



MODULE 4: LOCAL PAYMENT INTEGRATION

LESSON 11: Integrating PayHere Checkout

Learning Objective

By the end of this lesson, you will integrate PayHere payment gateway into your CeylonMart site and process test payments.

11.1 Creating the Checkout Page

Step 1: Create `app/checkout/page.js`

javascript

1. `"use client";`
- 2.
3. `import { useState } from 'react';`

```
4. import { useRouter } from 'next/navigation';
5. import Link from 'next/link';
6. import Header from '@app/components/Header';
7. import Footer from '@app/components/Footer';
8. import { useCart } from '@app/context/CartContext';
9.
10. // PayHere sandbox credentials (replace with yours)
11. const PAYHERE_MERCHANT_ID = '121XXX'; // Get from PayHere dashboard
12. const PAYHERE_SANDBOX_URL = 'https://sandbox.payhere.lk/pay/checkout';
13.
14. export default function CheckoutPage() {
15.   const router = useRouter();
16.   const { cart, cartTotal, clearCart } = useCart();
17.   const [formData, setFormData] = useState({
18.     firstName: "",
19.     lastName: "",
20.     email: "",
21.     phone: "",
22.     address: "",
23.     city: "",
24.     deliveryNotes: ""
25.   });
26.   const [isProcessing, setIsProcessing] = useState(false);
27.
28.   if (cart.length === 0) {
29.     router.push('/cart');
30.     return null;
31.   }
32.
33.   const handleInputChange = (e) => {
34.     const { name, value } = e.target;
35.     setFormData(prev => ({ ...prev, [name]: value }));
36.   };
37.
38.   const generateOrderId = () => {
39.     return 'ORD-' + Date.now() + '-' + Math.random().toString(36).substr(2, 9);
40.   };
```

```
41.
42. const handlePayHerePayment = () => {
43.   setIsProcessing(true);
44.
45.   // Generate unique order ID
46.   const orderId = generateOrderId();
47.
48.   // Prepare payment data for PayHere
49.   const paymentData = {
50.     sandbox: true, // Use false for live
51.     merchant_id: PAYHERE_MERCHANT_ID,
52.     return_url: `${window.location.origin}/payment/success`,
53.     cancel_url: `${window.location.origin}/payment/cancel`,
54.     notify_url: `${window.location.origin}/api/payment/notify`, // We'll create this later
55.     order_id: orderId,
56.     items: cart.map(item => item.name).join(', '),
57.     currency: 'LKR',
58.     amount: (cartTotal + 250).toFixed(2), // Including shipping
59.     first_name: formData.firstName,
60.     last_name: formData.lastName,
61.     email: formData.email,
62.     phone: formData.phone,
63.     address: formData.address,
64.     city: formData.city,
65.     country: 'Sri Lanka',
66.     // Custom parameters
67.     custom_1: JSON.stringify(cart.map(item => ({
68.       id: item.id,
69.       quantity: item.quantity,
70.       price: item.price
71.     })))
72.   };
73.
74.   // Create a form and submit to PayHere
75.   const form = document.createElement('form');
76.   form.method = 'POST';
77.   form.action = PAYHERE_SANDBOX_URL;
```

```
78.
79. Object.keys(paymentData).forEach(key => {
80.   const input = document.createElement('input');
81.   input.type = 'hidden';
82.   input.name = key;
83.   input.value = paymentData[key];
84.   form.appendChild(input);
85. });
86.
87. document.body.appendChild(form);
88. form.submit();
89. };
90.
91. const handleSubmit = (e) => {
92.   e.preventDefault();
93.
94.   // Basic validation
95.   const requiredFields = ['firstName', 'lastName', 'email', 'phone', 'address', 'city'];
96.   const missingFields = requiredFields.filter(field => !formData[field]);
97.
98.   if (missingFields.length > 0) {
99.     alert('Please fill in all required fields');
100.    return;
101.   }
102.
103.   // Proceed to payment
104.   handlePayHerePayment();
105. };
106.
107. return (
108.   <div>
109.     <Header />
110.
111.     <main style={{ padding: '2rem', maxWidth: '1200px', margin: '0 auto' }}>
112.       <h1 style={{ fontSize: '2rem', marginBottom: '2rem' }}>Checkout</h1>
113.
114.       <div style={{
```

```
115.     display: 'grid',
116.     gridTemplateColumns: '2fr 1fr',
117.     gap: '2rem'
118.   }}>
119.   { /* Checkout Form */ }
120.   <div>
121.     <form onSubmit={handleSubmit}>
122.       <div style={{
123.         backgroundColor: 'white',
124.         padding: '1.5rem',
125.         borderRadius: '8px',
126.         boxShadow: '0 2px 4px rgba(0,0,0,0.1)',
127.         marginBottom: '2rem'
128.       }}>
129.         <h2 style={{ fontSize: '1.3rem', marginBottom: '1rem' }}>Contact
    Information</h2>
130.
131.         <div style={{ display: 'grid', gap: '1rem' }}>
132.           <div style={{ display: 'grid', gridTemplateColumns: '1fr 1fr', gap: '1rem' }}>
133.             <div>
134.               <label style={{ display: 'block', marginBottom: '0.5rem' }}>
135.                 First Name *
136.               </label>
137.               <input
138.                 type="text"
139.                 name="firstName"
140.                 value={formData.firstName}
141.                 onChange={handleInputChange}
142.                 required
143.                 style={{
144.                   width: '100%',
145.                   padding: '0.75rem',
146.                   border: '1px solid #cbd5e1',
147.                   borderRadius: '4px'
148.                 }}
149.               />
150.             </div>
```

```
151.
152.     <div>
153.         <label style={{ display: 'block', marginBottom: '0.5rem' }}>
154.             Last Name *
155.         </label>
156.         <input
157.             type="text"
158.             name="lastName"
159.             value={formData.lastName}
160.             onChange={handleInputChange}
161.             required
162.             style={{
163.                 width: '100%',
164.                 padding: '0.75rem',
165.                 border: '1px solid #cbd5e1',
166.                 borderRadius: '4px'
167.             }}
168.         />
169.     </div>
170. </div>
171.
172. <div>
173.     <label style={{ display: 'block', marginBottom: '0.5rem' }}>
174.         Email Address *
175.     </label>
176.     <input
177.         type="email"
178.         name="email"
179.         value={formData.email}
180.         onChange={handleInputChange}
181.         required
182.         style={{
183.             width: '100%',
184.             padding: '0.75rem',
185.             border: '1px solid #cbd5e1',
186.             borderRadius: '4px'
187.         }}

