

```
// Frontend logic for checkout using Stripe Payment Intents
const PUBLIC_KEY = 'REPLACE_WITH_PUBLISHABLE_KEY'; //
optional; server can inject
const niches = [
  { id: 'restaurants', name: 'Restaurants & Food Services', price:
29700 },
  { id: 'lawyers', name: 'Law Firms & Attorneys', price: 29700 },
  { id: 'dentists', name: 'Dental Practices', price: 29700 }
];
```

```
const nicheGrid = document.getElementById('niche-grid');
niches.forEach(n => {
  const el = document.createElement('div');
  el.className = 'niche card';
  el.dataset.id = n.id;
  el.innerHTML = `

### `${n.name}`</h3><p class="muted">500 verified contacts</p><strong>`${(n.price/100).toFixed(2)} </strong>`; el.addEventListener('click', () => startCheckout(n)); nicheGrid.appendChild(el); });


```

```
let stripe, elements, card;
async function startCheckout(niche){
  // Show checkout form
  document.getElementById('selected-title').textContent =
`${niche.name} — 500 leads`;
```

```
document.getElementById('checkout').classList.remove('hidden');
```

```
// Create Payment Intent on server to collect ephemeral  
clientSecret  
const resp = await fetch('/api/create-payment-intent', {  
  method: 'POST',  
  headers: {'Content-Type': 'application/json'},  
  body: JSON.stringify({ niche: niche.id, amount: niche.price })  
});  
const data = await resp.json();  
if (!data.clientSecret) { alert('Failed to start payment'); return; }
```

```
// Initialize Stripe if not already  
if (!stripe) {  
  stripe = Stripe(data.publishableKey || PUBLIC_KEY);  
  elements = stripe.elements();  
  card = elements.create('card');  
  card.mount('#card-element');  
}
```

```
const form = document.getElementById('checkout-form');  
form.onsubmit = async (e) => {  
  e.preventDefault();  
  document.getElementById('status').textContent = 'Processing  
payment...';  
  const email = document.getElementById('email').value;
```

```
const { error, paymentIntent } = await
stripe.confirmCardPayment(data.clientSecret, {
  payment_method: { card, billing_details: { email } }
});
if (error) {
  document.getElementById('status').textContent =
error.message;
  return;
}
// On success, inform server to attach metadata and deliver
leads
const finalize = await fetch('/api/finalize-order', {
  method: 'POST',
  headers: {'Content-Type': 'application/json'},
  body: JSON.stringify({ paymentIntentId: paymentIntent.id,
email, niche: niche.id })
});
const r = await finalize.json();
if (r.success) location.href = '/success.html';
else { document.getElementById('status').textContent =
r.message || 'Order failed' }
};
}
```