```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Speranza Saturday Market</title>
    <!-- Load Tailwind CSS for styling -->
    <script src="https://cdn.tailwindcss.com"></script>
    <!-- Use Rajdhani (Primary Sci-Fi) and Space Mono (Monospace Data) fonts -->
    <style>
        @import
url('https://fonts.googleapis.com/css2?family=Rajdhani:wght@400;600;700&family=Space+Mon
o:wght@400;700&display=swap');

        /* Changed body font to Rajdhani for a futuristic, angular look */
        body { font-family: 'Rajdhani', sans-serif; background-color: #1a202c; color: #e2e8f0; }

        /* Ensure monospaced elements (like Gamer Tags) use Space Mono for a data terminal
look */
        .font-mono { font-family: 'Space Mono', monospace; }

        /* Custom scrollbar for better visual on dark mode */
        ::-webkit-scrollbar { width: 8px; }
        ::-webkit-scrollbar-track { background: #2d3748; }
        ::-webkit-scrollbar-thumb { background: #4a5568; border-radius: 4px; }
        ::-webkit-scrollbar-thumb:hover { background: #718096; }

        /* Card changed to a slightly lighter dark-metal shade */
        .card { background-color: #2d3748; border: 1px solid #4a5568; }

        /* Primary Button: Sci-Fi Neon Cyan/Blue */
        .btn-primary {
            background-color: #00ffff;
            color: #1a202c; /* Dark text on bright button */
            font-weight: 700;
            transition: background-color 0.15s, transform 0.15s;
            box-shadow: 0 4px 6px -1px rgba(0, 255, 255, 0.3), 0 2px 4px -1px rgba(0, 255, 255,
0.1);
        }
        .btn-primary:hover { background-color: #00e5e5; transform: translateY(-1px); }

        /* Secondary Action Button (Mark Complete) */
        .btn-action {
            background-color: #a0522d; /* Sienna/Rust color */
```

```css
    color: #e2e8f0;
    padding: 0.5rem 1rem;
    border-radius: 0.5rem;
    font-weight: 600;
    transition: background-color 0.15s;
}
.btn-action:hover { background-color: #8b4513; }

/* Modal button style (Confirm/Cancel) */
.btn-confirm {
    padding: 0.75rem 1.5rem;
    border-radius: 0.75rem;
    font-weight: 700;
    transition: background-color 0.2s;
}

/* Input Field: Dark, high contrast border on focus */
.input-field { background-color: #1a202c; border: 1px solid #4a5568; color: #e2e8f0; }
.input-field:focus { border-color: #00ffff; box-shadow: 0 0 0 2px rgba(0, 255, 255, 0.5); }

.trade-grid { grid-template-columns: 1fr 10px 1fr; }
@media (min-width: 640px) {
    .trade-grid { grid-template-columns: 1fr 30px 1fr; }
}

/* Modal specific styling */
.modal-overlay {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.8);
    display: flex;
    justify-content: center;
    align-items: center;
    z-index: 50;
}

.modal-content {
    background-color: #1a202c;
    border: 2px solid #00ffff;
    padding: 2rem;
    border-radius: 1rem;
```

```css
        box-shadow: 0 0 20px rgba(0, 255, 255, 0.5);
        max-width: 90%;
        width: 400px;
        animation: fadeIn 0.3s ease-out;
    }

    @keyframes fadeIn {
        from { opacity: 0; transform: scale(0.95); }
        to { opacity: 1; transform: scale(1); }
    }
</style>
```
```html
<!-- Firebase Imports -->
<script type="module">
    import { initializeApp } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-app.js";
    import { getAuth, signInAnonymously, signInWithCustomToken, onAuthStateChanged } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-auth.js";
    import { getFirestore, doc, addDoc, onSnapshot, collection, query, orderBy, deleteDoc, Timestamp, setLogLevel, updateDoc, where } from "https://www.gstatic.com/firebasejs/11.6.1/firebase-firestore.js";

    // IMPORTANT: Set log level for debugging
    setLogLevel('error');

    // Global variables provided by the environment
    const appId = typeof __app_id !== 'undefined' ? __app_id : 'default-app-id';
    const firebaseConfig = typeof __firebase_config !== 'undefined' ?
JSON.parse(__firebase_config) : {};
    const initialAuthToken = typeof __initial_auth_token !== 'undefined' ? __initial_auth_token :
null;

    let db;
    let auth;
    let userId = null;
    let tradesCollectionRef;
    let isAuthReady = false;
    let currentView = 'live'; // 'live' or 'history'
    let tradeToComplete = null; // Store the ID of the trade being confirmed


    // --- Item Catalog for Search and Selection (COMPREHENSIVE LIST) ---
    const ARC_RAIDERS_ITEMS = [
        "Advanced ARC Powercell", "Advanced Electrical Components", "Advanced Mechanical
Components", "Agave",
```

