

**LUCRARE PENTRU DOBÂNDIREA COMPETENȚELOR  
PROFESIONALE**

**BlockFund**

**Elev: Bleotu Alexandru-David**

**Clasa a XII-a D**

**Profesor îndrumător: Șerban Manuela**

# Cuprins

<b>Cuprins</b>	<b>1</b>
<b>1. Linkuri importante</b>	<b>2</b>
<b>2. Motivația alegerii temei lucrării</b>	<b>2</b>
2.1 Context și elemente fundamentale	2
2.2 Probleme identificate în soluțiile clasice	2
2.3 Avantajele unei platforme de crowdfunding pe blockchain	2
<b>3. Utilitatea aplicației</b>	<b>3</b>
3.1 Scenarii de utilizare (User Stories)	3
3.2 Avantaje pentru utilizatori	4
3.3 Beneficii pentru dezvoltare	4
<b>4. Structura aplicației</b>	<b>5</b>
4.1 Arhitectura generală	5
4.2 Organizarea codului	5
4.3 Fișiere de configurare	6
4.4 Prezentare vizuală a aplicației	8
<b>5. Detalii de implementare</b>	<b>10</b>
5.1 Smart Contract (Hardhat + Solidity)	10
5.2 Front-end (React + Tailwind + Ethers.js + LinguiJS + Context API)	11
5.3 Back-office și stocare off-chain (Supabase)	14
<b>6. Resurse hardware și software necesare</b>	<b>15</b>
6.1 Hardware minim recomandat	15
6.2 Software și unelte	15
<b>7. Posibilități de dezvoltare</b>	<b>16</b>
<b>8. Bibliografie</b>	<b>16</b>

# 1. Linkuri importante

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

## 2. Motivația alegerii temei lucrării

### 2.1 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.

### 2.2 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.

### 2.3 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.

### 3. Utilitatea aplicației

#### 3.1 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.

##### 2. 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.

##### 3. 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.

##### 4. 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.

### 3.2 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 mână creatorilor și a susținătorilor.
- **Experiență fluidă:** Toate operațiunile se fac direct în browser, fără redirectionări către procese de plată externe.
- **Încredere sporită:** Oricine poate urmări în timp real cum sunt gestionate fondurile.

### 3.3 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.

#### 2. Ciclu de dezvoltare rapid

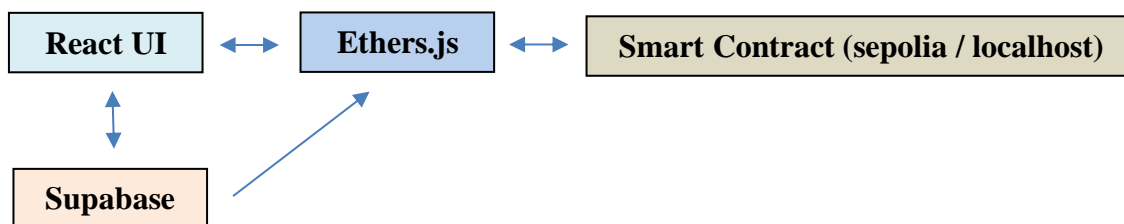
- Hot-reload prin Vite pentru front-end și rețea locală Hardhat pentru iterații imediate.
- Deploy direct pe Sepolia cu o singura comandă, în vederea demo-urilor.

