

Blockchain-based eCommerce warranty system using NFTs

Team Name: Netajihackers

Institute Name: Netaji Subhash University of Technology, Delhi

Team members details

Team Name	Netajihackers		
Institute Name	Netaji Subhash University of Technology, Delhi		
Team Members >	1 (Leader)	2	3
Name	Lakshay Gulati	Satyam Mishra	Siddharth Majumdar
Batch	2024	2024	2024

Deliverables/Expectations for Level 2 (Idea + Code Submission)

The solution should focus on:

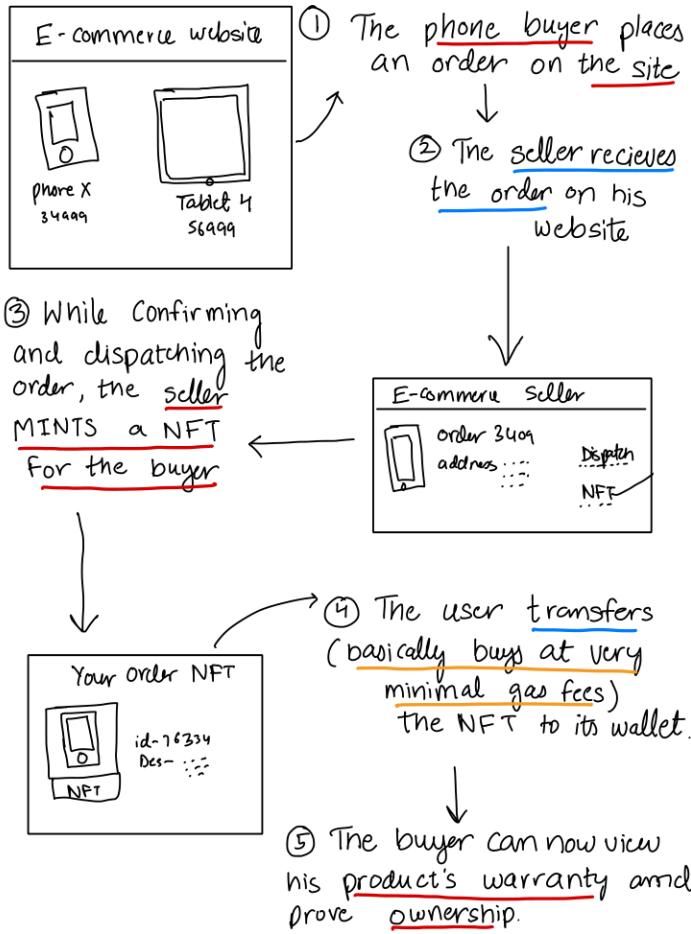
- The blockchain smart contract should allow users to prove ownership
- Provide the purchasing history, warranty period, and other item information
- The warranty card should include the item's serial number and upon purchase be sent to the customer's smartphone.
- The NFTs should be decaying in nature, in that, after a certain period their use for the redemption of warranty benefits offered by the brand/retailer will expire
- **Bonus** - GUI-based tool that doesn't require knowledge of any Blockchain programming to use by Brands and Retailers.
- **Bonus** - Usage of Soulbound NFTs
- **Bonus** - Add any engagement/gamification construct to the loyalty program

