

```
.cart-item-details {  
flex-grow: 1;  
}
```

```
.cart-item-title {  
font-weight: 600;  
margin-bottom: 5px;  
}
```

```
.cart-item-price {  
color: #3498db;  
font-weight: 700;  
margin-bottom: 10px;  
}
```

```
.cart-item-quantity {  
display: flex;  
align-items: center;
```

```
}
```

```
.quantity-btn {  
    width: 30px;  
    height: 30px;  
    background: #222;  
    border: none;  
    color: #fff;  
    font-size: 16px;  
    cursor: pointer;  
    transition: background 0.3s;  
}
```

```
.quantity-btn:hover {  
    background: #3498db;  
    color: #000;  
}
```

```
.cart-item-quantity input {  
    width: 40px;  
    height: 30px;  
    text-align: center;  
    background: #222;  
    border: none;  
    color: #fff;  
    margin: 0 5px;  
}
```

```
.cart-item-remove {  
    background: none;  
    border: none;  
    color: #e74c3c;  
    cursor: pointer;  
    margin-left: 10px;  
}
```

```
.cart-footer {
```

```
padding: 20px;  
border-top: 1px solid #222;  
}
```

```
.cart-total {  
display: flex;  
justify-content: space-between;  
margin-bottom: 20px;  
}
```

```
.cart-total-title {  
font-size: 18px;  
font-weight: 600;  
}
```

```
.cart-total-price {  
font-size: 18px;  
font-weight: 700;  
color: #3498db;  
}
```

```
.checkout-btn {  
display: block;  
width: 100%;  
padding: 15px;  
background: #3498db;  
color: #000;  
text-align: center;  
text-decoration: none;  
font-weight: 600;  
text-transform: uppercase;  
border: none;  
cursor: pointer;  
transition: background 0.3s;  
}
```

```
.checkout-btn:hover {
```

```
background: #2980b9;  
}  
  
.cart-overlay {  
    position: fixed;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
    background: rgba(0,0,0,0.7);  
    z-index: 999;  
    opacity: 0;  
    visibility: hidden;  
    transition: all 0.4s ease;  
}  
  
.cart-overlay.active {  
    opacity: 1;  
    visibility: visible;  
}  
  
.cart-toggle {  
    position: fixed;  
    top: 20px;  
    right: 20px;  
    background: #3498db;  
    color: #000;  
    border: none;  
    width: 50px;  
    height: 50px;  
    border-radius: 50%;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    cursor: pointer;  
    z-index: 998;  
    box-shadow: 0 5px 20px rgba(0,0,0,0.3);
```

```
}

.cart-count {
    position: absolute;
    top: -5px;
    right: -5px;
    background: #e74c3c;
    color: #fff;
    width: 20px;
    height: 20px;
    border-radius: 50%;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 12px;
    font-weight: 600;
}
</style>
```

```
<!-- Ajout du HTML pour le panier -->
<div class="cart-toggle" id="cart-toggle">
    <span class="cart-count" id="cart-count">0</span>
</div>

<div class="cart-overlay" id="cart-overlay"></div>

<div class="cart-popup" id="cart-popup">
    <div class="cart-header">
        <div class="cart-title">Votre Panier</div>
        <button class="close-cart" id="close-cart">X</button>
    </div>
    <div class="cart-items" id="cart-items">
        <!-- Les articles du panier seront ajoutés dynamiquement ici -->
    </div>
    <div class="cart-footer">
        <div class="cart-total">
            <div class="cart-total-title">Total:</div>

```

```

<div class="cart-total-price" id="cart-total">0 FCFA</div>
</div>
<a href="/checkout" class="checkout-btn">Passer à la caisse</a>
</div>
</div>

<script type="text/javascript">
    // Attendre que le DOM soit chargé
    document.addEventListener('DOMContentLoaded', function() {
        console.log('DOM chargé - initialisation du panier');

        var Panier = {};

        //Récupérer le panier existant du localStorage s'il existe
        if (localStorage.getItem('Panier') !== null) {
            Panier = JSON.parse(localStorage.getItem('Panier'));

            // Mettre à jour le compteur du panier au chargement
            updateCartCount(Panier);
        }

        //Gérer le clic sur les boutons d'ajout au panier
        document.querySelectorAll('.add-to-cart').forEach(function(button) {
            button.addEventListener('click', function() {
                var item_id = this.id.toString();
                console.log('Ajout au panier - ID:', item_id);

                //Récupérer le panier du localStorage ou créer un nouvel objet
                //var Panier = JSON.parse(localStorage.getItem('Panier')) || {};
                if (Panier[item_id] !== undefined) {
                    // Augmenter la quantité si l'article existe déjà
                    Panier[item_id][0] += 1;
                    Panier[item_id][2] += parseFloat(document.getElementById("price"+item_id).getAttribute('data-price'));
                } else {
                    // Ajouter un nouvel article au panier
                    var quantite = 1;
                    var prix = parseFloat(document.getElementById("price"+item_id).getAttribute('data-price'));
                    var nom = document.getElementById("aa"+item_id).innerHTML;
                }
            });
        });
    });
</script>

