

BLOCKCHAIN PROJECT REPORT

TOPIC: TOKENIZED LOYALTY AND REWARD SYSTEM- CREATING A BLOCKCHAIN-BASED REWARD POINTS SYSTEM FOR BUSINESSES.

INTRODUCTION

This project is a decentralized loyalty rewards system built on the Ethereum blockchain. The main goal is to let users earn, transfer, and redeem reward points (in the form of tokens) using a simple and transparent system. By using blockchain, we ensure security, transparency, and user control over reward points without depending on a central authority.

We used Solidity for the smart contract and React for the front-end interface. The DApp allows users to earn tokens when they buy items, redeem tokens to get discounts, and transfer tokens to other users. The system is managed through a user-friendly dashboard.

The app connects with MetaMask for wallet integration and shows transaction details on Etherscan after confirmation. It's simple, efficient, and showcases how blockchain can be used in a real-world reward system.

PROBLEM STATEMENT

Our problem statement was to create a tokenized reward and loyalty system which could also be used by businesses.

Traditional loyalty systems often come with a lot of problems like lack of transparency, limited usability, and trust issues. Users usually can't see what's going on in the background, and most of the time, their points are locked to just one platform.

Our goal was to solve this by creating a decentralized app where users can:

- Earn tokens based on purchases
- Redeem tokens to get discounts
- Transfer tokens to friends

This approach puts the power back in the user's hands, removes the need of a middleman and uses smart contracts to automate everything.

BLOCKCHAIN IMPLEMENTATION DETAILS

We wrote a smart contract named LoyaltyToken using Solidity. It is based on the ERC-20 token standard and makes use of OpenZeppelin's libraries for security and ease of development.

The key functions in the contract are:

- constructor: Sets the token name as "LoyaltyToken" and symbol as "LTY". It also assigns the deployer as the contract owner.
- issuePoints(address to, uint256 amount): This function lets only the contract owner mint new tokens and assign them to any user address. This is used when a customer makes a purchase and earns points.
- redeemPoints(uint256 amount): This function allows a user to burn a certain amount of their own tokens. This is called when the user wants to redeem points for a discount.

There is no separate function for transferring points because it uses the default transfer method from the ERC-20 contract.

APPLICATION ARCHITECTURE AND FRONTEND IMPLEMENTATION

Key Components:

- Login and Wallet Integration (MetaMask): The app requires users to connect their Ethereum wallet via MetaMask to interact with the blockchain. Once connected, users can view their token balance, earn points, and redeem tokens. MetaMask handles transaction signing and ensures that users have control over their private keys.
- Homepage: The homepage displays the choices given to a user once they enter the Loyalty shop. They can either Buy and Earn Points, Redeem Points or Transfer Points. Based on this choice, they will be led on to the subsequent page.
- Buy Page: This page allows users to make purchases and earn tokens based on their spending. When a user purchases an item, the corresponding amount of tokens is issued to their wallet using the issuePoints function from the smart contract.
- Redeem Page: The redeem page allows users to burn their tokens in exchange for discounts on future purchases. Users can select the amount of tokens they wish to

redeem, and the app will interact with the smart contract to trigger the redeemPoints function.

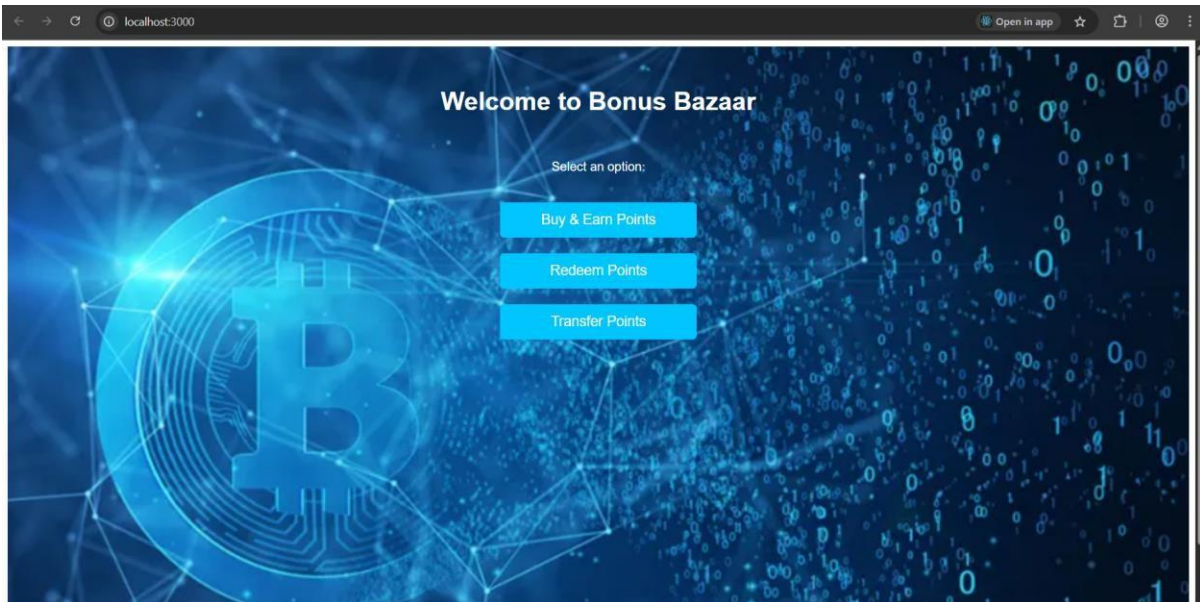
- Transfer Page: On the transfer page, users can send tokens to other users. This is achieved by utilizing the ERC-20 transfer function, which ensures that tokens are securely transferred from one address to another.
- Transaction History: Users can view their complete transaction history on the Ethereum blockchain, which is displayed in the app. The transaction details are pulled from Etherscan, allowing users to check the status of each transaction in real-time.

TECHNOLOGY STACK USED

- React: For building the user interface.
- Web3.js: A library that connects the React frontend with the Ethereum blockchain and interacts with smart contracts.
- MetaMask: A browser extension that serves as the user's Ethereum wallet, allowing them to sign transactions and manage their tokens.
- Solidity: Used to write the smart contract for the loyalty token.
- OpenZeppelin: Security libraries used for writing the smart contract in a secure and standardized manner.