```

```
188.     />
189.     </div>
190.
191.     <div>
192.         <label style={{ display: 'block', marginBottom: '0.5rem' }}>
193.             Phone Number *
194.         </label>
195.         <input
196.             type="tel"
197.             name="phone"
198.             value={formData.phone}
199.             onChange={handleInputChange}
200.             required
201.             placeholder="07X XXX XXXX"
202.             style={{
203.                 width: '100%',
204.                 padding: '0.75rem',
205.                 border: '1px solid #cbd5e1',
206.                 borderRadius: '4px'
207.             }}
208.         />
209.     </div>
210. </div>
211. </div>
212.
213. <div style={{
214.     backgroundColor: 'white',
215.     padding: '1.5rem',
216.     borderRadius: '8px',
217.     boxShadow: '0 2px 4px rgba(0,0,0,0.1)',
218.     marginBottom: '2rem'
219. }}>
220.     <h2 style={{ fontSize: '1.3rem', marginBottom: '1rem' }}>Delivery
    Address</h2>
221.
222.     <div style={{ display: 'grid', gap: '1rem' }}>
223.         <div>
```

```
224.     <label style={{ display: 'block', marginBottom: '0.5rem' }}>
225.       Address Line *
226.     </label>
227.     <input
228.       type="text"
229.       name="address"
230.       value={formData.address}
231.       onChange={handleInputChange}
232.       required
233.       style={{
234.         width: '100%',
235.         padding: '0.75rem',
236.         border: '1px solid #cbd5e1',
237.         borderRadius: '4px'
238.       }}
239.     />
240.   </div>
241.
242.   <div style={{ display: 'grid', gridTemplateColumns: '1fr 1fr', gap: '1rem' }}>
243.     <div>
244.       <label style={{ display: 'block', marginBottom: '0.5rem' }}>
245.         City *
246.       </label>
247.       <input
248.         type="text"
249.         name="city"
250.         value={formData.city}
251.         onChange={handleInputChange}
252.         required
253.         style={{
254.           width: '100%',
255.           padding: '0.75rem',
256.           border: '1px solid #cbd5e1',
257.           borderRadius: '4px'
258.         }}
259.       />
260.     </div>
```

```
261.
262.     <div>
263.         <label style={{ display: 'block', marginBottom: '0.5rem' }}>
264.             Delivery Notes
265.         </label>
266.         <input
267.             type="text"
268.             name="deliveryNotes"
269.             value={formData.deliveryNotes}
270.             onChange={handleInputChange}
271.             placeholder="E.g., Landmark, apartment number"
272.             style={{
273.                 width: '100%',
274.                 padding: '0.75rem',
275.                 border: '1px solid #cbd5e1',
276.                 borderRadius: '4px'
277.             }}
278.         />
279.     </div>
280. </div>
281. </div>
282. </div>
283.
284. <button
285.     type="submit"
286.     disabled={isProcessing}
287.     style={{
288.         width: '100%',
289.         backgroundColor: isProcessing ? '#94a3b8' : '#1e40af',
290.         color: 'white',
291.         border: 'none',
292.         padding: '1rem',
293.         borderRadius: '4px',
294.         fontSize: '1.1rem',
295.         fontWeight: 'bold',
296.         cursor: isProcessing ? 'not-allowed' : 'pointer'
297.     }}

