# BlockReceipt.ai – Enhanced Smart Contract Documentation

📄 Version: Enhanced ERC-1155 Receipt NFT (Receipt1155Enhanced.sol)

📅 Last Updated: May 2025

🛡️ Security Contact: security@blockreceipt.ai

## 🔧 Overview

BlockReceipt.ai is a full-stack Web3 platform that allows users and businesses to mint digital purchase receipts as NFTs on the blockchain. The enhanced smart contract (`Receipt1155Enhanced.sol`) leverages the ERC-1155 standard to support batch minting, on-chain metadata, role-based access control, and robust event logging. It is designed to be scalable, secure, and interoperable with major NFT platforms and wallets.

## 🧠 Smart Contract Features

- \*\*ERC-1155 Compliant\*\*: Supports flexible token types and batch operations.  
- \*\*AccessControl Roles\*\*: `ADMIN\_ROLE`, `MINTER\_ROLE`, and `DEFAULT\_ADMIN\_ROLE`.  
- \*\*On-Chain Metadata\*\*: Stores timestamp, receiptType, and revocation status.  
- \*\*URI Management\*\*: Supports base URI and token-specific overrides.  
- \*\*Event Emission\*\*: Custom events for `ReceiptMinted`, `ReceiptRevoked`, `ReceiptBurned`, and `BaseURISet`.  
- \*\*Security Features\*\*: `Pausable`, controlled minting/burning, and revocation without burning.

## 🔐 Roles

- `DEFAULT\_ADMIN\_ROLE`: Full control over role assignment and critical actions.  
- `ADMIN\_ROLE`: Can pause/unpause the contract, revoke receipts, update base URI.  
- `MINTER\_ROLE`: Can mint new receipts or batch mint using validated metadata.

## 📡 Events

- `ReceiptMinted(address to, uint256 tokenId, string uri, string receiptType)`  
- `ReceiptRevoked(uint256 tokenId)`  
- `ReceiptBurned(address owner, uint256 tokenId, uint256 amount)`  
- `BaseURISet(string newBaseURI)`

## 🛠 Public Functions

- `mintReceipt(address to, uint256 tokenId, string calldata uri\_, string calldata receiptType)`  
- `mintNewReceipt(address to, string calldata uri\_, string calldata receiptType)`  
- `revokeReceipt(uint256 tokenId)`  
- `burn(address from, uint256 tokenId, uint256 amount)`  
- `pause()` / `unpause()`  
- `setBaseURI(string calldata newBaseURI)` (Admin only)  
- `getReceiptMetadata(uint256 tokenId)` → `ReceiptMetadata` struct  
- `isValidReceipt(uint256 tokenId)` → `bool`

## 🧪 Suggested Tests

- Mint with valid/invalid role  
- Revoke and re-query `isValidReceipt`  
- Pause contract and test mint failure  
- Burn receipt and emit `ReceiptBurned`  
- Update base URI and verify result  
- Verify role-based access control enforcement

## 📦 Frontend Integration

- ABI located at `artifacts/contracts/Receipt1155Enhanced.sol/Receipt1155Enhanced.json`  
- Recommended to copy to: `client/src/abis/Receipt1155Enhanced.json`  
- Use `getReceiptMetadata` in React to fetch timestamp, status, and type of receipts

## 🚀 Deployment Notes

- Use `scripts/deploy-enhanced-receipt.js` to deploy  
- Script automatically grants roles and stores contract address in `.env`  
- Recommended to verify on Polygonscan for public visibility  
- After deployment, run `update-after-deployment.cjs` if needed

## 📈 Roadmap

- Add subgraph support for event indexing (e.g., ReceiptMinted)  
- Add UI for verifying receipt hash or ownership  
- Add integration with IPFS for storing receipt metadata securely  
- Develop merchant dashboard for issuing and revoking receipts