```

```
Panier[item_id] = [quantite, nom, prix];  
}  
  
// Sauvegarder le panier dans localStorage  
localStorage.setItem('Panier', JSON.stringify(Panier));  
  
// Mettre à jour l'affichage du panier  
updateCartDisplay(Panier);  
updateCartCount(Panier);  
  
// Animation pour montrer que l'article a été ajouté  
showAddedToCartAnimation(this);  
});  
});  
  
// Initialiser l'affichage du panier  
updateCartDisplay(Panier);  
  
// Gérer l'ouverture et la fermeture du panier  
document.getElementById('cart-toggle').addEventListener('click', function() {  
    document.getElementById('cart-popup').classList.add('active');  
    document.getElementById('cart-overlay').classList.add('active');  
});  
  
document.getElementById('close-cart').addEventListener('click', function() {  
    document.getElementById('cart-popup').classList.remove('active');  
    document.getElementById('cart-overlay').classList.remove('active');  
});  
  
document.getElementById('cart-overlay').addEventListener('click', function() {  
    document.getElementById('cart-popup').classList.remove('active');  
    document.getElementById('cart-overlay').classList.remove('active');  
});  
  
// Fonction pour mettre à jour le contenu du panier  
function updateCartDisplay(Panier) {  
    var cartItemsContainer = document.getElementById('cart-items');
```

```

var cartTotal = 0;

// Vider le contenu actuel
cartItemsContainer.innerHTML = "";

// Si le panier est vide
if (Object.keys(Panier).length === 0) {
    cartItemsContainer.innerHTML = '<p>Votre panier est vide</p>';
    document.getElementById('cart-total').textContent = '0 FCFA';
    return;
}

// Ajouter chaque article au panier
for (var item_id in Panier) {
    var item = Panier[item_id];
    var quantity = item[0];
    var name = item[1];
    var price = item[2];
    cartTotal += price;

    var cartItemHTML = `
        <div class="cart-item" data-id="${item_id}">
            <div class="cart-item-details">
                <div class="cart-item-title">${name}</div>
                <div class="cart-item-price">${price} FCFA</div>
                <div class="cart-item-quantity">
                    <button class="quantity-btn decrease-quantity" data-id="${item_id}">-</button>
                    <input type="text" value="${quantity}" readonly>
                    <button class="quantity-btn increase-quantity" data-id="${item_id}">+</button>
                    <button class="cart-item-remove" data-id="${item_id}">✖</button>
                </div>
            </div>
        </div>
    `;

    cartItemsContainer.innerHTML += cartItemHTML;
}

```

```

// Mettre à jour le total
document.getElementById('cart-total').textContent = cartTotal + ' FCFA';

// Ajouter les écouteurs d'événements pour les boutons de quantité et de suppression
addCartItemEventListeners();

}

// Fonction pour ajouter les écouteurs d'événements aux éléments du panier
function addCartItemEventListeners() {
    // Augmenter la quantité
    document.querySelectorAll('.increase-quantity').forEach(function(button) {
        button.addEventListener('click', function() {
            var item_id = this.getAttribute('data-id');
            updateItemQuantity(item_id, 1);
        });
    });
}

// Diminuer la quantité
document.querySelectorAll('.decrease-quantity').forEach(function(button) {
    button.addEventListener('click', function() {
        var item_id = this.getAttribute('data-id');
        updateItemQuantity(item_id, -1);
    });
});

// Supprimer l'article
document.querySelectorAll('.cart-item-remove').forEach(function(button) {
    button.addEventListener('click', function() {
        var item_id = this.getAttribute('data-id');
        removeCartItem(item_id);
    });
});

// Fonction pour mettre à jour la quantité d'un article
function updateItemQuantity(item_id, change) {

