# 📋 BlockReceipt.ai – Developer Task Queue for Replit

## 🔄 Project Context

BlockReceipt.ai is a privacy-first, receipt-to-NFT platform. The backend and smart contracts are deployed, but key integrations remain incomplete or unconnected. The following tasks represent critical next steps to bridge the backend, encryption layer, NFT bot, and frontend experience.

## ✅ Core Integration Tasks

1. 🔗 Upload Flow Rewrite  
- Enforce connected wallet in frontend + backend (via wagmi or similar)  
- Validate presence of wallet in `uploadReceipt.ts`  
- Call `extractReceiptData()` and verify line items + categories  
- Call `encryptLineItems(userPublicKey, receiptItems)` post-parsing  
- Save `ciphertext`, `capsule`, `policyPublicKey` with tokenId in DB  
- Trigger NFT bot with `wallet + tier` and return result to frontend

2. 🤖 NFT Purchase Bot Activation  
- Wire up `nftPurchaseBot()` to actually buy or mint NFT  
- Ensure contract supports gasless or backend-funded execution  
- Assign tokenId after success  
- Fallback to local mint if no NFT under $0.10 is found

3. 🔐 Metadata Encryption  
- Store encrypted receipt details (items + merchant + total) in backend  
- Ensure this data is gated: only visible to NFT owner (via wallet match)  
- Optionally implement `/decrypt/:tokenId` route or decrypt in frontend

4. 🖼 NFT Gallery Feature  
- Create `/gallery/:wallet` route to return owned NFTs  
- For each NFT, include metadataLocked: true/false  
- Build `NFTGallery.jsx` frontend to display card-style grid  
- Add 🔒 overlay and unlock button to attempt decryption via TACo

5. 💬 UX and Messaging Updates  
- Update copy in all components to reflect real capabilities  
- Change 'Upload' button to 'Immortalize Receipt' or similar  
- Add tooltip or link: 'Why is this metadata locked? Powered by Threshold TACo.'

## 📁 Files to Review or Add

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

## 🧪 Validation

- [ ] Can a user upload a receipt and receive an NFT?  
- [ ] Is the NFT stored on-chain and sent to the correct wallet?  
- [ ] Is the encrypted metadata stored securely?  
- [ ] Does the frontend show locked metadata?  
- [ ] Can the wallet owner unlock the metadata?  
- [ ] Does the gallery work for previously claimed NFTs?

## 📦 Additional Suggestions

- Consider pre-loading a few NFTs below $0.10 for testing  
- Use Polygon Amoy or Mumbai for demo purposes  
- Add retry or fallback if OpenSea API fails