

Blockchain final project

InvoiceFund -Decentralized working capital financing

Team:

Name: Daniyal Adilbekov

Group: SE-2435

Name: Bibifatima Bisesheva

Group: SE-2437

Name: Ataniyaz Mutigolla

Group: SE-2437

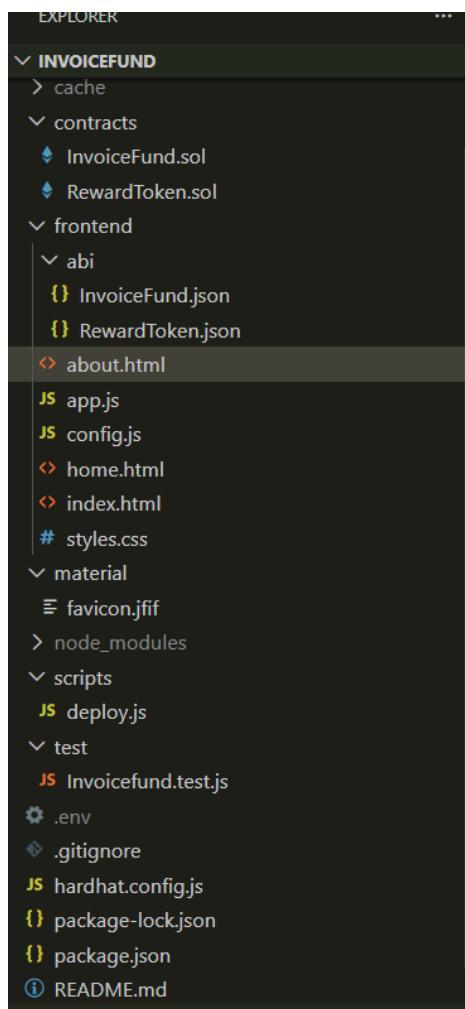
Project Overview

So, basically, InvoiceFund is a decentralized crowdfunding platform designed to provide early liquidity to businesses awaiting invoice payments. Instead of relying on banks or intermediaries, funding is raised through smart contracts. Participants receive ERC-20 reward tokens, which demonstrate proportional participation and accountability on the blockchain. The system operates entirely on the Ethereum Sepolia testnet using MetaMask.

This project was developed as a final project and operates exclusively on an Ethereum test network using free test tokens.

This project was developed as a blockchain final project

Project architecture



Home.html where is main information about project

Home.html where is main information about project

127.0.0.1:5173/frontend/home.html

Home DApp About

What is InvoiceFund?

So, basically, InvoiceFund is a decentralized crowdfunding platform designed to provide early liquidity to businesses awaiting invoice payments. Instead of relying on banks or intermediaries, funding is raised through smart contracts. Participants receive ERC-20 reward tokens, which demonstrate proportional participation and accountability on the blockchain. The system operates entirely on the Ethereum Sepolia testnet using MetaMask.

Why blockchain?

- Tracking contributions on-chain
- Reward token system
- Using Sepolia testnet

What is the meaning of reward tokens?

They are automatically minted by the smart contract during contributions and serve as an on-chain proof of involvement, ensuring transparency, traceability, and accountability without intermediaries. While the tokens have no real monetary value, they demonstrate core tokenization concepts.