```

```

var Panier = JSON.parse(localStorage.getItem('Panier')) || {};

if (Panier[item_id]) {
    var currentQuantity = Panier[item_id][0];
    var pricePerItem = Panier[item_id][2] / currentQuantity;

    // Nouvelle quantité après changement
    var newQuantity = currentQuantity + change;

    if (newQuantity <= 0) {
        // Si la quantité devient 0 ou moins, supprimer l'article
        removeCartItem(item_id);
    } else {
        // Mettre à jour la quantité et le prix total de l'article
        Panier[item_id][0] = newQuantity;
        Panier[item_id][2] = pricePerItem * newQuantity;

        // Sauvegarder le panier dans localStorage
        localStorage.setItem('Panier', JSON.stringify(Panier));
    }
}

// Fonction pour supprimer un article du panier
function removeCartItem(item_id) {
    var Panier = JSON.parse(localStorage.getItem('Panier')) || {};

    if (Panier[item_id]) {
        delete Panier[item_id];

        // Sauvegarder le panier dans localStorage
        localStorage.setItem('Panier', JSON.stringify(Panier));
    }
}

```

```

// Mettre à jour l'affichage
updateCartDisplay(Panier);
updateCartCount(Panier);

}

}

// Mettre à jour le compteur du panier
function updateCartCount(Panier) {
var count = 0;

// Compter le nombre total d'articles (pas seulement les types d'articles)
for (var item_id in Panier) {
count += Panier[item_id][0];
}

document.getElementById('cart-count').textContent = count;
}

// Animation pour montrer que l'article a été ajouté au panier
function showAddedToCartAnimation(button) {
button.textContent = "Ajouté ✓";
button.style.backgroundColor = "#27ae60";

setTimeout(function() {
button.textContent = "Ajouter au panier";
button.style.backgroundColor = "#3498db";
}, 1500);
}

});

</script>
{% endblock %}

```

## Login.html

```

{% extends "base.html" %}

{% block content %}

<div class="auth-container" style="max-width: 500px; margin: 2rem auto; padding: 2rem; background: #161616; border-radius: 8px;">

<h2 style="text-align: center; margin-bottom: 1.5rem;">Connexion</h2>

```

```

{% if messages %}

    {% for message in messages %}

        <div class="alert" style="color: #e74c3c; margin-bottom: 1rem;">

            {{ message }}

        </div>

    {% endfor %}

{% endif %}

<form method="post" action="{% url 'login' %}{% if request.GET.next %}?next={{ request.GET.next }}{% endif %}" style="display: flex; flex-direction: column; gap: 1rem;">

    {% csrf_token %}

    {% if request.GET.next %}

        <input type="hidden" name="next" value="{{ request.GET.next }}">

    {% endif %}

    <div class="form-group">

        <label for="{{ form.username.id_for_label }}>Nom d'utilisateur</label>
        {{ form.username }}

    </div>

    <div class="form-group">

        <label for="{{ form.password.id_for_label }}>Mot de passe</label>
        {{ form.password }}

    </div>

    <button type="submit" class="btn" style="width: 100%; padding: 12px; background: #3498db; color: #000; border: none; border-radius: 5px; cursor: pointer; margin-top: 1rem;">

        Se connecter

    </button>

</form>

<p style="text-align: center; margin-top: 1.5rem;">

    Pas encore de compte?

    <a href="{% url 'register' %}" style="color: #3498db; text-decoration: none;">S'inscrire</a>

</p>

</div>

```

```
{% endblock %},
```

## Panier.html

```
{% extends "base.html" %}

{% block content %}

<section class="cart">

    <div class="container">

        <h2 class="section-title">Votre panier</h2>

        <div class="cart-items">

            {% for item in cart_items %}

                <div class="cart-item">

                    

                    <div class="cart-item-info">

                        <h3>{{ item.product.title }}</h3>

                        <p>{{ item.product.price }} FCFA</p>

                        <div class="quantity-control">

                            <button class="decrease-quantity" data-id="{{ item.product.id }}"-></button>

                            <span class="quantity">{{ item.quantity }}</span>

                            <button class="increase-quantity" data-id="{{ item.product.id }}"+></button>

                        </div>

                        <button class="remove-item" data-id="{{ item.product.id }}>Supprimer</button>

                    </div>

                </div>

            {% endfor %}

        </div>

        <div class="cart-summary">

            <h3>Total: <span id="cart-total">{{ cart_total }}</span> FCFA</h3>

            {% if user.is_authenticated %}

                <a href="{% url 'checkout' %}" class="btn">Passer la commande</a>

            {% else %}

                <a href="{% url 'login' %}?next={% url 'cart' %}" class="btn">Connectez-vous pour commander</a>

            {% endif %}

        </div>

    </div>

</section>
```

```
{% endblock %}
```