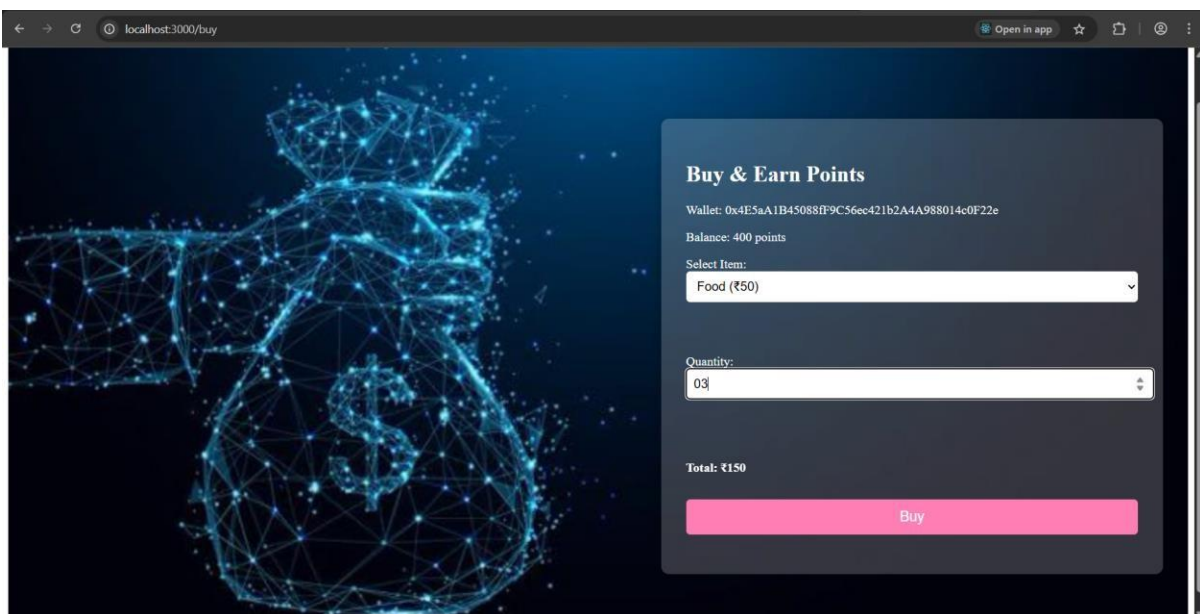
SCREENSHOTS OF THE WORKING APPLICATION

- HOME PAGE

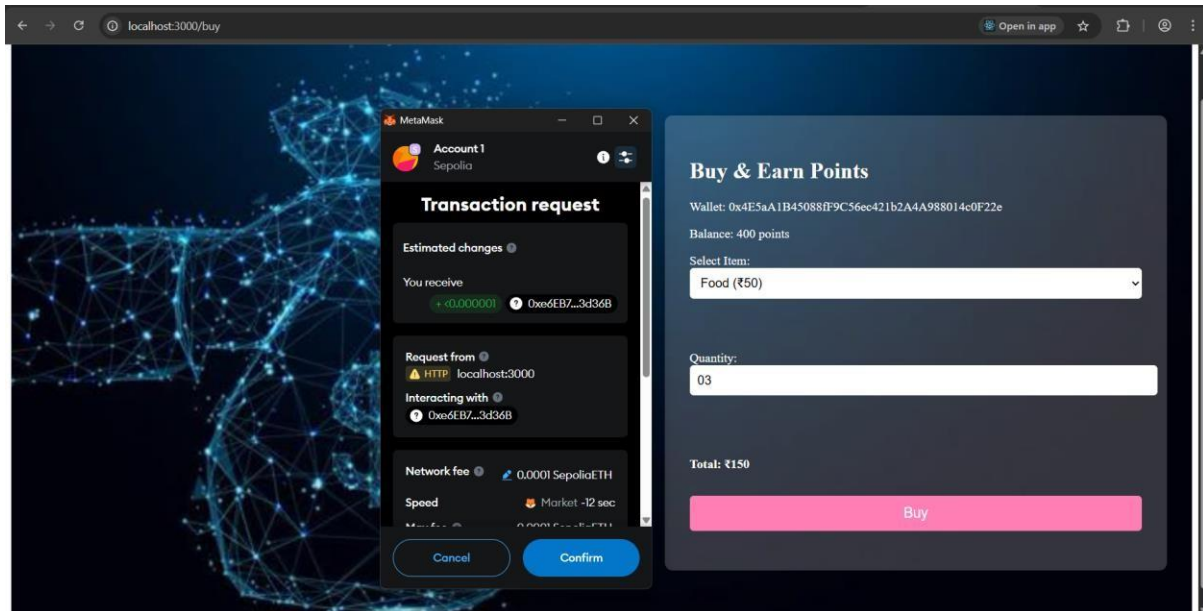


- This is our home page which welcomes the users to the Loyalty shop where they can choose either of the three options. Buy and earn points helps them get points by making purchases. Redeem points helps to get discount on the total bill by redeeming/burning points. Transfer points helps the users to transfer/send points from their account to another user's account.