"Air Freshener", "Alarm Clock", "Angled Grip II Blueprint", "Angled Grip III", "Anvil Blueprint",

"Anvil Splitter", "Antiseptic", "Apricot", "ARC Alloy", "ARC Circuitry", "ARC Coolant", "ARC Flex Rubber",

"ARC Motion Core", "ARC Performance Steel", "ARC Powercell", "ARC Synthetic Resin", "ARC Thermo Lining",

"Assorted Seeds", "Barrcade Kit Blueprint", "Bastion Cell", "Battery", "Bettina Blueprint", "Bicycle Pump",

"Blaze Grenade Blueprint", "Bloated Tuna Can", "Blue Light Stick", "Blown Fuses", "Bobcat",

"Bombardier Cell", "Breathtaking Snow Globe", "Broken Flashlight", "Broken Guidance System",

"Broken Handheld Radio", "Broken Taser", "Burletta Blueprint", "Burned ARC Circuitry", "Camera Lens",

"Candle Holder", "Canister", "Cat Bed", "Chemicals", "Coffee Pot", "Combat MK3 Blueprint",

"Combat MK3 Flanking Blueprint", "Compensator II Blueprint", "Compensator III", "Complex Gun Parts",

"Complex Gun Parts Blueprint", "Coolant", "Cooling Coil", "Cooling Fan", "Cracked Bioscanner",

"Crude Explosives", "Crumpled Plastic Bottle", "Damaged ARC Motion Core", "Damaged ARC Powercell",

"Damaged Fireball Burner", "Damaged Heat Sink", "Damaged Hornet Driver", "Damaged Rocketeer Driver",

"Damaged Wasp Driver", "Dart Board", "Defibrillator Blueprint", "Deflated Football", "Defibrillator",

"Degraded ARC Rubber", "Diving Googles", "Dog Collar", "Dried-Out ARC Resin", "Duct Tape",

"Durable Cloth", "Electrical Components", "Empty Wine Bottle", "Equalizer Blueprint", "Exodus Modules",

"Expired Pasta", "Expired Respirator", "Explosive Compound", "Explosive Mine Blueprint",

"Extended Barrel Blueprint", "Extended Light Magazine II", "Extended Light Magazine III",

"Extended Medium Mag II Blueprint", "Extended Medium Mag III", "Extended Shotgun Magazine II",

"Extended Shotgun Magazine III", "Fabric", "Faded Photograph", "Fertilizer", "Film Reel",

"Fine Wristwatch", "Fireball Burner", "Fried Motherboard", "Frying Pan", "Garlic Press",

"Green Light Stick Blueprint", "Great Mullein", "Headphones", "Heavy Gun Parts",

"Heavy Gun Parts Blueprint", "Hornet Driver", "Horizontal Grip", "Household Cleaner", "Hullcracker",

"Hullcracker Blueprint", "Humidifier", "Ice Cream Scooper", "IL Toro Blueprint", "Impure ARC Coolant",

"Industrial Battery", "Industrial Charger", "Industrial Magnet", "Jolt Mine Blueprint", "Jupiter Blueprint",

"Kinetic Converter", "Laboratory Reagents", "Lance's Mixtape (5th Edition)", "Leaper Pulse Unit",

"Lemon", "Light Bulb", "Light Gun Parts", "Light Gun Parts Blueprint", "Lightweight Stock",

"Looting MK3 Survivor Blueprint", "Lure Grenade Blueprint", "Magnet", "Magnetic Accelerator",

"Mechanical Components", "Medium Gun Parts", "Medium Gun Parts Blueprint", "Metal Brackets",

"Metal Parts", "Mod Components", "Moss", "Motor", "Mushroom", "Music Album", "Music Box",