## Products.html

```
{% extends "base.html" %}

{% block content %}

<section class="products">

<div class="container">

<h2 class="section-title">Tous nos produits</h2>

<div class="product-grid">

{% for product in products %}

<div class="product">

<div class="product-image">



{% if product.is_new %}<span class="product-tag">Nouveau</span>{% endif %}

<div class="product-actions">

{% if user.is_authenticated %}

{% if product.is_favorite(user) %}

<form method="post" action="{% url 'remove_favorite' product.id %}" style="display: inline;">

{% csrf_token %}

<button type="submit" style="background: none; border: none; cursor: pointer; font-size: 1.2em; color: #ff3b3b;">

♥

</button>

</form>

{% else %}

<form method="post" action="{% url 'add_favorite' product.id %}" style="display: inline;">

{% csrf_token %}

<button type="submit" style="background: none; border: none; cursor: pointer; font-size: 1.2em; color: #fff;">

♥

</button>

</form>

{% endif %}

{% else %}

<a href="{% url 'login' %}?next={{ request.path }}" style="background: none; border: none; cursor: pointer; font-size: 1.2em; color: #fff; text-decoration: none;">

♥

</a>


```

```

        </a>
    {% endif %}
    <a href="{% url 'product_detail' product.id %}" style="background: none; border: none; cursor: pointer; font-size: 1.2em; color: #fff; text-decoration: none;">
        
    </a>
</div>
</div>
<div class="product-info">
    <div class="product-category">{{ product.category.name }}</div>
    <h3 class="product-title">{{ product.title }}</h3>
    <div class="product-price">
        <span class="current-price">{{ product.price }} FCFA</span>
        {% if product.old_price %}
            <span class="old-price">{{ product.old_price }} FCFA</span>
        {% endif %}
    </div>
    <button class="add-to-cart" data-id="{{ product.id }}">Ajouter au panier</button>
</div>
</div>
{% endfor %}
</div>


<div class="pagination">
    
</div>
</div>
</section>


<script>
document.addEventListener('DOMContentLoaded', function() {
    // Animation au survol des boutons favoris
    const favoriteButtons = document.querySelectorAll('.product-actions button, .product-actions a');
    favoriteButtons.forEach(button => {
        button.addEventListener('mouseenter', function() {

```

```

        this.style.transform = 'scale(1.2)';
    });
    button.addEventListener('mouseleave', function() {
        this.style.transform = 'scale(1)';
    });
});
});

</script>
{% endblock %}

```

## Profile.html

```

{% extends "base.html" %}

{% block content %}

<div class="profile-container">

<h2>Mon Profil</h2>

<div class="profile-pic">
    {% if user.profile.profile_pic %}
        
    {% endif %}
</div>

<div class="profile-info">
    <p><strong>Nom:</strong> {{ user.get_full_name }}</p>
    <p><strong>Email:</strong> {{ user.email }}</p>
    {% if user.profile.phone %}
        <p><strong>Téléphone:</strong> {{ user.profile.phone }}</p>
    {% endif %}
    {% if user.profile.address %}
        <p><strong>Adresse:</strong> {{ user.profile.address }}</p>
        <p><strong>Ville:</strong> {{ user.profile.ville }} {{ user.profile.zipcode }}</p>
    {% endif %}
</div>

<a href="{% url 'edit_profile' %}" class="btn">Modifier le profil</a>
</div>