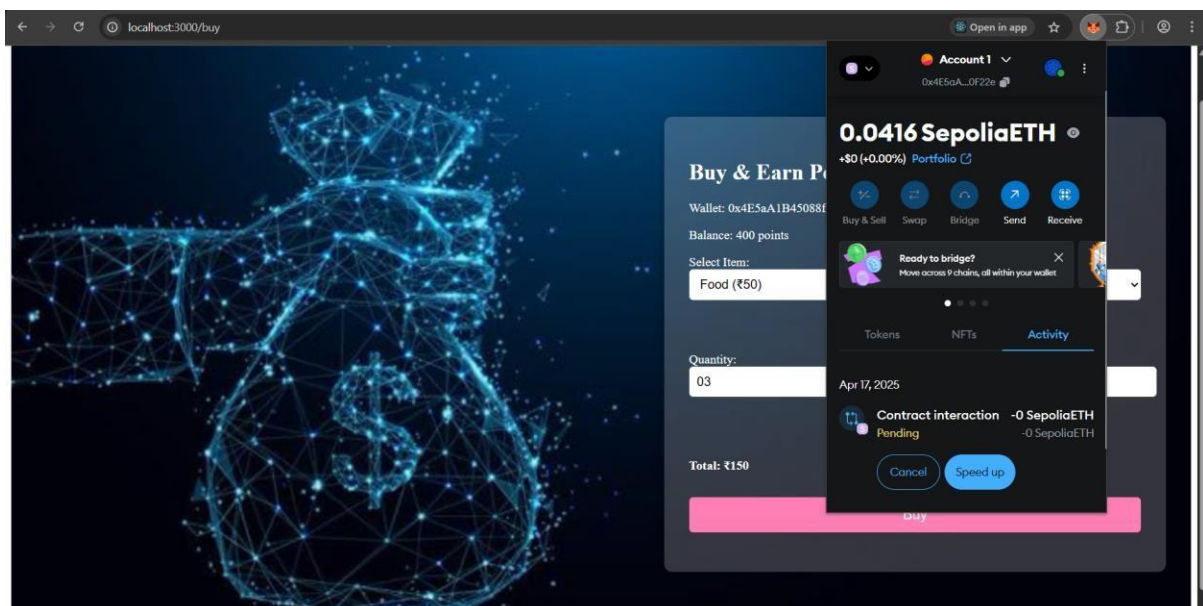
- BUY AND EARN POINTS PAGE



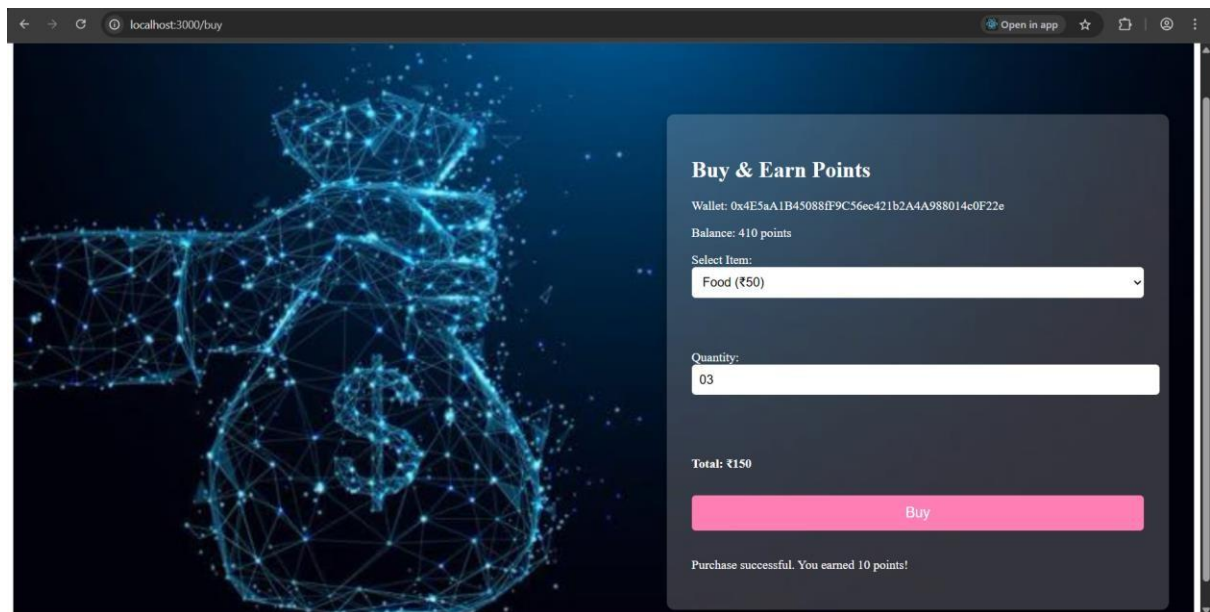
This is the Buy and Earn Points page. Here the user can make a purchase by choosing items from the dropdown menu, selecting the quantity for the chosen item and then buying it after viewing the total bill. We can also see that the user's current account balance holds 400 points.



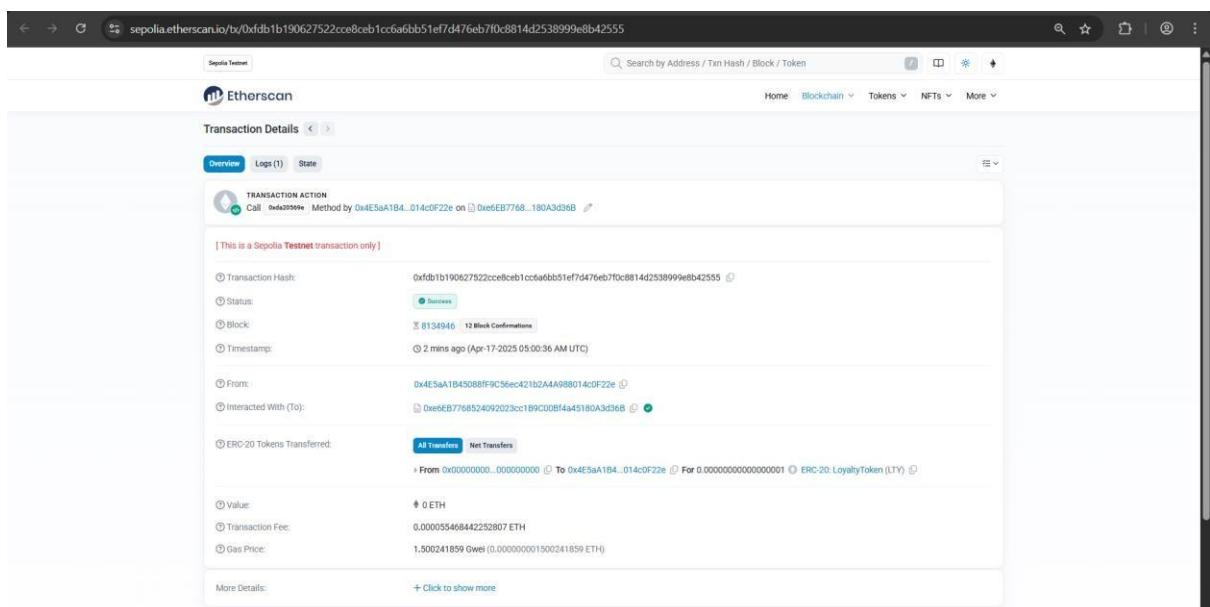
Once the user clicks on Buy, it gets connected to their MetaMask wallet which prompts the user to either confirm or cancel the requested transaction.



