**COLEGIUL NAȚIONAL DE INFORMATICĂ „GR. MOISIL” BRAȘOV**

**LUCRARE PENTRU DOBÂNDIREA COMPETENȚELOR PROFESIONALE**

**BlockFund**

**Elev: Bleotu Alexandru-David**

**Clasa a XII-a D**

**Profesor îndrumător: Șerban Manuela**

**Brașov – Mai 2025**

# **Cuprins**

[**Cuprins** 1](#_Toc196722648)

[**1.** **Linkuri importante** 2](#_Toc196722649)

[**2.** **Motivația alegerii temei lucrării** 2](#_Toc196722650)

[**2.1** **Context și elemente fundamentale** 2](#_Toc196722651)

[**2.2** **Probleme identificate în soluțiile clasice** 2](#_Toc196722652)

[**2.3** **Avantajele unei platforme de crowdfunding pe blockchain** 2](#_Toc196722653)

[**3.** **Utilitatea aplicației** 3](#_Toc196722654)

[**3.1** **Scenarii de utilizare (User Stories)** 3](#_Toc196722655)

[**3.2** **Avantaje pentru utilizatori** 4](#_Toc196722656)

[**3.3** **Beneficii pentru dezvoltare** 4](#_Toc196722657)

[**4.** **Structura aplicației** 5](#_Toc196722658)

[**4.1** **Arhitectura generală** 5](#_Toc196722659)

[**4.2** **Organizarea codului** 5](#_Toc196722660)

[**4.3** **Fișiere de configurare** 6](#_Toc196722661)

[**4.4** **Prezentare vizuală a aplicației** 8](#_Toc196722662)

[**5.** **Detalii de implementare** 10](#_Toc196722663)

[**5.1** **Smart Contract (Hardhat + Solidity)** 10](#_Toc196722664)

[**5.2** **Front-end (React + Tailwind + Ethers.js + LinguiJS + Context API)** 11](#_Toc196722665)

[**5.3** **Back-office și stocare off-chain (Supabase)** 14](#_Toc196722666)

[**6.** **Resurse hardware și software necesare** 15](#_Toc196722667)

[**6.1** **Hardware minim recomandat** 15](#_Toc196722668)

[**6.2** **Software și unelte** 15](#_Toc196722669)

[**7.** **Posibilități de dezvoltare** 16](#_Toc196722670)

[**8.** **Bibliografie** 16](#_Toc196722671)

# **Linkuri importante**

* **Web app -** <https://blockfund.alexbleotu.com/>
* **GitHub -** <https://github.com/alex-bleotu/BlockFund>

# **Motivația alegerii temei lucrării**

## **Context și elemente fundamentale**

* **Dezvoltarea Web3:** În ultimii ani, arhitectura descentralizată a Web3 (blockchain, smart contracts) a câștigat tot mai mult teren în fața soluțiilor centralizate. Nevoia de transparență și de eliminare a intermediarilor a stimulat apariția unor aplicații care se bazează exclusiv pe cod public și validat de rețea.
* **Popularitatea crowdfunding-ului:** Crowdfunding-ul a făcut mult mai simplă strângerea de bani pentru afaceri noi, proiecte sociale sau artistice. Însă platformele clasice iau comisioane între 5% și 10% și cer verificări de identitate care pot îngreuna participarea celor din țările în dezvoltare.

## **Probleme identificate în soluțiile clasice**

* **Lipsă de transparență:** Donatorii nu pot verifica direct modul în care sunt gestionate fondurile, depind de rapoartele proprietarilor de campanie sau de auditurile periodice ale platformelor.
* **Comisioane ridicate:** Taxele de procesare și de platformă pot ajunge la 15–20% din suma strânsă.
* **Acces limitat:** Utilizatorii din țări cu sisteme bancare subdezvoltate sau sancțiuni internaționale întâmpină bariere la transferuri de fonduri.
* **Dependență de infrastructură centrală:** Întreruperile de serviciu, atacurile DDoS sau blocarea conturilor pot împiedica campaniile să-și atingă obiectivele.

## **Avantajele unei platforme de crowdfunding pe blockchain**

* **Transparență totală:** Toate tranzacțiile sunt publice, inspectabile oricând pe exploratorul de blocuri (Etherscan).
* **Costuri reduse:** Smart contract-ul rulează independent, eliminând majoritatea comisioanelor de intermediere.
* **Acces global și instantaneu:** Orice persoană cu MetaMask și Ether poate contribui imediat, fără KYC extins.
* **Rezistență la cenzură:** Campaniile nu pot fi închise unilateral de către o autoritate centrală, atâta timp cât smart contract-ul este activ pe rețea.

# **Utilitatea aplicației**

## **Scenarii de utilizare (User Stories)**

1. **Creator de campanie**

* Context: Un utilizator dorește să își promoveze proiectul și să strângă fonduri.
* Flux de lucru:
  + Completează printr-un formular intuitiv din interfața web, titlul, descrierea, obiectivul financiar (suma țintă), data de încheiere și categoria campaniei, apoi încarcă imagini reprezentative.
  + La lansarea campaniei, tranzacția este trimisă către smart contract-ul de pe rețeaua Mainnet, iar utilizatorul achită taxa de gas aferentă.
  + Campania creată apare imediat în lista de proiecte, afișând atât datele on-chain, cât și elementele off-chain.
* Rezultat: Oricine poate lansa o campanie în câțiva pași simpli.

1. **Contribuția la campanii**

* Context: Un susținător dorește să ofere suport financiar.
* Flux de lucru:
  + Navighează pe pagina de campanii și selectează proiectul la care vrea să contribuie.
  + Apasă „Contribuie”, introduce suma dorită și confirmă tranzacția în MetaMask.
  + După validare, totalul on-chain se actualizează automat.
  + Trimite un mesaj direct creatorului campaniei pentru a pune întrebări suplimentare.
* Rezultat: Orice utilizator cu un portofel MetaMask poate trimite fonduri rapid și transparent.

