Pametni ugovor za Crowdfunding

Blockchain Solidity

|  |  |  |
| --- | --- | --- |
| Ime I prezime | Email | GitHub |
| Luka Pavlović | lukampavlovic03@gmail.com | https://github.com/alexazivkovic/CrowdfundingSolidity |
| Aleksa Živković | aleksa.zivkovic.1208@gmail.com |

Sadržaj

[1 Opis aplikacije 3](#_Toc178355384)

[1.1 Funkcionalnosti aplikacije 3](#_Toc178355385)

[1.2 Interakcije korisnika sa aplikacijom 3](#_Toc178355386)

[2 Implementacija 3](#_Toc178355387)

[2.1 Arhitektura pametnog ugovora 3](#_Toc178355388)

[2.1.1 Crowdfunding.sol 3](#_Toc178355389)

[2.1.2 CrowdfundingFactory.sol 4](#_Toc178355390)

[3 Korisničko uputstvo 5](#_Toc178355391)

[3.1 Uputstvo za korišćenje aplikacije 5](#_Toc178355392)

[3.2 Foto prilog uz dokumentaciju 6](#_Toc178355393)

# Opis aplikacije

Svrha aplikacije je omogućavanje vlasnicima projekata da kreiraju crowdfunding kampanje za izbore, dok korisnici mogu donirati sredstva u obliku Ether-a. Ova aplikacija koristi pametne ugovore na Ethereum blockchain-u kako bi osigurala transparentnost i sigurnost u prikupljanju sredstava.

## Funkcionalnosti aplikacije

1. **Kreiranje projekta**: Vlasnik može da kreira novi projekat sa sledećim podacima:
   * **Opis**: Kratak opis projekta.
   * **Rok**: Vreme do kada donacije traju.
   * **Potrebna suma**: Ukupan iznos koji je potrebno prikupiti.
2. **Doniranje**: Korisnici mogu da doniraju sredstva u Ether-u. Donacije se beleže na blockchain-u, a korisnici mogu pratiti svoj doprinos kampanji.
3. **Provera statusa projekta**: Bilo koji korisnik može proveriti koliko je sredstava prikupljeno za određeni projekat, kao i trenutni status procesa doniranja.
4. **Emitovanje događaja**: Svaka donacija i povlačenje sredstava emituje događaj koji sadrži:
   * **Adresu korisnika**: Ko je izvršio donaciju.
   * **Iznos donacije**: Koliko je sredstava donirano.
   * **Status kampanje**: Trenutni status prikupljanja sredstava.
5. **Konstruktor**: Postavlja vlasnika kampanje prilikom kreiranja ugovora.

## Interakcije korisnika sa aplikacijom

* **Vlasnici**: Kreiraju projekte putem CrowdfundingFactory ugovora, unoseći potrebne podatke.
* **Korisnici**: Doniraju sredstva, pregledaju istoriju svojih donacija i prate status procesa doniranja.

# Implementacija

## Arhitektura pametnog ugovora

Aplikacija se sastoji od dva pametna ugovora: Crowdfunding i CrowdfundingFactory.

### Crowdfunding.sol

solidity

Copy code

// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.8.2 <0.9.0;

contract Crowdfunding {

address payable public vlasnikKampanje;

uint public potrebnaSuma;

uint public trenutnaSuma;

string public nazivProjekta;

string public opisProjekta;

uint public rokProjekta;

mapping (address => uint) individualniDoprinosi;

event EvidencijaUplate(address uplatilac, uint iznos, uint status);

constructor(address payable vlasnik, uint potSuma, string memory naziv, string memory opis, uint rok) {

vlasnikKampanje = vlasnik;

potrebnaSuma = potSuma;

trenutnaSuma = 0;

nazivProjekta = naziv;

opisProjekta = opis;

rokProjekta = block.timestamp + rok;

}

function doniraj() external payable {

if (msg.value <= 0) {

revert("Ne mozete uplatiti iznos manji od nule.");

}

if (block.timestamp > rokProjekta || trenutnaSuma >= potrebnaSuma) {

revert("Donacije su zavrsene.");

}

individualniDoprinosi[msg.sender] += msg.value;

trenutnaSuma += msg.value;

emit EvidencijaUplate(msg.sender, msg.value, trenutnaSuma);

vlasnikKampanje.transfer(msg.value);

}

function istorijaDonacija() external view returns (uint) {

return individualniDoprinosi[msg.sender];

}

}

**Ključni delovi koda:**

* **Konstruktor**: Inicijalizuje vlasnika, potrebnu sumu, naziv, opis i rok projekta.
* **Doniraj funkcija**: Omogućava korisnicima da doniraju sredstva i beleži donacije.
* **Istorija donacija**: Vraća ukupan iznos donacija od strane tog konkretnog korisnika.

