



INVOICEFUND – DECENTRALIZED INVOICE CROWDFUNDING PLATFORM

Daniyal Adilbekov SE-2435
Bibifatima Bisesheva SE-2437
Ataniyaz Mutigolla SE-2437

PROBLEM

- **BUSINESSES WAIT LONG FOR INVOICE PAYMENTS**
- **CASH-FLOW PROBLEMS**
- **BANKS/INTERMEDIARIES = SLOW & EXPENSIVE**

SOLUTION

INVOICEFUND — DECENTRALIZED CROWDFUNDING PLATFORM

- **EARLY LIQUIDITY FOR BUSINESSES**
- **SMART CONTRACTS INSTEAD OF BANKS**
- **REWARD TOKENS FOR CONTRIBUTORS**



INVOICEFUND OVERVIEW



InvoiceFund is a decentralized crowdfunding platform that:

- **Operates on Ethereum Sepolia Testnet**
- **Uses Smart Contracts**
- **Uses MetaMask for wallet interaction**
- **Issues ERC-20 reward tokens**
- **Works only with free test ETH**

Participants fund campaigns and receive reward tokens as proof of contribution.

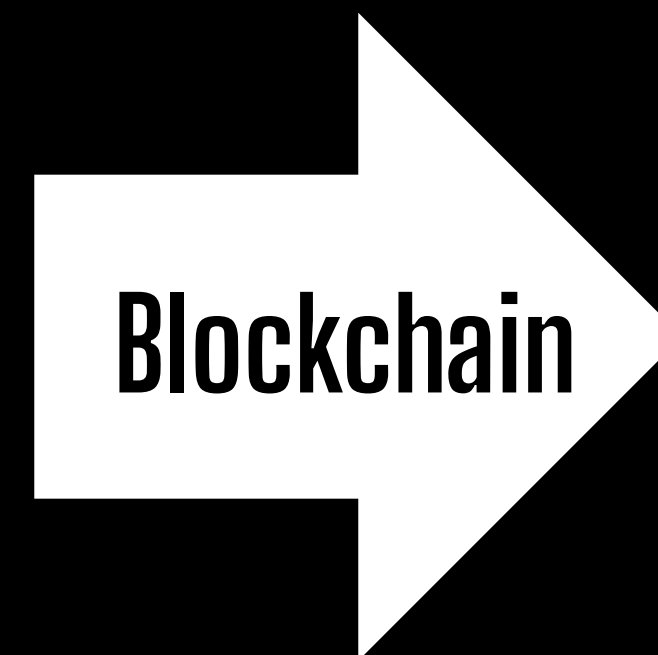
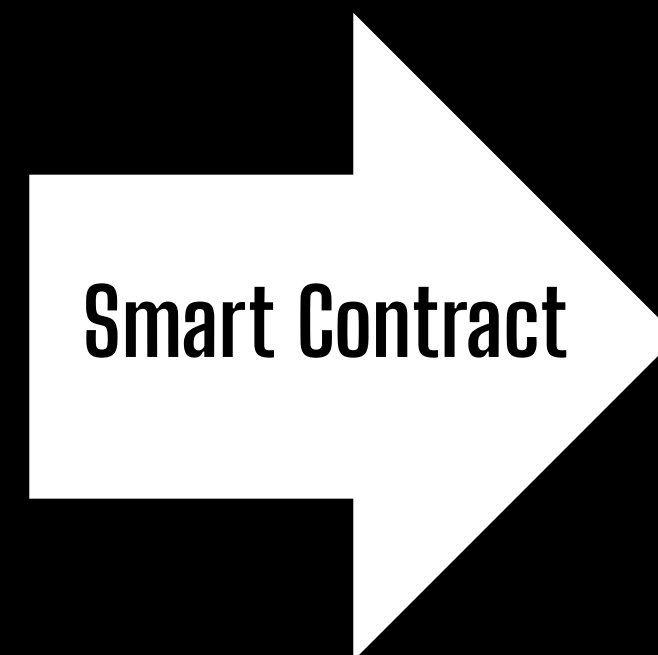
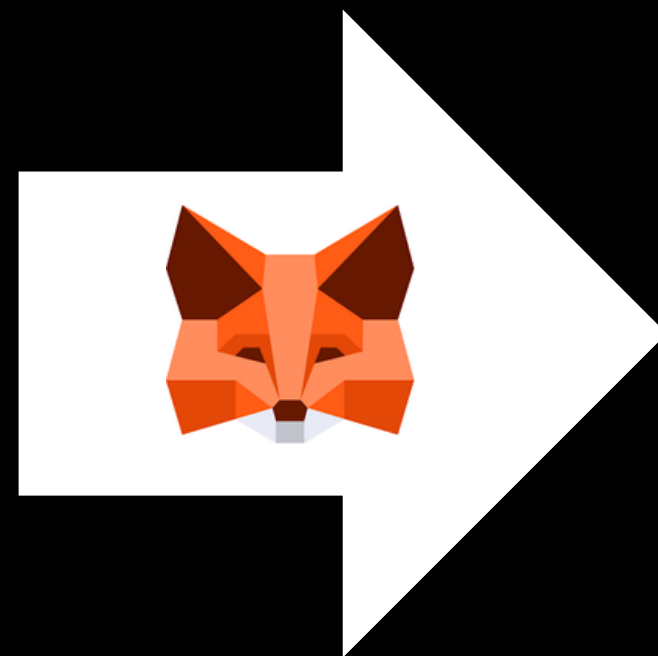
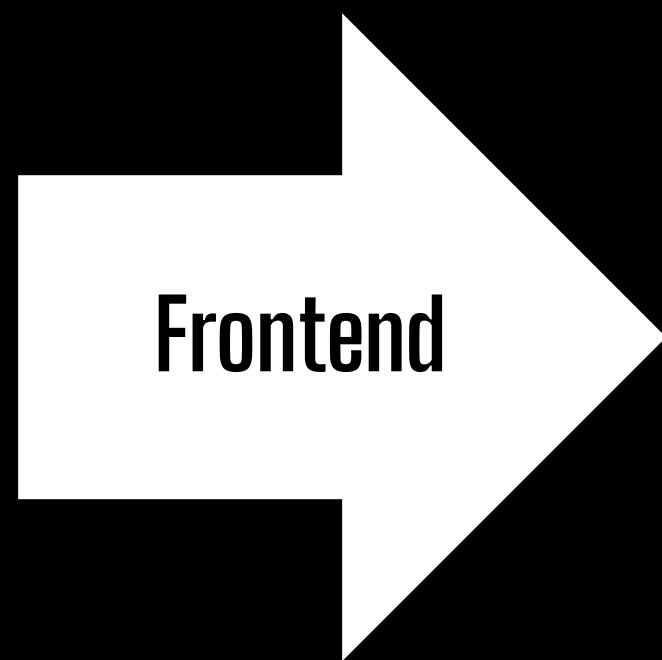
TECHNOLOGY STACK

