Natcha Karnsakultorn (SID: 20816364)

Green Trace App Functionality and Test Logs

Introduction

GreenTrace is an eco-friendly activity logging platform designed to encourage users to track and manage their carbon footprint. By logging various activities, users can gain insights into their environmental impact and earn rewards in the form of eco-tokens. The application utilizes blockchain technology to ensure transparency, security, and reliability in tracking users' activities and rewards.

Technology Stack

The smart contract code is written in Solidity and deployed using Remix. The front end is developed with HTML and JavaScript, and the application is hosted on GitHub Pages.

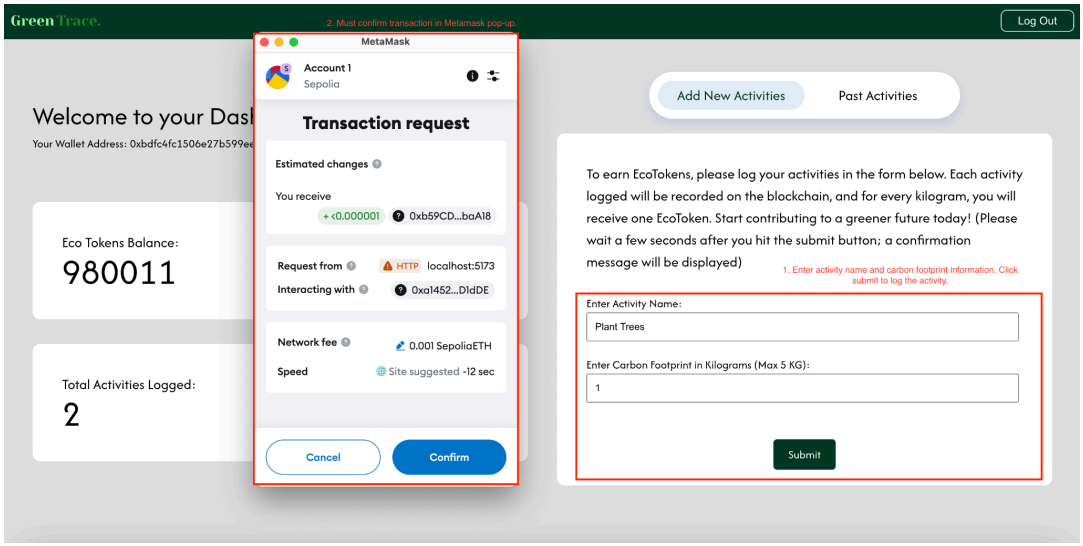
Deployed Link: https://natchakarnsa.github.io/
ETH Faucet Used: Ethereum Sepolia (obtained from this faucet)

Key Functionalities

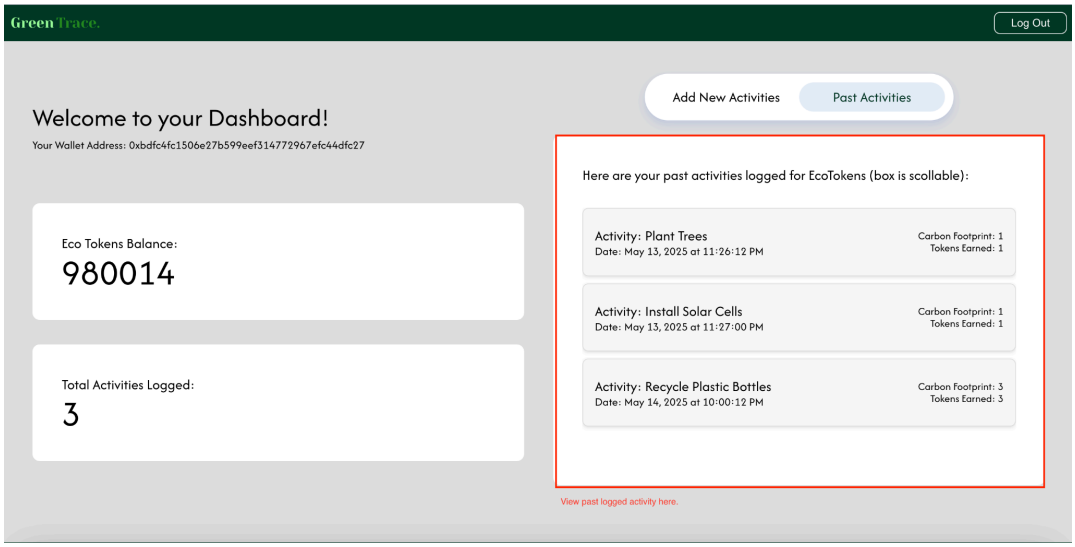
1. **User Registration and Authentication:** Users can connect to their metamask wallet directly from the user interface **(must have metamask extension installed)**