### CrowdfundingFactory.sol

solidity

Copy code

// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.8.2 <0.9.0;

import "./Crowdfunding.sol";

contract CrowdfundingFactory {

Crowdfunding[] public projekti;

function kreirajProjekat(address payable vlasnik, uint pSuma, string memory naziv, string memory opis, uint rok) public {

projekti.push(new Crowdfunding(vlasnik, pSuma, naziv, opis, rok));

}

function prikaziSveProjekte() public view returns (Crowdfunding[] memory) {

return projekti;

}

}

**Ključni delovi koda:**

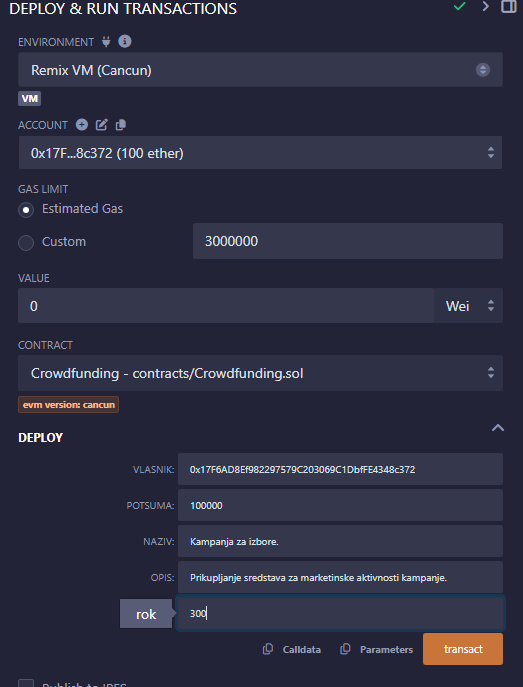
* **Kreiraj projekat**: Funkcija koja stvara novi Crowdfunding ugovor i dodaje ga u niz projekti.
* **Prikazi sve projekte**: Vraća sve kreirane projekte.

# Korisničko uputstvo

## Uputstvo za korišćenje aplikacije

1. **Kreiranje projekta**:
   * Korisnik poziva funkciju kreirajProjekat na CrowdfundingFactory ugovoru.
   * Unosi potrebne parametre: adresu vlasnika, potrebnu sumu, naziv, opis i rok.
2. **Doniranje**:
   * Korisnici mogu koristiti funkciju doniraj u Crowdfunding ugovoru sa određenim iznosom Ether-a.
   * Svaka donacija se beleži i emitovaće se događaj koji sadrži informacije o donaciji.
3. **Provera statusa projekta**:
   * Korisnici mogu proveriti status prikupljanja sredstava koristeći javne funkcije dostupne u Crowdfunding ugovoru.

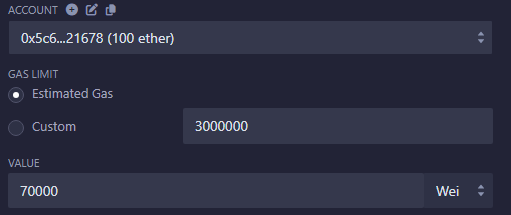
## Foto prilog uz dokumentaciju



Klikom na dugme *transact* kreira se projekat.

Capture2.PNG

U konzoli se prikazuje potvrda o uspresno kreiranoj transakciji.



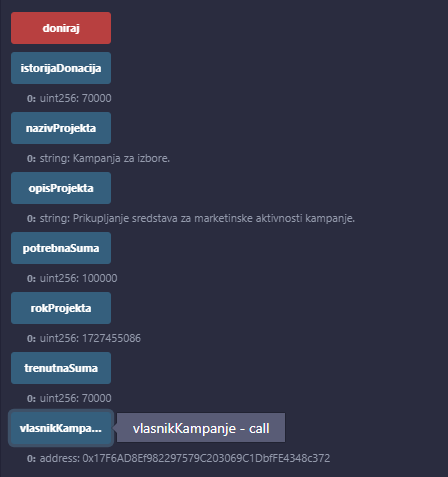
Unos parametara kako bi gorenavedeni korisnik izvršio donaciju od 70000 Wei.

Capture4.PNG

Potvrda prethodnog koraka.

Capture5.PNG

U konzoli se prikazuje potvrda o uspresno kreiranoj transakciji.



Prikaz ažuriranih podataka.