#### 3. 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.

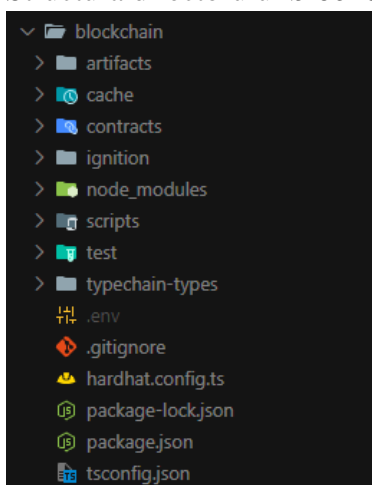
## 4. Structura aplicației

### 4.1 Arhitectura generală

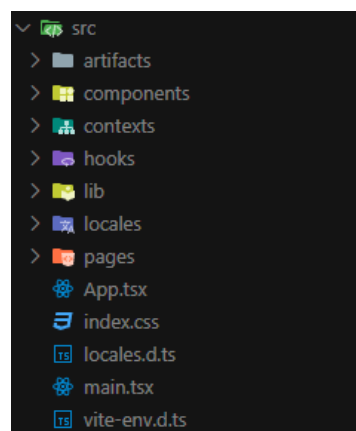
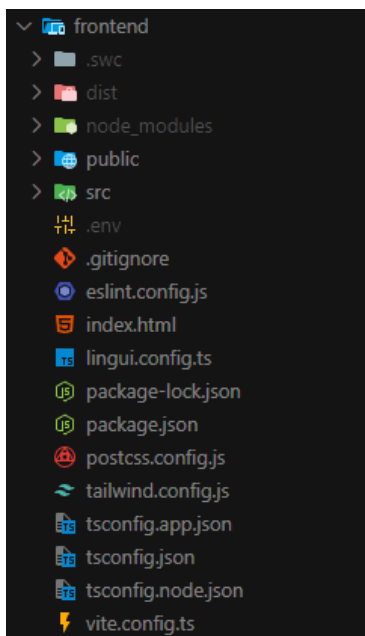


### 4.2 Organizarea codului

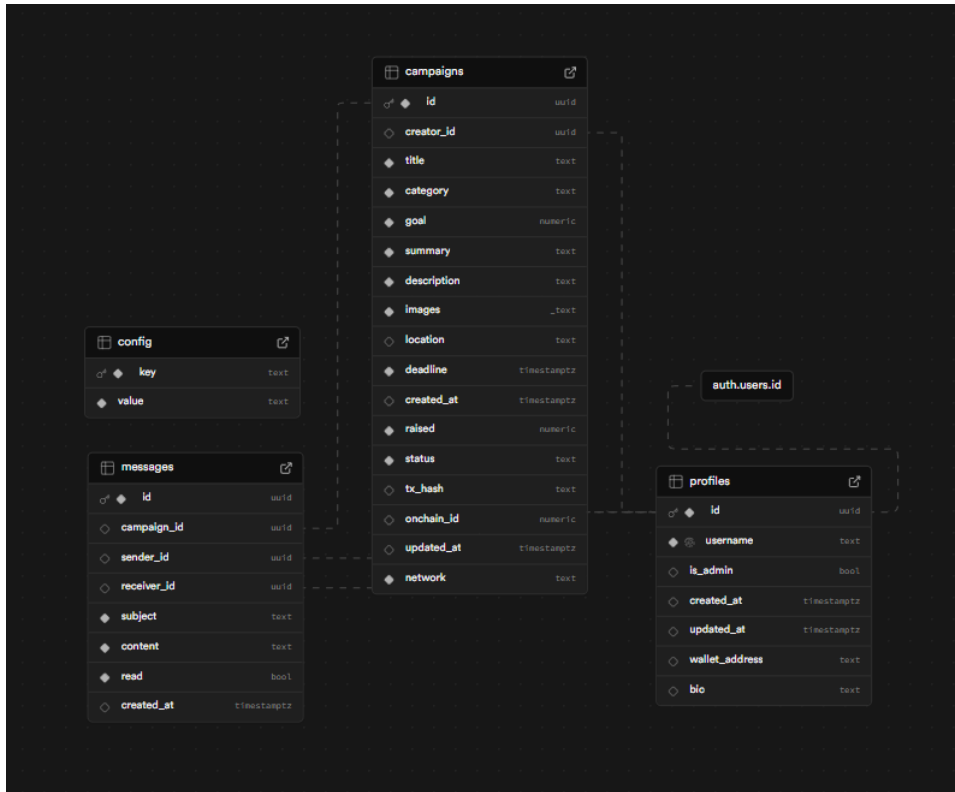
#### 1. Structura directorului **blockchain**



#### 2. Structura directorului **frontend**



### 3. Structura bazei de date de pe supabase



## 4.3 Fișiere de configurare

- hardhat.config.js

```
1 import "@nomicfoundation/hardhat-toolbox";
2 import * as dotenv from "dotenv";
3 import { HardhatUserConfig } from "hardhat/config";
4
5 dotenv.config();
6
7 const { ALCHEMY_API_KEY, WALLET_PRIVATE_KEY, ETHERSCAN_API_KEY } = process.env;
8
9 const config: HardhatUserConfig = {
10   solidity: "0.8.28",
11   networks: {
12     sepolia: {
13       url: `https://eth-sepolia.g.alchemy.com/v2/${ALCHEMY_API_KEY}`,
14       accounts: WALLET_PRIVATE_KEY ? [WALLET_PRIVATE_KEY] : [],
15     },
16     mainnet: {
17       url: `https://eth-mainnet.g.alchemy.com/v2/${ALCHEMY_API_KEY}`,
18       accounts: WALLET_PRIVATE_KEY ? [WALLET_PRIVATE_KEY] : [],
19     },
20   },
21   etherscan: {
22     apiKey: ETHERSCAN_API_KEY,
23   },
24 };
25
26 export default config;
```

