

```
<!-- index.html -->

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width,initial-scale=1" />

<title>LUMEXA TRADECONNECT</title>

<meta name="description" content="LUMEXA TradeConnect - AI P2P marketplace" />

<style>

/* compact styles for mobile-first UI */

:root{--bg:#041022;--accent:#06b6d4;--muted:#9fb0c7}

body{margin:0;font-family:Inter,system-ui;background:linear-gradient(#041022,#071827);color:#eaf6ff}

.wrap{max-width:980px;margin:12px auto;padding:12px}

.card{background:rgba(255,255,255,0.03);padding:12px;border-radius:12px;margin-bottom:12px}

.logo{width:56px;height:56px;border-radius:8px;background:linear-gradient(135deg,var(--accent),#0ea5a3);display:inline-flex;align-items:center;justify-content:center;font-weight:800;color:#041022}

.row{display:flex;justify-content:space-between;align-items:center}

.btn{padding:8px 12px;border-radius:10px;border:0;cursor:pointer}

.btn-primary{background:var(--accent);color:#041022}

input{padding:10px;border-radius:8px;border:1px solid rgba(255,255,255,0.03);width:100%;background:transparent;color:inherit}

@media(min-width:900px){ .layout{display:grid;grid-template-columns:1fr 360px;gap:12px} }

</style>

</head>
```

```
<body>

<div class="wrap">

<div style="display:flex;gap:12px;align-items:center;margin-bottom:8px">

<div class="logo">LT</div>

<div>

<h2 style="margin:0">LUMEXA TRADECONNECT</h2>

<div style="color:#9fb0c7;font-size:13px">AI-assisted global P2P marketplace</div>

</div>

</div>

<div class="layout">

<main>

<div class="card" id="walletCard">

<div class="row">

<div>

<div style="font-size:12px;color:#9fb0c7">Available balance</div>

<div id="balance" style="font-size:22px;font-weight:800">₦0.00</div>

<div id="userInfo" style="font-size:13px;color:#9fb0c7">Guest</div>

</div>

<div>

<button class="btn btn-primary" id="signinBtn">Sign in</button>

</div>

</div>

</div>

<div class="card">
```

```
<h3 style="margin:0 0 8px">Send Money</h3>
<form id="transferForm">
  <input id="to" placeholder="Recipient phone or email" required />
  <div style="display:flex;gap:8px;margin-top:8px">
    <input id="amount" placeholder="Amount" type="number" />
    <select id="currency"><option>NGN</option><option>USD</option></select>
  </div>
  <input id="note" placeholder="Note (optional)" style="margin-top:8px" />
  <div style="margin-top:8px;display:flex;gap:8px">
    <button class="btn btn-primary" type="submit">Send</button>
    <button class="btn" type="button" id="clearBtn">Clear</button>
  </div>
  <div id="transferStatus" style="margin-top:8px;color:#9fb0c7;font-size:13px"></div>
</form>
</div>
```

```
<div class="card">
  <h3 style="margin:0 0 8px">AI Assistant</h3>
  <div id="messages" style="min-height:120px;max-height:260px;overflow:auto;padding:6px;border-radius:8px;background:rgba(255,255,255,0.01)"></div>
  <div style="display:flex;gap:8px;margin-top:8px">
    <input id="chatInput" placeholder="Ask Lumexa — check scam, write listing, bargain..." />
    <button class="btn btn-primary" id="askBtn">Ask</button>
  </div>
</div>
```

```
<div class="card">

  <h3 style="margin:0 0 8px">Trading Signals</h3>
  <div id="signals" style="min-height:80px;color:#9fb0c7">Loading...</div>
</div>

</main>

<aside>

  <div class="card">

    <h4 style="margin:0 0 8px">Transactions</h4>
    <div id="txList" style="color:#9fb0c7">No transactions</div>
  </div>

<div class="card">

  <h4 style="margin:0 0 8px">Quick Actions</h4>
  <div style="display:flex;flex-direction:column;gap:6px">
    <button class="btn btn-primary" id="topup">Top up</button>
    <button class="btn" id="withdraw">Withdraw</button>
    <button class="btn" id="browse">Browse listings</button>
  </div>
</div>

<div class="card">
  <div style="font-size:13px;color:#9fb0c7">Version 1.0 · LUMEXA
  TRADECONNECT</div>
</div>
```

```
</aside>

</div>

<div style="text-align:center;color:#9fb0c7;margin-top:10px;font-size:13px">Replace
<code>BASE_API</code> with your backend URL after deployment</div>

</div>

<script>

const BASE_API = 'https://REPLACE_WITH_YOUR_BACKEND_URL'; // set this after you
deploy the backend