Glossary

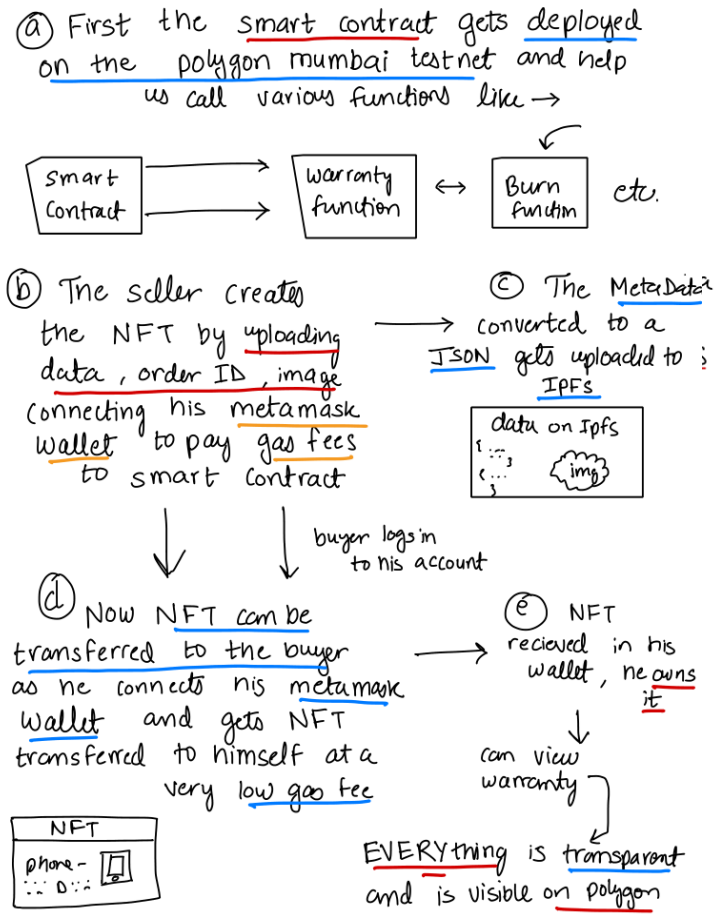
- **IPFS** – **Inter Planetary File System** (IPFS) is a protocol and peer-to-peer network for storing and sharing data in a distributed file system. **STORES THE METADATA AND IMAGE ~ The NFT Permanently**
- **Smart Contract** - **Smart contracts** are simply programs stored on a blockchain that run when predetermined conditions are met
- **ERC – 721** - is a **standard smart contract for representing ownership of NFTs**, that is, where each token is unique and can be decayed
- **MetaMask** - cryptocurrency wallet used to interact with the Ethereum blockchain used to interact with decentralized applications.
- **Polygon** - **Ethereum blockchain** — allowing speedy transactions and low fees , uses MATIC as currency
- **Gas Fees** - the cost necessary to perform a transaction on the Ethereum network
- **Buyer** – The customer placing the order on E-Commerce Website
- **Seller** – The creator of NFT and selling the order

Proposed approach

Happening at the Frontend



Happening at the backend



Smart Contract

Functions present –

- Burning the NFT token
- Viewing the Warranty

Other basic ERC 721 functions -

- Fetching the NFT from blockchain with address
- Token URI (unique id for every NFT token created)
- Listing NFT , Transfer approval from owner
- Secure Functions to check ownership

```
function checkUpkeep(bytes memory)
    public
    returns (bool needsUpkeep, bytes memory)
    {
        bool timePassed = (expireDate >= currentDate);
        needsUpkeep = (timePassed);
    }

    //burns token once condition is met
    function performUpkeep(bytes calldata,uint256 tokenId) external {
        (bool needsUpkeep, ) = checkUpkeep("");
        require(!_isApprovedOrOwner(_msgSender(), tokenId), "ERC721: caller
is not token owner or approved");
        require(needsUpkeep == true, "Upkeep not needed.");
        _burn(tokenId);
    }
}
```

The **smart contract running in hardhat environment gets deployed to the Polygon Mumbai test-net** and has an address on which all the functions work.

You **can view** this [here](#).

And **smart contract** [here](#).

This function creates the NFT

```
function createToken(string memory tokenURI, uint256 price) public payable
returns (uint) {
    _tokenIds.increment();
    uint256 newTokenId = _tokenIds.current();

    _mint(msg.sender, newTokenId);
    _setTokenURI(newTokenId, tokenURI);
    createWarranty(newTokenId, price);
    return newTokenId;
}
```

The Workflow

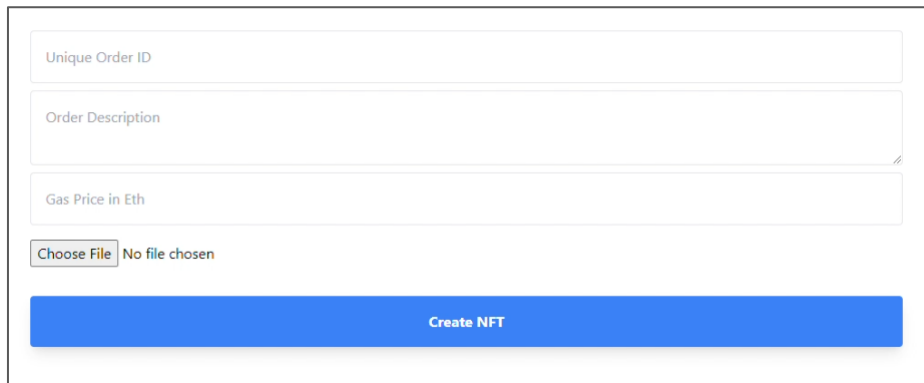
- The smart contract gets deployed on the polygon test-net using hardhat , gas fees is charged from the seller / creator. The **address** where the contract is deployed is saved as a constant in the project file.

```
lakshayx9@Lakshay:/mnt/d/nft-marketplace$ npx hardhat run scripts/deploy.js
nftMarketplace deployed to: 0x18aFa9F8d1DAa8302A4E0f02736C47Fc98736eC8

> nft-marketplace@0.1.0 dev /mnt/d/nft-marketplace
> next dev

ready - started server on 0.0.0.0:3000, url: http://localhost:3000
info - SWC minify release candidate enabled. https://nextjs.link/swcmin
```