1. **Vizualizare fără conectare**

* Context: Un potențial susținător sau vizitator pur și simplu explorează campaniile disponibile.
* Flux de lucru:
  + Accesează pagina „Explore” fără a conecta un wallet.
  + Consultă detaliile campaniilor (titlu, descriere, nivel de finanțare) generate prin interogări read-only la smart contract.
* Rezultat: Informațiile sunt disponibile tuturor, chiar dacă nu dețin sau nu au conectat un portofel.

1. **Administrare și întreținere**

* Context: Administratorul trebuie să verifice și să reseteze starea campaniilor sau să schimbe rețeaua Ethereum utilizată de BlockFund.
* Flux de lucru:
  + Accesează interfața de administrare Supabase, unde găsește tabele cu campanii, utilizatori sau mesaje.
  + Rulează, dacă este necesar, scriptul Hardhat pentru resetarea contractului pe rețeaua locală.
* Rezultat: Monitorizarea și întreținerea aplicației se fac rapid, cu instrumentele deja existente.

## **Avantaje pentru utilizatori**

|  |  |  |
| --- | --- | --- |
| Caracteristică | Platformă tradițională | BlockFund |
| Comisioane | 5 – 15% per tranzacție | Gas fee + 2.5% la retragere |
| Timp de procesare | 1 – 3 zile (plăți bancare) | Minute (confirmare blockchain) |
| Transparență | Rapoarte manual | Tranzacții publice |
| Acces global | KYC/AML, limitări geografice | Orice wallet MetaMask |
| Reziliență | Dependent de servere | Imuabilitate a contractului |

* **Economii:** Eliminarea comisioanelor mari de platformă lasă mai mulți bani în mâna creatorilor și a susținătorilor.
* **Experiență fluidă:** Toate operațiunile se fac direct în browser, fără redirecționări către procese de plată externe.
* **Încredere sporită:** Oricine poate urmări în timp real cum sunt gestionate fondurile.

## **Beneficii pentru dezvoltare**

1. **Arhitectură modulară**

* Front-end React + Tailwind, smart contracts în Solidity gestionate cu Hardhat și back-office pe Supabase.
* Fiecare funcție a contractului are propriul test și poate fi updatată independent.

1. **Ciclu de dezvoltare rapid**

* Hot-reload prin Vite pentru front-end și rețea locală Hardhat pentru iterații imediate.
* Deploy direct pe Sepolia cu o singura comandă, în vederea demo-urilor.

1. **Costuri reduse de infrastructură**

* Folosirea testnet-ului Sepolia minimizează cheltuielile cu gas în faza de dezvoltare.
* Planul gratuit Supabase acoperă nevoile de stocare off-chain pentru MVP.

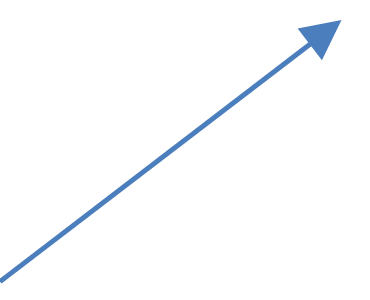
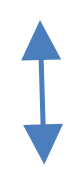
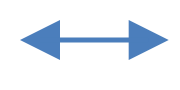
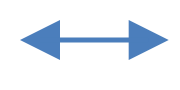
# **Structura aplicației**

## **Arhitectura generală**

**React UI**

**Ethers.js**

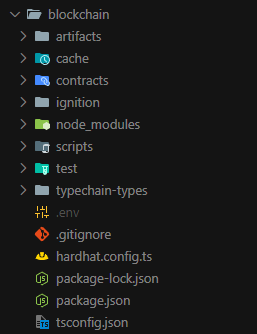
**Smart Contract (sepolia / localhost)**



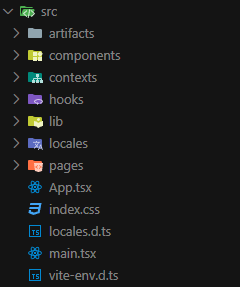
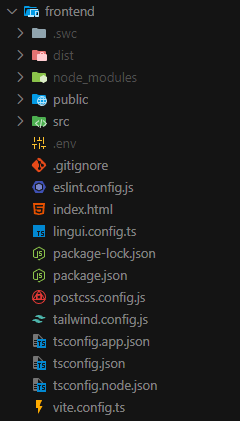
**Supabase**

## **Organizarea codului**

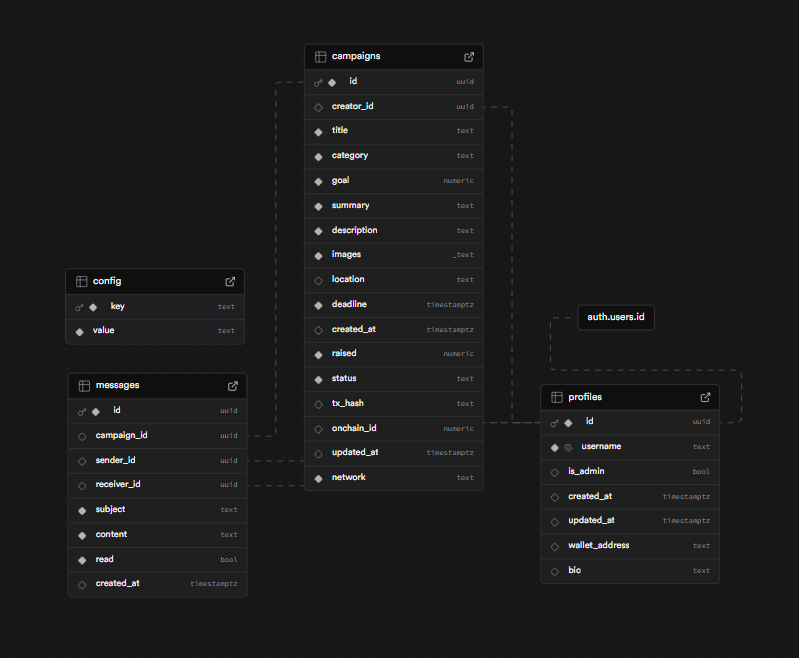
1. Structura directorului **blockchain**