Technologies Used

- Solidity – Smart contracts
- Hardhat – Development & testing
- Ethereum Sepolia Testnet
- ERC-20 Token Standard
- JavaScript
- MetaMask
- HTML & CSS

SYSTEM ARCHITECTURE

Frontend ↔ MetaMask ↔ Smart Contracts ↔ Blockchain



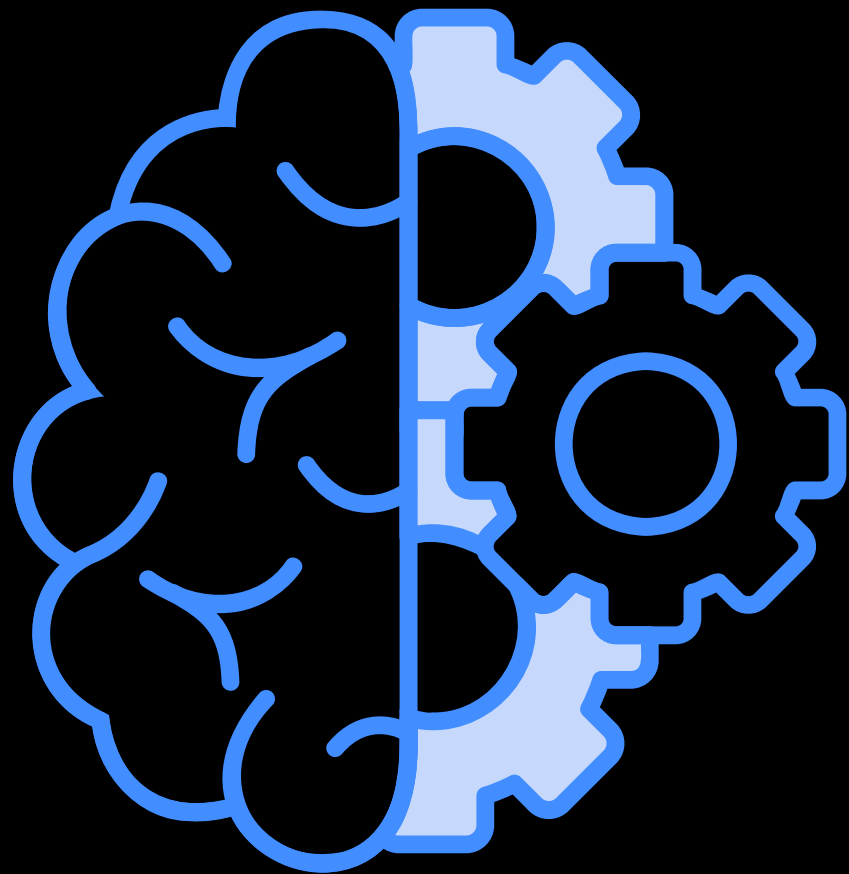
SMART CONTRACTS

InvoiceFund Contract

Responsibilities:

- Create crowdfunding campaigns
- Accept ETH contributions
- Track individual contributions
- Finalize campaigns
- Mint reward tokens





CROWDFUNDING LOGIC

Campaign Parameters:

- Title
- Funding Goal
- Deadline

Functionality:

- Contribution in test ETH
- Mapping for tracking contributions
- Campaign finalization after deadline

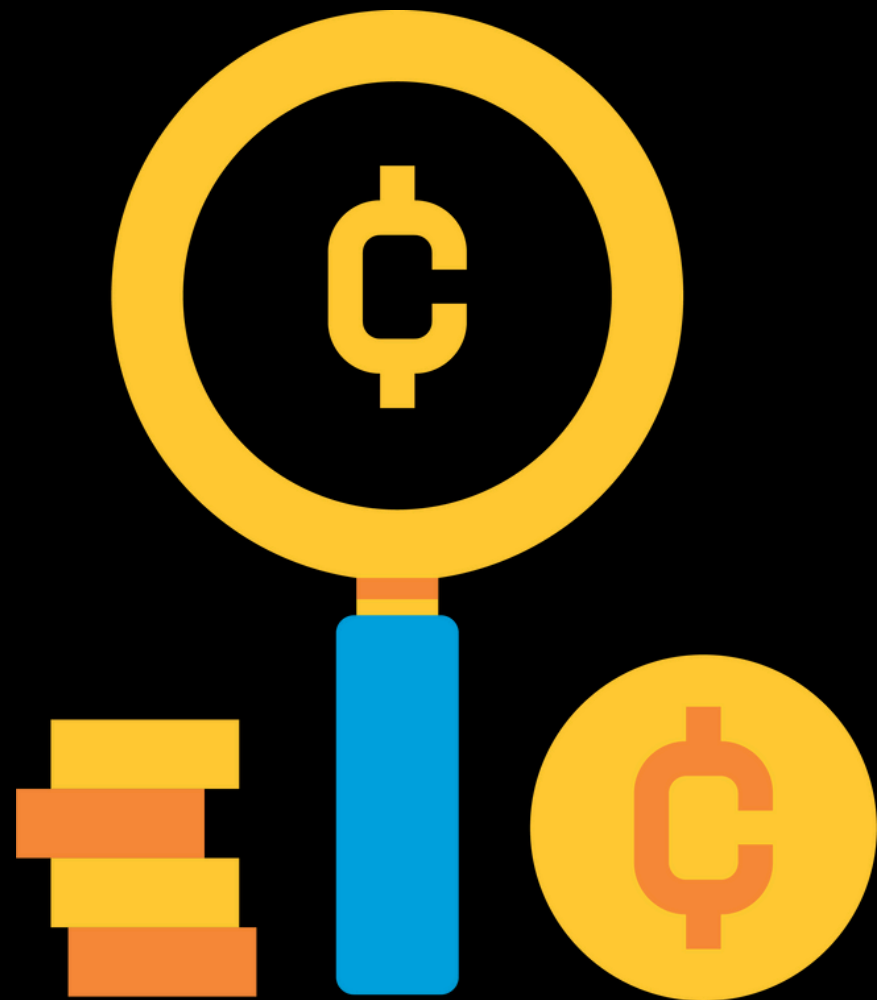
Each contribution is recorded on-chain and is transparently accessible.