"Muzzle Brake II Blueprint", "Muzzle Brake III", "Number Plate", "Oil", "Olives", "Ospris BP",

"Padded Stock", "Painted Box", "Plastic Parts", "Playing Cards", "Polluted Air Filter", "Pop Trigger",

"Portable TV", "Poster of Natural Wonders", "Pottery", "Power Cable", "Power Bank", "Power Rod",

"Prickly Pear", "Processor", "Projector", "Queen Reactor", "Radio", "Red Coral Jewelry",

"Red Light Stick Blueprint", "Remote Control", "Remote Raider Flare Blueprint", "Resin",

"Ripped Safety Vest", "Rocket Thruster", "Rocketeer Driver", "Roots", "Rope", "Rosary",

"Rubber Duck", "Rubber Pad", "Rubber Parts", "Ruined Accordion", "Ruined Augment", "Ruined Baton",

"Ruined Handcuffs", "Ruined Parachute", "Ruined Riot Shield", "Ruined Tactical Vest", "Rusted Bolts",

"Rusted Gear", "Rusted Shut Medical Kit", "Rusted Tools", "Rusty ARC Steel", "Sensors",

"Sentinel Firing Core", "Shotgun Choke II Blueprint", "Shotgun Choke III", "Shotgun Silencer",

"Showstopper", "Silencer I", "Silencer II Blueprint", "Silencer III", "Silver Teaspoon Set",

"Simple Gun Parts", "Smoke Grenade Blueprint", "Snap Hook", "Snitch Scanner", "Speaker Component",

"Spring Cushion", "Spotter Relay", "Stable Stock II Blueprint", "Stable Stock III Blueprint",

"Statuette", "Steel Spring", "Surveyor Vault", "Synthesized Fuel", "Syringe",

"Tactical Mk3 Defensive Blueprint", "Tactical MK3 Healing Blueprint", "Tagging Grenade",

"Tattered ARC Lining", "Tattered Clothes", "Tempest", "Thermostat", "Tick Pod", "Toaster",

"Torrente Blueprint", "Torn Blanket", "Torn Book", "Trigger Nade Blueprint", "Turbo Pump",

"Unusable Weapon", "Vase", "Venator Blueprint", "Very Comfortable Pillow", "Vertical Grip II Blueprint",