1. Structura directorului **frontend**



1. Structura bazei de date de pe **supabase**



## **Fișiere de configurare**

* hardhat.config.js



* vite.config.js

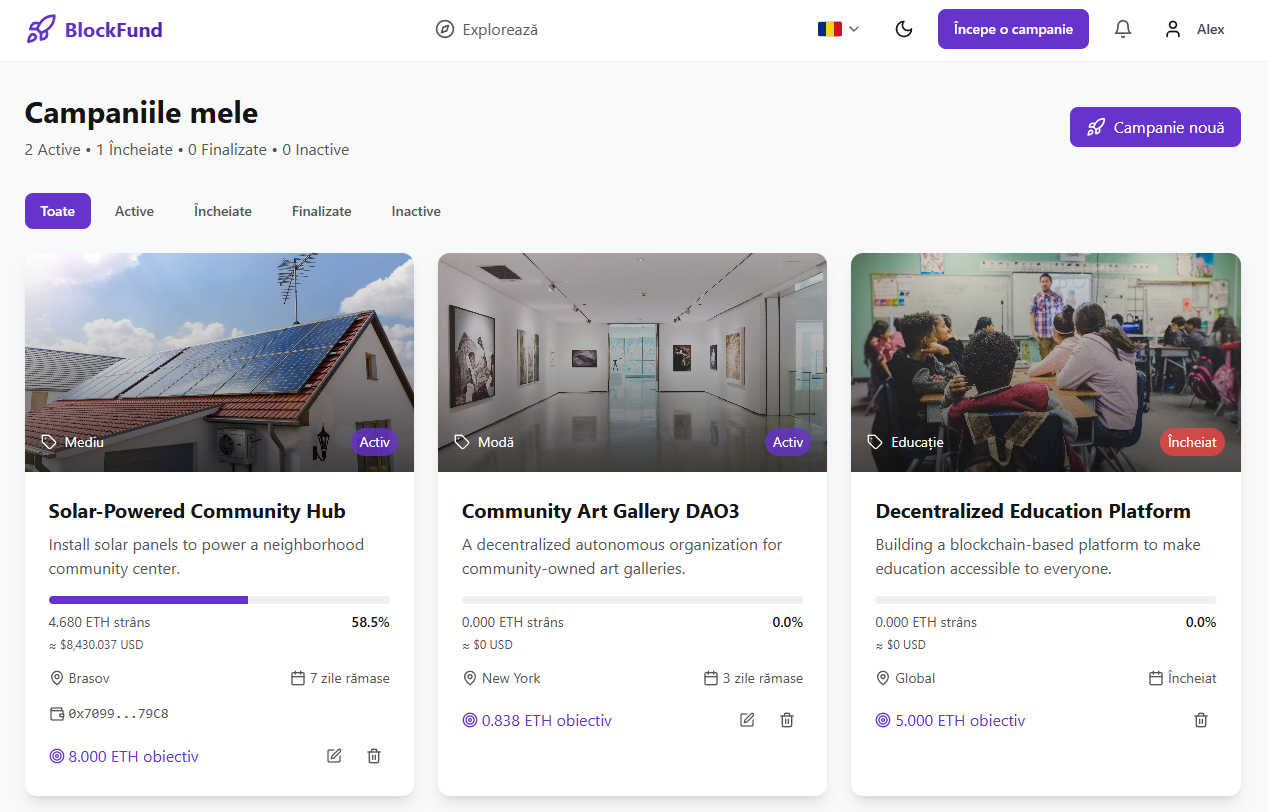


* package.json (front-end)

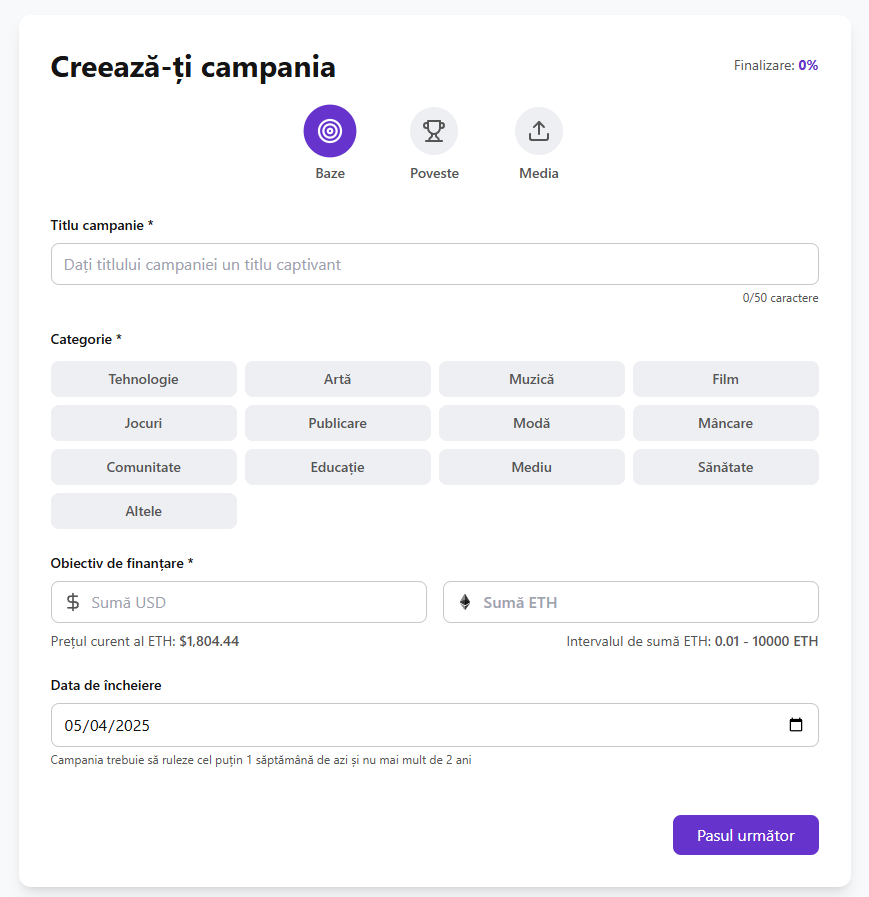
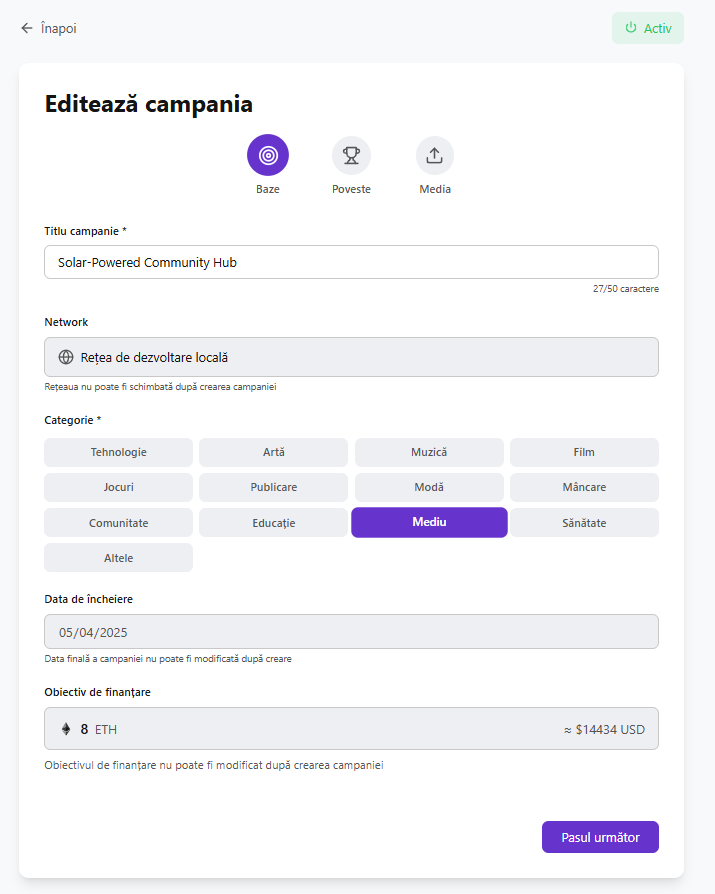