const state = { token: null, user: null, balance: 0, tx: [] };

function fmt(n,c='NGN'){ try { return new Intl.NumberFormat('en-
NG',{style:'currency',currency:c}).format(Number(n)); } catch(e){ return c+''
+Number(n).toFixed(2); } }

function render(){

  document.getElementById('balance').innerText = fmt(state.balance||0);

  document.getElementById('userInfo').innerText = state.user ?
(state.user.name||state.user.email) : 'Guest';

  const tx = document.getElementById('txList'); tx.innerHTML = state.tx.length ?
state.tx.map(t=>`<div style="padding:6px;border-
radius:6px;background:rgba(255,255,255,0.01);margin-bottom:6px">${t.to} —
${fmt(t.amount,t.currency)}</div>`).join("") : 'No transactions';

}

document.getElementById('signinBtn').addEventListener('click', async () => {
```

```
const email = prompt('Enter email for demo sign-in:');

if(!email) return;

// In production: call /api/auth/request-otp and /api/auth/verify-otp

state.user = { name: email.split('@')[0], email };

state.token = 'demo-token';

state.balance = 1500;

render();

alert('Demo sign-in complete. Implement server auth for production.');

});
```

```
document.getElementById('transferForm').addEventListener('submit', async (e) => {

e.preventDefault();

const to = document.getElementById('to').value.trim();

const amount = Number(document.getElementById('amount').value);

const currency = document.getElementById('currency').value;

const note = document.getElementById('note').value;

if(!to || !amount) { alert('Enter recipient and amount'); return; }

// optimistic

state.tx.push({ to, amount, currency, note, date: Date.now() }); state.balance -= amount;

render();

document.getElementById('transferStatus').innerText = 'Sending...';

try {

// PRODUCTION: call backend

// await fetch(` ${BASE_API}/api/transfer` , { method:'POST', headers:{'Content-Type':'application/json', 'Authorization': 'Bearer '+state.token}, body: JSON.stringify({ to, amount, currency, note }) });

await new Promise(r=>setTimeout(r,800));
```

```
document.getElementById('transferStatus').innerText = 'Sent (demo)';
} catch(e){
  document.getElementById('transferStatus').innerText = 'Failed';
}
});

document.getElementById('askBtn').addEventListener('click', async () => {
  const text = document.getElementById('chatInput').value.trim();
  if(!text) return;
  addMessage('You', text);
  addMessage('Lumexa', 'Thinking...');

  try {
    // PRODUCTION: call your backend API:
    // const res = await fetch(` ${BASE_API}/api/chat` , { method:'POST', headers:{'Content-Type':'application/json','Authorization':'Bearer '+state.token}, body: JSON.stringify({ message: text })});
    // const data = await res.json();

    await new Promise(r=>setTimeout(r,900));
    replaceLastRobot('Demo reply: integrate your backend /api/chat to get real AI answers.');
  } catch(e){
    replaceLastRobot('Error contacting backend. Check BASE_API.');
  }
});

function addMessage(who, text){
  const m = document.createElement('div'); m.style.marginBottom='8px';
```

```

m.innerHTML = `<strong>${who}</strong><div style="margin-top:4px">${escapeHtml(text)}</div>`;

document.getElementById('messages').appendChild(m);
document.getElementById('messages').scrollTop = 9999;

}

function replaceLastRobot(text){ const m =
document.getElementById('messages').lastChild; if(m) m.innerHTML =
`<strong>Lumexa</strong><div style="margin-top:4px">${escapeHtml(text)}</div>` ; }

function escapeHtml(s){ return
String(s).replace(/&/g,'&').replace(/</g,'<').replace(/>/g,'>'); }

async function loadSignals(){

const el = document.getElementById('signals'); el.innerText = 'Loading...';

try{

// PRODUCTION: fetch(` ${BASE_API}/api/signals`)

await new Promise(r=>setTimeout(r,400));

const sample = [{ text: 'BTC: Short-term buy opportunity' }, { text: 'ETH: Watch 1800
support' }];

el.innerHTML = sample.map(s=>`<div style="padding:6px;border-radius:8px;background:rgba(255,255,255,0.01);margin-bottom:6px">${s.text}</div>`).join("");

} catch(e){

el.innerText = 'Unable to load signals';

}

}

loadSignals();

render();

```

</script>

</body>

</html>