↙ Address where deployed



- We created a simple GUI for seller for him to **create warranty** easily , he can upload the order image and data , with **warranty being automatically created** for One year from date of order

Now, this how the order and NFT will be created

1. The Customer / Buyer **places the order** on the E-commerce website and his details and order gets notified to the seller

Shipping Address
Rahul
H num 1234
Delhi
Delhi Delhi
Phone : 9875631254
Owner NFT wallet address : 0x405bf53aA1c39E5342d430fc0D4be3a44dD8615b

Order

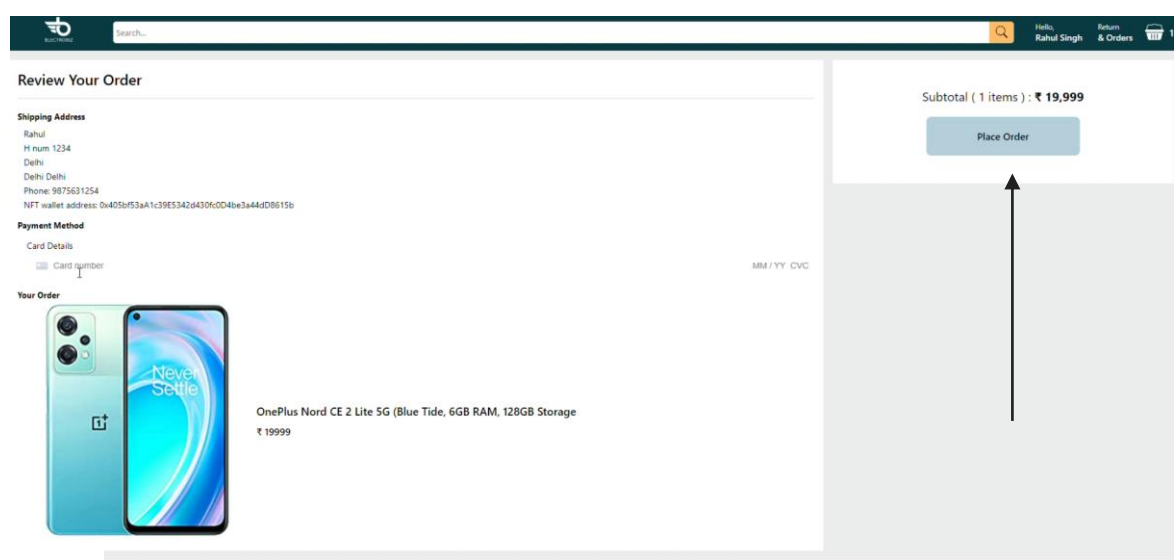
Subtotal : ₹ 19999



OnePlus Nord CE 2 Lite 5G (Blue Tide, 6GB RAM, 128GB Storage)
₹ 19999

Mint NFT

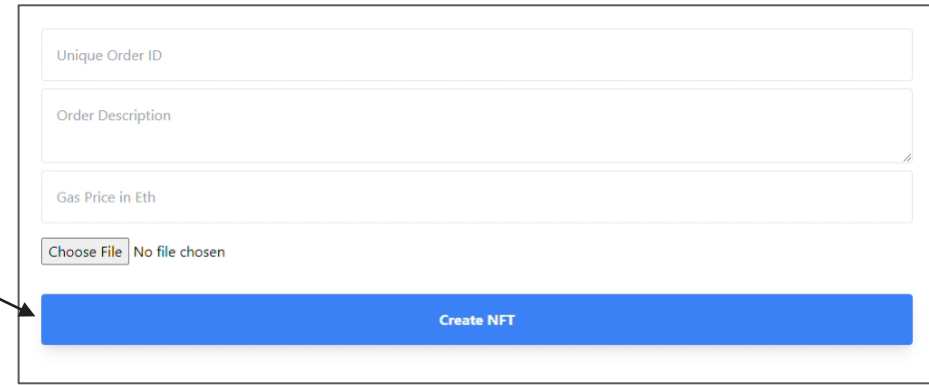
Dispatch Order



2. The seller sees that he has received the order and now dispatches it along with **creating NFT** when he clicks the **Create NFT button** which redirects it to the site we created.