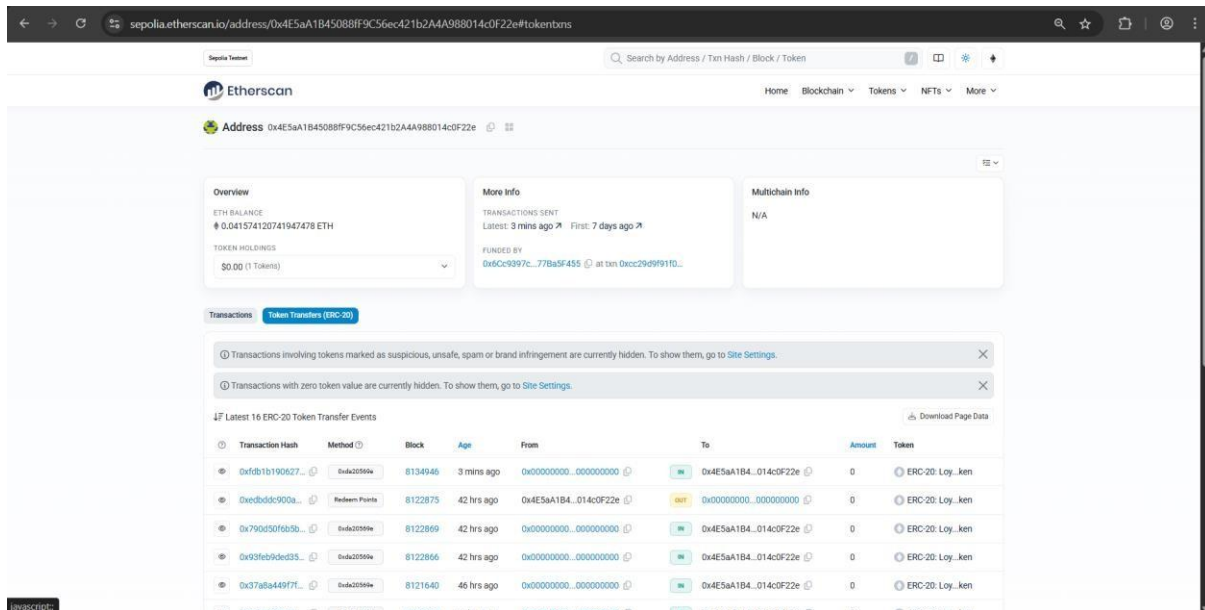
Transaction status is pending which confirms that it is in process.



Transaction has been confirmed. It displays a success message and also simultaneously updates the user's account balance.

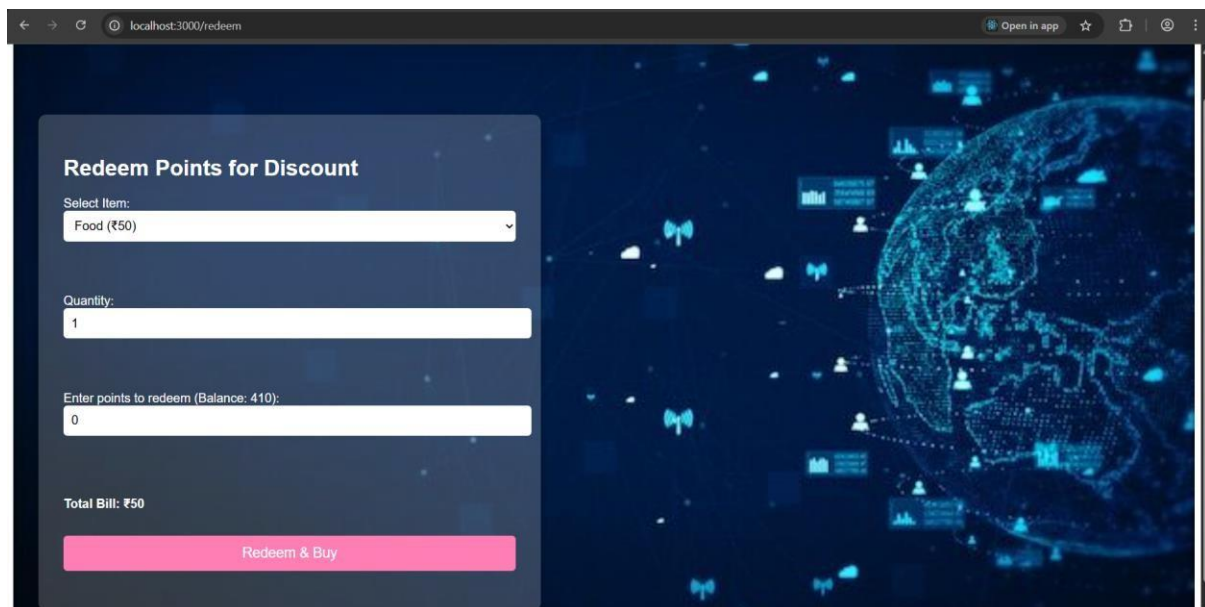


Transaction details are also visible on Etherscan.

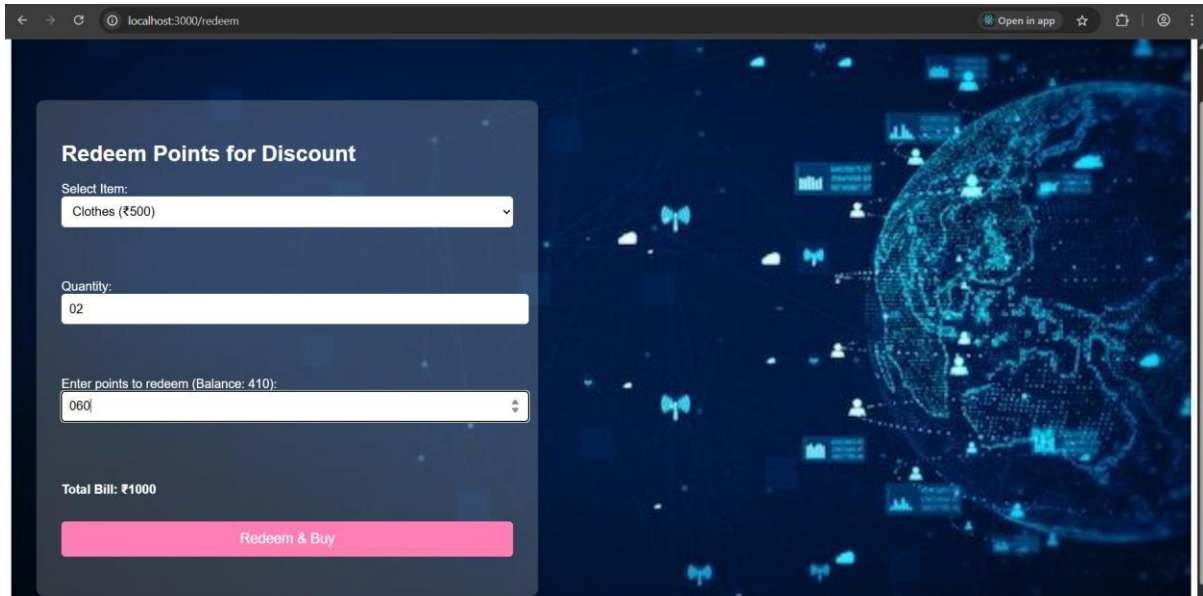


For the user's wallet address, we can also view the token transfers made, the top being the most recent one which says 'in' indicating that points have been successfully earned.

