



help



DECENTRALIZED  
**CROWDFUNDING PLATFORM**

BY AIBEK, AIZADA, VALERIY



[https://github.com/KhanOfTheMoon/Blockchain\\_final](https://github.com/KhanOfTheMoon/Blockchain_final)



2026



# OUR TEAM



**AIBEK**



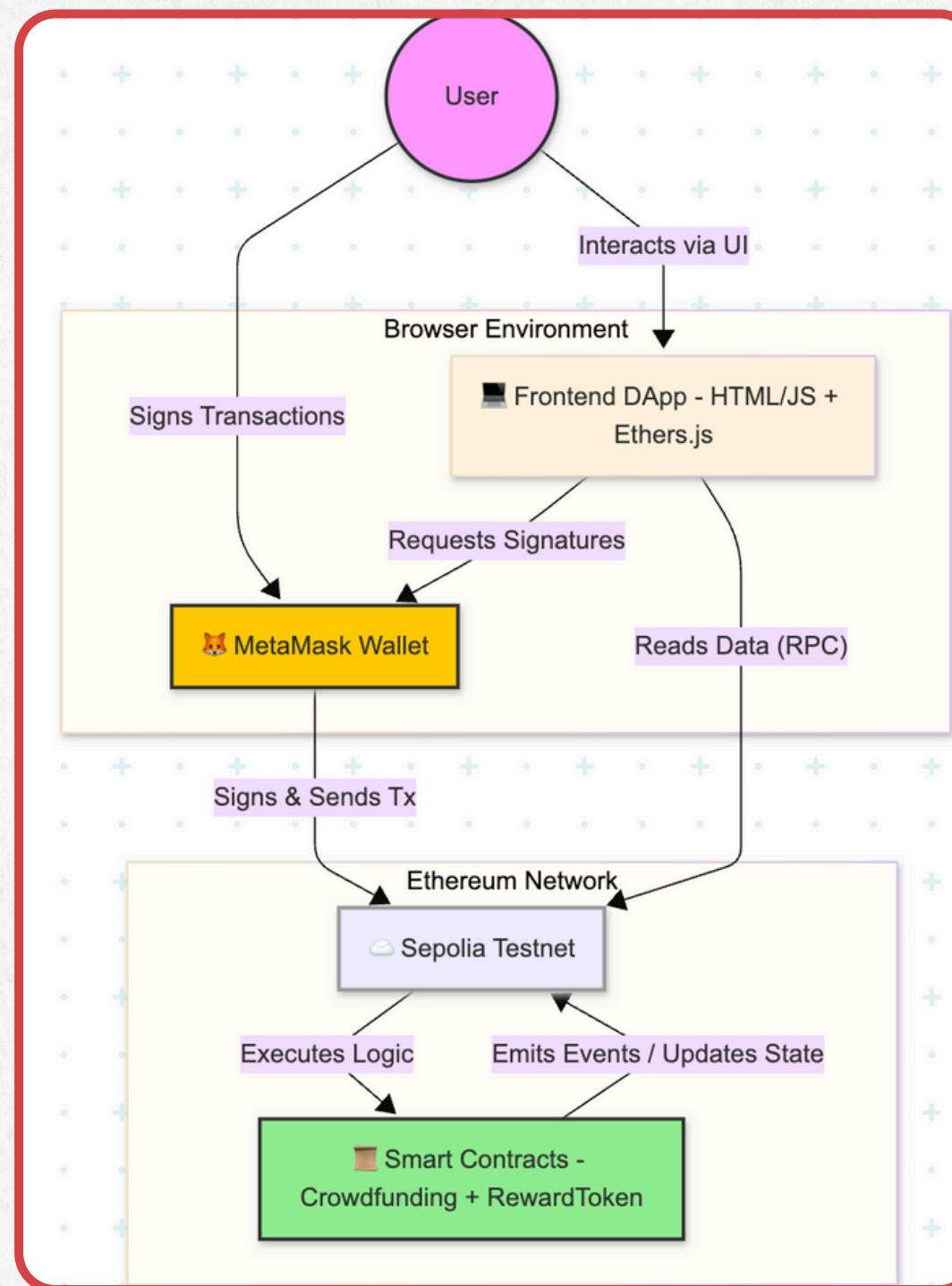
**AIZADA**



**VALERIY**

# PROJECT OVERVIEW & ARCHITECTURE

blockchain



## GOAL

A DApp allowing users to create fundraising campaigns and receive "RewardTokens" (RWD) for donating ETH.