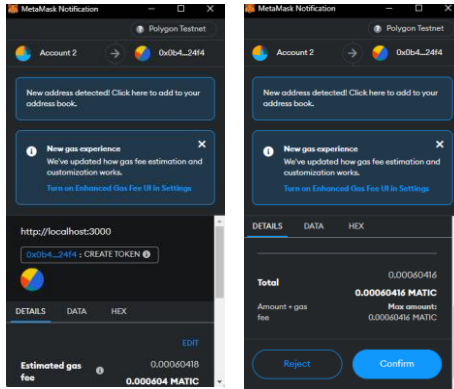
3. Now the seller gets redirected to the warranty NFT creation site where he **enters the details and clicks on create NFT and approves transaction through metamask wallet**

Also uploads the image / product svg

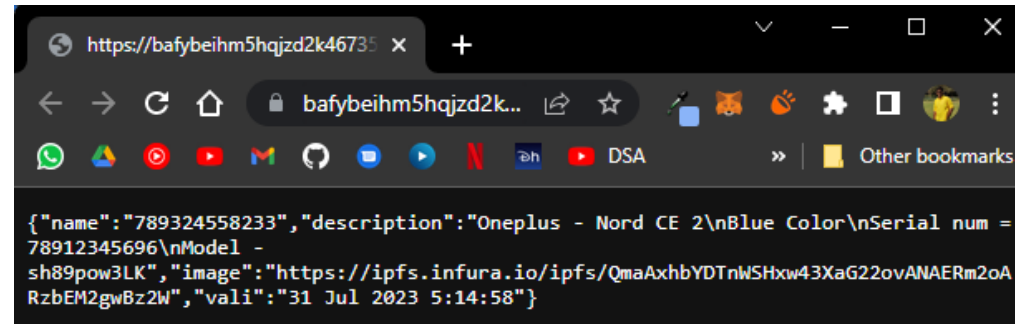


A screenshot of a web form for creating an NFT. It contains four input fields: 'Unique Order ID', 'Order Description', 'Gas Price in Eth', and a file upload section with a 'Choose File' button and 'No file chosen' text. At the bottom is a large blue button labeled 'Create NFT'. An arrow from the text 'enters the details and clicks on create NFT' points to this button.

4. The details entered by the seller **gets converted to a JSON file** and act as **NFT Metadata**, Also the **warranty date is automatically taken in the metadata** (one year after the creation) and **Uploaded on IPFS** just like what platforms like OpenSea do so it can't be deleted and thus NFT is created , you can view one [here](#).



5. Now the NFT can be **accessed by seller** and he can **transfer the NFT to his own wallet address** and get owner ship and claim warranty



Shipping Address

1. Rahul
H num 1234
Delhi
Delhi Delhi
Phone : 9875631254
Your NFT wallet address : 0x405bf53a1c39E5342d430fc0D4be3a44dD6615b

Order

Subtotal : ₹ 19999



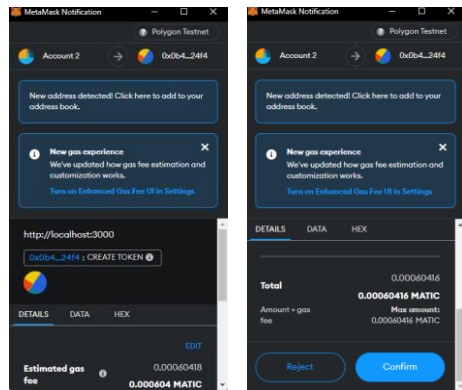
OnePlus Nord CE 2 Lite 5G (Blue Tide, 6GB RAM, 128GB Storage)
₹ 19999

6. The buyer can now **view the NFT** of the order he placed and can now transfer it to himself

Your NFT based warranty

Flipkart Warranty Site

[Get NFTs](#) [Create NFT Warranty](#) [My Orders](#) [Dashboard](#)



7. Now the **buyer connects his own meta mask wallet** and clicks on **Fetch My NFT** and pays a nominal gas fee to get the ownership of the NFT



789324558233

Oneplus - Nord CE 2 Blue Color Serial num = 78912345696 Model - sh89pow3LK

Fetch My Nft Order

7. The user can now view his **NFT** on IPFS and can also see the warranty

Flipkart Warranty Site

[Get NFTs](#) [Create NFT Warranty](#) [My Orders](#) [Dashboard](#)