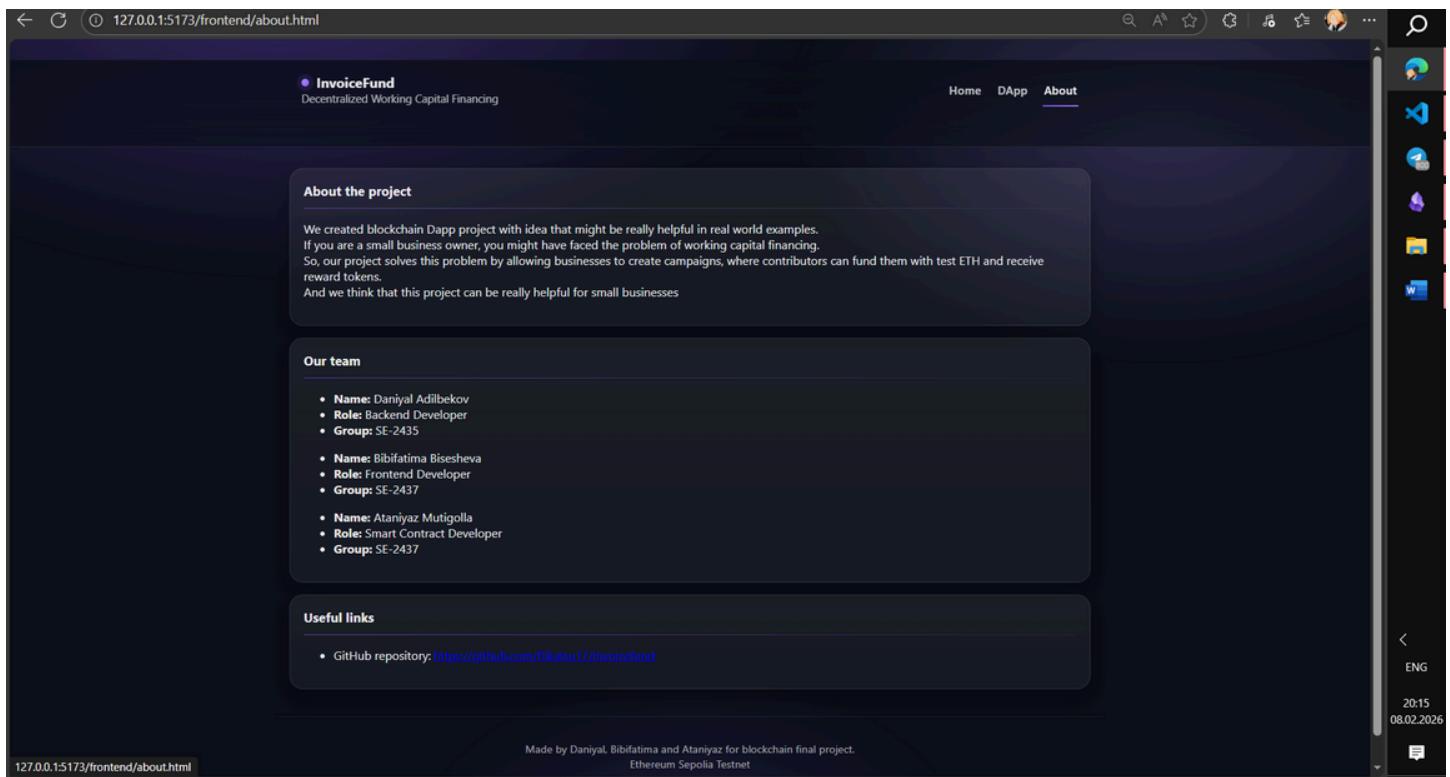
Try the DApp

Go to the DApp page to connect MetaMask, create campaigns, and contribute

Open DApp

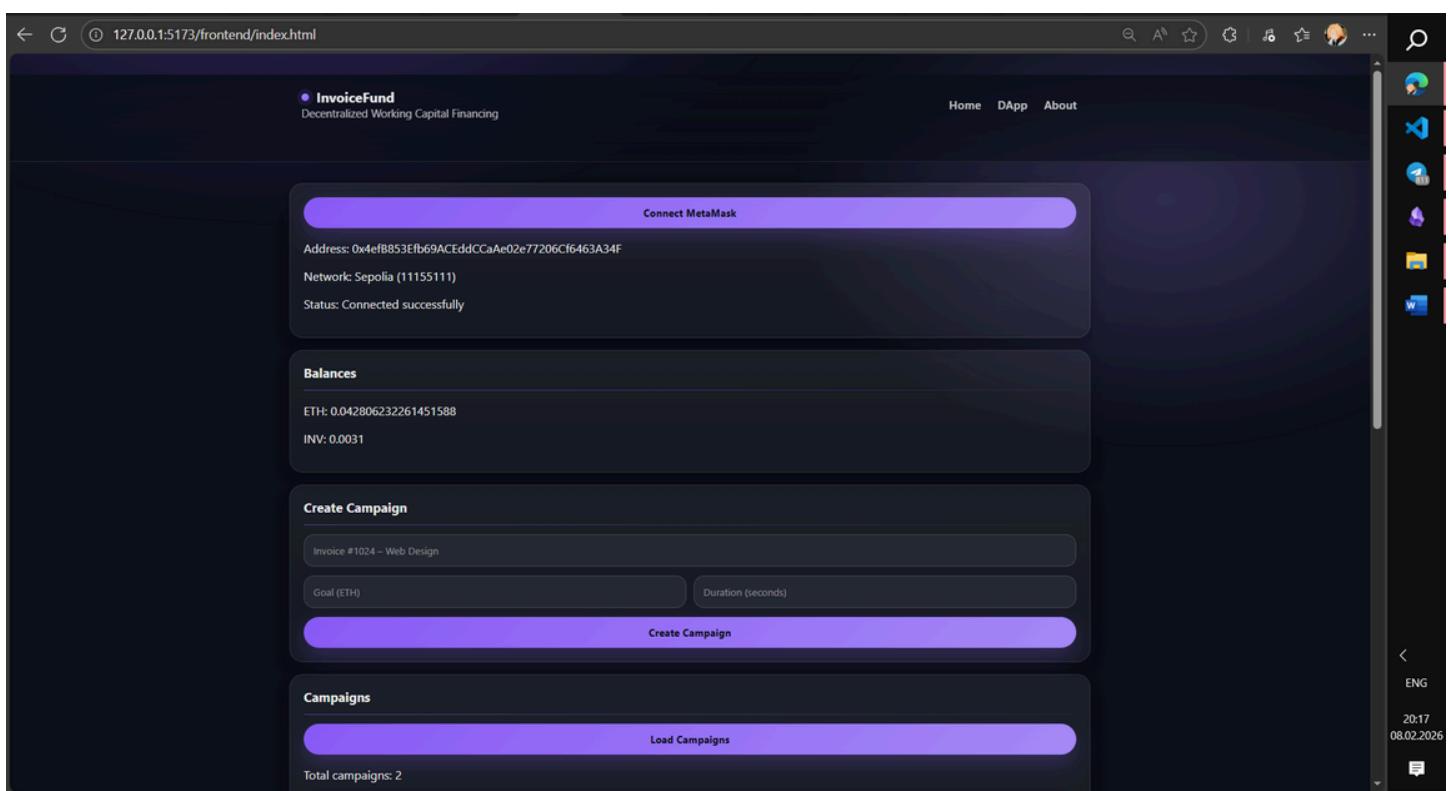
Made by Daniyal, Bibafatima and Ataniyaz for blockchain final project.
Ethereum Sepolia Testnet

About.html where you can see information about creators and links



The screenshot shows the 'About' page of the InvoiceFund DApp. At the top, there's a header with the project name 'InvoiceFund' and its description 'Decentralized Working Capital Financing'. Below the header, there are three main sections: 'About the project', 'Our team', and 'Useful links'. The 'About the project' section contains a paragraph about solving the problem of working capital financing for small business owners. The 'Our team' section lists three members with their names, roles, and group numbers. The 'Useful links' section includes a link to the GitHub repository. At the bottom of the page, it says 'Made by Daniyal, Bibifatima and Ataniyaz for blockchain final project. Ethereum Sepolia Testnet'.

And there is main file index.html where are functions like create campaign, list of campaigns, contribute function, and finalizing any campaign.



The screenshot shows the main index.html page of the InvoiceFund DApp. It features a 'Connect MetaMask' button at the top. Below it, there's a 'Balances' section showing ETH and INV balances. The 'Create Campaign' section allows users to input an invoice ID, set a goal in ETH, and specify a duration in seconds, followed by a 'Create Campaign' button. The 'Campaigns' section has a 'Load Campaigns' button and displays a message indicating there are 2 total campaigns.

