# 🚨 BlockReceipt.ai – Final Implementation Corrections for Replit

## 🔧 UploadReceipt.ts – Critical Fixes Required

- ✅ Validate wallet presence + format with regex ✅  
- ❌ Add web3 signature verification (optional for now, but needed soon)  
- ❌ Call `extractReceiptData()` and verify it returns categorized items  
- ❌ Handle case where `items.length === 0`  
- ❌ Display parsed receipt preview in frontend for user confirmation

## 🤖 NFT Bot Trigger

- Call `nftPurchaseBot(walletAddress, tierInfo)` after successful parsing  
- If it fails, fallback to internal NFT mint logic  
- Ensure returned tokenId is saved and tied to the user's wallet

## 🔐 TACo Integration Fixes

- After extracting line items, call `encryptLineItems(userPublicKey, parsedData)`  
- Save `{ capsule, ciphertext, policyPublicKey }` to DB, linked to tokenId and wallet  
- Store tokenId result from NFT bot to tie to metadata

## 🖼 NFT Gallery Fixes

- Add `/gallery/:wallet` backend route to return user's NFTs  
- Each NFT should include:  
 - tokenId  
 - image  
 - metadataLocked: true/false  
- Frontend must call this and display NFTs in a grid  
- If locked, show 🔒 and a button to 'Unlock Metadata'

## 🚫 What’s Currently Broken

- `encryptLineItems` is imported but never called  
- `nftPurchaseBot` is imported but not actually triggered post-upload  
- OCR fails silently if no line items found  
- No gallery frontend shows issued NFTs  
- No UI indication of TACo-encrypted metadata (locked receipts)

## ✅ Deliverables to Implement

- [ ] Rewrite `uploadReceipt.ts` to call OCR → encrypt → trigger NFT bot → store everything  
- [ ] Add backend `/gallery/:wallet` to return NFTs with lock status  
- [ ] Add frontend `NFTGallery.jsx` with unlock logic  
- [ ] Display receipt items in UI before NFT is minted  
- [ ] Finalize fallback NFT mint from local art if bot fails

## 📦 Files to Add or Update

- `server/routes/uploadReceipt.ts`  
- `server/routes/gallery.ts`  
- `server/utils/encryptLineItems.js`  
- `components/NFTGallery.jsx`  
- `components/UnlockMetadataButton.jsx`

## 🧠 Final Notes for Replit

This app is now positioned as a next-gen, privacy-first NFT reward engine. But to fully deliver on its UX promise, all of these backend and frontend components must be correctly implemented, wired, and validated. Please audit the entire UX and backend for consistency, test NFT delivery, and ensure metadata is encrypted and usable.