🧪 KYC Testing Guide: UnifiedKycAgent

This guide will help you test the `unifiedKycAgent` to ensure it works correctly with your Chainlink Functions and GetWalletKYC.sol contract.

📋 Prerequisites Checklist

Before testing, ensure you have completed:

- ☐ ✅ Supabase database with `wallet_kyc` table
- ☐ ✅ DON hosted secrets uploaded (`node upload-secrets.js`)
- ☐ ✅ GetWalletKYC.sol contract deployed
- ☐ ✅ Chainlink Functions subscription with sufficient LINK
- ☐ ✅ Test wallet address added to Supabase database
- ☐ ✅ Environment variables configured

🔧 Step 1: Environment Setup

1.1 Copy Environment File

```
cp .env.local.example .env.local
```

1.2 Update .env.local with Your Values

```
# After running upload-secrets.js
DON_HOSTED_SECRETS_VERSION=YOUR_ACTUAL_VERSION_NUMBER

# After deploying GetWalletKYC.sol
NEXT_PUBLIC_KYC_CONTRACT_AVALANCHE=0xYourDeployedContractAddress

# Your Chainlink subscription ID
CHAINLINK_SUBSCRIPTION_ID=YOUR_SUBSCRIPTION_ID
```

1.3 Verify Supabase Database

Ensure your `wallet_kyc` table has this structure:

```
CREATE TABLE wallet_kyc (
    id SERIAL PRIMARY KEY,
    wallet_address TEXT UNIQUE NOT NULL,
    is_kyc_verified BOOLEAN DEFAULT FALSE,
    kyc_level INTEGER DEFAULT 1,
    kyc_status TEXT DEFAULT 'pending', -- 'pending', 'approved',
```

```
  'rejected'
      nft_token_id INTEGER,
      created_at TIMESTAMP DEFAULT NOW(),
      updated_at TIMESTAMP DEFAULT NOW()
  );
```

🧪 Step 2: Test Scenarios

Test Scenario 1: 🚫 No Contract Deployed (Expected Failure)

Purpose: Verify system correctly handles missing smart contract

Setup:

- Use a fake contract address in .env.local
- Or don't deploy the contract yet

Expected Results:

```
✅ Wallet Connection
🔄 NFT Access Check → ❌ Smart contract not deployed at 0x... Please deploy GetWalletKYC.sol first.
```

✅ Success Criteria:

- ☐ NFT Access Check step fails with proper error message
- ☐ No fake success messages
- ☐ Clear instruction to deploy contract

Test Scenario 2: 📋 Database Not Ready (Expected Failure)

Purpose: Verify system handles missing database records

Setup:

- Deploy your GetWalletKYC.sol contract
- Update contract address in .env.local
- DON'T add your wallet to Supabase database

Expected Results:

```
✅ Wallet Connection
✅ NFT Access Check → No existing NFT found via Chainlink
❌ Database Verification → KYC not verified via Chainlink Functions
```

✅ Success Criteria:

- ☐ NFT check completes (no existing NFT)
- ☐ Database verification fails with proper error
- ☐ No fake "KYC Level 1 verified" messages

Test Scenario 3: 🎯 Full Success Flow (Expected Success)

Purpose: Test complete KYC flow with all components ready

Setup:

1. Deploy GetWalletKYC.sol contract ✅
2. Update contract address in .env.local ✅
3. Add your test wallet to Supabase:

```
INSERT INTO wallet_kyc (wallet_address, is_kyc_verified, kyc_level,
kyc_status)
VALUES ('0xYourTestWalletAddress', true, 2, 'approved');
```

4. Ensure Chainlink subscription has sufficient LINK ✅

Expected Results:

```
✅ Wallet Connection
✅ NFT Access Check → No existing NFT found via Chainlink
✅ Database Verification → ✅ KYC Level 2 verified via Chainlink
✅ Chainlink Functions → Chainlink request submitted
🔄 NFT Minting → Waiting for Chainlink NFT minting...
[After 2-5 minutes]
✅ NFT Minting → ✅ NFT Minted! Token ID: X
✅ Access Granted → KYC verification complete
```

✅ **Success Criteria:**

- ☐ All steps complete successfully
- ☐ Real transaction hash appears
- ☐ NFT is actually minted on blockchain
- ☐ "Proceed to Questionnaire" button appears
- ☐ Can verify NFT in wallet/block explorer