ERC-20 REWARD TOKEN

RewardToken Contract

- Custom ERC-20 token (INV)
- Minted automatically after contribution
- No real monetary value
- Used for educational purposes
- Demonstrates tokenization concept





REWARD FORMULA

Reward Calculation

Reward = contributionWei × REWARD_RATE / 1e18

This ensures:

Fair distribution

Transparency

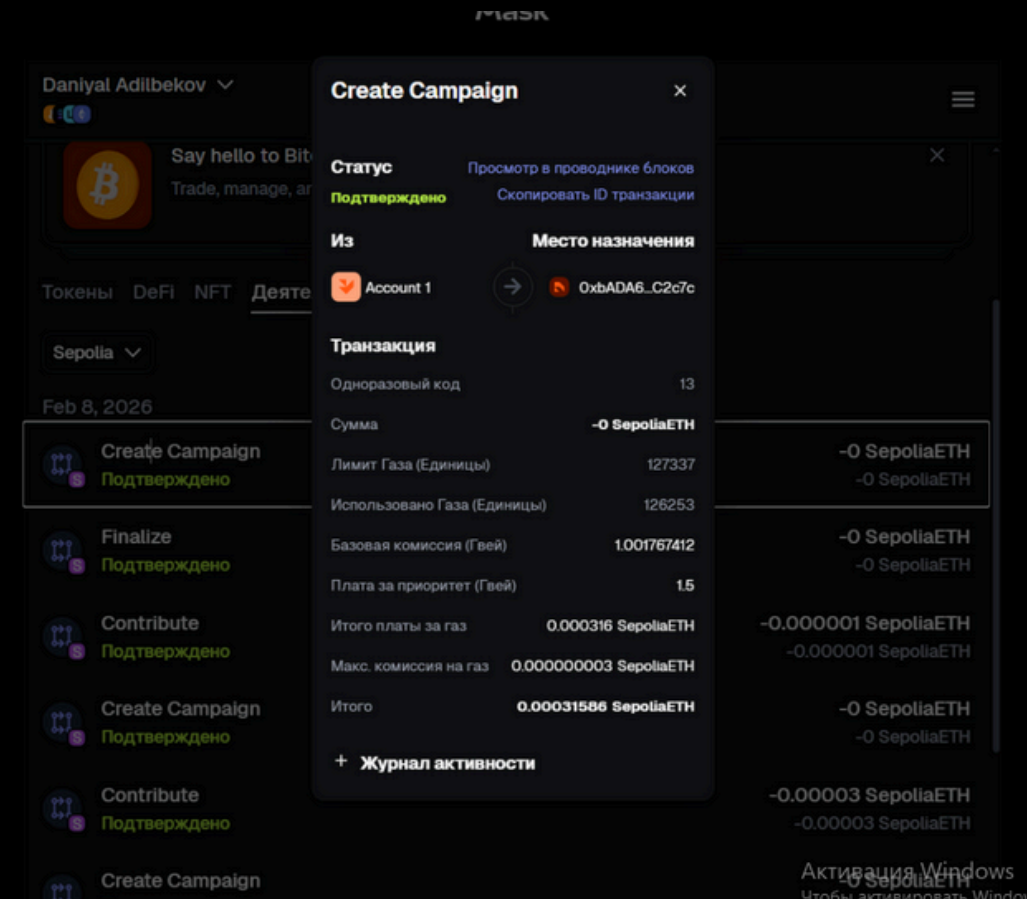
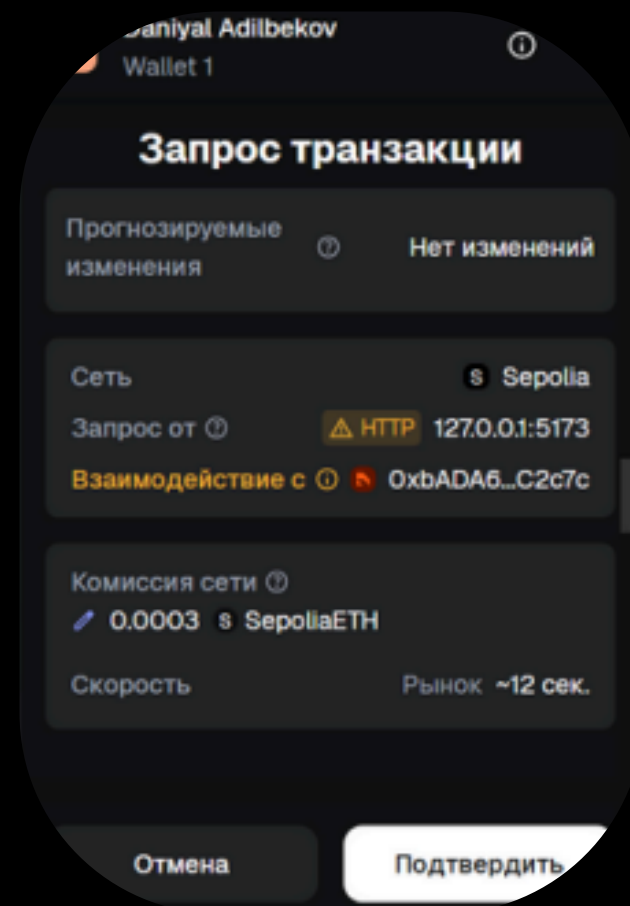
Proportional rewards