"Vertical Grip III", "Vita Shot", "Vita Spray", "Volcanic Rock", "Voltage Converter",

```javascript
            "Vulcano Blueprint", "Wasp Driver", "Water Filter", "Water Pump", "Wires", "Wolfpack
Blueprint",
            "Yellow Light Stick Blueprint"
        ].sort();

        // --- Utility Functions ---
        function showMessage(message, isError = false) {
            const messageBox = document.getElementById('message-box');
            messageBox.textContent = message;
            messageBox.className = `p-3 my-4 rounded-lg text-center ${isError ? 'bg-red-800
text-red-200' : 'bg-green-800 text-green-200'}`;
            messageBox.classList.remove('hidden');
            setTimeout(() => messageBox.classList.add('hidden'), 5000);
        }

        function createItemSelect(id) {
            const select = document.createElement('select');
            select.id = id;
            select.className = 'input-field w-full p-2.5 rounded-lg focus:outline-none focus:ring-2
focus:ring-cyan-500';

            const defaultOption = document.createElement('option');
            defaultOption.value = "";
            defaultOption.textContent = "Select Item or Blueprint...";
            defaultOption.disabled = true;
            defaultOption.selected = true;
            select.appendChild(defaultOption);

            ARC_RAIDERS_ITEMS.forEach(item => {
                const option = document.createElement('option');
                option.value = item;
                option.textContent = item;
                select.appendChild(option);
            });
            return select;
        }

        // --- Modal Functions ---
        function showCompletionModal(tradeId, giving, seeking) {
            tradeToComplete = tradeId; // Store the ID

            document.getElementById('modal-giving-item').textContent = `${giving.quantity}
${giving.item}`;
```

```javascript
        document.getElementById('modal-seeking-item').textContent = `${seeking.quantity}
${seeking.item}`;
        document.getElementById('completion-modal').classList.remove('hidden');
    }

    function hideCompletionModal() {
        tradeToComplete = null;
        document.getElementById('completion-modal').classList.add('hidden');
    }

    // --- View Management ---
    function switchView(viewName) {
        currentView = viewName;
        document.getElementById('live-tab').classList.remove('border-b-4', 'border-cyan-400');
        document.getElementById('history-tab').classList.remove('border-b-4',
'border-cyan-400');

        document.getElementById('live-feed-content').classList.add('hidden');
        document.getElementById('trade-history-content').classList.add('hidden');

        // Set active tab style
        document.getElementById(`${viewName}-tab`).classList.add('border-b-4',
'border-cyan-400');

        // Show correct content
        document.getElementById(`${viewName}-content`).classList.remove('hidden');

        // Trigger listener for the new view
        if (viewName === 'live') {
            setupLiveListener();
        } else {
            setupHistoryListener();
        }
    }

    // --- Database Setup and Authentication ---
    async function initializeFirebase() {
        try {
            const app = initializeApp(firebaseConfig);
            auth = getAuth(app);
            db = getFirestore(app);

            const collectionPath = `/artifacts/${appId}/public/data/arcraiders_trades`;
            tradesCollectionRef = collection(db, collectionPath);
```

```javascript
        if (initialAuthToken) {
            await signInWithCustomToken(auth, initialAuthToken);
        } else {
            await signInAnonymously(auth);
        }

        onAuthStateChanged(auth, (user) => {
            if (user) {
                userId = user.uid;
                document.getElementById('post-form').classList.remove('hidden');
                document.getElementById('user-id-display').textContent = `Your ID: ${userId}`;
            } else {
                userId = null;
            }
            isAuthReady = true;
            // Start on the live feed
            switchView('live');
        });
    } catch (e) {
        console.error("Error initializing Firebase:", e);
        showMessage("Failed to connect to the database. Check console for details.", true);
    }
}

// --- Trade Management ---

// Listener for Pending Trades (Live Feed)
function setupLiveListener() {
    if (!isAuthReady || !db) return;

    // Query for trades with status 'pending', ordered by timestamp descending
    const q = query(tradesCollectionRef, where("status", "==", "pending"),
orderBy("timestamp", "desc"));

    onSnapshot(q, (snapshot) => {
        const trades = [];
        snapshot.forEach(doc => {
            trades.push({ id: doc.id, ...doc.data() });
        });
        renderLiveTrades(trades);
    }, (error) => {
        console.error("Error listening to live trades:", error);
