Interview

Project Goal Summary
Your goal is to write code to subscribe (and unsubscribe) to real-time solana token wallet balances and display the data. This balance data for a certain token may be displayed in more than one component at the same time.

Libraries needed:
@solana/web3.js
@solana/spl-token

Wallet address to use:
5QRZKZ65CuQPu3XwKaNnK9L2ASuzvrz4NGpAUrWEDoQm

Token addresses to listen to:
BONK: DezXAZ8z7PnrnRJjz3wXBoRgixCa6xjnB7YaB1pPB263
  Decimals: 5
WIF: 21AErpiB8uSb94oQKRcwuHqyHF93njAxBSbdUrpupump
  Decimals: 6

Connect to RPC:
```
import { Connection, PublicKey } from '@solana/web3.js';
const RPC_URL = 'https://gene-v4mswe-fast-mainnet.helius-rpc.com';
const SOLANA_CONNECTION = new Connection(RPC_URL);

const LAMPORT_PER_SOL = 1000000000;
```

Get SOL balance:
```
const walletPublicKey = new PublicKey(WALLET_ADDRESS);
SOLANA_CONNECTION.onAccountChange(
  walletPublicKey,
  (info) => {
    console.log('SOL in lamports', info.lamports);
  },
  { commitment: 'processed' }
);
```

Continued next page

Get SPL Token Balance:
Function to get associated token address:

```
import { getAssociatedTokenAddress } from '@solana/spl-token';

// get associated token address

SOLANA_CONNECTION.onAccountChange(
  ASSOCIATED_TOKEN_ADDRESS,
  (info) => {
    const d = info.data;
    if (d.length === 165 || d.length === 182) {
     const low = d.readUInt32LE(64);
     const high = d.readUInt32LE(68);
     const raw = high * 2 ** 32 + low;
     const divisor = 10 ** (DECIMALS_HERE);
     const balance = raw / divisor;
    }
  },
  { commitment: 'processed' }
);
```