```

```
{% endblock %}
```

## Register.html

```
{% extends "base.html" %}
```

```
{% block content %}
```

```
<div class="auth-container" style="max-width: 500px; margin: 2rem auto; padding: 2rem; background: #161616; border-radius: 8px;">
```

```
    <h2 style="text-align: center; margin-bottom: 1.5rem;">Créer un compte</h2>
```

```
{% if form.errors %}
```

```
    <div class="alert" style="color: #e74c3c; margin-bottom: 1rem;">
```

```
        Veuillez corriger les erreurs ci-dessous.
```

```
    </div>
```

```
{% endif %}
```

```
<form method="post" enctype="multipart/form-data" style="display: flex; flex-direction: column; gap: 1rem;">
```

```
    {% csrf_token %}
```

```
    <div class="form-group">
```

```
        <label for="{{ form.username.id_for_label }}">Nom d'utilisateur</label>
```

```
        {{ form.username }}
```

```
        {% if form.username.errors %}
```

```
            <div class="error" style="color: #e74c3c;">{{ form.username.errors }}</div>
```

```
        {% endif %}
```

```
    </div>
```

```
    <div class="form-group">
```

```
        <label for="{{ form.email.id_for_label }}">Email</label>
```

```
        {{ form.email }}
```

```
        {% if form.email.errors %}
```

```
            <div class="error" style="color: #e74c3c;">{{ form.email.errors }}</div>
```

```
        {% endif %}
```

```
    </div>
```

```
<div class="form-row" style="display: flex; gap: 1rem;">
```

```
    <div class="form-group" style="flex: 1;">
```

```
        <label for="{{ form.first_name.id_for_label }}">Prénom</label>
```

```

    {{ form.first_name }}

</div>

<div class="form-group" style="flex: 1;">
    <label for="{{ form.last_name.id_for_label }}>Nom</label>
    {{ form.last_name }}
</div>

</div>

<div class="form-group">
    <label for="{{ form.phone.id_for_label }}>Téléphone</label>
    {{ form.phone }}
</div>

<div class="form-group">
    <label for="{{ form.profile_pic.id_for_label }}>Photo de profil (facultatif)</label>
    {{ form.profile_pic }}
</div>

<div class="form-group">
    <label for="{{ form.password1.id_for_label }}>Mot de passe</label>
    {{ form.password1 }}

    {% if form.password1.errors %}
        <div class="error" style="color: #e74c3c;">{{ form.password1.errors }}</div>
    {% endif %}

</div>

<div class="form-group">
    <label for="{{ form.password2.id_for_label }}>Confirmation du mot de passe</label>
    {{ form.password2 }}
</div>

<button type="submit" class="btn" style="width: 100%; padding: 12px; background: #3498db; color: #000; border: none; border-radius: 5px; cursor: pointer; margin-top: 1rem;">
    S'inscrire
</button>
</form>

```

```

<p style="text-align: center; margin-top: 1.5rem;">
    Déjà un compte?
    <a href="{% url 'login' %}" style="color: #3498db; text-decoration: none;">Se connecter</a>
</p>
</div>
{% endblock %}

```

## Admin.py

```

from django.contrib import admin
from .models import Categorie, Product, Commande, UserProfile, User
from django.contrib.auth.admin import UserAdmin
# Register your models here.
# Register your models here.
admin.site.site_header = "E-commerce"
admin.site.site_title = "Mynia"
admin.site.index_title = "Manageur"

class AdminCategorie(admin.ModelAdmin):
    list_display = ('name', 'date_ajout')

class AdminProduct(admin.ModelAdmin):
    list_display = ('title', 'price', 'Categorie', 'date_ajout')
    search_fields = ('title',)
    list_editable = ('price',)

class AdminCommande(admin.ModelAdmin):
    list_display = ('id', 'user', 'nom', 'email', 'items', 'total', 'ville', 'pays', 'date_commande')
    search_fields = ('nom', 'email', 'ville')

class UserProfileInline(admin.StackedInline):
    model = UserProfile
    extra = 1

class CustomUserAdmin(UserAdmin):
    inlines = [UserProfileInline]
    list_display = ('username', 'email', 'first_name', 'last_name', 'is_staff')

admin.site.register(Product, AdminProduct)
admin.site.register(Categorie, AdminCategorie)

```

```
admin.site.register(Commande, AdminCommande)
admin.site.unregister(User)
admin.site.register(User, CustomUserAdmin)
```

## Apps.py

```
from django.apps import AppConfig

class MonMarchéConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'mon_marché'
```

## Forms.py

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User # Import manquant
from .models import UserProfile
from django.core.exceptions import ValidationError

class RegisterForm(UserCreationForm):
    phone = forms.CharField(max_length=20, required=True)
    profile_pic = forms.ImageField(required=False) # Uniformisez ce nom

    class Meta:
        model = User
        fields = ['username', 'email', 'password1', 'password2', 'first_name', 'last_name']

    def save(self, commit=True):
        user = super().save(commit=commit) # Modification cruciale
        UserProfile.objects.create(
            user=user,
            phone=self.cleaned_data['phone'],
            profile_pic=self.cleaned_data.get('profile_pic', None)
        )
        return user

    def clean_email(self):
        email = self.cleaned_data.get('email')
        if User.objects.filter(email=email).exists():
            raise ValidationError("Cet email est déjà utilisé.")
        return email
```