        showMessage("Error fetching live trades. Please refresh.", true);
```

```
        });
    }

    // Listener for Accepted Trades (Trade Archive)
    function setupHistoryListener() {
        if (!isAuthReady || !db) return;

        // Query for trades with status 'accepted', ordered by acceptedAt descending (most
recent first)
        const q = query(tradesCollectionRef, where("status", "==", "accepted"),
orderBy("acceptedAt", "desc"));

        onSnapshot(q, (snapshot) => {
            const trades = [];
            snapshot.forEach(doc => {
                trades.push({ id: doc.id, ...doc.data() });
            });
            renderHistoryList(trades);
        }, (error) => {
            console.error("Error listening to trade history:", error);
            showMessage("Error fetching trade history. Please refresh.", true);
        });
    }


    async function postNewTrade(event) {
        event.preventDefault();
        if (!userId) {
            showMessage("Authentication not ready. Please wait a moment.", true);
            return;
        }

        const form = event.target;
        const givingItem = form.elements['givingItem'].value;
        const givingQuantity = parseInt(form.elements['givingQuantity'].value);
        const seekingItem = form.elements['seekingItem'].value;
        const seekingQuantity = parseInt(form.elements['seekingQuantity'].value);
        const platform = form.elements['platform'].value;
        const gamerTag = form.elements['gamerTag'].value.trim();

        if (!givingItem || isNaN(givingQuantity) || givingQuantity <= 0 || !seekingItem ||
isNaN(seekingQuantity) || seekingQuantity <= 0 || !platform || !gamerTag) {
            showMessage("Please fill out all trade details correctly.", true);
            return;
```

```
            }

            try {
                await addDoc(tradesCollectionRef, {
                    userId: userId,
                    platform: platform,
                    gamerTag: gamerTag,
                    givingItem: givingItem,
                    givingQuantity: givingQuantity,
                    seekingItem: seekingItem,
                    seekingQuantity: seekingQuantity,
                    status: 'pending', // New status field
                    timestamp: Timestamp.now()
                });

                form.reset();
                form.elements['givingItem'].value = "";
                form.elements['seekingItem'].value = "";

                showMessage("Trade posted successfully!");
            } catch (e) {
                console.error("Error adding document: ", e);
                showMessage("Failed to post trade. Check console for error.", true);
            }
        }

        async function deleteTrade(tradeId) {
            if (!userId) return;

            try {
                const tradeDocRef = doc(db, tradesCollectionRef.path, tradeId);
                await deleteDoc(tradeDocRef);
                showMessage("Trade removed.");
            } catch (e) {
                console.error("Error removing document: ", e);
                showMessage("Failed to remove trade. Only the poster can delete it.", true);
            }
        }

        // Global function to trigger the completion process (after modal confirmation)
        async function confirmCompletion() {
            if (!userId || !tradeToComplete) {
                showMessage("Invalid trade or user state.", true);
                return;
```

```
          }

          try {
              const tradeDocRef = doc(db, tradesCollectionRef.path, tradeToComplete);
              await updateDoc(tradeDocRef, {
                  status: 'accepted',
                  acceptedAt: Timestamp.now() // Record when the trade was completed
              });

              hideCompletionModal();
              showMessage("Trade marked as completed and archived!", false);
          } catch (e) {
              console.error("Error completing document: ", e);
              showMessage("Failed to complete trade. Check console for error.", true);
          }
      }

      // Wrapper function to show the modal (called from the HTML button)
      function triggerCompleteTrade(tradeId, givingItem, givingQuantity, seekingItem,
seekingQuantity) {
          showCompletionModal(tradeId,
              { item: givingItem, quantity: givingQuantity },
              { item: seekingItem, quantity: seekingQuantity }
          );
      }

      // --- Rendering Logic ---

      function renderLiveTrades(trades) {
          const tradeListElement = document.getElementById('live-trade-list');
          const filterText = document.getElementById('search-item-live').value.toLowerCase();
          tradeListElement.innerHTML = '';

          const filteredTrades = trades.filter(trade =>
              trade.givingItem.toLowerCase().includes(filterText) ||