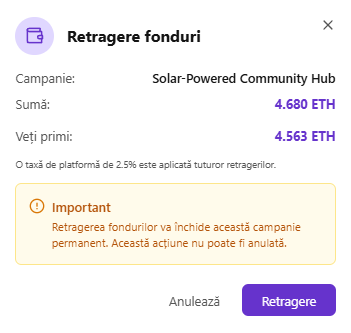
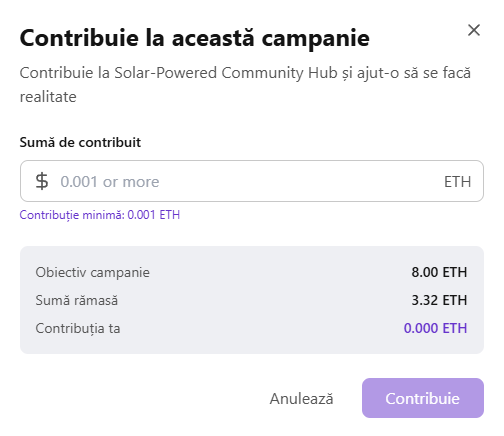
## **Prezentare vizuală a aplicației**

* **Pagina cu campaniile mele**



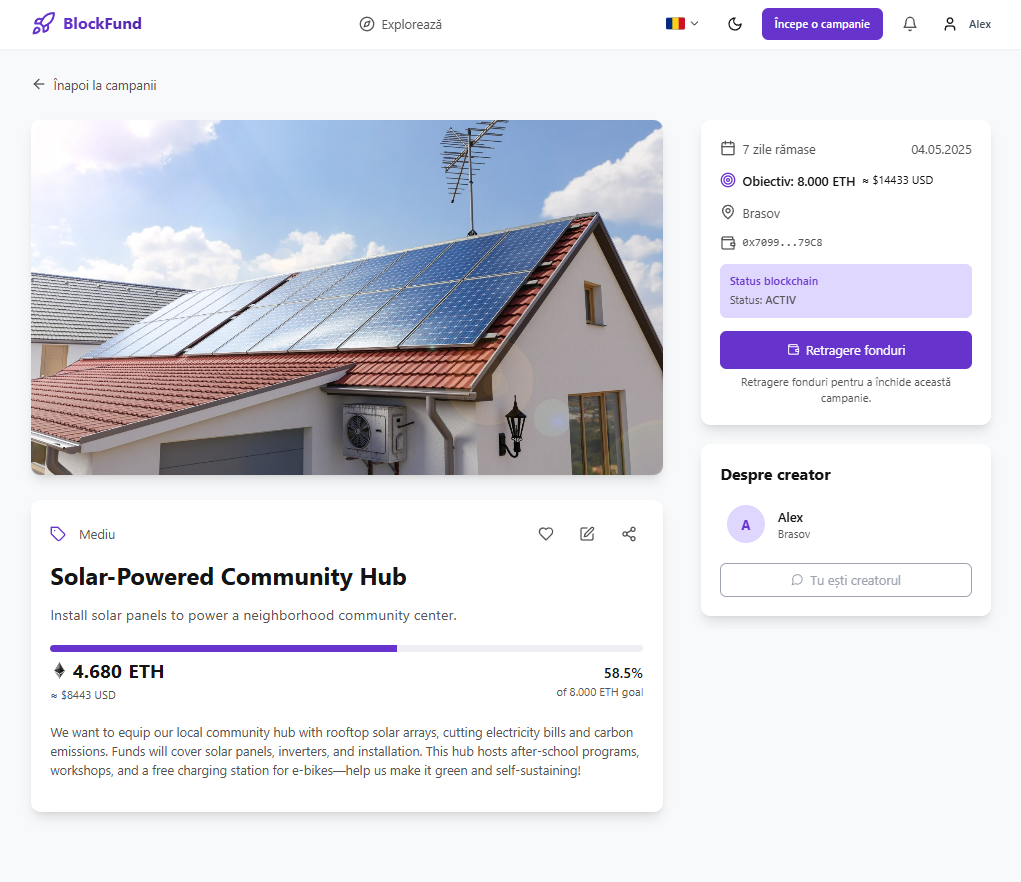
* **Pagina de creare și editare a campaniei**

