# 🧭 BlockReceipt.ai – Response to Iteration 5 Implementation Update

## 📣 Great Work & Acknowledgement

Appreciate your thoughtful summary and all the features you've now delivered. You've brought the product within striking distance of a complete proof-of-concept MVP.

## 🎯 Priority Next Steps (In Order)

1. 🔄 Enhance Upload Process  
- [ ] Store encrypted metadata from `encryptLineItems()` during receipt upload  
- [ ] Persist capsule, ciphertext, policyPublicKey using `MetadataService`  
- [ ] Link metadata to NFT tokenId returned from mint  
- [ ] Ensure this happens only after mint succeeds

2. 🧪 End-to-End Workflow Testing  
- [ ] Simulate entire user journey:  
 - Upload → Encrypt → Mint → Store → View → Unlock  
- [ ] Validate that tokenId is returned and can be queried via gallery/metadata/unlock routes  
- [ ] Confirm frontend unlock attempts retrieve and decrypt expected data

3. 🧱 Sample NFT Generator (Optional)  
- [ ] Add temporary route to mint test NFTs with dummy encrypted payloads  
- [ ] Helps populate the gallery and rapidly iterate on frontend visual logic  
- [ ] Route should insert dummy metadata and link to a known test wallet

## 🧠 Notes

We’ll likely want to iterate on the unlock experience and metadata UX next. For now, this phase is about closing the loop on technical functionality and ensuring the app works in production-like scenarios.

## ✅ Summary Instruction

Please prioritize storing encrypted metadata in the upload flow, ensure it's tied to minted tokenId, and test the full lifecycle. Once confirmed, we can begin polishing the UX and exploring smart NFT preview methods or wallet ownership badges.