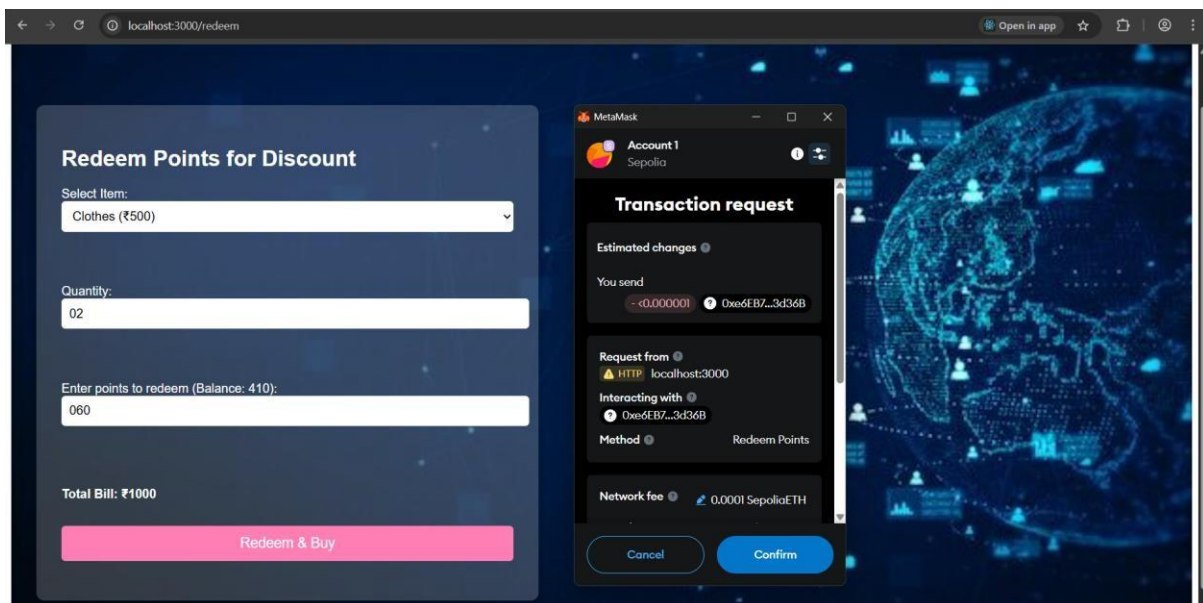
- REDEEM POINTS PAGE



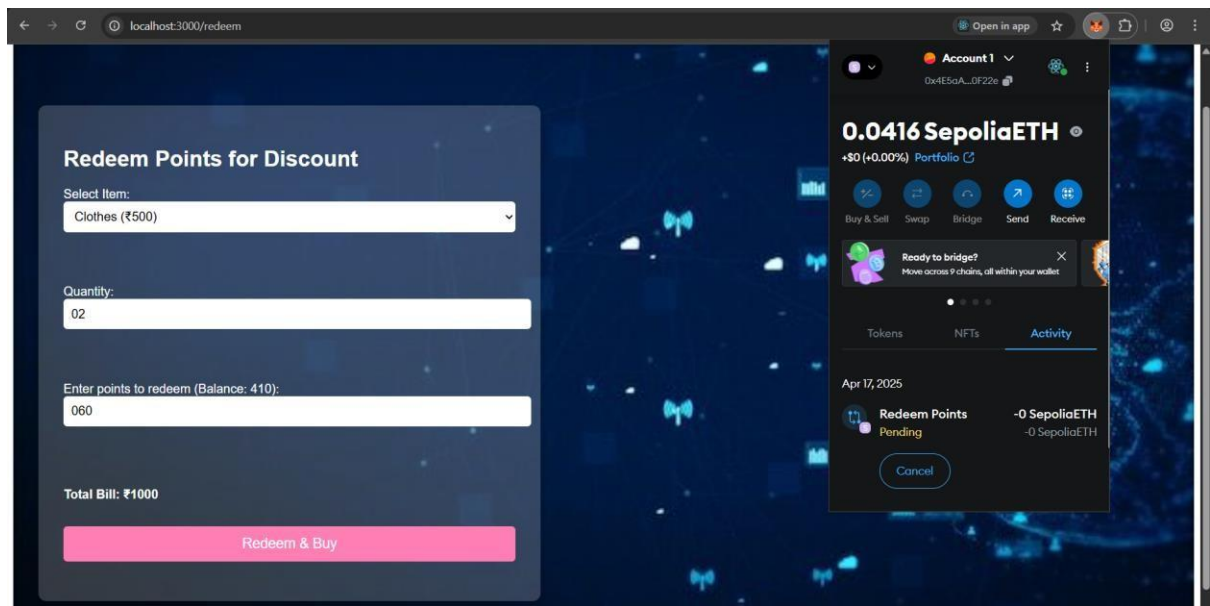
This is the Redeem Points page. Here the user can burn/redeem their points to earn a discount on their total bill after making the purchase. Once again, the user is given a choice to make their purchase from the dropdown menu and choose the quantity, after which they can select the number of points they want to redeem.