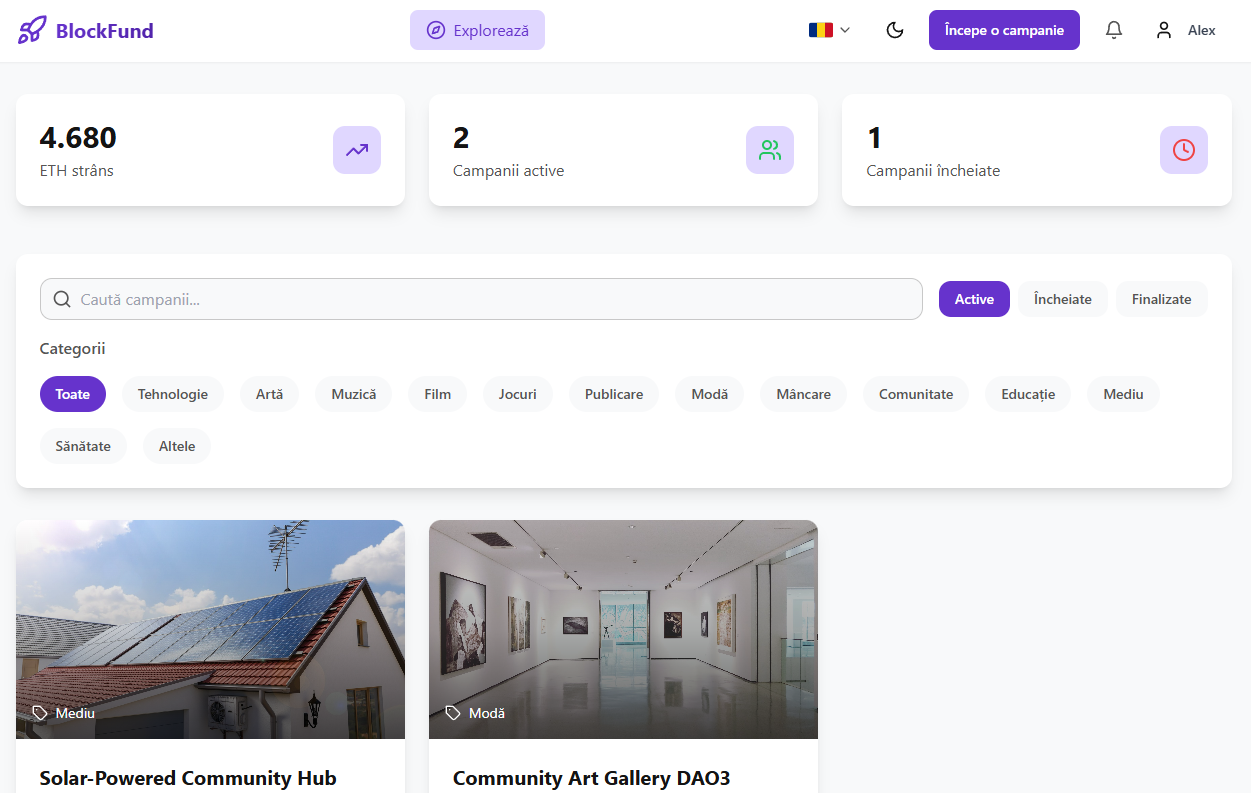
* **Acțiunile campaniei**



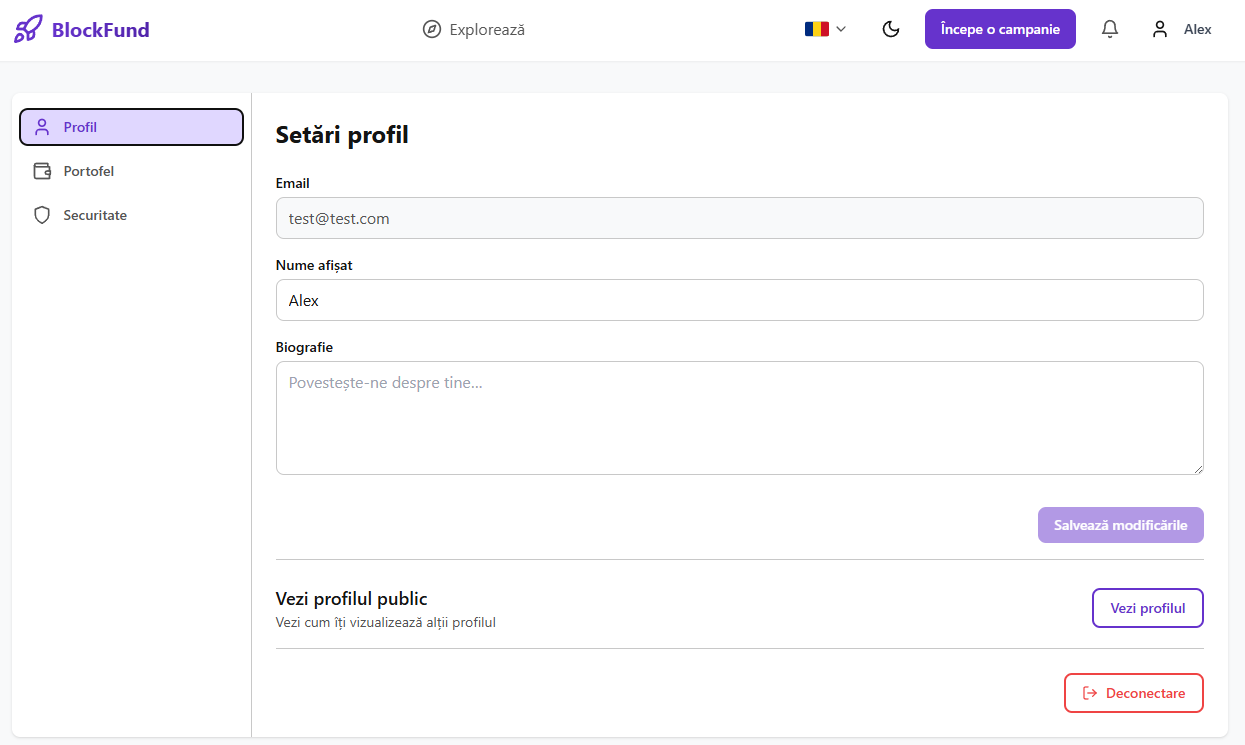
* **Pagina cu detaliile campaniei**



* **Pagina de explore**



* Pagina de setări



# **Detalii de implementare**

## **Smart Contract (Hardhat + Solidity)**

1. Campaign.sol

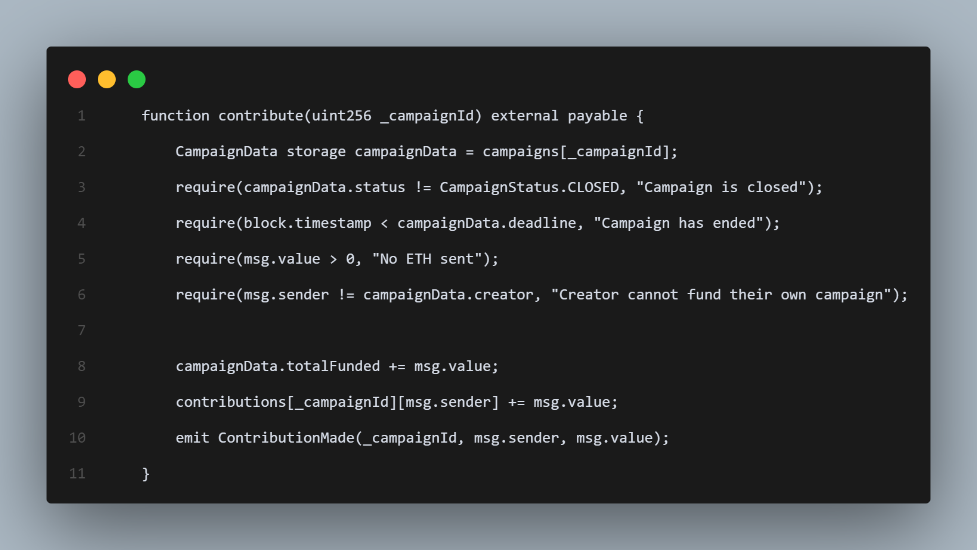
* Funcții cheie:
  + function createCampaign(uint256 \_goal, uint256 \_deadline,

string calldata \_metadataCID)

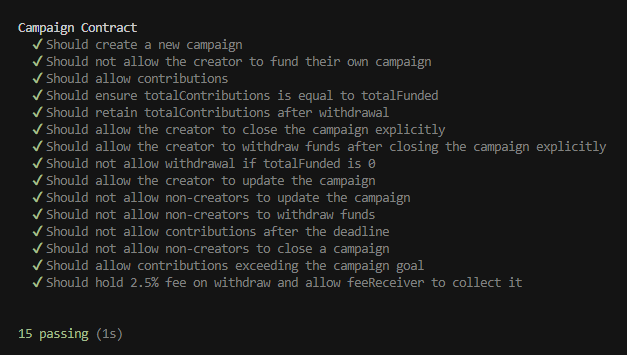
* + function contribute(uint256 \_campaignId)
  + function closeCampaign(uint256 \_campaignId)
  + function withdraw(uint256 \_campaignId)