Campaigns

[Load Campaigns](#)

Total campaigns: 2

ID 0

Title: invoice 1025
Owner: 0x4efB853Ef69ACEddCCaAe02e77206Cf6463A34F
Goal: 0.0003 ETH
Raised: 0.00003 ETH
Deadline: 08.02.2026, 14:58:36
Finalized: true

ID 1

Title: invoice 666
Owner: 0x4efB853Ef69ACEddCCaAe02e77206Cf6463A34F
Goal: 0.0003 ETH
Raised: 0.000001 ETH
Deadline: 08.02.2026, 16:15:24
Finalized: false

Contribute

Contribute

Campaign ID

Amount (ETH)

[Contribute](#)

Finalize

Campaign ID

[Finalize Campaign](#)

How it works

The screenshot shows two overlapping interfaces. On the left is the 'InvoiceFund' web application at 127.0.0.1:5173/frontend/index.html. It displays a 'Create Campaign' form with the title 'Creating website for client', a goal of '0.00004' ETH, and a deadline of '08.02.2026'. Below the form is a section titled 'Campaigns' with a single entry: ID 0, Title: invoice 1025, Owner: 0x4efB853Efb69ACEddCCaAe02e77206Cf6463A34F, Goal: 0.0003 ETH, Raised: 0.00003 ETH, Deadline: 08.02.2026, 14:58:36, and Finalized: true. On the right is the 'MetaMask' extension for a browser window titled 'Daniyal Adilbekov'. It shows a transaction request for a 'Sepolia' network interaction with address 0xbADA6...C2c7c, amount 0.0003 s SepoliaETH, and a gas fee of ~12 sec. The transaction status is 'Прогнозируемые изменения' (Predicted changes) with 'Нет изменений' (No changes). The transaction hash is visible in the URL.

There is created campaign with full information

ID 2

Title: Creating website for client

Owner: 0x4efB853Efb69ACEddCCaAe02e77206Cf6463A34F

Goal: 0.00004 ETH

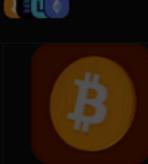
Raised: 0.0 ETH

Deadline: 08.02.2026, 23:06:52

Finalized: false

There is creating campaing in metamask

Daniyal Adilbekov ▾



Say hello to Bitcoin!
Trade, manage, earn.

Токены DeFi NFT Действия Sepolia ▾ Feb 8, 2026

Create Campaign Подтверждено

Finalize Подтверждено

Contribute Подтверждено

Create Campaign Подтверждено

Contribute Подтверждено

Create Campaign Подтверждено

Create Campaign Подтверждено

Create Campaign

Статус Просмотр в проводнике блоков
Подтверждено Скопировать ID транзакции

Из Account 1 → **Место назначения** 0xbADA6...C2c7c

Транзакция

Одноразовый код 13

Сумма -0 SepoliaETH

Лимит Газа (Единицы) 127337

Использовано Газа (Единицы) 126253

Базовая комиссия (Гвей) 1.001767412

Плата за приоритет (Гвей) 1.5

Итого платы за газ 0.000316 SepoliaETH

Макс. комиссия на газ 0.000000003 SepoliaETH

Итого 0.00031586 SepoliaETH

+ Журнал активности

Now if people want to contribute to this campaign

Goal: 0.0003 ETH
Raised: 0.000001 ETH
Deadline: 08.02.2026, 16:15:24
Finalized: false

ID 2
Title: Creating website for client
Owner: 0x4efB853EfB69ACEddCCaAe02e77206Cf6463A34F
Goal: 0.0004 ETH
Raised: 0.0 ETH
Deadline: 08.02.2026, 23:06:52
Finalized: false

Contribute

2 0.00001

Contribute

Finalize

Campaign ID

Finalize Campaign

Запрос транзакции

Прогнозируемые изменения ?
Вы отправляете - 0,00001 \$ SepoliaETH
Вы получаете + 0,001 \$ 0xaC3C1...f569e

Сеть Sepolia
Запрос от HTTP 127.0.0.1:5173
Взаимодействие с OxbADA6..C2c7c

Комиссия сети ? 0.0002 \$ SepoliaETH

Скорость Пример: ~12 сек

Отмена Подтвердить

Made by Daniyal, Bibifatima and Ataniyaz for blockchain final project.
Ethereum Sepolia Testnet

Changes total of INV

Balances

ETH: 0.042223874374651905

INV: 0.0041

And in campaign itself shows how much raised

ID 2

Title: Creating website for client

Owner: 0x4efB853Efb69ACEddCCaAe02e77206Cf6463A34F

Goal: 0.00004 ETH

Raised: 0.00001 ETH

Deadline: 08.02.2026, 23:06:52

Finalized: false

About finalizing. I can finalize the campaign only after deadline

The screenshot shows a web application interface for a campaign titled "ID 2". The campaign details are as follows:

- Title: Creating website for client
- Owner: 0x4efB853Efb69ACEddCCaAe02e77206Cf6463A34F
- Goal: 0.00004 ETH
- Raised: 0.00001 ETH
- Deadline: 08.02.2026, 23:06:52
- Finalized: false

Below the campaign details, there are two buttons: "Contribute" and "Finalize".