              trade.seekingItem.toLowerCase().includes(filterText)
          );

          if (filteredTrades.length === 0) {
              tradeListElement.innerHTML = `<div class="text-center p-8 text-gray-500">
                  No live trades found matching your search.
              </div>`;
              return;
          }
```

```javascript
filteredTrades.forEach(trade => {
    const tradeElement = document.createElement('div');
    tradeElement.className = 'card p-4 rounded-xl mb-4 shadow-lg flex flex-col space-y-3';
    tradeElement.innerHTML = `
        <div class="flex justify-between items-start">
            <span class="text-sm font-semibold px-3 py-1 rounded-full text-zinc-900
${trade.platform === 'PC' ? 'bg-cyan-400' : trade.platform === 'PS5' ? 'bg-blue-500' :
'bg-orange-500'}">
                ${trade.platform}
            </span>

            <div class="flex space-x-2">
                ${trade.userId === userId ? `
                  <button
                    onclick="window.triggerCompleteTrade(
                        '${trade.id}',
                        '${trade.givingItem}',
                        ${trade.givingQuantity},
                        '${trade.seekingItem}',
                        ${trade.seekingQuantity}
                    )"
                    class="btn-action text-xs leading-none">
                    Mark Completed
                  </button>
                  <button onclick="window.deleteTrade('${trade.id}')" class="text-red-500
hover:text-red-400 transition-colors text-lg" aria-label="Delete Trade">
                    &times;
                  </button>` : ''}
            </div>
        </div>

        <div class="trade-grid items-center gap-2">
            <!-- Giving: Rust/Copper accent for the item being offered -->
            <div class="text-center p-3 rounded-lg bg-orange-900/40">
                <p class="text-xl font-bold
text-orange-300">${trade.givingQuantity.toLocaleString()}</p>
                <p class="text-sm text-orange-200">${trade.givingItem}</p>
                <p class="text-xs text-gray-400 font-medium mt-1">GIVING</p>
            </div>

            <!-- Arrow/Swap Icon: Neon Cyan Accent -->
            <div class="text-center">
```

```
                <svg xmlns="http://www.w3.org/2000/svg" class="h-6 w-6 text-cyan-400
mx-auto" viewBox="0 0 20 20" fill="currentColor">
                    <path fill-rule="evenodd" d="M10 2a1 1 0 011 1v3.757l3.182-3.182a1 1 0
111.414 1.414L12.414 8h3.757a1 1 0 110 2h-3.757l3.182 3.182a1 1 0 11-1.414 1.414L11
11.243V15a1 1 0 11-2 0v-3.757l-3.182 3.182a1 1 0 11-1.414-1.414L7.586 10H3.829a1 1 0
110-2h3.757L4.364 4.818a1 1 0 111.414-1.414L9 6.757V3a1 1 0 011-1z" clip-rule="evenodd" />
                </svg>
            </div>

            <!-- Seeking: Electric Blue accent for the desired item -->
            <div class="text-center p-3 rounded-lg bg-blue-900/40">
                <p class="text-xl font-bold
text-blue-300">${trade.seekingQuantity.toLocaleString()}</p>
                <p class="text-sm text-blue-200">${trade.seekingItem}</p>
                <p class="text-xs text-gray-400 font-medium mt-1">SEEKING</p>
            </div>
        </div>

        <div class="pt-2 border-t border-gray-700 mt-2">
            <p class="text-xs text-gray-400">Gamer Tag / Embark ID:</p>
            <p class="text-lg font-mono text-cyan-300 break-words">${trade.gamerTag}</p>
            <p class="text-xs text-gray-500 mt-1">Posted: ${new
Date(trade.timestamp.seconds * 1000).toLocaleDateString()}</p>
        </div>
        `;
        tradeListElement.appendChild(tradeElement);
    });
}

function renderHistoryList(trades) {
    const historyListElement = document.getElementById('history-trade-list');
    const filterText = document.getElementById('search-item-history').value.toLowerCase();
    historyListElement.innerHTML = '';

    const filteredTrades = trades.filter(trade =>
        trade.givingItem.toLowerCase().includes(filterText) ||
        trade.seekingItem.toLowerCase().includes(filterText)
    );

    if (filteredTrades.length === 0) {
        historyListElement.innerHTML = `<div class="text-center p-8 text-gray-500">
            The Trade Archive is empty. Complete a trade to establish an item's value!
        </div>`;
        return;
```

```
        }

        filteredTrades.forEach(trade => {
            const acceptedDate = trade.acceptedAt ? new Date(trade.acceptedAt.seconds *
1000).toLocaleDateString() : 'N/A';

            const tradeElement = document.createElement('div');
            // Muted/Archived look for history trades
            tradeElement.className = 'card p-4 rounded-xl mb-4 shadow-inner flex flex-col
space-y-3 bg-gray-800/50 border-gray-700';
            tradeElement.innerHTML = `
                <div class="flex justify-between items-center text-gray-400">
                    <span class="text-xs font-semibold px-2 py-0.5 rounded text-gray-900