  + function collectFees(uint256 \_campaignId)
  + function getCampaign(uint256 \_campaignId)
  + function updateCampaign(uint256 \_campaignId, uint256 \_newGoal, uint256

\_newDeadline, string calldata \_newMetadataCID)

* + function getCampaignCount()



1. Testare smart contract





1. Deployment

* blockchain/scripts/**deploy.ts**

****

* comandă - **npx hardhat run scripts/deploy.ts --network mainnet**

## **Front-end (React + Tailwind + Ethers.js + LinguiJS + Context API)**

1. Configurare proiect

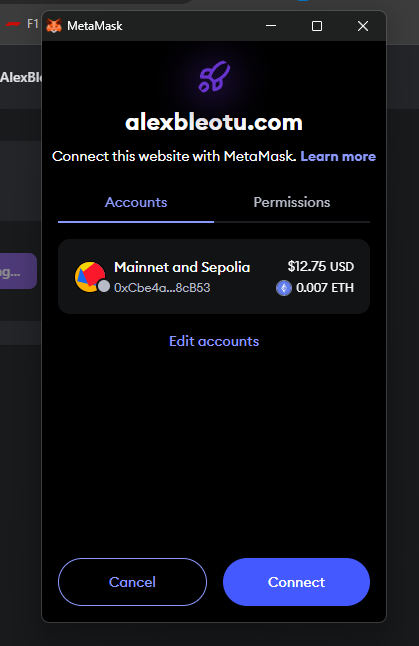
* vite.config.js
* tailwind.config.js
* postcss.config.js
* lingui.config.ts

