# BlockReceipt.ai – NFT Claim Flow After Receipt Upload

## ✨ Vision

After a user uploads a receipt, BlockReceipt.ai offers them 5–10 unique, tier-appropriate NFTs to choose from. These NFTs are free to mint for the user but carry a strong collectible appeal, and are optionally customized based on what was purchased. Minting is gasless for the user and cost-efficient for the platform via batching, relayers, or low-cost chains like Polygon.

## 🧱 Strategy

1. Use extracted tags from receipts (e.g. 'snacks', 'electronics')  
2. Map those tags to NFT art pools  
3. Display 5–10 randomly chosen art options to the user  
4. Upon selection, mint the NFT on Polygon using a sponsored wallet  
5. Optionally, use AI-generated artwork from Replicate or Stable Diffusion based on the purchase

## 🔧 Backend: NFT Art Pool Mapper

File: shared/utils/nftArtSelector.js

const fs = require('fs');  
const path = require('path');  
  
// Mocked pool of NFT art organized by tag  
const nftArtPool = {  
 snacks: ['twix1.png', 'chips2.png', 'candy3.png'],  
 grooming: ['shampoo1.png', 'lotion2.png'],  
 tech: ['headphones1.png', 'charger2.png'],  
 random: ['cat1.png', 'pixel\_dog2.png', 'tiny\_turtle3.png']  
};  
  
function getNFTArtByTags(tags = []) {  
 const matches = new Set();  
 tags.forEach(tag => {  
 if (nftArtPool[tag]) {  
 nftArtPool[tag].forEach(img => matches.add(img));  
 }  
 });  
  
 const fallback = nftArtPool['random'] || [];  
 const resultPool = [...matches].length ? [...matches] : fallback;  
  
 // Return up to 10 random images  
 const shuffled = resultPool.sort(() => 0.5 - Math.random());  
 return shuffled.slice(0, 10);  
}  
  
module.exports = { getNFTArtByTags };

## 🧾 Frontend Component: NFT Art Picker

File: client/components/NFTArtPicker.jsx

import React, { useEffect, useState } from 'react';  
  
export default function NFTArtPicker({ tags, onSelect }) {  
 const [options, setOptions] = useState([]);  
  
 useEffect(() => {  
 fetch('/api/nft-options', {  
 method: 'POST',  
 headers: { 'Content-Type': 'application/json' },  
 body: JSON.stringify({ tags }),  
 })  
 .then(res => res.json())  
 .then(data => setOptions(data.options || []));  
 }, [tags]);  
  
 return (  
 <div className="grid grid-cols-2 gap-4">  
 {options.map((src, i) => (  
 <div key={i} onClick={() => onSelect(src)} className="cursor-pointer border p-2 rounded hover:bg-gray-100">  
 <img src={`/nft-images/${src}`} alt={`NFT ${i}`} className="w-full h-auto rounded" />  
 </div>  
 ))}  
 </div>  
 );  
}

## 📡 Backend: NFT Options API

File: server/routes/nftOptions.js

const express = require('express');  
const { getNFTArtByTags } = require('../../shared/utils/nftArtSelector');  
  
const router = express.Router();  
  
router.post('/nft-options', (req, res) => {  
 const { tags = [] } = req.body;  
 const options = getNFTArtByTags(tags);  
 res.json({ success: true, options });  
});  
  
module.exports = router;

## ⚙️ Minting Function (Backend Helper)

Use this to mint the selected art NFT via Polygon. The backend handles gas and mints from a shared wallet.

const ethers = require('ethers');  
const CONTRACT\_ABI = require('./YourContractABI.json');  
const CONTRACT\_ADDRESS = '0xYourContractAddress';  
  
async function mintNFT(to, metadataURI) {  
 const provider = new ethers.JsonRpcProvider('https://polygon-rpc.com');  
 const wallet = new ethers.Wallet(process.env.MINTER\_PRIVATE\_KEY, provider);  
 const contract = new ethers.Contract(CONTRACT\_ADDRESS, CONTRACT\_ABI, wallet);  
  
 const tx = await contract.mintReceipt(to, metadataURI);  
 return await tx.wait();  
}

## 🎨 Optional: AI Generation API (e.g., Replicate)

You can enhance the experience by dynamically generating art with prompts based on receipt content. Use Replicate or Stable Diffusion APIs and store results temporarily before minting.

// Sample Prompt from Receipt  
const prompt = `a happy cartoon of a bottle of shampoo and a Twix bar in a grocery aisle`;  
  
// Send to Replicate API or local generator and save output to IPFS or server

## 🔐 Final Note

This system keeps user experience fun, NFT issuance affordable, and art contextually relevant. You can later expand this to use AI-generated art, brand partnerships, or gamified tiers for NFT collectibles.