## TECH STACK

- Smart Contracts:** Solidity 0.8.20 (Hardhat Framework).
- Frontend:** Vanilla JS, Ethers.js (v6).
- Network:** Ethereum Sepolia Testnet.
- Wallet:** MetaMask Integration.



# SMART CONTRACT STRUCTURE

blockchain

## CORE LOGIC: CAMPAIGN STRUCTURE

- The Campaign struct stores vital information for each project.
- campaigns mapping enables O(1) access to data.

```
struct Campaign {  
    string title;  
    address payable creator;  
    uint256 goalWei;  
    uint256 deadline;  
    uint256 raisedWei;  
    bool finalized;  
    bool successful;  
}  
  
uint256 public campaignCount;  
mapping(uint256 => Campaign) public campaigns;
```





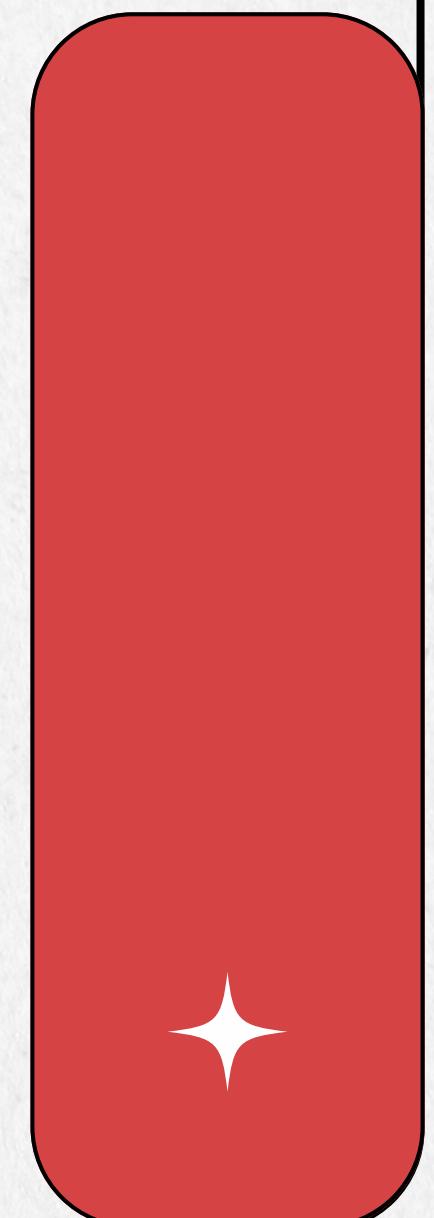
# CONTRIBUTION & REWARD LOGIC

blockchain

## CONTRIBUTIONS & TOKEN REWARDS

- Users donate ETH and automatically receive ERC-20 Reward Tokens.
- Rate: 1 ETH = 100 Reward Tokens.
- Checks deadline and finalized status before execution.

```
function contribute(uint256 id) external payable nonReentrant {  
    c.raisedWei += msg.value;  
  
    uint256 rewardAmount = 0;  
    if (address(rewardToken) != address(0)) {  
        rewardAmount = (msg.value * rewardPerEth) / 1 ether;  
        rewardToken.mint(msg.sender, rewardAmount);  
    }  
}
```





# FRONTEND INTEGRATION (WEB3)

blockchain

## FRONTEND-BLOCKCHAIN INTERACTION

```
connectWallet: async () => {
  if (!window.ethereum) return alert("Install MetaMask");
  AppState.provider = new ethers.BrowserProvider(window.ethereum);
  AppState.signer = await AppState.provider.getSigner();
  AppState.address = await AppState.signer.getAddress();

  AppState.cfContract = new ethers.Contract(
    CONFIG.CROWDFUNDING_ADDRESS,
    ABIS.CROWDFUNDING,
    AppState.signer
);
```

- Using Ethers.js v6 for wallet connection and contract calls.
- BrowserProvider connects to the window.ethereum object (MetaMask).