- vite.config.js

```
1 import { lingui } from "@lingui/vite-plugin";
2 import react from "@vitejs/plugin-react-swc";
3 import { defineConfig } from "vite";
4
5 export default defineConfig({
6   plugins: [
7     react({
8       plugins: [["@lingui/swc-plugin", {}]],
9     }),
10    lingui(),
11  ],
12  optimizeDeps: {
13    exclude: ["lucide-react"],
14  },
15 });
16
```

- package.json (front-end)

```
1 {
2   "name": "project",
3   "private": true,
4   "version": "0.0.0",
5   "type": "module",
6   "scripts": {
7     "dev": "vite --host",
8     "build": "tsc && vite build",
9     "lint": "eslint . --ext ts,tsx --report-unused-disable-directives --max-warnings 0",
10    "preview": "vite preview"
11  },
12  "dependencies": {
13    "@emailjs/browser": "^4.4.1",
14    "@lingui/cli": "^5.3.1",
15    "@lingui/core": "^5.3.1",
16    "@lingui/core/macro": "^5.3.1",
17    "@lingui/react": "^5.3.1",
18    "@lingui/swc-plugin": "^5.5.2",
19    "@supabase/supabase-js": "^2.39.3",
20    "@vitejs/plugin-react-swc": "^3.9.0",
21    "date-fns": "^4.1.0",
22    "ethers": "^6.9.0",
23    "framer-motion": "^11.0.3",
24    "lucide-react": "^0.309.0",
25    "react": "^18.2.0",
26    "react-dom": "^18.2.0",
27    "react-hot-toast": "^2.4.1",
28    "react-router-dom": "^7.5.2"
29  },
30  "devDependencies": {
31    "@lingui/vite-plugin": "^5.3.1",
32    "@types/react": "^18.2.43",
33    "@types/react-dom": "^18.2.17",
34    "@typescript-eslint/eslint-plugin": "^6.14.0",
35    "@typescript-eslint/parser": "^6.14.0",
36    "@vitejs/plugin-react": "^4.2.1",
37    "autoprefixer": "^10.4.17",
38    "eslint": "^8.55.0",
39    "eslint-plugin-react-hooks": "^4.6.0",
40    "eslint-plugin-react-refresh": "^0.4.5",
41    "postcss": "^8.4.33",
42    "tailwindcss": "^3.4.1",
43    "typescript": "^5.2.2",
44    "vite": "^5.0.8"
45  }
46 }
47
```



## 4.4 Prezentare vizuală a aplicației

- Pagina cu campaniile mele

**Campaniile mele**  
2 Active • 1 Încheiate • 0 Finalizate • 0 Inactive

**Solar-Powered Community Hub**  
Install solar panels to power a neighborhood community center.  
4.680 ETH strâns ≈ \$8,430.037 USD 58.5%  
Brasov 7 zile rămase  
8x7099...79C8  
8.000 ETH obiectiv

**Community Art Gallery DAO3**  
A decentralized autonomous organization for community-owned art galleries.  
0.000 ETH strâns ≈ \$0 USD 0.0%  
New York 3 zile rămase  
0.838 ETH obiectiv

**Decentralized Education Platform**  
Building a blockchain-based platform to make education accessible to everyone.  
0.000 ETH strâns ≈ \$0 USD 0.0%  
Global Încheiat  
5.000 ETH obiectiv

- Pagina de creare și editare a campaniei

**Creează-ți campania** Finalizare: 0%

Titlu campanie \*  
Dăți titlului campaniei un titlu captivant (0/50 caractere)

Categorie \*  
Tehnologie, Artă, Muzică, Film, Jocuri, Publicare, Modă, Măncare, Comunitate, Educație, Mediu, Sănătate, Altele

Obiectiv de finanțare \*  
Sumă USD, Sumă ETH  
Prețul curent al ETH: \$1,804.44 Intervalul de sumă ETH: 0.01 - 10000 ETH

Data de încheiere  
05/04/2025  
Campania trebuie să ruleze cel puțin 1 săptămână de azi și nu mai mult de 2 ani