1. Conectare la wallet prin MetaMask

* frontend/src/hooks/**useWallet.ts**

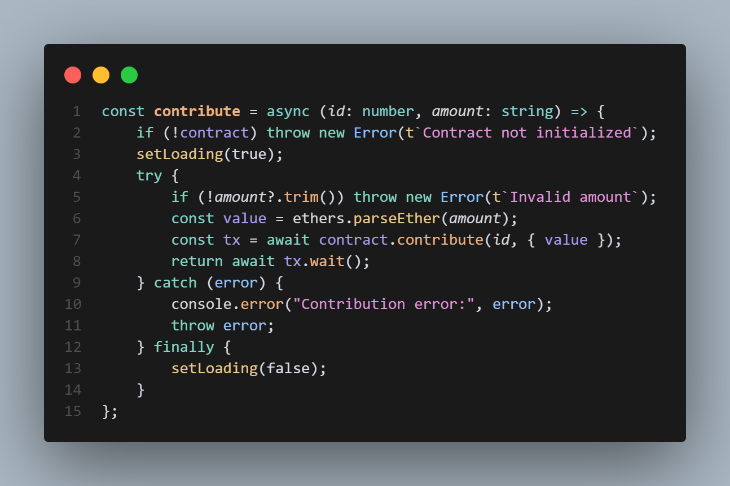


* frontend/src/hooks/**useMetaMask.ts**



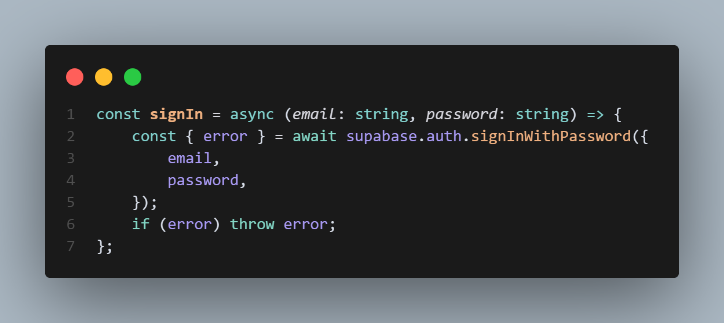
1. Comunicare cu smart contract prin Ethers.js

* frontend/src/hooks/**useCampaignContract.ts**



1. Conectarea utilizatorului

* frontend/src/contexts/**AuthContext.tsx**

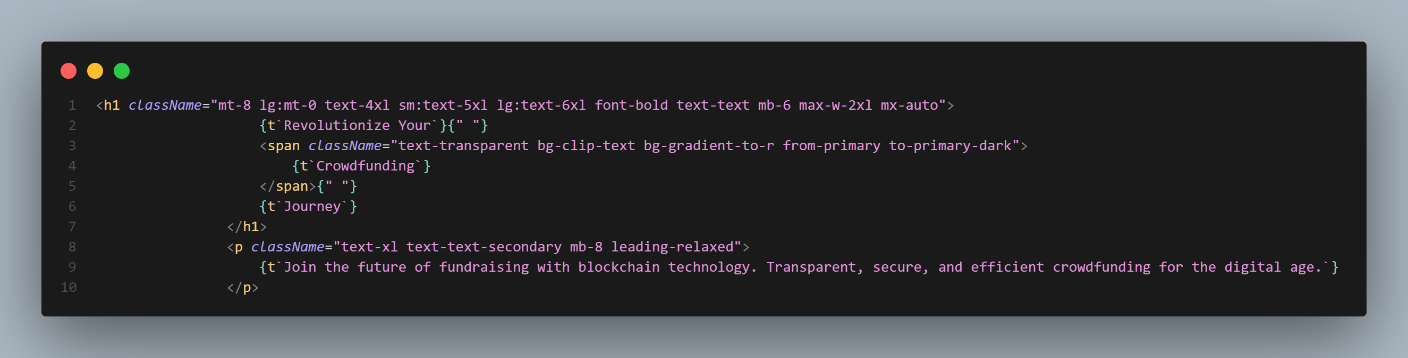


* frontend/src/hooks/**useAuth.ts**



1. Internaționalizarea aplicației cu LinguiJS

* Marcarea textului drept translatabil cu tagul **t`…`**



* Extragere de mesaje – **npx lingui extract** (genereaza fisiere .po)
  + frontend\src\locales\en\messages.po
  + frontend\src\locales\ro\messages.po
* frontend/src/contexts/**LanguageContext.tsx**

