

CROWDFUNDING DAPP ON ETHEREUM TESTNET

BLOCKCHAIN TECHNOLOGIES 1 — FINAL
EXAMINATION PROJECT

Students:

Nauryzbay Tomiris, Aisulu Azimkhan, Medina Klyumova

Instructor: Sayakulova Zarina

ASTANA IT UNIVERSITY

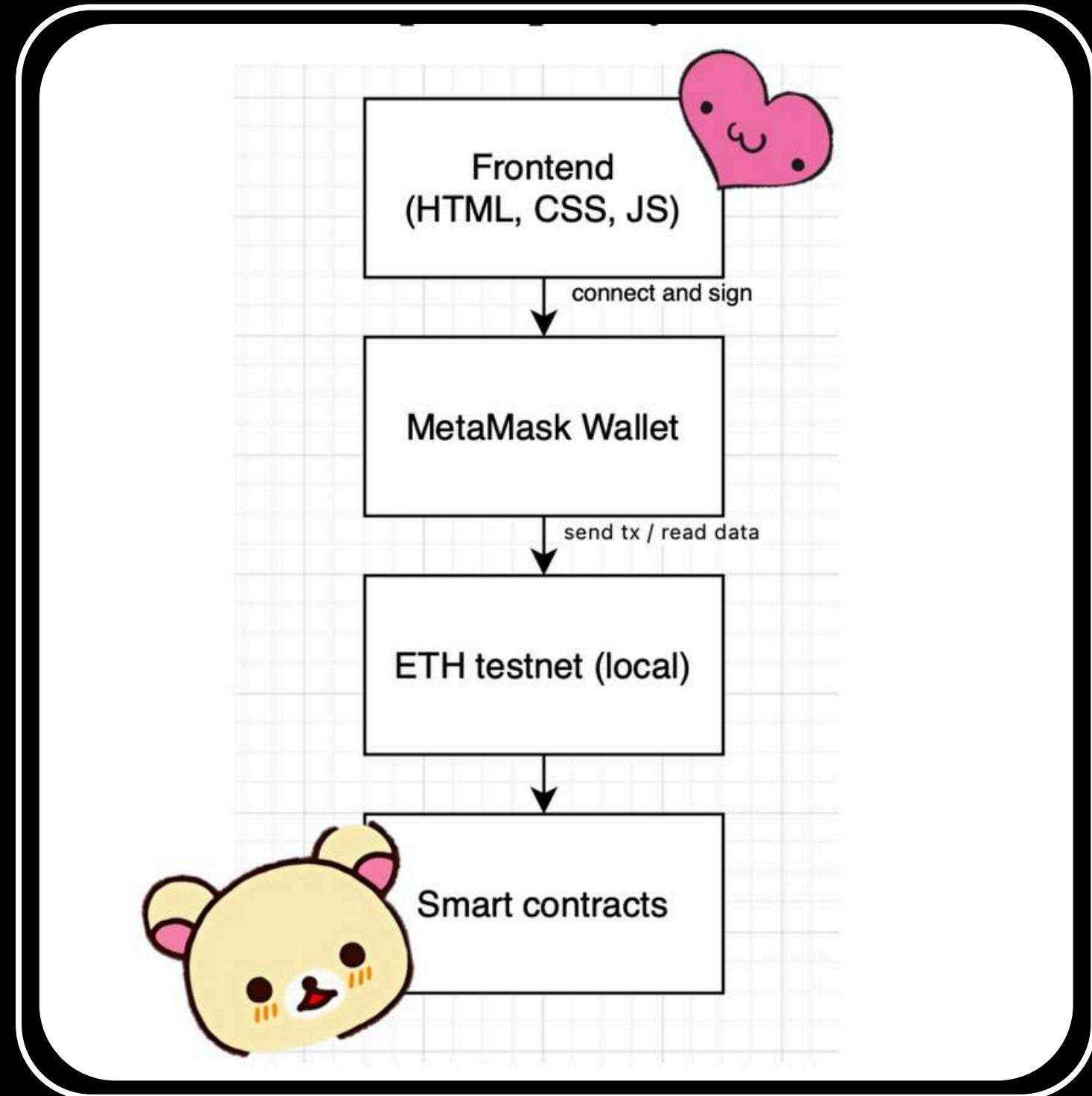
PROJECT GOAL

- Build a decentralized crowdfunding application
 - Replace centralized intermediaries with smart contracts
- Enable campaign creation and ETH contributions
 - Use Ethereum test networks for safe experimentation

TECHNOLOGY STACK

- Solidity — smart contract development
- JavaScript — frontend logic and interaction
- MetaMask — wallet connection and transaction signing
- Ethereum (Testnet / Localhost) — blockchain execution
- Hardhat — development and deployment framework