Test Scenario 4: 🔄 Already Has NFT (Expected Skip)

Purpose: Test system behavior when NFT already exists

Setup:

- Use wallet that already has KYC NFT from previous test

- Or manually mint NFT to test wallet

Expected Results:

```
✅ Wallet Connection
✅ NFT Access Check → ✅ NFT Found - Level X
✅ Database Verification → Verified via Chainlink
✅ Chainlink Functions → Already verified
✅ NFT Minting → Token ID: X
✅ Access Granted → KYC verification complete
```

✅ **Success Criteria:**

- ☐ All steps marked complete immediately
- ☐ No redundant Chainlink calls
- ☐ Shows existing token ID
- ☐ "Proceed to Questionnaire" button appears

🛠️ Step 3: Debugging Tools

3.1 Browser Console Logs

Open browser console to see detailed logs:

```
// Expected logs for successful flow:
🔗 Attempting to check KYC status via deployed smart contract...
📋 Contract Address: 0xYourContractAddress
✅ Contract is deployed! HasAccess: false, Level: 0
🔗 Attempting to trigger Chainlink Functions for database verification...
🔗 Attempting to trigger Chainlink Functions for NFT minting...
✅ Smart contract found! Calling requestKYCVerification...
📤 Transaction submitted: 0xTransactionHash
🔗 View on explorer: https://testnet.snowtrace.io/tx/0xTransactionHash
```

3.2 Transaction Verification

Check your transactions on Avalanche Fuji testnet:

- **Snowtrace:** https://testnet.snowtrace.io/tx/YOUR_TX_HASH
- Look for `KYCVerificationRequested` event
- Look for `NFTMinted` event after 2-5 minutes

3.3 Database Verification

Check if NFT info is updated in Supabase:

```
SELECT * FROM wallet_kyc WHERE wallet_address = '0xYourWalletAddress';
```

❌ Common Issues & Solutions

Issue 1: "Contract not deployed" Error

Problem: Contract address is wrong or contract not deployed **Solution:**

1. Verify contract deployed on Avalanche Fuji
2. Update `NEXT_PUBLIC_KYC_CONTRACT_AVALANCHE` in .env.local
3. Restart development server

Issue 2: "Database verification failed"

Problem: Wallet not in Supabase or wrong status **Solution:**

1. Add wallet to database with `is_kyc_verified=true` and `kyc_status='approved'`
2. Check wallet address case sensitivity
3. Verify Supabase connection

Issue 3: "Transaction submitted but NFT not minted"

Problem: Chainlink Functions request failed or insufficient LINK **Solution:**

1. Check Chainlink subscription balance
2. Verify DON hosted secrets are correct
3. Wait 5-10 minutes for processing
4. Check contract events on Snowtrace

Issue 4: Steps showing fake success

Problem: Code is using local checks instead of Chainlink **Solution:**

1. Verify you're using the updated `unifiedKycAgent.ts`
2. Check contract address in constants.ts
3. Clear browser cache and restart

✅ Final Verification Checklist

After testing, verify:

- ☐ **No Fake Data:** All success messages come from real blockchain/database calls
- ☐ **Proper Error Handling:** Appropriate errors when contract/database not ready
- ☐ **Real Transactions:** Actual transaction hashes and NFT minting
- ☐ **UI Flow:** Proper step progression and completion indicators
- ☐ **Blockchain Verification:** NFT exists in wallet and on block explorer

---

🎯 Expected Timeline

Phase	Expected Duration	What Happens
Wallet Connection	Instant	Connect wallet
NFT Access Check	2-5 seconds	Check blockchain for existing NFT
Database Verification	1-3 seconds	Query Supabase via Chainlink
Chainlink Functions	10-30 seconds	Submit transaction
NFT Minting	2-5 minutes	Wait for Chainlink processing
Access Granted	Instant	Show completion

Total Expected Time: 3-6 minutes for complete flow

📞 Support

If tests fail:

1. Check browser console for detailed error logs
2. Verify all environment variables are correct
3. Ensure Chainlink subscription has sufficient LINK
4. Check Snowtrace for transaction details
5. Verify Supabase database structure and data

Happy Testing! 🚀