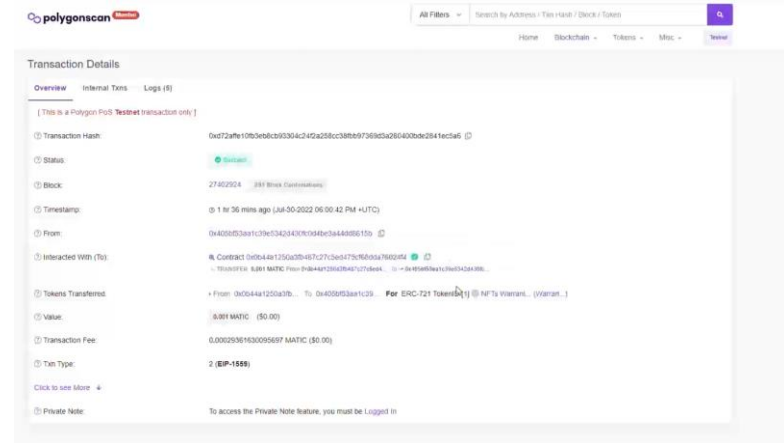
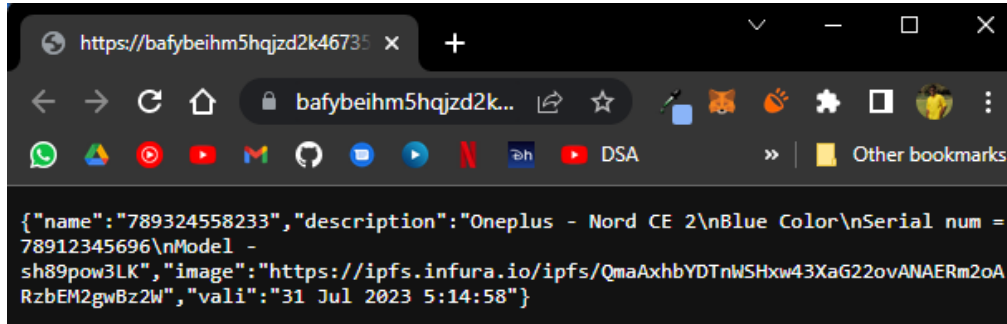


8. Can view his NFT on the IPFS and Polygon Scan



Show My NFT on IPFS

31 Jul 2023 5:14:58



Use-cases

- Customers who have trust issues with **expensive items being bought online are authentic** , can now authenticate the product with the NFT metadata.
- Customers can now **claim warranty (Paperless) easily as the NFT** behaves like a warranty card and is sent to their account and can even **prove the ownership** of the product as they are also the owner of the NFT.
- The seller **doesn't require any knowledge of blockchain** and smart-contract , he just needs to follow easy steps and create NFT with a click of a button.
- Individual sellers / MSMEs can now **ease the process of giving warranty** and authenticating their product
- Customer can **easily transfer ownership** of Luxury items like watches when he also transfers the NFT.

Limitations

- The seller has to **manually upload the data to create the warranty** based NFT , as when he gets 1000s of orders , he has to manually connect his wallet and create the NFT
- As of now there is **no authentication for the seller** to login and create NFT on our platform.
- The Platform is working on **Mumbai Test-net** and uses Test Matic as currency and has not been tested on **Main Ethereum Net**.
- For every transaction a **Gas fees has to paid** which will add to the product's cost but is very minimal , as low as 0.0003 MATIC = 0.024 Rupee

Timestamp:	13 hrs 8 mins ago (Jul-30-2022 06:00:42 PM +UTC)
From:	0x405bf53aa1c39e5342d430fc0d4be3a44dd8615b
Interacted With (To):	Contract 0x0b44a1250a3fb487c27c5ed475cf68dda76024f4 TRANSFER 0.001 MATIC From 0x0b44a1250a3fb487c27c5ed4... To → 0x405bf53aa1c39e5342d430fc...
Tokens Transferred:	From 0x0b44a1250a3fb... To 0x405bf53aa1c39... For ERC-721 TokenID [1] NFTs Warrant... (Warran...)
Value:	0.001 MATIC (\$0.00)
Transaction Fee:	0.00029361630095697 MATIC (\$0.00)
Txn Type:	2 (EIP-1559)

Future Scope

- The Digital NFT is **tied to the warranty program** but the owners still **can not track repairs** and replacements to the original item so in future we will create repair tracking on NFTs.
- **Transfer the ownership of** the NFT from our platform and we can also gain loyalty from it
- **Automate the process** of Minting/Creating of NFTs , the data will be automatically fetched from our database along with image and seller won't have to create it on its own.
- **Bulk Uploading** of the NFTs