2024





# DESIGN & SECURITY DECISIONS

blockchain

## KEY DESIGN & SECURITY DECISIONS

- **Reentrancy Protection:** Used OpenZeppelin's ReentrancyGuard on the contribute function to prevent recursive attacks.
- **Access Control:** Used Ownable pattern to restrict administrative functions (e.g., setting the reward token address).
- **Checks-Effects-Interactions:** State is updated (`raisedWei += msg.value`) before external calls (minting tokens).
- **Testnet Only:** Strict checks to ensure deployment on Sepolia, not Mainnet.





# DEPLOYMENT & TEST ETH

blockchain

## DEPLOYMENT & EXECUTION

- **Network:** Sepolia Testnet.
- **Crowdfunding Address:**  
0x14B2bfDbF6413FaF3448b28a49028  
73f99290c0b

- **RewardToken Address:**  
0x0cbC71037f588fC662868B062a6d  
69AC5CbDcc48
- **Test ETH:** Obtained via Google  
Cloud Web3 Faucet and Alchemy  
Sepolia Faucet.

## SCREENSHOT

```
PS C:\Users\Valeriy\Desktop\Desktop\ДЗ\Blockchain\FINAL> npx hardhat run scripts/deploy.js --network sepolia
[dotenv@17.2.3] injecting env (2) from .env -- tip: ⚡ add observability to secrets: https://dotenvx.com/ops
[dotenv@17.2.3] injecting env (0) from .env -- tip: ⚡ enable debug logging with { debug: true }
TOKEN_ADDRESS = 0x0cbC71037f588fC662868B062a6d69AC5CbDcc48
CROWDFUNDING_ADDRESS = 0x14B2bfDbF6413FaF3448b28a4902873f99290c0b
```