```

```

298.         >
299.           {isProcessing ? 'Redirecting to PayHere...' : 'Proceed to Payment'}
300.         </button>
301. </form>
302. </div>
303.
304.     {/ * Order Summary */}
305.     <div style={{
306.         backgroundColor: '#f8f9fc',
307.         padding: '1.5rem',
308.         borderRadius: '8px',
309.         height: 'fit-content'
310.     }}>
311.     <h2 style={{ fontSize: '1.3rem', marginBottom: '1rem' }}>Order Summary</h2>
312.
313.     {cart.map(item => (
314.     <div
315.     key={item.id}
316.     style={{
317.         display: 'flex',
318.         justifyContent: 'space-between',
319.         marginBottom: '0.5rem',
320.         fontSize: '0.9rem'
321.     }}
322.     >
323.         <span>{item.name} x {item.quantity}</span>
324.         <span>Rs. {(item.price * item.quantity).toLocaleString()}</span>
325.     </div>
326.     )))}
327.
328.     <div style={{
329.         borderTop: '1px solid #e2e8f0',
330.         marginTop: '1rem',
331.     padding: '1rem'
332.     }}>
333.     <div style={{
334.         display: 'flex',

```

```
335.         justifyContent: 'space-between',
336.         marginBottom: '0.5rem'
337.     }}>
338.     <span>Subtotal</span>
339.     <span>Rs. {cartTotal.toLocaleString()}</span>
340. </div>
341.
342.     <div style={{
343.         display: 'flex',
344.         justifyContent: 'space-between',
345.         marginBottom: '0.5rem'
346.     }}>
347.         <span>Shipping</span>
348.         <span>Rs. 250</span>
349.     </div>
350.
351.     <div style={{
352.         display: 'flex',
353.         justifyContent: 'space-between',
354.         fontWeight: 'bold',
355.         fontSize: '1.2rem',
356.         marginTop: '1rem',
357.         paddingTop: '1rem',
358.         borderTop: '1px solid #e2e8f0'
359.     }}>
360.         <span>Total</span>
361.         <span>Rs. {(cartTotal + 250).toLocaleString()}</span>
362.     </div>
363. </div>
364. </div>
365. </div>
366. </main>
367.
368. <Footer />
369. </div>
370. );
371. }
```

11.2 Creating Payment Notification Handler

PayHere sends a notification to your server after payment. We need to handle this.