METAMASK INTEGRATION

MetaMask Functions

- Request wallet permission
- Validate Sepolia network
- Sign transactions
- Listen to account changes
- Listen to network changes

Ensures secure blockchain interaction.

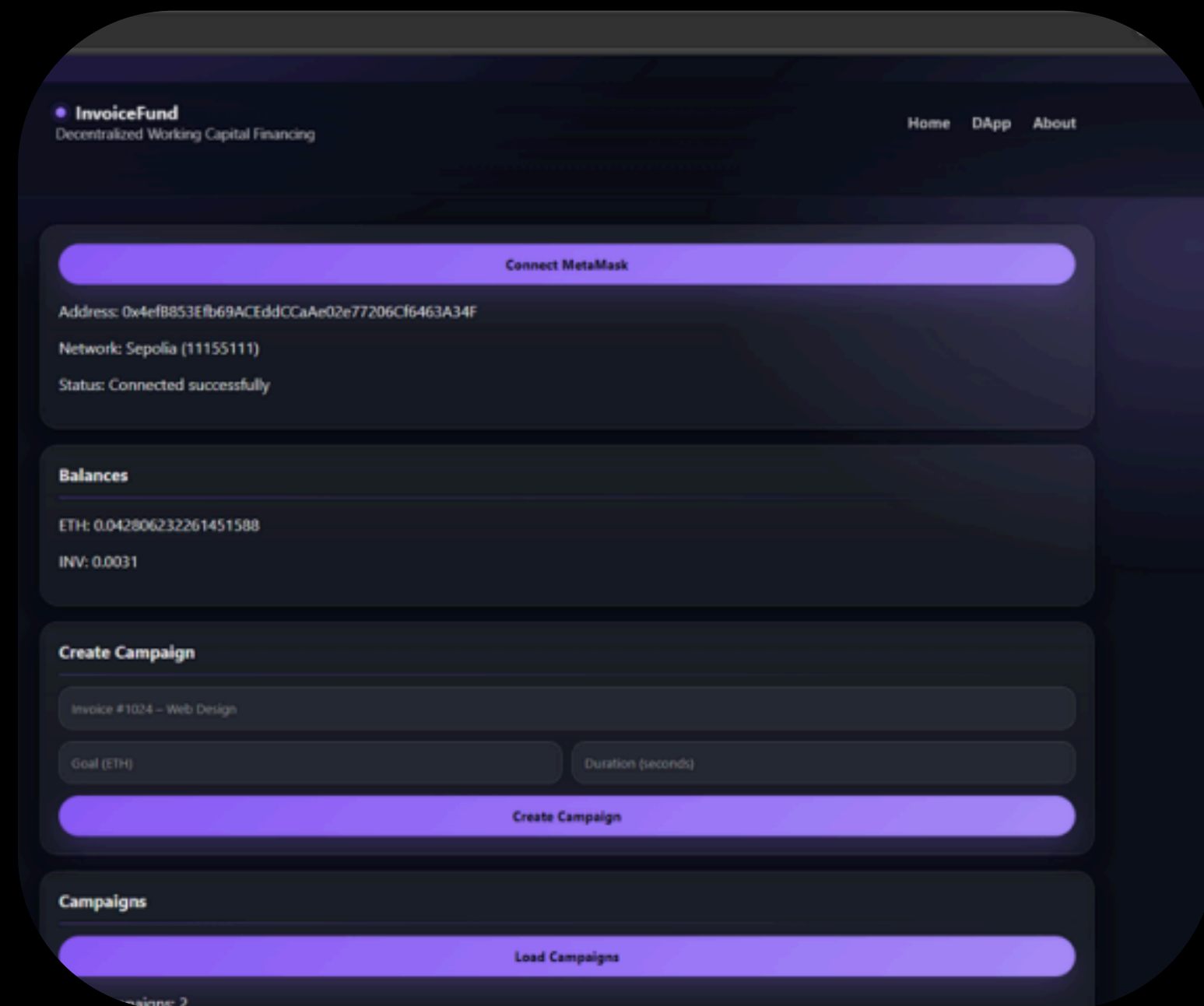
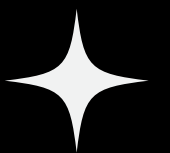