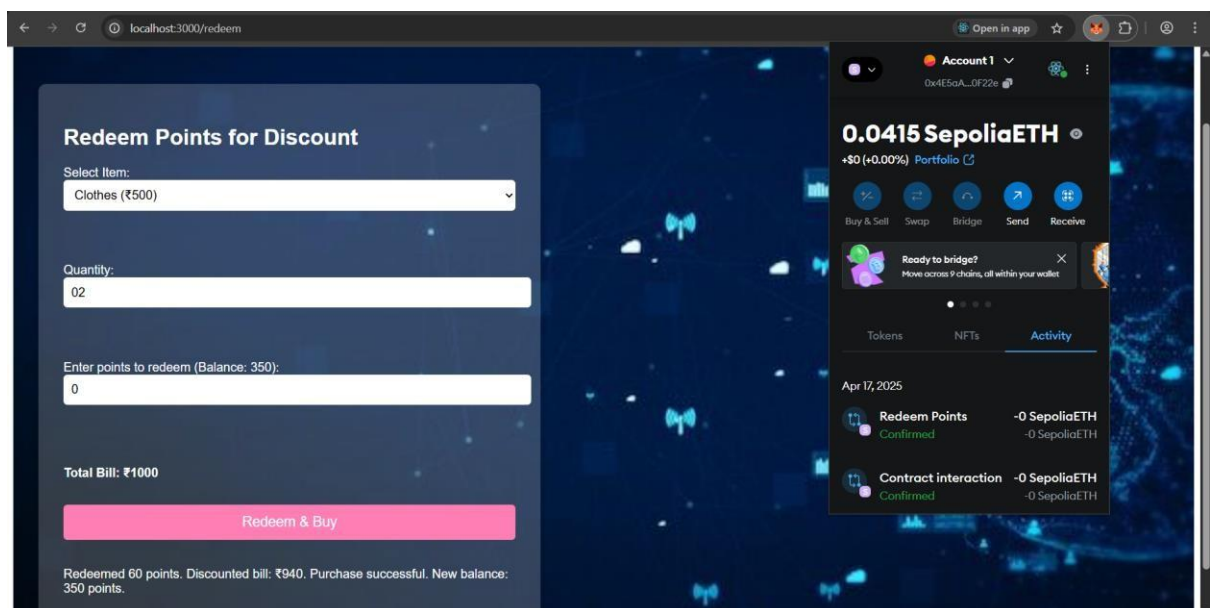
After making all the choices, the total bill amount is visible to the user (without discount).



Once the user clicks on Redeem and Buy, it gets connected to their MetaMask wallet which prompts the user to either confirm or cancel the requested transaction.

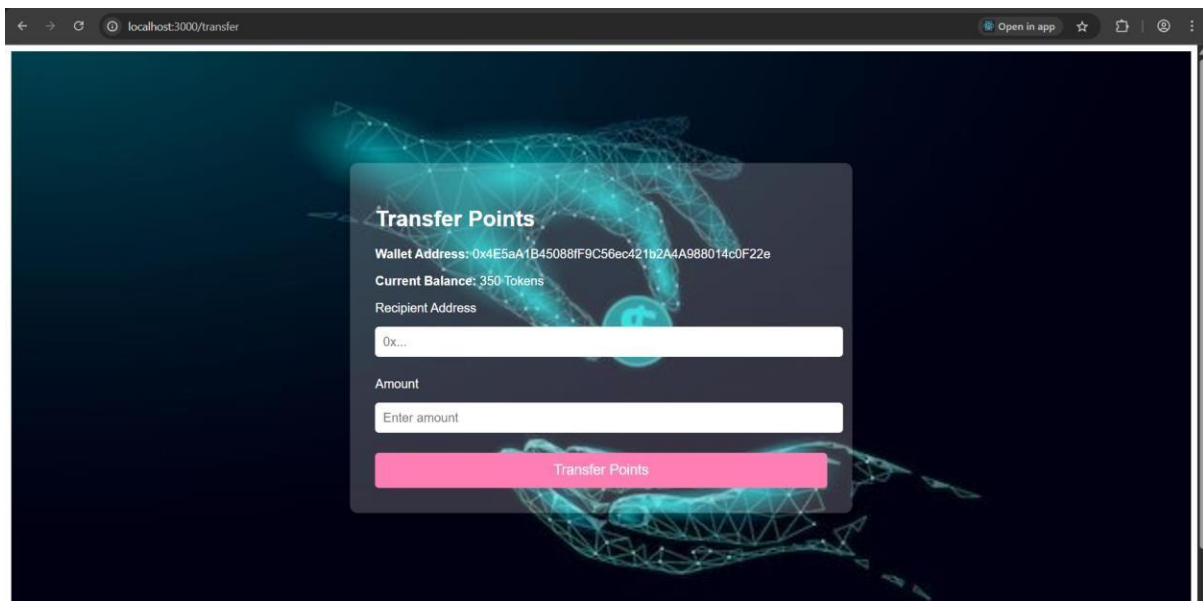


Transaction status is pending which confirms that it is in process



Once the transaction is confirmed on MetaMask, the UI updates by giving a successful purchase message along with the discounted bill and the new account balance after redeeming the chosen number of points.

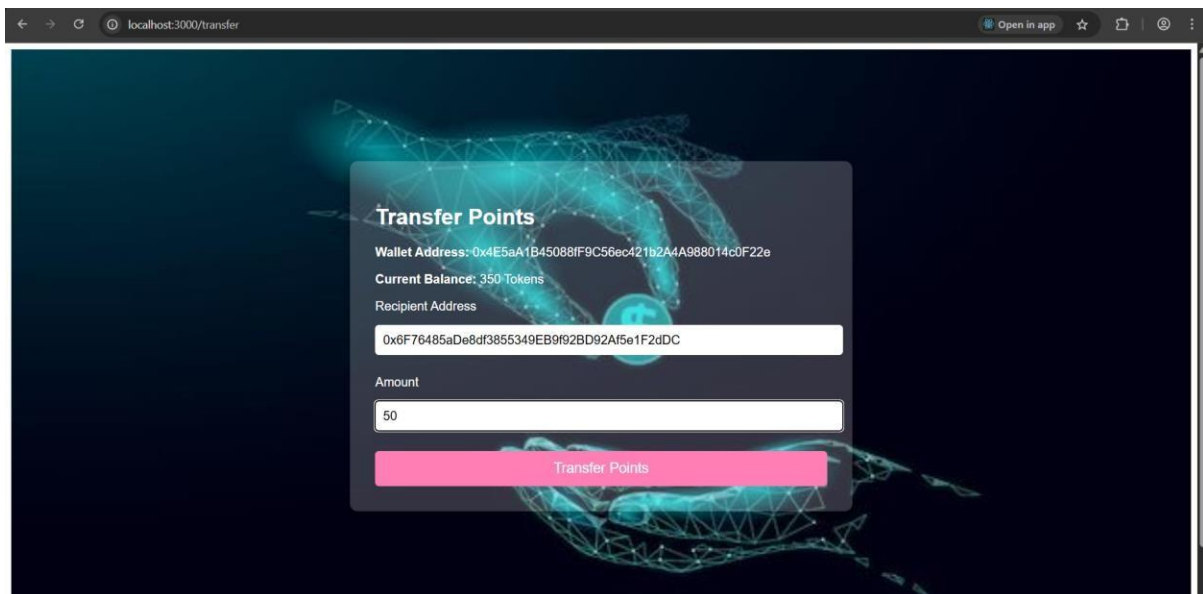
- TRANSFER POINTS PAGE



The screenshot shows a web browser window with the address bar displaying 'localhost:3000/transfer'. The page features a dark background with a glowing, wireframe hand holding a coin. A central form titled 'Transfer Points' contains the following information:

- Wallet Address:** 0x4E5a1B45088f9C56ec421b2A4A988014c0F22e
- Current Balance:** 350 Tokens
- Recipient Address:** A text input field with the placeholder '0x...'.
- Amount:** A text input field with the placeholder 'Enter amount'.
- Transfer Points:** A pink button at the bottom of the form.

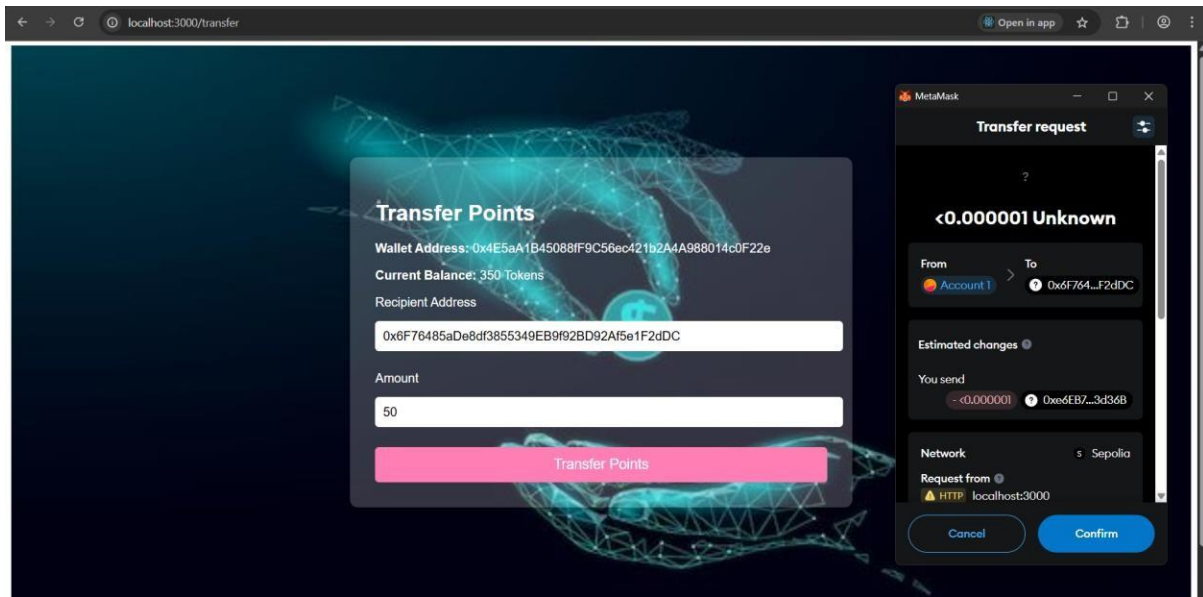
This is the Transfer Points page where the user's wallet address and current balance are visible. In addition to this, the user can transfer points by entering the wallet's address to which they want to transfer points as well as choosing the number of points to be transferred.



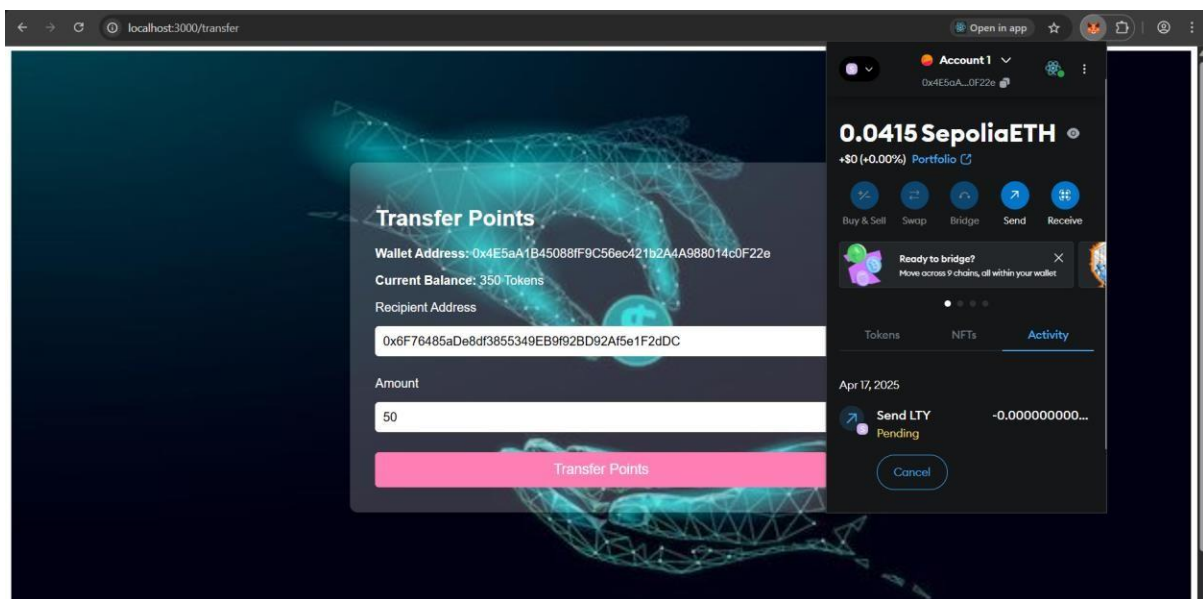
This screenshot shows the same 'Transfer Points' form, but with the following changes:

- Recipient Address:** The text input field now contains the address '0xbF76485aDe8df3855349EB9f92BD92A15e1F2dDC'.
- Amount:** The text input field now contains the value '50'.
- Transfer Points:** The pink button remains at the bottom of the form.