**Editează campania**

Titlu campanie \*  
Solar-Powered Community Hub (27/50 caractere)

Network  
Rețea de dezvoltare locală  
Rețeaua nu poate fi schimbată după crearea campaniei

Categorie \*  
Tehnologie, Artă, Muzică, Film, Jocuri, Publicare, Modă, Măncare, Comunitate, Educație, Mediu, Sănătate, Altele

Data de încheiere  
05/04/2025  
Data finală a campaniei nu poate fi modificată după creare

Obiectiv de finanțare  
8 ETH ≈ \$14,434 USD  
Obiectivul de finanțare nu poate fi modificat după crearea campaniei

- Acțiunile campaniei

**Șterge campanie**  
Sunteți sigur că doriți să ștergeți Decentralized Education Platform? Această acțiune va închide campania pe blockchain și nu poate fi anulată.  
**Important**  
Te rugăm să îți retragi orice fonduri rămase înainte de a șterge această campanie.  
Anulează Șterge

**Contribuie la această campanie**  
Contribuie la Solar-Powered Community Hub și ajută-o să se facă realitate  
Sumă de contribuit  
\$ 0.001 or more ETH  
Contribuție minimă: 0.001 ETH  
Obiectiv campanie 8.00 ETH  
Sumă rămasă 3.32 ETH  
Contribuția ta 0.000 ETH  
Anulează Contribuie

**Retragere fonduri**  
Campanie: Solar-Powered Community Hub  
Sumă: 4.680 ETH  
Veți primi: 4.563 ETH  
O taxă de platformă de 2.5% este aplicată tuturor retragerilor.  
**Important**  
Retragerea fondurilor va închide această campanie permanent. Această acțiune nu poate fi anulată.  
Anulează Retragere

- Pagina cu detaliile campaniei

Explorează

🇷🇴
🔄
Începe o campanie
🔔
Alex

← Înapoi la campanii

7 zile rămase

04.05.2025

Obiectiv: 8.000 ETH ≈ \$14433 USD

Brasov

0x7899...79C8

Status blockchain

Status: ACTIV

Retragere fonduri

Retragere fonduri pentru a încheia această campanie.

Despre creator

A

Alex

Brasov

Tu ești creatorul

Mediu

Solar-Powered Community Hub

Install solar panels to power a neighborhood community center.

4.680 ETH

≈ \$8443 USD

58.5%

of 8.000 ETH goal

We want to equip our local community hub with rooftop solar arrays, cutting electricity bills and carbon

- Pagina de explore

Explorează

🇷🇴
🔄
Începe o campanie
🔔
Alex

4.680 ETH strâns

2 Campanii active

1 Campanii încheiate

Caută campanii...

Active Încheiate Finalizate

Categorii

Toate

Tehnologie

Artă

Muzică

Film

Jocuri

Publicare

Modă

Mâncare

Comunitate

Educație

Mediu

Sănătate

Altele

Mediu

Solar-Powered Community Hub

Modă

Community Art Gallery DAO3

- Pagina de setări

Explorează

🇷🇴
🔄
Începe o campanie
🔔
Alex

Profil

Portofel

Securitate

Setări profil

Email

test@test.com

Nume aliat

Alex

Biografie

Povesteste-ne despre tine...

Salvează modificările

Vezi profilul public

Vezi cum îți vizualizează alții profilul

Vezi profilul

Deconectare

9

## 5. Detalii de implementare

### 5.1 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  function contribute(uint256 _campaignId) external payable {
2      CampaignData storage campaignData = campaigns[_campaignId];
3      require(campaignData.status != CampaignStatus.CLOSED, "Campaign is closed");
4      require(block.timestamp < campaignData.deadline, "Campaign has ended");
5      require(msg.value > 0, "No ETH sent");
6      require(msg.sender != campaignData.creator, "Creator cannot fund their own campaign");
7
8      campaignData.totalFunded += msg.value;
9      contributions[_campaignId][msg.sender] += msg.value;
10     emit ContributionMade(_campaignId, msg.sender, msg.value);
11 }
```

#### 2. Testare smart contract

