Project Summary: LinkedIn to Zoho Recruit Integration
Project Overview
Goal: Build an automated system to extract LinkedIn profile data from screenshots and create candidates in Zoho Recruit, bypassing expensive third-party tools like Make.com.
Tech Stack:

Frontend: Tempo Labs (no-code platform)
Backend: Node.js/Express on Vercel
APIs: Google Vision (OCR), Claude AI (parsing), Zoho Recruit (CRM)
Current Status: 95% complete - everything works except automatic token refresh

What Works ✅

File Upload: Multipart form upload from frontend to backend
OCR Extraction: Google Vision API successfully extracts text (4000+ characters)
AI Parsing: Claude accurately parses profile details into structured JSON
Zoho Integration: Creates candidates successfully when valid access token exists
Manual Token Refresh: Using Postman to refresh tokens works perfectly

The Challenge 🚫
Long-term Operation: The app needs to run for months without manual intervention, but:

Access tokens expire after 1 hour
Automatic token refresh returns "No access token received"
Manual refresh via Postman works, but automatic refresh fails

Solutions Attempted
❌ Failed Attempts:

Initial Implementation: Sent refresh request as JSON instead of form-encoded

Error: 500 Internal Server Error with HTML response
Fix: Changed to URLSearchParams and form-encoded content type

Missing Functions: getUserCredentials and createCandidate were missing

Fix: Removed unnecessary calls and added missing methods

Class vs Module: Code was importing class as module

Fix: Changed to instantiate class properly

✅ Successful Fixes:

Converted refresh request from JSON to form-encoded format
Properly instantiated ZohoService class
Added token management to handle missing access tokens

Current Issue
The refresh token request appears successful (returns 200 status) but:

Response contains no access_token field
Might be returning HTML error page instead of JSON
Same credentials work perfectly in Postman

SYMPTOMS:

1. Request returns 200 status (appears successful)
2. Response.data exists but has no access_token field
3. Might be HTML instead of JSON (not confirmed)
4. Same refresh_token works in Postman

POSTMAN SUCCESS:

- Method: POST
- URL: https://accounts.zoho.in/oauth/v2/token
- Body: x-www-form-urlencoded with same parameters
- Returns: Valid JSON with access_token

ENVIRONMENT VARIABLES (confirmed set):

- ZOHO_REFRESH_TOKEN: 1000.509534c4d7f1b2e...
- ZOHO_CLIENT_ID: 1000.ZS23YFTDD5MXZ07...
- ZOHO_CLIENT_SECRET: ebc3f242e2a1880bfd84...

QUESTION: Why would the same refresh token work in Postman but return no access_token in the Node.js/Axios request? What differences between Postman and Axios could cause this?

Please help identify:

1. Potential differences in how Postman vs Axios send requests
2. Hidden headers or settings Postman might be adding
3. Zoho-specific requirements I might be missing
4. Debugging steps to see the actual response content