SYSTEM ARCHITECTURE

WEB3 ARCHITECTURE

→ METAMASK

→ ETHEREUM TESTNET

→ Smart Contracts

SMART CONTRACTS OVERVIEW

Two smart contracts are used:

- Crowdfunding Contract
- ERC-20 Reward Token Contract



Crowdfunding.sol



RewardToken.sol

CAMPAIGN DATA STRUCTURE

EACH CAMPAIGN STORES:

- FUNDING GOAL
- DEADLINE
- OWNER ADDRESS
- TOTAL RAISED AMOUNT
- FINALIZATION STATUS

```
Crowdfunding.sol ^0.8.20
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

interface IRewardToken {
    function mint(address to, uint256 value) external returns (bool);
}

/// @title Crowdfunding - simple crowdfunding with reward token
contract Crowdfunding {
    struct Campaign {
        string title;
        uint256 goal;
        uint256 deadline;
        address owner;
        uint256 totalRaised;
        bool finalized;
    }

    uint256 public campaignCount;
    mapping(uint256 => Campaign) public campaigns;
    mapping(uint256 => mapping(address => uint256)) public contributions;

    IRewardToken public immutable rewardToken;

    // 1000 tokens per 1 ETH (1e18 wei)
    uint256 public constant TOKENS_PER_ETH = 1000;
```

CAMPAIN DATA STRUCTURE

On-chain storage model for crowdfunding
campaigns and contributions

```
struct Campaign {  
    string title;  
    uint256 goal;  
    uint256 deadline;  
    address owner;  
    uint256 totalRaised;  
    bool finalized;  
}  
  
uint256 public campaignCount;  
mapping(uint256 => Campaign) public campaigns;  
mapping(uint256 => mapping(address => uint256)) public contributions;  
  
IRewardToken public immutable rewardToken;
```

CAMPAIGN CREATION LOGIC

New campaigns are created on-chain with a funding goal and a time-based deadline.

```
function createCampaign(string calldata title, uint256 goal,  
uint256 durationSeconds) external returns (uint256) {  
    require(bytes(title).length > 0, "TITLE_EMPTY");  
    require(goal > 0, "GOAL_ZERO");  
    require(durationSeconds > 0, "DURATION_ZERO");  
  
    uint256 id = ++campaignCount;  
    uint256 deadline = block.timestamp + durationSeconds;  
  
    campaigns[id] = Campaign({  
        title: title,  
        goal: goal,  
        deadline: deadline,  
        owner: msg.sender,  
        totalRaised: 0,  
        finalized: false  
    });  
  
    emit CampaignCreated(id, msg.sender, title, goal, deadline);  
    return id;  
}
```

CONTRIBUTION AND REWARD LOGIC

Users send test ETH contributions via MetaMask. Each contribution is recorded on-chain and automatically triggers reward token minting.

The image consists of two side-by-side screenshots. On the left, a dark-themed web interface shows a 'Contribute' button, a progress bar indicating 6 contributions at 0.2 each, and a message 'Contribution sent'. On the right, a white-themed MetaMask transaction history shows two entries: one for 'Contribute' on Feb 9, 2026, and another for 'Create Campaign' on the same date. Both transactions are marked as 'Подтверждено' (Confirmed) and show a value of -0.2 ETH and -0 ETH respectively, with a corresponding amount of -422,26 \$.

Date	Action	Status	Value (ETH)	Value (\$)
Feb 9, 2026	Contribute	Подтверждено	-0.2 ETH	-422,26 \$
Feb 9, 2026	Create Campaign	Подтверждено	-0 ETH	-0,00 \$

ERC-20 REWARD TOKEN

- Custom ERC-20 token is used as a reward mechanism
- Tokens are minted automatically during user contribution
- The token has no real monetary value
- Used only for educational and demonstration purposes

The screenshot shows a dark-themed blockchain application interface. On the left, there's a "Finalize / Refund" section with a progress bar. The first segment is grey with the number "6" and the word "Finalized". The second segment is orange with the word "Finalize". To its right is a grey segment with the word "Refund". Above this section, there are navigation tabs: "Токены", "DeFi", "NFT", and "Деятельность", with "Деятельность" being underlined. To the right of the tabs is a dropdown menu showing "localhost 8545". Below the tabs, a timestamp "Feb 9, 2026" is displayed. A campaign entry is listed: "Finalize Campaign" with a status of "Подтверждено" (Confirmed) and a note "-0 ETH -0,00 \$".

METAMASK INTEGRATION

The image displays two side-by-side screenshots illustrating the integration of MetaMask with a blockchain application.