FRONTEND FEATURES

User Can:

- Connect MetaMask wallet
- View wallet address
- Check ETH balance
- Check INV token balance
- Create campaigns
- Browse campaigns
- Contribute ETH
- Finalize campaigns

All transactions are confirmed via MetaMask.



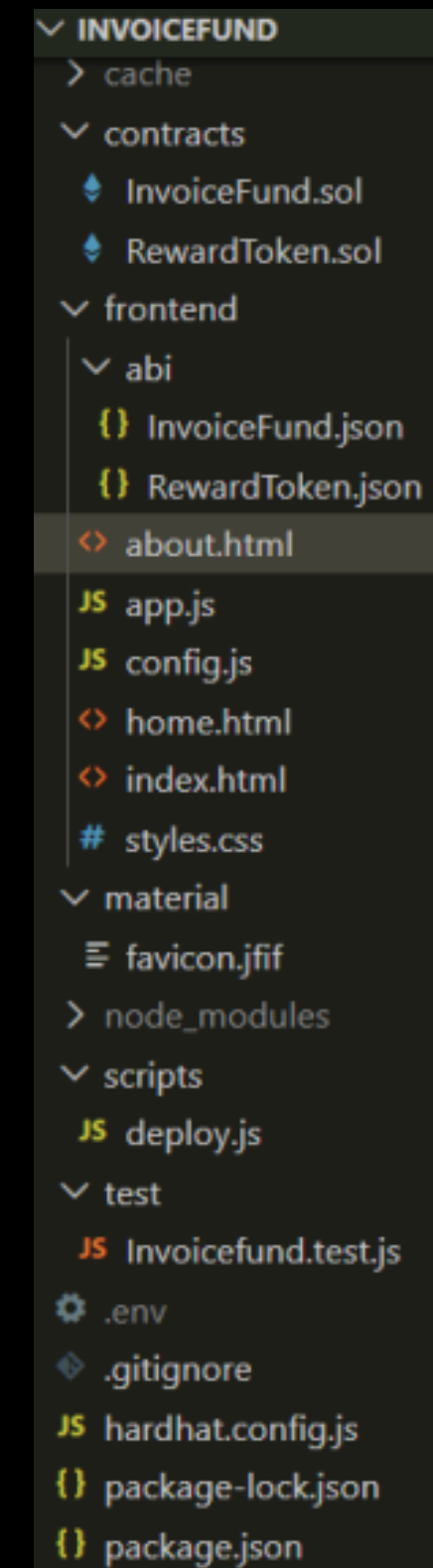
PROJECT STRUCTURE

contracts/
InvoiceFund.sol
RewardToken.sol

frontend/
app.js
index.html
styles.css

scripts/
deploy.js

test/
InvoiceFund.test.js



```
INVOICEFUND
├── cache
├── contracts
│   ├── InvoiceFund.sol
│   └── RewardToken.sol
├── frontend
│   ├── abi
│   │   ├── InvoiceFund.json
│   │   └── RewardToken.json
│   ├── about.html
│   ├── app.js
│   ├── config.js
│   ├── home.html
│   ├── index.html
│   └── styles.css
├── material
│   └── favicon.jfif
├── node_modules
├── scripts
│   └── deploy.js
├── test
│   └── Invoicefund.test.js
├── .env
├── .gitignore
├── hardhat.config.js
├── package-lock.json
└── package.json
```

DEPLOYMENT