```

class LoginForm(forms.Form):
    username = forms.CharField(label='Nom d\'utilisateur')
    password = forms.CharField(widget=forms.PasswordInput)

Models.py

from django.db import models
from django.db.models.fields.related import ForeignKey
from django.contrib.auth.models import User
from django.contrib.auth.models import AbstractUser, BaseUserManager
from django.utils.translation import gettext_lazy as _

class Categorie(models.Model):
    name = models.CharField(max_length=200)
    date_ajout = models.DateTimeField(auto_now=True)

    class Meta:
        ordering = ['-date_ajout']

    def __str__(self):
        return self.name

class Product(models.Model):
    title = models.CharField(max_length=200)
    price = models.FloatField()
    description = models.TextField()

    Categorie = ForeignKey(Categorie, related_name='category', on_delete=models.CASCADE)
    image = models.ImageField(upload_to='products/')
    date_ajout = models.DateTimeField(auto_now=True)
    is_new = models.BooleanField(default=False) # Ajouté
    old_price = models.FloatField(null=True, blank=True) # Ajouté

    def is_favorite(self, user):
        if not user.is_authenticated:
            return False
        return Favorite.objects.filter(user=user, product=self).exists()

    class Meta:
        ordering = ['-date_ajout']

```

```

def __str__(self):
    return self.title

class Commande(models.Model):
    items = models.TextField(max_length=300)
    total = models.CharField(max_length=200)
    nom = models.CharField(max_length=150)
    email = models.EmailField()
    address = models.CharField(max_length=200)
    ville = models.CharField(max_length=200)
    pays = models.CharField(max_length=300)
    zipcode = models.CharField(max_length=300)
    date_commande = models.DateTimeField(auto_now=True)
    user = models.ForeignKey(User, on_delete=models.SET_NULL, null=True, blank=True) # Nouveau champ
    payment_method = models.CharField(max_length=50, blank=True) # Nouveau
    payment_status = models.CharField(max_length=50, default='pending') # Nouveau

    class Meta:
        ordering = ['-date_commande']

    def __str__(self):
        return self.nom

class UserProfile(models.Model):
    user = models.OneToOneField(User, on_delete=models.CASCADE, related_name='profile')
    phone = models.CharField(max_length=20, blank=True)
    address = models.CharField(max_length=200, blank=True)
    ville = models.CharField(max_length=100, blank=True)
    zipcode = models.CharField(max_length=10, blank=True)
    profile_pic = models.ImageField(upload_to='profile_pics/', blank=True, null=True)

    def __str__(self):
        return f"Profil de {self.user.username}"

class Favorite(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    product = models.ForeignKey(Product, on_delete=models.CASCADE)

```

```

date_added = models.DateTimeField(auto_now_add=True)

class Meta: # Nouveau Meta
    unique_together = ('user', 'product') # Empêche les doublons

class Cart(models.Model):
    user = models.ForeignKey(User, on_delete=models.CASCADE)
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    quantity = models.PositiveIntegerField(default=1)
    date_added = models.DateTimeField(auto_now_add=True)

    class Meta: # Nouveau Meta
        ordering = ['-date_added']

```

## Urls.py

```

from django.urls import path
from mon_marché.views import (
    index,
    detail,
    checkout,
    confirmation,
    initiate_orange_payment, # Ajout des nouvelles vues
    initiate_wave_payment,
    register,
    user_login,
    user_logout,
    profile,
    favorites,
    add_favorite,
    remove_favorite

```

```
)
```

```

urlpatterns = [
    # URLs existantes (conservées telles quelles)
    path("", index, name='home'),

```

```

path('<int:myid>', detail, name="detail"),
path('checkout/', checkout, name="checkout"), # Ajout du / pour la cohérence
path('confirmation/', confirmation, name='confirmation'),


# Nouveaux endpoints de paiement
path('payment/orange/', initiate_orange_payment, name="initiate_orange"),
path('payment/wave/', initiate_wave_payment, name="initiate_wave"),


# URLs d'authentification
path('register/', register, name='register'),
path('login/', user_login, name='login'),
path('logout/', user_logout, name='logout'),
path('profile/', profile, name='profile'),
path('favorites/', favorites, name='favorites'),


path('favorites/add/<int:product_id>', add_favorite, name='add_favorite'),
path('favorites/remove/<int:product_id>', remove_favorite, name='remove_favorite'),
]

```

## Views.py