2. **Activity Logging:** Users can log various eco-friendly activities by entering the activity name and the associated carbon footprint (up to 5 kg). The application captures the date and time of each

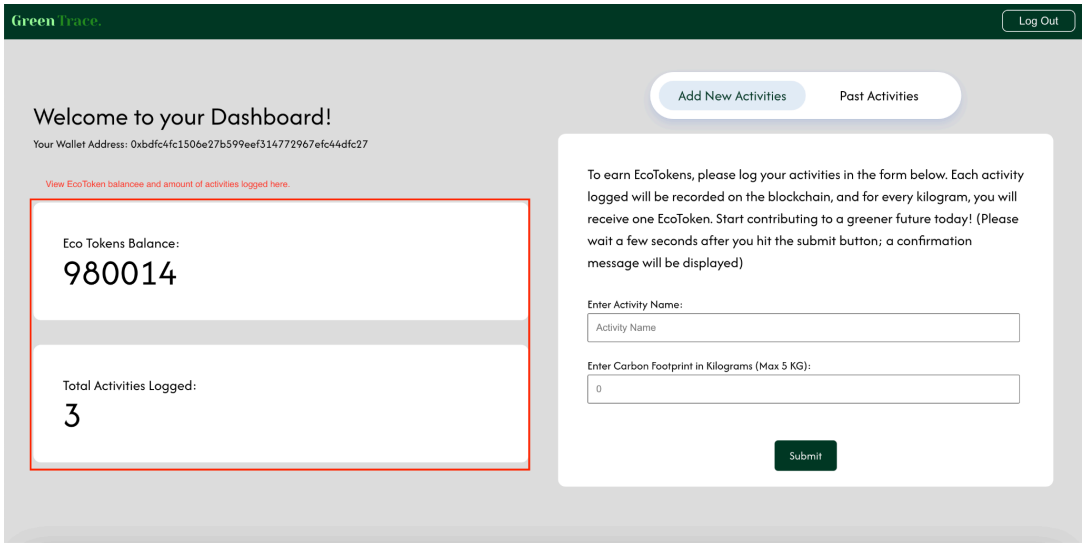
logged activity.



3. **Carbon Footprint Tracking:** Users can view a history of their logged activities including details such as activity name, carbon footprint, and date of logging.



4. **Reward System:** Users will earn one eco token per each kilogram of carbon footprint logged. The tokens will automatically be transferred to their wallet upon successful logging.



5. **Eco Token Balance:** Users can check their current balance of eco tokens. The idea is that in the future, these can be exchanged for various rewards or eco-friendly initiatives.
6. **Smart Contract Integration:** The application connects to a smart contract via web3 to handle activity logging, token distribution, and maintaining transparency.

Testing (From https://mycontract.fun/ Platform):

Testing was divided into 2 parts: EcoToken.sol and GreenTrace.sol

1. EcoToken.sol (0xb59CD39288438AD31ae0D19096D4eAFf3f3baA18) : Note that import "@openzeppelin/contracts/token/ERC20/ERC20.sol"; was changed to import { ERC20 } from "../lib/openzeppelin-contracts/contracts/token/ERC20/ERC20.sol"; for testing purposes

   Test Result: View Here
   Generated Test Cases: View Here

2. GreenTrace.sol (0xa14524EFa1A6C04Ad1c31055F4D8ff5C359D1dDE): Note that instead of using import "./EcoToken.sol"; the code was manually pasted inside GreenTrace.sol for testing purposes.

   Test result: View Here
   Generated Test Cases: View Here

Security Considerations

1. EcoToken.sol:
    a. The initial supply is minted to the initial deployment address to ensure safety (token distribution starts with a known entity).
    b. Utilized the ERC20 standard, which is designed to minimize vulnerabilities, including protection against reentrancy attacks.
    c. The ERC20 implementation handles pre-transfer validation checks, ensuring the sender has sufficient balance when using transferFrom
2. GreenTrace.sol:
    a. The contract uses custom errors (e.g., ERC20InsufficientBalance) to help prevent unexpected behavior during token transfers.
    b. Prior to transferring tokens, the contract checks its own balance to ensure it can fulfill the reward.
    c. Events are emitted for critical actions (e.g., activity logging) to enhance transparency.
    d. Each logged activity includes a timestamp, providing a historical context for users.