DEPLOYMENT STEPS

1. NPM INSTALL
 2. NPX HARDHAT COMPILE
 3. NPX HARDHAT RUN SCRIPTS/DEPLOY.JS --NETWORK SEPOLIA
- GET SEPOLIA ETH FROM FAUCET:
GOOGLE CLOUD WEB3 FAUCET

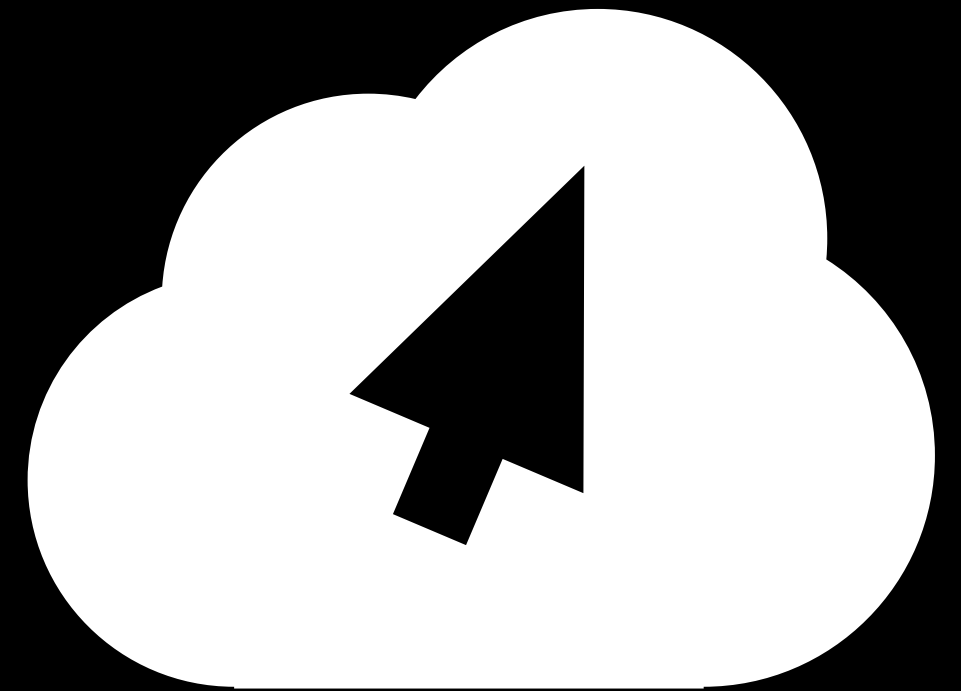
TESTING

WE USED HARDHAT AUTOMATED TESTS TO VERIFY:

- CONTRACT DEPLOYMENT
- CAMPAIGN CREATION
- CONTRIBUTION LOGIC
- REWARD MINTING
- FINALIZATION

COMMAND:

NPX HARDHAT TEST



DEPLOYED CONTRACTS

Sepolia Addresses

InvoiceFund:

0xbADA66D973aa12c43f06F8f46b846F04f3CC2c7c

RewardToken:

0xaC3C1F55973Ea2c1137b3295D52fBB556C11569e

CONCLUSION



InvoiceFund demonstrates a full decentralized crowdfunding workflow.

The project successfully shows:

- **Smart contract logic**
- **ERC-20 token usage**
- **MetaMask integration**
- **Frontend-blockchain interaction**
- **Secure transactions**
- **Real blockchain deployment on testnet**

This project proves practical understanding of blockchain development.

THANKYOU