A MetaMask extension window is overlaid on the browser window, titled "Запрос транзакции" (Transaction Request). It displays the following information:

- Прогнозируемые изменения (Predicted changes): Нет изменений (No changes)
- Сеть (Network): Sepolia
- Запрос от (From): 127.0.0.1:5173
- Взаимодействие с (Interaction with): OxbADA6...C2c7c
- Комиссия сети (Network fee): 0.0001 s SepoliaETH
- Скорость (Speed): Рынок ~12 сек. (Market ~12 sec.)

At the bottom right of the MetaMask window are two buttons: "Отмена" (Cancel) and "Подтвердить" (Confirm).

At the bottom of the browser window, it says: "Made by Danyal, Bibifatima and Ataniyaz for blockchain final project. Ethereum Sepolia Testnet".

There it shows in metamask and campaign with id finalized changes to true

The screenshot shows a web application for 'InvoiceFund' running at 127.0.0.1:15173. The main interface displays three campaign entries:

- ID 0**: Title: invoice 1025, Owner: 0x4efB853EfB69ACeddCcAe02e77206Cf6463A34F, Goal: 0.0003 ETH, Raised: 0.00003 ETH, Deadline: 08.02.2026, 14:58:36, Finalized: true.
- ID 1**: Title: invoice 666, Owner: 0x4efB853EfB69ACeddCcAe02e77206Cf6463A34F, Goal: 0.0003 ETH, Raised: 0.00001 ETH, Deadline: 08.02.2026, 16:15:24, Finalized: true.
- ID 2**: Title: Creating website for client, Owner: 0x4efB853EfB69ACeddCcAe02e77206Cf6463A34F, Goal: 0.00004 ETH, Raised: 0.00001 ETH, Deadline: 08.02.2026, 23:06:52, Finalized: false.

A purple button labeled 'Load Campaigns' is visible above the campaign list. Below the campaigns, a 'Contribute' section allows users to input a contribution amount (2) and click the 'Contribute' button.

In the top right corner, a MetaMask extension window is open for the user 'Daniyal Adilbekov'. The window displays a Bitcoin logo and the text 'Say hello to Bitcoin: Trade, manage, and buy BTC directly on MetaMask'. It also lists recent transactions under the 'Deятельность' tab:

Action	Status	Value
Finalize	Подтверждено	-0 SepoliaETH -0 SepoliaETH
Contribute	Подтверждено	-0.00001 Sepol... -0.00001 SepoliaETH
Create Campaign	Подтверждено	-0 SepoliaETH -0 SepoliaETH
Finalize	Подтверждено	-0 SepoliaETH -0 SepoliaETH

if i try to finalize before deadline, it will show as error

Purpose of the Project

The main objective of this project is to demonstrate practical knowledge of:

- Using Solidity for smart contracts
- ERC-20 tokens integration
- Client-side blockchain interaction using JavaScript
- MetaMask wallet integration

Interaction with sepolia test networks

Basic dapp architecture

The application addresses the problem of delayed invoice payments by simulating decentralized invoice crowdfunding.

Technology Stack Solidity

Solidity

Hardhat Ethereum Sepolia Test Network

ERC-20 Token Standard

JavaScript

MetaMask

HTML & CSS

Smart Contracts

The project consists of two smart contracts:

1. InvoiceFund

Responsible for:

Creating crowdfunding campaigns

Accepting ETH contributions

Tracking individual and total contributions

Finalizing campaigns after the deadline

Minting reward tokens for contributors

2. RewardToken (ERC-20)

Custom ERC-20 token(inv)

Minted automatically during campaign participation

Has no real monetary value

Used strictly for educational demonstration

Reward formula

Reward = contributionWei × REWARD_RATE / 1e18

Frontend Features

The client-side application allows users to:

Connect a MetaMask wallet

Validate the active blockchain network (Sepolia)

View wallet address and balances (ETH & INV)

Create crowdfunding campaigns

Browse existing campaigns

Contribute test ETH to campaigns
Automatically receive ERC-20 reward tokens
Finalizing campaigns
All blockchain transactions are executed securely through MetaMask.