Left Screenshot: Simple Crowdfunding Application

- Header:** BLOCKCHAIN TECHNOLOGIES 1
- Title:** Simple Crowdfunding
- Description:** A minimal on-chain crowdfunding demo with reward tokens. Built for Ethereum test networks and MetaMask.
- Buttons:** Solidity, MetaMask, Testnet
- Connect MetaMask:** A prominent orange button with the text "Connect MetaMask". Below it, a smaller text says "Connect your wallet to start."
- Information Boxes:**
 - Wallet: 0x70997970C51812dc3AO10C7d01b50e0d17dc79C8
 - Network: unknown (31337)
 - ETH Balance: 9999.7980380106027961 38 ETH
 - Reward Balance: 0.00000000000000000000000000000008 С RWD
- Create Campaign:** A section with three input fields: "Campaign title", "Goal in ETH", and "Duration in seconds". Below them is a large orange "Create" button.
- Contribute:** A section with two input fields: "Campaign ID" and "Amount in ETH". Below them is a large orange "Contribute" button.

Right Screenshot: MetaMask Wallet Interface

- Header:** MetaMask
- Title:** Imported Account 2
- Balance:** 0,00 \$ +0,00 \$ (+0,00 %) [Обзор](#)
- Actions:** Купить, Обмен..., Отправ..., Получи...
- Bitcoin Advertisement:** Say hello to Bitcoin. Trade, manage, and buy BTC directly on MetaMask.
- Activity Tab:** Токены, DeFi, NFT, Деятельность (selected)
- Activity Log:** Localhost 8545
- | Date | Action | Value |
|-------------|-------------------|---------------------|
| Feb 9, 2026 | Refund | -0 ETH -0,00 \$ |
| | Finalize Campaign | -0 ETH -0,00 \$ |
| | Contribute | -0.2 ETH -422,26 \$ |
| | Create Campaign | -0 ETH -0,00 \$ |

FRONTEND INTERFACE

Campaign List

Load All Campaigns

Loaded 6 campaigns.

#1 — tomitomi

Goal: 1.0 ETH
Raised: 0.2 ETH
Deadline: 2026-02-08T20:59:32.000Z
Owner: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
Finalized: true

#2 — Tomi

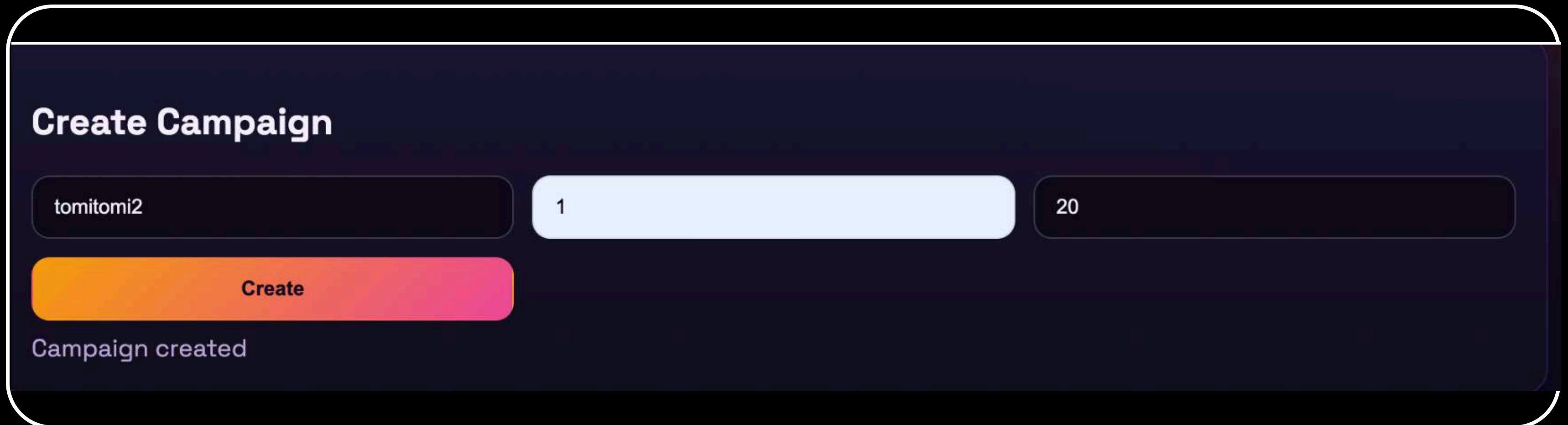
Goal: 1.0 ETH
Raised: 0.0 ETH
Deadline: 2026-02-08T22:37:05.000Z
Owner: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
Finalized: true

#3 — sloniktomi

Goal: 1.0 ETH
Raised: 0.2 ETH
Deadline: 2026-02-08T22:40:58.000Z
Owner: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
Finalized: true

#4 — sloniktomi1

Goal: 1.0 ETH
Raised: 0.2 ETH



CAMPAIN CREATION (DEMO)

This demo demonstrates how a user creates a crowdfunding campaign by specifying a title, funding goal, and duration. The transaction is sent via MetaMask and stored on the blockchain.