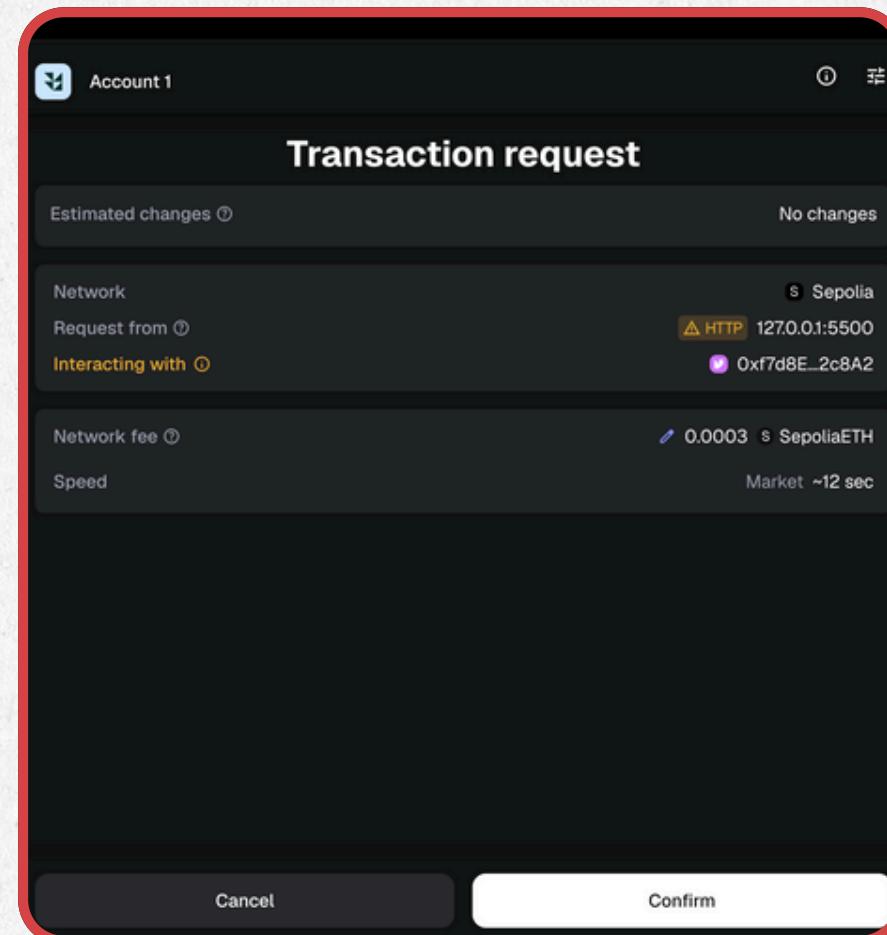


# USER INTERFACE (DEMO)

blockchain

## APPLICATION INTERFACE

1. Dashboard:
2. Active Campaigns:
3. MetaMask Popup:

A screenshot of the CrowdStarter dashboard. It displays:

- ETH BALANCE: 0.0225 ETH
- Reward TOKENS: 1.7 RWD
- Launch a Campaign section with fields for Campaign Title, Goal (ETH), Duration (seconds), and a "Create Campaign" button.
- Active Campaigns section showing three campaigns:
  - #4 Test tset: FAILED, Goal: 0.03 ETH, Raised: 0.004 ETH, Deadline: 07.02.2026, 23:42:36
  - #3 Test Campaign3: SUCCESS, Goal: 0.002 ETH, Raised: 0.003 ETH, Deadline: 07.02.2026, 23:02:00
  - #2 Test Campaign2: SUCCESS, Goal: 0.01 ETH, Raised: 0.01 ETH, Deadline: 07.02.2026, 22:11:24

A screenshot of a campaign detail view for "Test Campaign5". It shows:

- Campaign status: ACTIVE
- Campaign name: Test Campaign5
- Goal: 0.001 ETH
- Raised: 0.0 ETH
- Deadline: 08.02.2026, 21:03:48
- A input field with value "0.01" and a "Donate" button.





# QUALITY ASSURANCE & TESTING

blockchain



**Automated Testing:** Developed unit tests using Hardhat & Chai framework.

**Coverage:** Tests verify critical functions:

- Campaign creation & validation.
- Contributions & Reward Token minting.
- Withdrawals & Campaign finalization.

**Security Checks:** Tested ReentrancyGuard and Ownable restrictions.

2024

```
PS C:\Users\Valeriy\Desktop\Desktop\ДЗ\Blockchain\FINAL> npx hardhat test test/RewardToken.test.cjs
[dotenv@17.2.3] injecting env (2) from .env -- tip: ⚡ add access controls to secrets: https://dotenvx.com/ops

RewardToken Contract
Deployment
  ✓ Should set the right owner
  ✓ Should have correct name and symbol
Minting
  ✓ Should allow owner to mint
  ✓ Should allow designated minter to mint
  ✓ Should fail if unauthorized account tries to mint
Access Control
  ✓ Should only allow owner to set minter

6 passing (145ms)

Solidity and Network Configuration
| Solidity: 0.8.20 · Optim: false · Runs: 200 · viaIR: false · Block: 60,000,000 gas
|
Methods
Contracts / Methods · Min · Max · Avg · # calls · usd (avg)
RewardToken
mint · 71,223 · 73,416 · 72,320 · 2 · -
setMinter · - · - · 46,406 · 1 · -
Deployments
RewardToken · - · - · 1,223,474 · 2 % · -
Key
  ⓘ Execution gas for this method does not include intrinsic gas overhead
  ⓘ Cost was non-zero but below the precision setting for the currency display (see options)
Toolchain: hardhat
```



blockchain →

THANK YOU  
FOR  
**ATTENTION**

[https://github.com/KhanOfTheMoon/Blockchain\\_final](https://github.com/KhanOfTheMoon/Blockchain_final) →

2026