```

from django.shortcuts import render, redirect, get_object_or_404

from django.http import JsonResponse

from django.conf import settings

from .models import Product, Commande, UserProfile, Favorite

from django.core.paginator import Paginator

import requests

import random

import string

from django.contrib.auth import login, authenticate, logout

from django.contrib.auth.models import User

from django.contrib import messages

from django.contrib.auth.decorators import login_required

from .forms import RegisterForm, LoginForm # Import des nouveaux formulaires

# Fonction pour générer un ID de commande aléatoire

def generate_order_id():

    return ''.join(random.choices(string.ascii_uppercase + string.digits, k=10))

```

```

# Fonction pour générer une référence client

def generate_client_reference():
    return 'CLIENT-' + ''.join(random.choices(string.digits, k=8))

def index(request):
    Product_object = Product.objects.all()

    item_name = request.GET.get('item-name')

    if item_name != "" and item_name is not None:
        Product_object = Product.objects.filter(title__icontains=item_name)

    paginator = Paginator(Product_object, 4)
    page = request.GET.get('page')

    Product_object = paginator.get_page(page)

    return render(request, 'index.html', {'product_object': Product_object})

def detail(request, myid):
    product_object = Product.objects.get(id=myid)
    return render(request, 'detail.html', {'product': product_object})

# Nouvelle version de checkout avec gestion authentifiée

def checkout(request):
    if not request.user.is_authenticated:
        return redirect(f'/login/?next={request.path}')

    if request.method == "POST":
        items = request.POST.get('items')
        total = request.POST.get('total')

        # Utilisation des infos du profil connecté

        com = Commande(
            items=items,
            total=total,
            nom=request.user.get_full_name(),
            email=request.user.email,
            address=request.POST.get('address'),
            ville=request.POST.get('ville'),
            pays=request.POST.get('pays'),
            zipcode=request.POST.get('zipcode'),
            user=request.user # Lier la commande à l'utilisateur

```

```

        )

com.save()

return redirect('confirmation')

# Pré-remplir avec les infos utilisateur si disponible

try:

    profile = request.user.profile

    initial_data = {

        'nom': request.user.get_full_name(),

        'email': request.user.email,

        'phone': profile.phone

    }

except UserProfile.DoesNotExist:

    initial_data = {}

return render(request, 'checkout.html', {'initial_data': initial_data})

def confirmation(request):

    if not request.user.is_authenticated:

        return redirect('login')

    # Récupère la dernière commande de l'utilisateur

    info = Commande.objects.filter(user=request.user).order_by('-date_commande')[1]

    if info:

        return render(request, 'confirmation.html', {'name': info[0].nom})

    return redirect('home')

def initiate_orange_payment(request, commande_id=None):

    if request.method == 'POST':

        amount = request.POST.get('amount')

        # Vérification que les settings sont bien configurés

        if not all([hasattr(settings, 'ORANGE_MERCHANT_KEY'),

                   hasattr(settings, 'ORANGE_API_TOKEN')]):

            return JsonResponse({'status': 'error', 'message': 'Configuration Orange Money manquante'}, status=500)

    payload = {

        'merchant_key': settings.ORANGE_MERCHANT_KEY,

```

```

'currency': 'XOF',
'order_id': generate_order_id(),
'amount': amount,
'return_url': getattr(settings, 'ORANGE_CALLBACK_URL', ''),
'cancel_url': getattr(settings, 'ORANGE_CANCEL_URL', ''),
'notif_url': getattr(settings, 'ORANGE_NOTIF_URL', ''),
'lang': 'fr'
}

try:
    response = requests.post(
        'https://api.orange.com/orange-money-webpay/dev/v1/webpayment',
        json=payload,
        headers={'Authorization': f'Bearer {settings.ORANGE_API_TOKEN}'},
        timeout=30
    )
    response.raise_for_status() # Lève une exception pour les codes 4XX/5XX
    return JsonResponse(response.json())
except requests.exceptions.RequestException as e:
    return JsonResponse({'status': 'error', 'message': str(e)}, status=500)
return JsonResponse({'status': 'error', 'message': 'Méthode non autorisée'}, status=405)

def initiate_wave_payment(request, commande_id=None):
    if request.method == 'POST':
        amount = request.POST.get('amount')
        # Vérification de la configuration Wave
        if not all([hasattr(settings, 'WAVE_API_KEY'),
                   hasattr(settings, 'WAVE_MERCHANT_ID')]):
            return JsonResponse({'status': 'error', 'message': 'Configuration Wave manquante'}, status=500)

        payload = {
            'amount': amount,
            'currency': 'XOF',
            'client_reference': generate_client_reference(),
            'error_url': getattr(settings, 'WAVE_ERROR_URL', ''),
            'success_url': getattr(settings, 'WAVE_SUCCESS_URL', '')
        }

```