Step 1: Create `app/api/payment/notify/route.js`

javascript

```
1. import { NextResponse } from 'next/server';
2.
3. export async function POST(request) {
4.   try {
5.     const formData = await request.formData();
6.     const data = Object.fromEntries(formData);
7.
8.     // Log for debugging
9.     console.log('Payment Notification:', data);
10.
11.    // Verify payment (in production, verify with md5 signature)
12.    const {
13.      merchant_id,
14.      order_id,
15.      payhere_amount,
16.      payhere_currency,
17.      status_code,
18.      md5sig
19.    } = data;
20.
21.    // Check if payment was successful
22.    if (status_code === '2') { // 2 means success
23.      // Payment successful
24.      // TODO: Update order status in database
25.      // TODO: Send confirmation email to customer
```

```
26. // TODO: Update inventory
27.
28. console.log(`Payment successful for order: ${order_id}`);
29.
30. // You can access cart data from custom_1
31. if (data.custom_1) {
32.   const cartData = JSON.parse(data.custom_1);
33.   console.log('Order items:', cartData);
34. }
35. } else {
36.   // Payment failed
37.   console.log(`Payment failed for order: ${order_id}, Status: ${status_code}`);
38. }
39.
40. // Always return success to PayHere
41. return new NextResponse('OK', { status: 200 });
42.
43. } catch (error) {
44.   console.error('Payment notification error:', error);
45.   return new NextResponse('Error', { status: 500 });
46. }
47. }
```

11.3 Creating Payment Result Pages

Step 1: Create `app/payment/success/page.js`

javascript

```
1. "use client";
2.
3. import { useEffect } from 'react';
4. import { useRouter } from 'next/navigation';
5. import Link from 'next/link';
```

```
6. import Header from '@app/components/Header';
7. import Footer from '@app/components/Footer';
8. import { useCart } from '@app/context/CartContext';
9.
10. export default function PaymentSuccessPage() {
11.   const router = useRouter();
12.   const { clearCart } = useCart();
13.
14.   useEffect(() => {
15.     // Clear cart after successful payment
16.     clearCart();
17.
18.     // Optional: Send order confirmation email
19.     // Optional: Track conversion
20.   }, [clearCart]);
21.
22.   return (
23.     <div>
24.       <Header />
25.
26.       <main style={{ padding: '4rem 2rem', textAlign: 'center', minHeight: '60vh' }}>
27.         <div style={{ maxWidth: '500px', margin: '0 auto' }}>
28.           <div style={{
29.             fontSize: '4rem',
30.             marginBottom: '1rem',
31.             color: '#059669'
32.           }}>
33.             
34.         </div>
35.
36.         <h1 style={{ fontSize: '2rem', marginBottom: '1rem', color: '#059669' }}>
37.           Payment Successful!
38.         </h1>
39.
40.         <p style={{ color: '#475569', marginBottom: '2rem', lineHeight: '1.6' }}>
41.           Thank you for your purchase! We've received your order and will send you a
           confirmation email shortly with delivery details.
```

```
42. </p>
43.
44. <div style={{
45.   backgroundColor: '#f8fafc',
46.   padding: '1.5rem',
47.   borderRadius: '8px',
48.   marginBottom: '2rem',
49.   textAlign: 'left'
50. }}>
51.   <h3 style={{ marginBottom: '1rem' }}>What happens next?</h3>
52.   <ul style={{ paddingLeft: '1.5rem', color: '#475569' }}>
53.     <li style={{ marginBottom: '0.5rem' }}>>✓ You'll receive an order confirmation
    email</li>
54.     <li style={{ marginBottom: '0.5rem' }}>>✓ We'll prepare your items for
    shipping</li>
55.     <li style={{ marginBottom: '0.5rem' }}>>✓ You'll get tracking information once
    shipped</li>
56.     <li>>✓ Delivery within 2-3 business days (Colombo) or 3-5 days (other
    areas)</li>
57.   </ul>
58. </div>
59.
60. <Link
61.   href="/products"
62.   style={{
63.     display: 'inline-block',
64.     backgroundColor: '#1e40af',
65.     color: 'white',
66.     padding: '0.75rem 2rem',
67.     textDecoration: 'none',
68.     borderRadius: '4px',
69.     marginRight: '1rem'
70.   }}
71. >
72.   Continue Shopping
73. </Link>
74.
```

```

75.   <Link
76.     href="/"
77.     style={{
78.       display: 'inline-block',
79.       color: '#1e40af',
80.       padding: '0.75rem 2rem',
81.       textDecoration: 'none'
82.     }}
83.   >
84.     Go to Home
85.   </Link>
86. </div>
87. </main>
88.
89. <Footer />
90. </div>
91. );
92. }