CONTRIBUTION RESULT (DEMO)

This demo shows a successful ETH contribution to an active campaign.

The transaction is signed and confirmed via MetaMask and recorded on the blockchain.

The image shows a mobile application interface. On the left, a dark-themed screen displays a "Contribute" button, a progress bar with the value "0.2", and a message "Contribution sent". On the right, a light-themed screen shows a transaction history for February 9, 2026. The first transaction is a "Contribute" entry with a blue icon, showing "-0.2 ETH" and "-422,26 \$". The second transaction is a "Create Campaign" entry with a blue icon, showing "-0 ETH" and "-0,00 \$". Both transactions have a green status indicator "Подтверждено" (Confirmed).

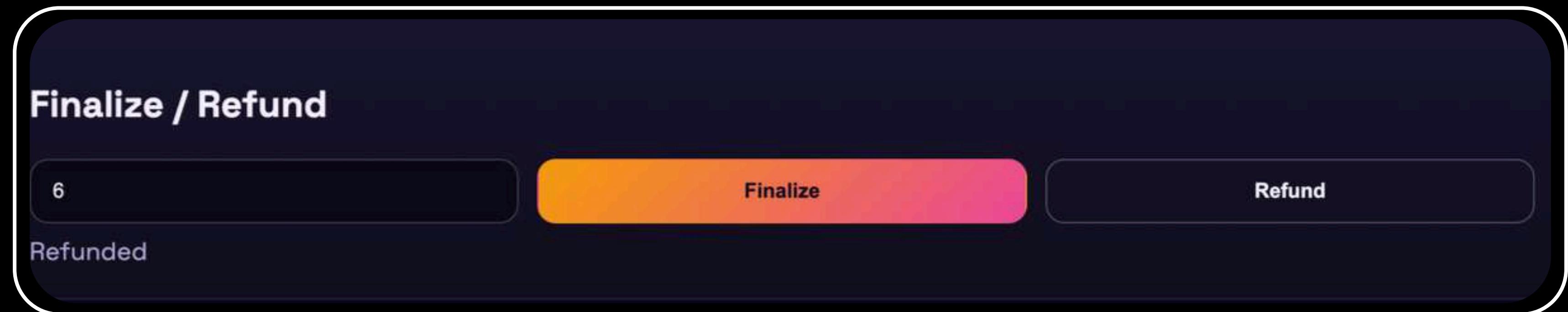
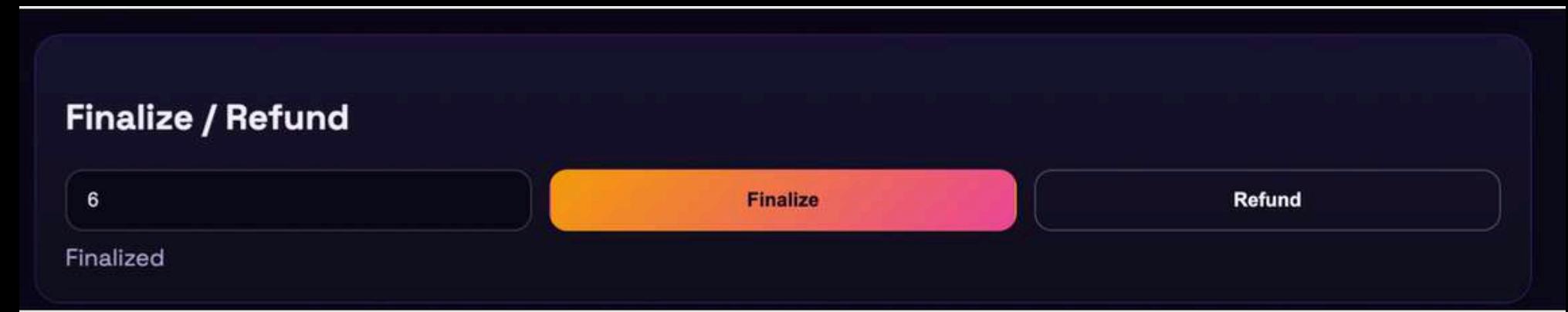
Date	Action	Value	Fee
Feb 9, 2026	Contribute	-0.2 ETH	-422,26 \$
Feb 9, 2026	Create Campaign	-0 ETH	-0,00 \$

FINALIZATION & REFUND (DEMO)

After the campaign deadline, the campaign can be finalized.

If the funding goal is reached, funds are transferred to the campaign owner.

Otherwise, contributors can refund their ETH.



CONCLUSION

This project demonstrates a fully functional decentralized crowdfunding application. Smart contracts manage campaign logic, contributions, and refunds transparently. ERC-20 reward tokens illustrate tokenization concepts.

The application operates entirely on an Ethereum test network.