Frontend to blockchain interaction

The frontend interacts with the Ethereum Sepolia test network using app.js and the MetaMask wallet. MetaMask provides the connection between the web application and the blockchain.

After the user connects MetaMask, the application initializes an app BrowserProvider and obtains a signer representing the active wallet account. Smart contracts are accessed using their deployed addresses and ABI files.

Read-only operations (loading campaigns, checking balances) are performed via the provider, while state-changing operations (creating campaigns, contributing ETH, finalizing campaigns) are executed as blockchain transactions and require user confirmation through MetaMask.

The frontend validates that the user is connected to the Sepolia test network and listens for MetaMask account or network changes to keep the application state consistent.

How to Run the Project

Steps

1) Navigate to the frontend directory:

```
cd frontend
```

```
● PS D:\invoicefund> cd frontend
○ PS D:\invoicefund\frontend>
```

2) Start a local static server:

```
http-server -p 5173 -c-1
```

```
Reward token ownership transferred to InvoiceFund successfully
PS D:\invoicefund> http-server -p 5173 -c-1
>>
Starting up http-server, serving ./

http-server version: 14.1.1

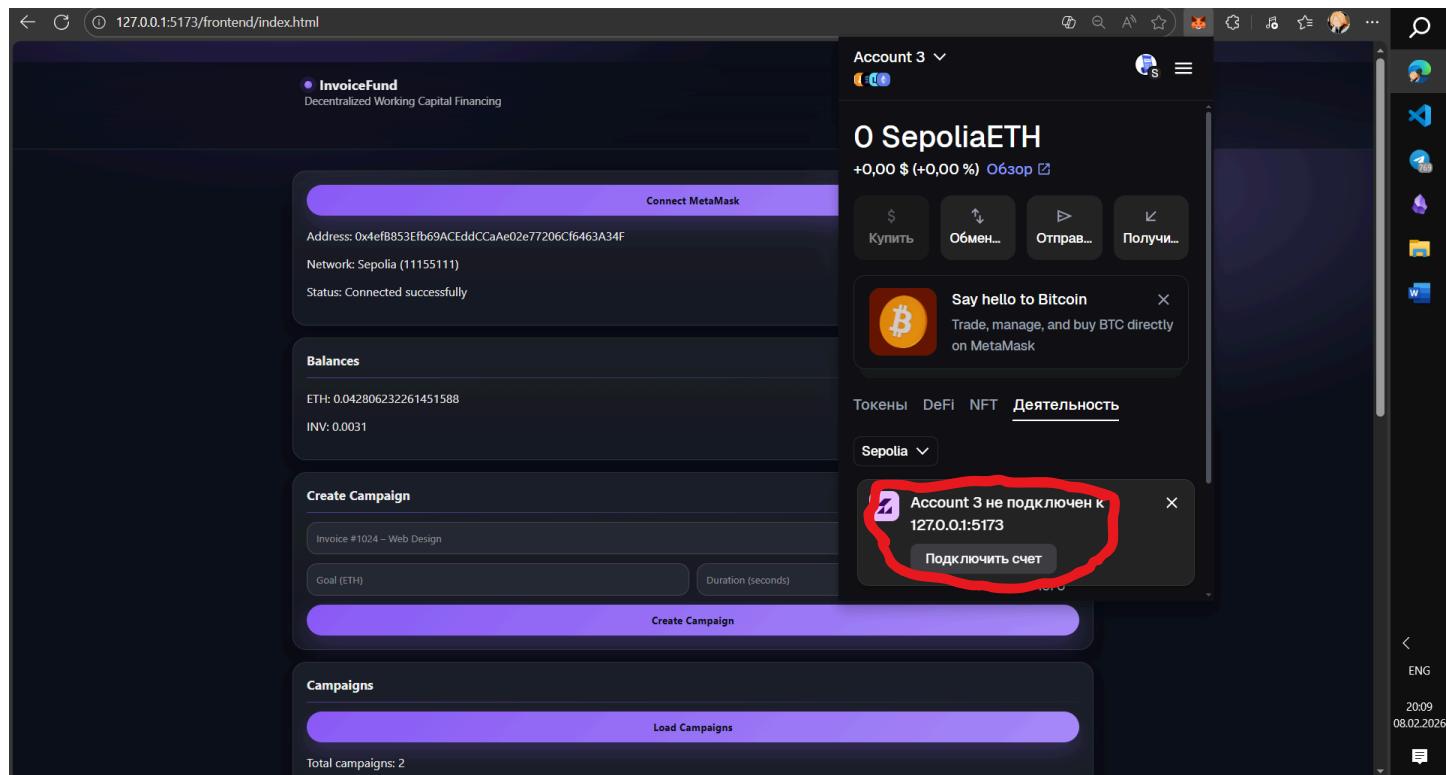
http-server settings:
CORS: disabled
Cache: -1 seconds
Connection Timeout: 120 seconds
Directory Listings: visible
AutoIndex: visible
Serve GZIP Files: false
Serve Brotli Files: false
Default File Extension: none

Available on:
http://26.0.74.107:5173
http://192.168.56.1:5173
http://192.168.8.4:5173
http://127.0.0.1:5173
Hit CTRL-C to stop the server
```