bg-gray-400">
                        ARCHIVED
                    </span>
                    <p class="text-xs">Completed: ${acceptedDate}</p>
                </div>

                <div class="trade-grid items-center gap-2">
                    <!-- Giving side (muted orange) -->
                    <div class="text-center p-3 rounded-lg bg-orange-900/20 border
border-orange-700/50">
                        <p class="text-lg font-bold
text-orange-400">${trade.givingQuantity.toLocaleString()}</p>
                        <p class="text-sm text-orange-300">${trade.givingItem}</p>
                    </div>

                    <!-- Arrow/Swap Icon: Grayed out -->
                    <div class="text-center">
                        <svg xmlns="http://www.w3.org/2000/svg" class="h-5 w-5 text-gray-500
mx-auto" viewBox="0 0 20 20" fill="currentColor">
                            <path fill-rule="evenodd" d="M10 2a1 1 0 011 1v3.757l3.182-3.182a1 1 0
011.414 1.414L12.414 8h3.757a1 1 0 110 2h-3.757l3.182 3.182a1 1 0 11-1.414 1.414L11
11.243V15a1 1 0 11-2 0v-3.757l-3.182 3.182a1 1 0 11-1.414-1.414L7.586 10H3.829a1 1 0
110-2h3.757L4.364 4.818a1 1 0 111.414-1.414L9 6.757V3a1 1 0 011-1z" clip-rule="evenodd" />
                        </svg>
                    </div>

                    <!-- Seeking side (muted blue) -->
                    <div class="text-center p-3 rounded-lg bg-blue-900/20 border
border-blue-700/50">
                        <p class="text-lg font-bold
text-blue-400">${trade.seekingQuantity.toLocaleString()}</p>
```

```
                <p class="text-sm text-blue-300">${trade.seekingItem}</p>
              </div>
            </div>

            <div class="pt-2 border-t border-gray-800 mt-2 text-center">
                <p class="text-xs text-gray-500 font-mono">Trader ID:
${trade.userId.substring(0, 8)}... | Platform: ${trade.platform}</p>
            </div>
          `;
          historyListElement.appendChild(tradeElement);
      });
    }


    // --- Initialization on Load ---
    document.addEventListener('DOMContentLoaded', () => {
      initializeFirebase();

      // Populate Item Selects

document.getElementById('givingItemContainer').appendChild(createItemSelect('givingItem'));

document.getElementById('seekingItemContainer').appendChild(createItemSelect('seekingItem'
));

      // Attach event listeners
      document.getElementById('post-form').addEventListener('submit', postNewTrade);

      // Expose global functions for HTML buttons
      window.deleteTrade = deleteTrade;
      window.triggerCompleteTrade = triggerCompleteTrade; // Show modal
      window.confirmCompletion = confirmCompletion; // Execute completion (called from
modal)
      window.hideCompletionModal = hideCompletionModal; // Hide modal (called from
modal)
      window.switchView = switchView; // Expose switchView

      // Setup search filtering for live and history views
      document.getElementById('search-item-live').addEventListener('input', () => {
        if (currentView === 'live') setupLiveListener();
      });
      document.getElementById('search-item-history').addEventListener('input', () => {
        if (currentView === 'history') setupHistoryListener();
      });
```

```html
        });
    </script>
</head>
<body class="p-4 sm:p-8">

    <div class="max-w-4xl mx-auto">
        <header class="text-center mb-8 p-4 rounded-xl shadow-2xl bg-[#0b1424] border
border-cyan-400/30">
            <h1 class="text-4xl font-extrabold text-cyan-400 tracking-wider uppercase">
                Speranza Saturday Market 🛠️
            </h1>
            <p class="text-sm text-gray-400 mt-1">
                A community-driven marketplace built by Raiders, for Raiders, celebrating the shared
spirit of Speranza.
            </p>
            <p id="user-id-display" class="text-xs mt-2 text-gray-500 font-mono"></p>
        </header>

        <!-- Tab Navigation -->
        <div class="flex border-b border-gray-700 mb-6">
            <button id="live-tab" onclick="switchView('live')" class="p-3 text-lg font-bold
tracking-wider text-cyan-300 hover:text-cyan-200 transition duration-150">
                LIVE FEED (PENDING)
            </button>
            <button id="history-tab" onclick="switchView('history')" class="p-3 text-lg font-bold
tracking-wider text-gray-400 hover:text-gray-200 transition duration-150">
                TRADE ARCHIVE (VALUE)
            </button>
        </div>

        <!-- Message Box -->
        <div id="message-box" class="hidden" role="alert"></div>

        <!-- Post New Trade Card (Always visible) -->
        <div class="card p-6 rounded-xl shadow-xl mb-8">
            <h2 class="text-2xl font-bold mb-4 text-cyan-400">Post a New Trade</h2>
            <form id="post-form" class="space-y-4 hidden">
                <div class="grid grid-cols-1 sm:grid-cols-3 gap-4">
                    <!-- Platform -->