```

try:
    response = requests.post(
        'https://api.wave.com/v1/checkout/sessions',
        json=payload,
        headers={
            'Authorization': f'Bearer {settings.WAVE_API_KEY}',
            'Content-Type': 'application/json'
        },
        timeout=30
    )
    response.raise_for_status()
    return JsonResponse(response.json())
except requests.exceptions.RequestException as e:
    return JsonResponse({'status': 'error', 'message': str(e)}, status=500)
    return JsonResponse({'status': 'error', 'message': 'Méthode non autorisée'}, status=405)

# Nouvelles vues d'authentification

def register(request):
    if request.method == 'POST':
        form = RegisterForm(request.POST, request.FILES)
        if form.is_valid():
            user = form.save()
            # Création automatique du profil
            UserProfile.objects.create(
                user=user,
                phone=form.cleaned_data['phone'],
                profile_picture=form.cleaned_data['profile_pic']
            )
            # Connexion automatique
            login(request, user)
            messages.success(request, "Inscription réussie !")
            return redirect('home')

    else:
        print(form.errors) # Debug
else:
    form = RegisterForm()

```

```

return render(request, 'register.html', {'form': form})

def user_login(request):
    if request.method == 'POST':
        form = LoginForm(request.POST)
        if form.is_valid():
            username = form.cleaned_data['username']
            password = form.cleaned_data['password']
            user = authenticate(request, username=username, password=password)
            if user is not None:
                login(request, user)
                # Récupérer le paramètre next du formulaire ou de l'URL
                next_url = request.POST.get('next', '') or request.GET.get('next', '')
                if next_url:
                    return redirect(next_url)
                else:
                    return redirect('home')
            else:
                messages.error(request, "Identifiants incorrects")
        else:
            # Afficher les erreurs de formulaire
            for field, errors in form.errors.items():
                for error in errors:
                    messages.error(request, f"{field}: {error}")
    else:
        form = LoginForm()

    return render(request, 'login.html', {'form': form})

def user_logout(request):
    logout(request)
    return redirect('home')

@login_required
def profile(request):
    try:
        # Essayez d'obtenir ou de créer le profil
        profile, created = UserProfile.objects.get_or_create(user=request.user)
    
```

```
    return render(request, 'profile.html', {'user': request.user, 'profile': profile})

except Exception as e:
    messages.error(request, f"Erreur lors de l'accès au profil: {str(e)}")
    return redirect('home')

@login_required
def favorites(request):
    # Récupérer les favoris de l'utilisateur connecté
    userFavorites = Favorite.objects.filter(user=request.user).select_related('product')
    return render(request, 'favorites.html', {'favorites': userFavorites})

@login_required
def add_favorite(request, product_id):
    product = get_object_or_404(Product, id=product_id)
    Favorite.objects.get_or_create(user=request.user, product=product)
    return redirect(request.META.get('HTTP_REFERER', 'home'))

@login_required
def remove_favorite(request, product_id):
    product = get_object_or_404(Product, id=product_id)
    Favorite.objects.filter(user=request.user, product=product).delete()
    return redirect('favorites')

@login_required
def checkout(request):
    if request.method == "POST":
        try:
            panier = request.session.get('panier', {})
            commande = Commande(
                nom=f"{request.user.first_name} {request.user.last_name}",
                items=str(panier),
                total=request.POST.get('total'),
                email=request.user.email,
                address=request.POST.get('address'),
                ville=request.POST.get('ville'),
                pays=request.POST.get('pays', ''),
                zipcode=request.POST.get('zipcode'),
            )
        except Exception as e:
            messages.error(request, f"Erreur lors de la création de la commande: {str(e)}")
            return redirect('checkout')
        else:
            commande.save()
            request.session['commande_id'] = commande.id
            return redirect('confirmation')
    else:
        return render(request, 'checkout.html')
```

```
        user=request.user,
        payment_method=request.POST.get('payment_method'),
        payment_status='pending'

    )
commande.save()

if request.POST.get('payment_method') == 'orange':
    return initiate_orange_payment(request, commande.id)
elif request.POST.get('payment_method') == 'wave':
    return initiate_wave_payment(request, commande.id)
else:
    return redirect('confirmation')

except Exception as e:
    messages.error(request, f"Erreur : {str(e)}")
    return redirect('checkout')

return render(request, 'checkout.html')
```