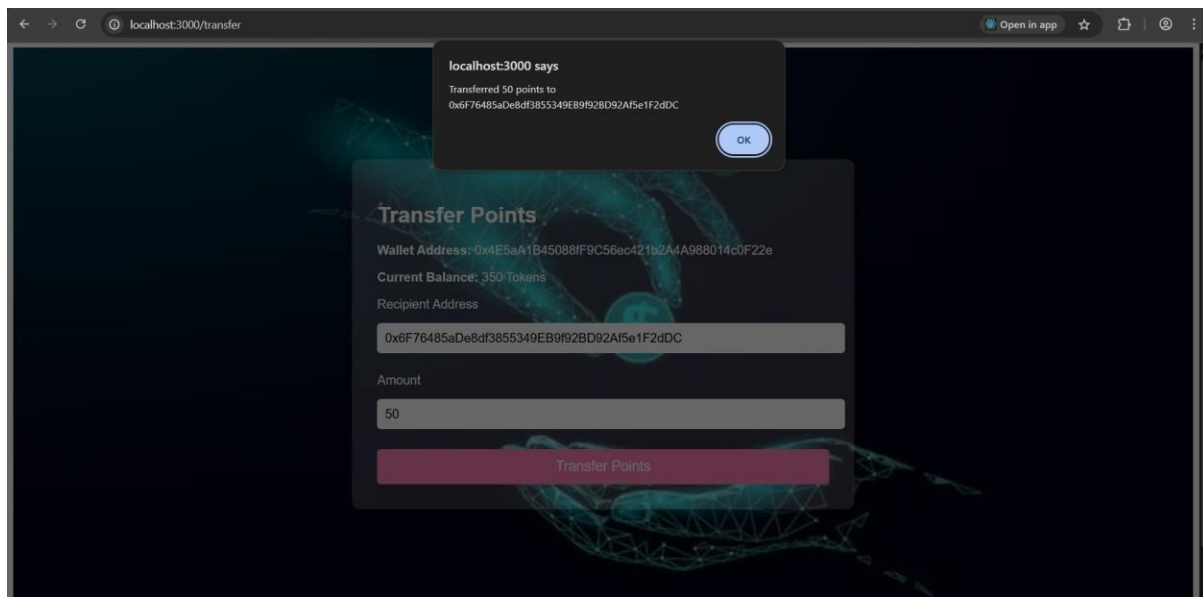
Here, the user has chosen the number of points to be transferred as well as the wallet's address to whom they want to transfer.



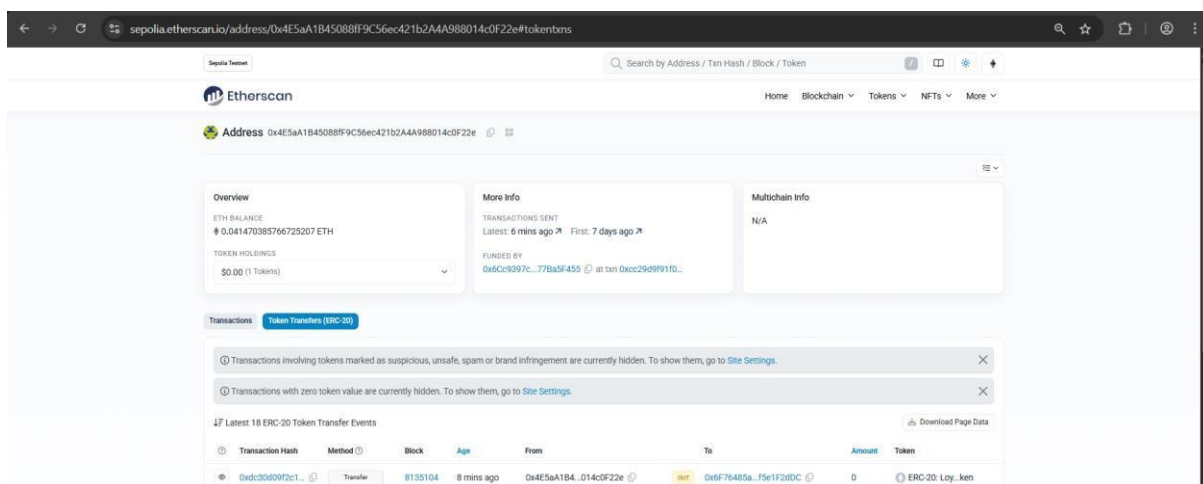
Here MetaMask shows a notification waiting for the user to either confirm or cancel the transfer.



Here the transfer is pending, indicating that it is in progress.



Upon confirmation, local host also gives a confirmation notification.



On Etherscan, it shows 'out' for the sender's wallet address indicating that points have been deducted and has been transferred to the receiver's wallet address.

The screenshot shows the Etherscan interface for the LoyaltyToken (LTY) token. The token is ERC-20. The overview section shows a max total supply of 0.00000000000000000462 LTY and 2 holders. The market section shows an on-chain market cap of \$0.00. The other info section shows the token contract address: 0xe6eb7768524092023cc1b9c00b4a45180a3d36b7a-0x4e5aa1b45088ff9c56ec421b2a4a988014c0f22e. The filtered by token holder section shows the address 0x4E5aA1B45088FF9C56ec421b2A4A988014c0F22e with a balance of 0.000000000000000003 LTY. The transactions section shows a list of transactions, with the most recent one being a transfer of 0.000000000000000005 LTY to the address 0x6F76485a...f5e1f2dDC.

Shows the total amount that has been transferred in terms of LTY.

The screenshot shows the Etherscan interface for the address 0x6F76485aDe8dF3855349EB9F92B092AF5e1F2dDC. The overview section shows an ETH balance of 0.039733983763223749 ETH and token holdings of \$0.00 (2 Tokens). The more info section shows the latest transaction sent 7 days ago. The multichain info section shows N/A. The transactions section shows a list of transactions, with the most recent one being a transfer of 0 LTY to the address 0x6F76485a...f5e1f2dDC. The transaction status is 'in', indicating that the points have been received successfully.

From the receiver's address, we can see that the latest transaction shows 'in', indicating that points have been received successfully.

FUTURE ENHANCEMENTS

- Reward Tiers:
Businesses could set up different levels of rewards (like silver, gold, or platinum) to encourage users to spend more and earn points at higher rates.
- Support for Multiple Currencies:
Adding support for more types of tokens or currencies, like NFTs (Non-Fungible Tokens), could make the platform more flexible and allow for more diverse rewards.
- Mobile App:
We could create a mobile version of the app using React Native, so users can access the loyalty system from their phones and easily manage their points from anywhere.