# 🛠 BlockReceipt.ai – NFT Auto-Purchase Bot Starter Code (Polygon)

## 🔐 Wallet & Blockchain Setup

- Use a dedicated hot wallet (backend-only, never expose to frontend)  
- Store the private key securely in `.env`  
- Use Polygon RPC (mainnet or Mumbai for testing)

## 📦 Required Dependencies

Install these in your Replit backend:  
```bash  
npm install ethers axios dotenv  
```

## 🧠 Backend NFT Purchase Script

File: `server/utils/purchaseNFT.js`

const { ethers } = require("ethers");  
require("dotenv").config();  
const axios = require("axios");  
  
const provider = new ethers.JsonRpcProvider("https://polygon-rpc.com");  
const wallet = new ethers.Wallet(process.env.PRIVATE\_KEY, provider);  
  
// NFT Contract ABI (simplified ERC-721 for transfer)  
const ERC721\_ABI = [  
 "function safeTransferFrom(address from, address to, uint256 tokenId) external",  
];  
  
// Example NFT purchase + transfer  
async function purchaseAndTransferNFT(userWallet) {  
 try {  
 // Simulate NFT query  
 const mockNFT = {  
 contractAddress: "0xExampleNFTContract",  
 tokenId: "123",  
 price: ethers.utils.parseUnits("0.05", "ether"),  
 };  
  
 const contract = new ethers.Contract(mockNFT.contractAddress, ERC721\_ABI, wallet);  
  
 // Send the NFT to the user  
 const tx = await contract.safeTransferFrom(wallet.address, userWallet, mockNFT.tokenId);  
 const receipt = await tx.wait();  
  
 return {  
 success: true,  
 txHash: receipt.transactionHash,  
 };  
 } catch (err) {  
 console.error("NFT Transfer Failed:", err.message);  
 return { success: false, error: err.message };  
 }  
}  
  
module.exports = { purchaseAndTransferNFT };

## 🚀 Trigger from Receipt Upload (Example)

Update `uploadReceipt.ts` to add:

const { purchaseAndTransferNFT } = require('../utils/purchaseNFT');  
  
// Inside receipt POST logic after receipt tier:  
if (tier !== "Basic") {  
 const nftResult = await purchaseAndTransferNFT(userWallet); // Pass connected wallet  
 if (!nftResult.success) console.log("NFT issue:", nftResult.error);  
}

## ✅ Next Steps Checklist

- [ ] Set up `.env` with PRIVATE\_KEY of funded hot wallet  
- [ ] Test on Polygon Mumbai first  
- [ ] Rate-limit NFT transfer per wallet (e.g. Redis, memory log)  
- [ ] Store a log of each wallet's NFT claims  
- [ ] Integrate a real NFT query function from OpenSea or Reservoir API

## 🧠 Optional (Later)

- Integrate fallback logic: if no external NFT is available, mint a custom NFT from your own contract  
- Add pixel art generation with Replicate API for personalized NFTs  
- Notify user via frontend of claim success/failure