```

Step 2: Create app/payment/cancel/page.js

javascript

```

1.  "use client";
2.
3.  import Link from 'next/link';
4.  import Header from '@app/components/Header';
5.  import Footer from '@app/components/Footer';
6.
7.  export default function PaymentCancelPage() {
8.    return (
9.      <div>
10.     <Header />
11.
12.     <main style={{ padding: '4rem 2rem', textAlign: 'center', minHeight: '60vh' }}>
13.       <div style={{ maxWidth: '500px', margin: '0 auto' }}>
14.         <div style={{
15.           fontSize: '4rem',

```

```
16.     marginBottom: '1rem',
17.     color: '#dc2626'
18.   }}>
19.     ✖
20. </div>
21.
22. <h1 style={{ fontSize: '2rem', marginBottom: '1rem', color: '#dc2626' }}>
23.   Payment Cancelled
24. </h1>
25.
26. <p style={{ color: '#475569', marginBottom: '2rem', lineHeight: '1.6' }}>
27.   Your payment was cancelled. No charges were made to your account.
28. </p>
29.
30. <div style={{
31.   backgroundColor: '#f8f9fc',
32.   padding: '1.5rem',
33.   borderRadius: '8px',
34.   marginBottom: '2rem'
35. }}>
36.   <h3 style={{ marginBottom: '1rem' }}>Why was payment cancelled?</h3>
37.   <p style={{ color: '#475569' }}>
38.     Payments can be cancelled for several reasons:
39.   </p>
40.   <ul style={{ paddingLeft: '1.5rem', color: '#475569', textAlign: 'left' }}>
41.     <li>You clicked "Cancel" on the payment page</li>
42.     <li>Your bank declined the transaction</li>
43.     <li>The payment session timed out</li>
44.     <li>Insufficient funds</li>
45.   </ul>
46. </div>
47.
48. <div>
49.   <Link
50.     href="/cart"
51.     style={{
52.       display: 'inline-block',
```

```
53.     backgroundColor: '#1e40af',
54.     color: 'white',
55.     padding: '0.75rem 2rem',
56.     textDecoration: 'none',
57.     borderRadius: '4px',
58.     marginRight: '1rem'
59.   }}
60. >
61.   Return to Cart
62. </Link>
63.
64. <Link
65.   href="/products"
66.   style={{
67.     display: 'inline-block',
68.     color: '#1e40af',
69.     padding: '0.75rem 2rem',
70.     textDecoration: 'none'
71.   }}
72. >
73.   Browse Products
74. </Link>
75. </div>
76. </div>
77. </main>
78.
79. <Footer />
80. </div>
81. );
82. }
```

11.4 Testing the Payment Flow

Step 1: Add items to cart

Step 2: Go to checkout

Step 3: Fill in test details

Step 4: Click "Proceed to Payment"

Step 5: On PayHere sandbox page, use test card:

- Card: 4916 9300 0000 0000
- Expiry: 12/25 (any future date)
- CVV: 123
- OTP: 123456

Step 6: Complete payment

Step 7: You should be redirected to success page

Congratulations! You've integrated a real payment gateway!

11.5 Security Best Practices

DO NOT:

- Store Secret Key in frontend code
- Trust client-side data without verification
- Skip MD5 signature verification in production

DO:

- Verify all notifications on server
- Use environment variables for credentials
- Log all payment notifications
- Implement idempotency (prevent duplicate orders)

Lesson 11 Summary

- PayHere integration requires sending form data
- Always validate on server side
- Handle both success and cancel scenarios
- Test thoroughly in sandbox mode
- Never expose sensitive keys

Practice Exercises

1. Add order summary email after successful payment
2. Implement MD5 signature verification for security
3. Create order history page for logged-in users
4. Challenge: Add support for multiple payment gateways