3)Open the application in a browser:

<http://127.0.0.1:5173>

4)Switch MetaMask to the Sepolia Test Network



5)Connect MetaMask and interact with the DApp

Daniyal Adilbekov

0,0428 SepoliaETH
+0,00 \$ (+0,00 %) [Обзор](#)

Address: 0x4efB853Ef69ACEddCCaAe02e77206Cf6463A34F
Network: Sepolia (11155111)
Status: Connected successfully

Balances
ETH: 0.042806232261451588
INV: 0.0031

Create Campaign
Invoice #1024 – Web Design
Goal (ETH) Duration (seconds)
Create Campaign

Campaigns
Load Campaigns
Total campaigns: 2

Tokens DeFi NFT Деятельность
Sepolia
Feb 8, 2026
Finalize Подтверждено -0 SepoliaETH -0 SepoliaETH
Contribute -0.000001 Sep...

20:10 08.02.2026

How to get Sepolia test ethereum

Firstly u need to get your account address, then go to websites where you can get test sepoliaETH

For example: <https://cloud.google.com/application/web3/faucet/ethereum/sepolia> deploy to sepolia

Google Cloud Web3

Get your first drip of OG's \$OG on the Galileo testnet today! [Open faucet →](#)

Ethereum Sepolia Faucet BETA

Get free Sepolia ETH sent directly to your wallet. Brought to you by [Google Cloud for Web3](#).

Select network* Ethereum Sepolia

*required

Wallet address or ENS name* 0x4efB853Ef69ACEddCCaAe02e77206Cf6463A34F

Enter the account address or ENS name where you want to receive tokens

Get 0.05 Sepolia ETH

FAQ

How does my data get used?
What is a faucet?
How does the faucet work?

Feedback
Privacy Policy

ENG
20:13 08.02.2026

Testing

Automated unit tests are implemented using Hardhat to verify contract deployment, creating campaigns, contribution logic, reward token minting and etc. To run it write in terminal:
npx hardhat test

```
● PS D:\invoicefund> npx hardhat test  
[dotenv@17.2.3] injecting env (2) from .env -- tip: ✓ audit secrets and track compliance: https://dotenvx.com/ops
```

InvoiceFund DApp

- ✓ Should deploy and transfer RewardToken ownership to InvoiceFund (2575ms)
- ✓ Should create a campaign with correct parameters (62ms)
- ✓ Should accept contributions and track raised amount (46ms)
- ✓ Should mint reward tokens proportional to contribution (49ms)
- ✓ Should not allow contribute to non-existing campaign (213ms)
- ✓ Should not allow finalize before deadline (54ms)
- ✓ Should allow finalize after deadline (52ms)
- ✓ Should not allow finalize twice
- ✓ Should not allow contribute after finalize (45ms)

9 passing (3s)

```
❖ PS D:\invoicefund> █
```

Conclusion

InvoiceFund demonstrates a complete decentralized crowdfunding workflow using Ethereum smart contracts, MetaMask integration, and a client-side DApp interface. The project highlights key blockchain concepts such as decentralization, transparency, tokenization, and secure user interaction on a test network.