Open Source **Funding DApp**

SE-2432, Astana IT University
Blockchain Technologies - Final Project

Alisher Amangeldi & Nurzhan Nurlybek

Search

What is it

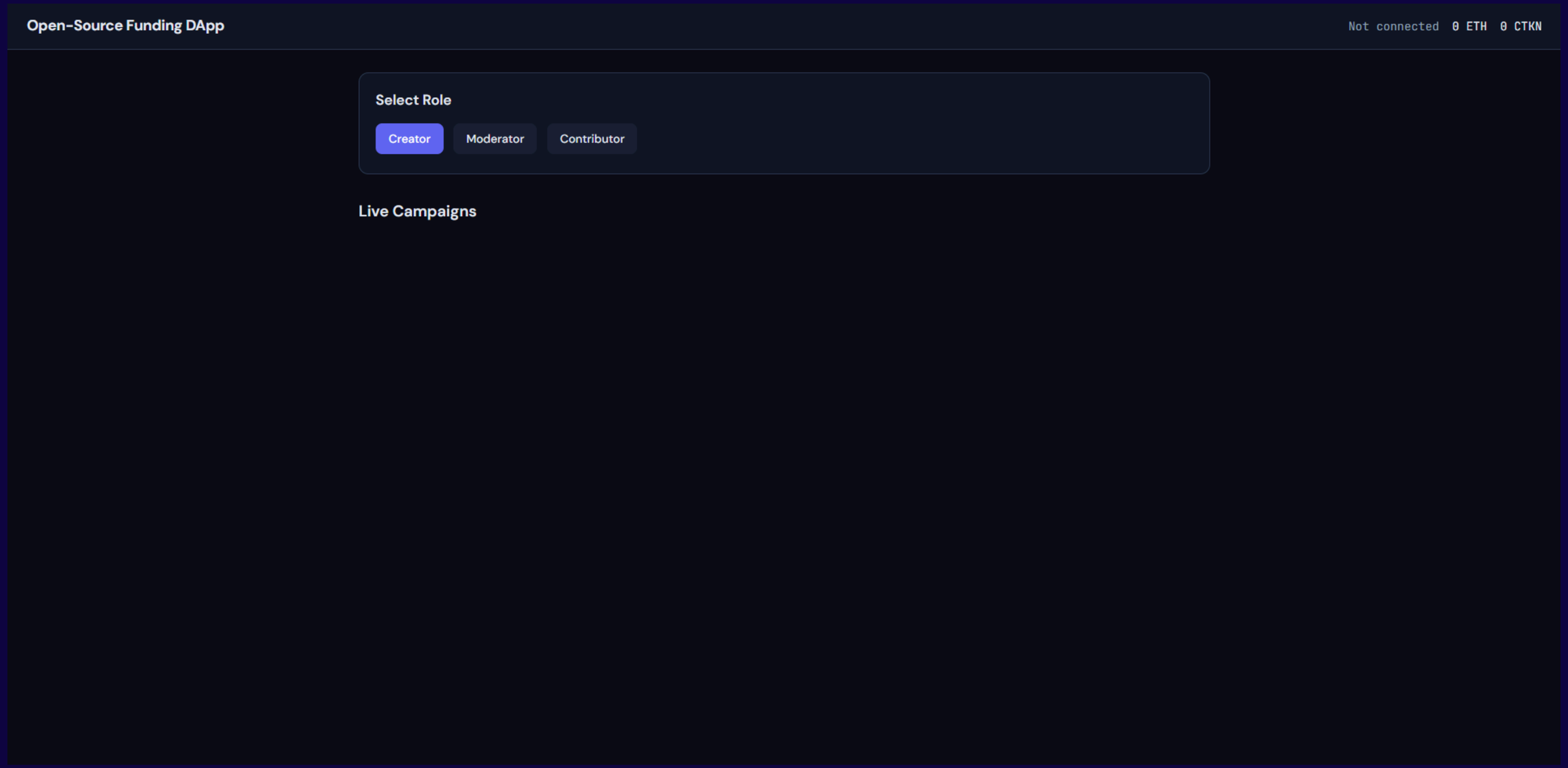
Decentralized crowdfunding platform on Ethereum

Reward tokens (ERC-20) + NFT badges (ERC-721)

3 roles: Creator, Moderator, Contributor

Creators submit campaigns
Moderator approves
Contributors fund with ETH

- Testnet only, no real money


Open-Source Funding DApp
Not connected
0 ETH
0 CTKN
Select Role
Creator
Moderator
Contributor
Live Campaigns

Search

Tech Stack

- Solidity 0.8.28 + OpenZeppelin (contracts)
- Hardhat (dev environment + testing)
- Ethers.js 6 (frontend-to-chain)
- MetaMask (wallet)
- Vanilla HTML/CSS/JS (frontend)
- Networks: Hardhat local (31337), Sepolia, Holesky

Architecture

Search

Metamask

Frontend
(app.js + Ethers.js)

Crowdfunding.sol

ContributorToken.sol
(ERC-20)

ContributorBadge.sol
(ERC-721)

Search

Smart Contracts

Crowdfunding.sol

- Campaign logic
- Contributions
- Refunds
- Withdrawals
- Calls token contract to mint/burn

ContributorToken.sol

- ERC-20 "CTKN"
- onlyOwner mint
- Anyone can burn their own

ContributorBadge.sol

- ERC-721 "CBADGE"
- Stores URI per token

Search

Ownership Transfer
how contracts connect

1. Deploy ContributorToken => deployer is owner
2. Deploy Crowdfunding(tokenAddr)
3. token.transferOwnership(crowdfundingAddr) => Crowdfunding is now owner