                    <div>
                        <label for="platform" class="block text-sm font-medium mb-1
text-gray-300">Your Platform</label>
                        <select id="platform" required class="input-field w-full p-2.5 rounded-lg
focus:outline-none focus:ring-2 focus:ring-cyan-500">
```

```html
                <option value="" disabled selected>Select Platform...</option>
                <option value="PC">PC (Steam/Epic)</option>
                <option value="PS5">PlayStation 5</option>
                <option value="Xbox">Xbox Series X/S</option>
            </select>
        </div>
        <!-- Gamer Tag -->
        <div class="sm:col-span-2">
            <label for="gamerTag" class="block text-sm font-medium mb-1 text-gray-300">Your Embark ID / Gamer Tag</label>
            <input type="text" id="gamerTag" required placeholder="e.g., RaiderName#1234" class="input-field w-full p-2.5 rounded-lg focus:outline-none focus:ring-2 focus:ring-cyan-500">
        </div>
    </div>

    <div class="grid grid-cols-1 md:grid-cols-2 gap-4">
        <!-- GIVING SECTION -->
        <fieldset class="p-4 border border-orange-500/50 rounded-lg">
            <legend class="text-md font-semibold text-orange-400 px-2">I am GIVING</legend>
            <div class="flex space-x-2 mt-2">
                <div id="givingItemContainer" class="w-2/3">
                    <!-- Item select will be injected here -->
                </div>
                <input type="number" id="givingQuantity" required min="1" placeholder="Qty (e.g., 100)" class="input-field w-1/3 p-2.5 rounded-lg text-center focus:outline-none focus:ring-2 focus:ring-orange-500">
            </div>
        </fieldset>

        <!-- SEEKING SECTION -->
        <fieldset class="p-4 border border-blue-500/50 rounded-lg">
            <legend class="text-md font-semibold text-blue-400 px-2">I am SEEKING</legend>
            <div class="flex space-x-2 mt-2">
                <div id="seekingItemContainer" class="w-2/3">
                    <!-- Item select will be injected here -->
                </div>
                <input type="number" id="seekingQuantity" required min="1" placeholder="Qty (e.g., 1)" class="input-field w-1/3 p-2.5 rounded-lg text-center focus:outline-none focus:ring-2 focus:ring-blue-500">
            </div>
        </fieldset>
```

```html
        </div>

        <button type="submit" class="btn-primary w-full p-3 rounded-xl font-bold
tracking-wider shadow-md hover:shadow-lg">
            POST TRADE DATA
        </button>
    </form>
</div>


<!-- LIVE FEED CONTENT -->
<div id="live-feed-content">
    <div class="mb-6">
        <input type="text" id="search-item-live" placeholder="Search live trades by item
name..." class="input-field w-full p-3 rounded-lg focus:outline-none focus:ring-2
focus:ring-cyan-500 shadow-inner">
    </div>
    <div id="live-trade-list">
        <div class="text-center p-8 text-gray-500">
            Loading trades...
        </div>
    </div>
</div>


<!-- TRADE HISTORY CONTENT -->
<div id="trade-history-content" class="hidden">
    <div class="mb-6">
        <p class="text-sm text-gray-400 mb-4">
            The Trade Archive shows recently completed exchanges, providing a snapshot of
the current market value for items.
        </p>
        <input type="text" id="search-item-history" placeholder="Search archive for completed
exchanges..." class="input-field w-full p-3 rounded-lg focus:outline-none focus:ring-2
focus:ring-cyan-500 shadow-inner">
    </div>
    <div id="history-trade-list">
        <div class="text-center p-8 text-gray-500">
            Loading archive...
        </div>
    </div>
</div>
</div>
```

```html
<!-- Confirmation Modal -->
<div id="completion-modal" class="modal-overlay hidden">
    <div class="modal-content">
        <h3 class="text-2xl font-bold text-cyan-400 mb-4 tracking-wider">CONFIRM TRANSACTION</h3>
        <p class="text-gray-300 mb-6">
            Are you sure this trade was successfully completed? Marking it as **accepted** will move the data to the Trade Archive.
        </p>

        <div class="p-4 rounded-lg bg-gray-800 border border-cyan-600/50 mb-6">
            <p class="text-sm font-semibold text-gray-400 mb-2">Trade Details:</p>
            <p class="text-orange-300 font-bold">GIVING: <span id="modal-giving-item"></span></p>
            <p class="text-blue-300 font-bold">SEEKING: <span id="modal-seeking-item"></span></p>
        </div>

        <div class="flex justify-end space-x-4">
            <button onclick="window.hideCompletionModal()" class="btn-confirm bg-gray-600 hover:bg-gray-500 text-white">
                Cancel
            </button>
            <button onclick="window.confirmCompletion()" class="btn-confirm bg-green-600 hover:bg-green-500 text-white">
                CONFIRM & ARCHIVE
            </button>
        </div>
    </div>
</div>

</body>
</html>
```