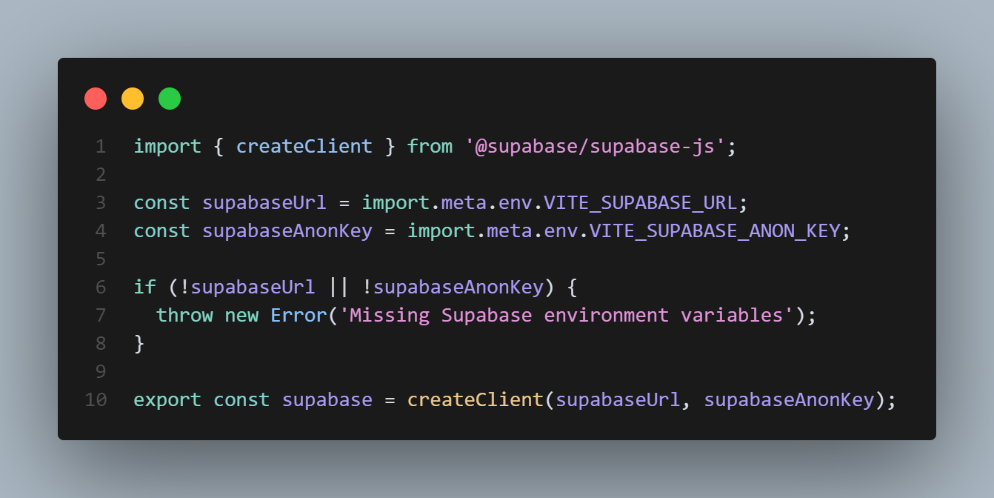
****

1. Notificările și mesajele din aplicație

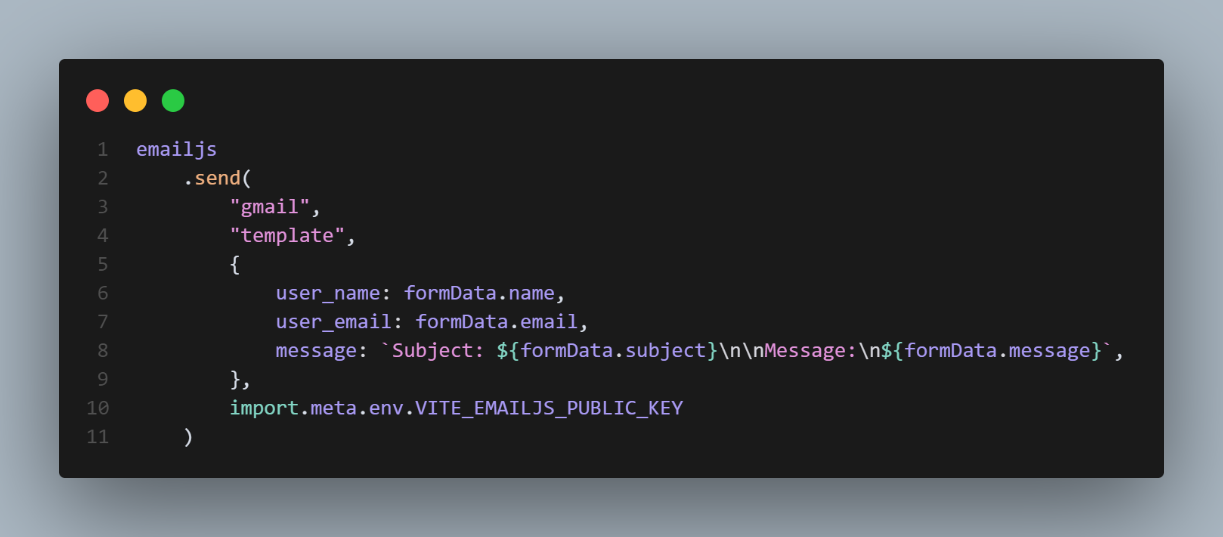
* frontend/src/contexts/**useMessages.ts**



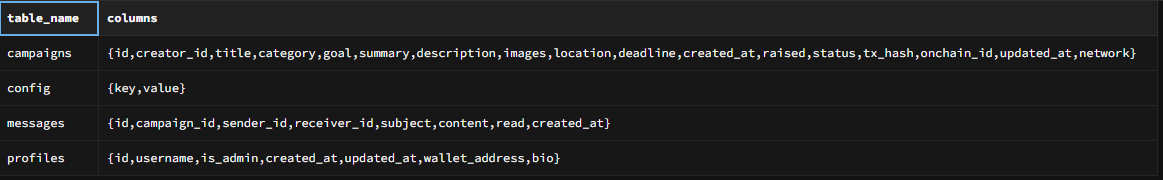
1. Conectare la supabase

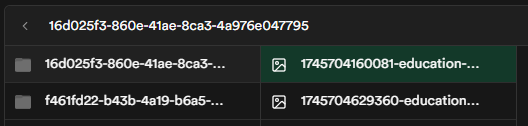
* frontend/src/lib/**supabase.ts**

1. Trimiterea de emailuri prin formularul de contact



## **Back-office și stocare off-chain (Supabase)**

1. Schema bazei de date
2. Bucket de stocare – **campaign-images**



# **Resurse hardware și software necesare**

## **Hardware minim recomandat**

Pentru dezvoltarea si rulare locala a proiectului.

|  |  |  |
| --- | --- | --- |
| Tip | Cerință minimă | Recomandat |
| CPU | 2 nuclee | 4 nuclee |
| RAM | 4 GB | 8GB |
| Spațiu | 10 GB | 20 GB |
| Conexiune internet | 10 Mbps | 20 Mbps |

## **Software și unelte**

|  |  |  |
| --- | --- | --- |
| Categorie | Componentă / Unealtă | Versiune |
| **Runtime** | Node.js | ≥ 18.x |
|  | Npm | ≥ 8.x |
| **Front-end** | React | ^ 18.x |
|  | Vite | ^ 5.x |
|  | TypeScript | ^ 5.x |
|  | Tailwind CSS | ^ 3.x |
|  | Ethers.js | ^ 6.x |
|  | Supabase JS | ^ 2.x |
|  | LinguiJS | ^ 5.x |
|  | Framer Motion | ^ 11.x |
|  | Lucide-React | ^ 0.x |
|  | React Hot Toast | ^ 2.x |
|  | React Router | ^ 7.x |
|  | Date-fns | ^ 4.x |
|  | EmailJS Browser | ^ 4.x |
| **Smart contract** | Hardhat | ^ 2.22.x |
|  | Hardhat Toolbox | ^ 5.0.0 |
|  | Hardhat Etherscan | ^ 3.1.8 |
|  | Dotenv | ^ 16.x |
| **Testare și Linting** | Mocha + Chai (integrate) | - |
|  | EsLint | ^ 8.x |
| **Off-chain și Back-end** | Supabase SDK | ^ 2.x |
|  | Supabase CLI | - |
| **Browser și extensii** | Chrome / Firefox / Edge | - |
|  | MetaMask | - |
| **IDE** | VS Code | - |
| **Versionare** | Git | - |

# **Posibilități de dezvoltare**

1. **Suport multi-chain**

Integrarea cu rețele populare precum Polygon, Binance Smart Chain sau Avalanche, folosind bridge-uri sau protocoale cross-chain

1. **Recompense sub formă de NFT**

Emiterea automată a NFT-urilor de recompensă în momentul contribuției.

1. **Contribuții recurente**

Smart contract care permite setarea unui abonament lunar către o campanie de strângere de fonduri.

1. **Dashboard**

Panou de board cu grafice de progres, rapoarte detaliate ale contribuțiilor.

1. **Back-end extern și API dedicat**

Extinderea aplicației cu un serviciu de back-end separat, care să asigure servirea datelor către front-end și către alte clienți (mobile, dashboard, webhook-uri), fără a folosi Supabase.

# **Bibliografie**

1. **Hardhat** - <https://hardhat.org/docs/>
2. **Ethers.js** - <https://docs.ethers.org/>
3. **Solidity** - <https://docs.soliditylang.org/>
4. **Ethereum Whitepaper -** <https://ethereum.org/en/whitepaper/>