```
Campaign Contract
✓ Should create a new campaign
✓ Should not allow the creator to fund their own campaign
✓ Should allow contributions
✓ Should ensure totalContributions is equal to totalFunded
✓ Should retain totalContributions after withdrawal
✓ Should allow the creator to close the campaign explicitly
✓ Should allow the creator to withdraw funds after closing the campaign explicitly
✓ Should not allow withdrawal if totalFunded is 0
✓ Should allow the creator to update the campaign
✓ Should not allow non-creators to update the campaign
✓ Should not allow non-creators to withdraw funds
✓ Should not allow contributions after the deadline
✓ Should not allow non-creators to close a campaign
✓ Should allow contributions exceeding the campaign goal
✓ Should hold 2.5% fee on withdraw and allow feeReceiver to collect it

15 passing (1s)
```

```

1  it("Should allow contributions", async function () {
2    const goal = ethers.parseEther("5");
3    const deadline = Math.floor(Date.now() / 1000) + 3600;
4    const metadataCID = "QmExampleCID";
5
6    await campaign.createCampaign(goal, deadline, metadataCID);
7
8    const contribution1 = ethers.parseEther("2");
9    await campaign.connect(addr1).contribute(1, { value: contribution1 });
10
11    const campaignData = await campaign.getCampaign(1);
12    expect(campaignData.totalFunded).to.equal(contribution1);
13
14    const contribution2 = ethers.parseEther("1");
15    await campaign.connect(addr2).contribute(1, { value: contribution2 });
16
17    const updatedData = await campaign.getCampaign(1);
18    expect(updatedData.totalFunded).to.equal(ethers.parseEther("3"));
19  });

```

### 3. Deployment

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

```

1  import { ethers } from "hardhat";
2
3  async function main() {
4    const Campaign = await ethers.getContractFactory("Campaign");
5
6    const campaign = await Campaign.deploy();
7
8    console.log("Campaign contract deployed to:", campaign.target);
9  }
10
11  main().catch((error) => {
12    console.error("Error deploying the contract:", error);
13    process.exitCode = 1;
14  });
15

```

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

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

### 1. Configurare proiect

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

### 2. Conectare la wallet prin MetaMask

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

```

1  if (!window as any).ethereum {
2    throw new Error('MetaMask is not installed');
3  }
4
5  const [newAddress] = (await (window as any).ethereum.request({
6    method: "eth_requestAccounts",
7  })) as string[];
8
9  if (!newAddress) {
10    throw new Error('No account found');
11  }
12
13  const { data, error: updateErr } = await supabase
14    .from("profiles")
15    .update({ wallet_address: newAddress })
16    .eq("id", user.id)
17    .select();

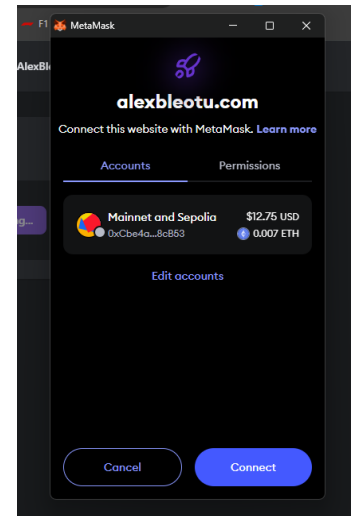
```

- frontend/src/hooks/useMetaMask.ts

```

1  try {
2      if (!localStorage.getItem("walletAddress")) {
3          await connectWallet();
4      }
5      if (localStorage.getItem("walletAddress")) {
6          setIsConnected(true);
7          setIsLocked(false);
8          setError(null);
9          window.location.reload();
10     }
11     } else {
12         setIsConnected(true);
13         setIsLocked(false);
14         setError(null);
15     }
16 } catch (err: any) {
17     setError(err.message || 'Failed to connect to MetaMask');
18 }

```



### 3. Comunicare cu smart contract prin Ethers.js

- frontend/src/hooks/useCampaignContract.ts

```

1  const contribute = async (id: number, amount: string) => {
2      if (!contract) throw new Error('Contract not initialized');
3      setLoading(true);
4      try {
5          if (!amount?.trim()) throw new Error('Invalid amount');
6          const value = ethers.parseEther(amount);
7          const tx = await contract.contribute(id, { value });
8          return await tx.wait();
9      } catch (error) {
10         console.error("Contribution error:", error);
11         throw error;
12     } finally {
13         setLoading(false);
14     }
15 };

```

### 4. Conectarea utilizatorului

- frontend/src/context/AuthContext.tsx

```

1  const signIn = async (email: string, password: string) => {
2      const { error } = await supabase.auth.signInWithPassword({
3          email,
4          password,
5      });
6      if (error) throw error;
7  };

```

- frontend/src/hooks/useAuth.ts