- mint() has onlyOwner => only Crowdfunding can mint
- contribute() internally calls rewardToken.mint(msg.sender, msg.value)
- refund() internally calls rewardToken.burn(msg.sender, amount)

Campaign Lifecycle

Search

Submitted → [moderator approves] → Active → [goal reached] → Successful → Withdrawn

↓

[deadline passes, goal not reached]

↓

Failed → Refund

Submitted → [moderator rejects] → Rejected

Deadline = block.timestamp + duration (set on approval)

Auto-success if raised >= goal during contribution

finalize() needed only if deadline passes while Active

Search

Contribution flow **what happens in one tx**

When contribute(id) is called with ETH:

1. Checks: campaign is Active, deadline not passed, value > 0
2. First-time contributor → added to campaignContributors[id]
3. raised += msg.value
4. contributions[id][msg.sender] += msg.value
5. rewardToken.mint(msg.sender, msg.value) — 1 CTKN per 1 wei
6. If raised >= goal → state = Successful

Search

MetaMask **Integration**

eth_requestAccounts — asks user for wallet access
Chain ID validation: 31337 / 11155111 / 17000
BrowserProvider + Signer from Ethers.js
View functions (free reads) vs transactions (need confirmation)
Balances update on every block: provider.on("block", updateBalance)

Frontend Features

Search

- Role selection screen
- Creator dashboard (submit form)
- Campaign cards with progress bar, countdown, action buttons
- NFT badge selection modal
- Wallet info in header (address, ETH balance, CTKN balance)

Open-Source Funding DApp

0xf39F… 9999.99198 ETH 192.0 CTKN

Creator Dashboard

Title

Campaign title

Description

Describe what the campaign is for, how funds will be used, and why backers should contribute.

Goal (ETH)

e.g. 1.5

Duration (seconds)

e.g. 86400 (24 hours)

Submit Campaign

NFT Badge

Trade 10 CTKN for NFT Badge

Live Campaigns

test

srytsdgfg

Did not succeed

0 / 1 ETH

View Contributors

Open-Source Funding DApp

90% − + Reset 0xf39F… 9999.99198 ETH 192.0 CTKN

Moderator Dashboard

Review submitted campaigns below and approve or reject them.

NFT Badge

Trade 10 CTKN for NFT Badge

Live Campaigns

test

srytsdgfg

Did not succeed

0 / 1 ETH

View Contributors

test

adrdopidxpfijsdo

Withdrawn

10 / 10 ETH

View Contributors

test

sfeasdasdasd

Withdrawn

212 / 122 ETH

View Contributors

ыкпывавы

ывывпып

Rejected

0 / 1 ETH

View Contributors

Open-Source Funding DApp

0xf39F… 9999.99198 ETH 192.0 CTKN

Contributor Dashboard

Campaign ID (0-based)

ETH amount

Contribute

Refund (failed campaign)

NFT Badge

Trade 10 CTKN for NFT Badge

Live Campaigns

test

srytsdgfg

Did not succeed

0 / 1 ETH

View Contributors

Search

Testing

- 10 test cases in crowdfunding.test.js
- Happy path: submit → approve → contribute → success
- Failed campaign: finalize + refund with token burn
- Access control: only moderator approves, only creator withdraws
- Input validation: zero goal, zero duration rejected
- Contributor tracking verification
- Time manipulation: evm_increaseTime for deadline testing

```
Nurzhan@Nurzhan MINGW64 ~/Favorites/Basa/Education/Programming/Blockchain Technologies/Block-Chain-Final (main)
$ npx hardhat test
Compiled 2 Solidity files successfully (evm target: paris).


  Crowdfunding basic flows
    ✓ allows submit -> approve -> contribute and mints tokens (41ms)
    ✓ finalizes failed campaign and allows refund with token burn (44ms)
    ✓ rejects submitCampaign with zero goal
    ✓ rejects submitCampaign with zero duration
    ✓ only moderator can approve campaign
    ✓ only moderator can reject campaign
    ✓ only creator can withdraw from successful campaign
    ✓ tracks campaign contributors and contribution amounts
    ✓ contribute requires active campaign and reverts for rejected
    ✓ refund reverts when campaign is not failed
    ✓ campaignCount returns correct number of campaigns


  11 passing (743ms)
```

Solidity and Network Configuration				
Solidity: 0.8.28	Optim: false	Runs: 200	viaIR: false	Block: 60,000,000 gas
Methods				
Contracts / Methods	**Min**	**Max**	**Avg**	**# calls**	**usd (avg)**
ContributorToken					
transferOwnership	29,052	29,064	29,063	11	-
Crowdfunding					
approveCampaign	-	-	72,975	6	-
contribute	123,756	178,149	167,537	6	-
finalize	-	-	35,661	1	-
refund	-	-	47,249	1	-
rejectCampaign	-	-	48,542	2	-
submitCampaign	128,198	145,322	143,601	10	-
withdraw	-	-	40,817	1	-
Deployments			**% of limit**		
ContributorToken	-	-	1,214,915	2 %	-
Crowdfunding	2,194,681	2,194,693	2,194,691	3.7 %	-
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					

Design Decisions

- Single moderator (deployer) — simple but centralized, would use multisig/DAO in production
- 1 CTKN per 1 wei — proportional, no rounding issues
- Token burn on refund — keeps supply in sync with real contributions
- NFT trade is 2 separate txs (burn + mint) — not atomic, but simple
- Rejected/withdrawn campaigns hidden from non-moderators

Search

Thank**you**

GitHub: github.com/Xurmayo/Block-Chain-Final