```

1  import { useContext } from 'react';
2  import { AuthContext } from '../contexts/AuthContext';
3
4  export function useAuth() {
5      const context = useContext(AuthContext);
6      if (context === undefined) {
7          throw new Error('useAuth must be used within an AuthProvider');
8      }
9      return context;
10 }

```

## 5. Internaționalizarea aplicației cu LinguiJS

- Marcarea textului drept translatabil cu `t`...``

```
1 <h1 className="mt-8 lg:mt-0 text-4xl sm:text-5xl lg:text-6xl font-bold text-text mb-6 max-w-2xl mx-auto">
2   {t`Revolutionize Your`}{" "}
3   <span className="text-transparent bg-clip-text bg-gradient-to-r from-primary to-primary-dark">
4     {t`Crowdfunding`}
5   </span>{" "}
6   {t`Journey`}
7 </h1>
8 <p className="text-xl text-text-secondary mb-8 leading-relaxed">
9   {t`Join the future of fundraising with blockchain technology. Transparent, secure, and efficient crowdfunding for the digital age.`}
10 </p>
```

- Extragere de mesaje – **npm lingui extract** (generează fișiere .po)
  - frontend/src/locales/en/messages.po
  - frontend/src/locales/ro/messages.po
- **LanguageContext.tsx**

```
1 async function dynamicActivate(locale: Locale) {
2   let messages;
3   switch (locale) {
4     case "en":
5       ({ messages } = await import("../locales/en/messages.po"));
6       break;
7     case "ro":
8       ({ messages } = await import("../locales/ro/messages.po"));
9       break;
10    default:
11      throw new Error(`Unsupported locale: ${locale}`);
12    }
13    i18n.load(locale, messages);
14    i18n.activate(locale);
15  }
16 }
```

## 6. Notificările și mesajele din aplicație

- frontend/src/contexts/useMessages.ts

```
1 const sendMessage = async (
2   campaignId: string | null,
3   receiverId: string,
4   subject: string,
5   content: string
6 ) => {
7   try {
8     const { error } = await supabase.from("messages").insert({
9       campaign_id: campaignId,
10      sender_id: user?.id,
11      receiver_id: receiverId,
12      subject,
13      content,
14    });
15
16    if (error) throw error;
17    return { success: true };
18  } catch (err) {
19    console.error("Error sending message:", err);
20    return { success: false, error: t`Failed to send message` };
21  }
22 }
```

## 7. Conectare la supabase

- frontend/src/lib/supabase.ts

```
1 import { createClient } from '@supabase/supabase-js';
2
3 const supabaseUrl = import.meta.env.VITE_SUPABASE_URL;
4 const supabaseAnonKey = import.meta.env.VITE_SUPABASE_ANON_KEY;
5
6 if (!supabaseUrl || !supabaseAnonKey) {
7   throw new Error('Missing Supabase environment variables');
8 }
9
10 export const supabase = createClient(supabaseUrl, supabaseAnonKey);
```

## 8. Trimiterea de emailuri prin formularul de contact

```
1 emailjs
2   .send(
3     "gmail",
4     "template",
5     {
6       user_name: formData.name,
7       user_email: formData.email,
8       message: `Subject: ${formData.subject}\n\nMessage: \n${formData.message}`,
9     },
10    import.meta.env.VITE_EMAILJS_PUBLIC_KEY
11  )
```

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

### 1. Schema bazei de date

table_name	columns
campaigns	{id,creator_id,title,category,goal,summary,description,images,location,deadline,created_at,raised,status,tx_hash,onchain_id,updated_at,network}
config	{key,value}
messages	{id,campaign_id,sender_id,receiver_id,subject,content,read,created_at}
profiles	{id,username,is_admin,created_at,updated_at,wallet_address,bio}

### 2. Bucket de stocare – campaign-images

< 16d025f3-860e-41ae-8ca3-4a976e047795	
16d025f3-860e-41ae-8ca3-...	1745704160081-education-...
f461fd22-b43b-4a19-b6a5-...	1745704629360-education...

## 6. Resurse hardware și software necesare

### 6.1 Hardware minim recomandat

Pentru dezvoltarea și rulare locală 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

### 6.2 Software și unelte

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



## 7. 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

### 2. Recompense sub formă de NFT

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

### 3. Contribuții recurente

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

### 4. Dashboard

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

### 5. 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.

## 8. 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/>