

BLOKZİNCİR TÜMLEŞİK HİZMET SİSTEMİ

Gelişen dünyada blokzinciri kamuya entegre etme zamanı!

PROJE YETKİLİLERİ

Y. Emre AKŞİTKÜTAY

B. A. Buğra ALTUNSOY

BİZ KİMİZ?

Projemizi tanıtmadan önce kendimizi tanıtmamızda fayda var.

İki kişilik bir ekibiz ve kripto merakımız yılları saran bir aktif saha tecrübesine sahip.



SUNULAN ÇÖZÜMLER



Çözüm # 1

Sertifikaları NFT'leştirerek ve ders başına alınan rozet parçalarını birleştirip oyunlaştırarak değerlerini ve devam yüzdesini arttırıyoruz.

Çözüm # 2

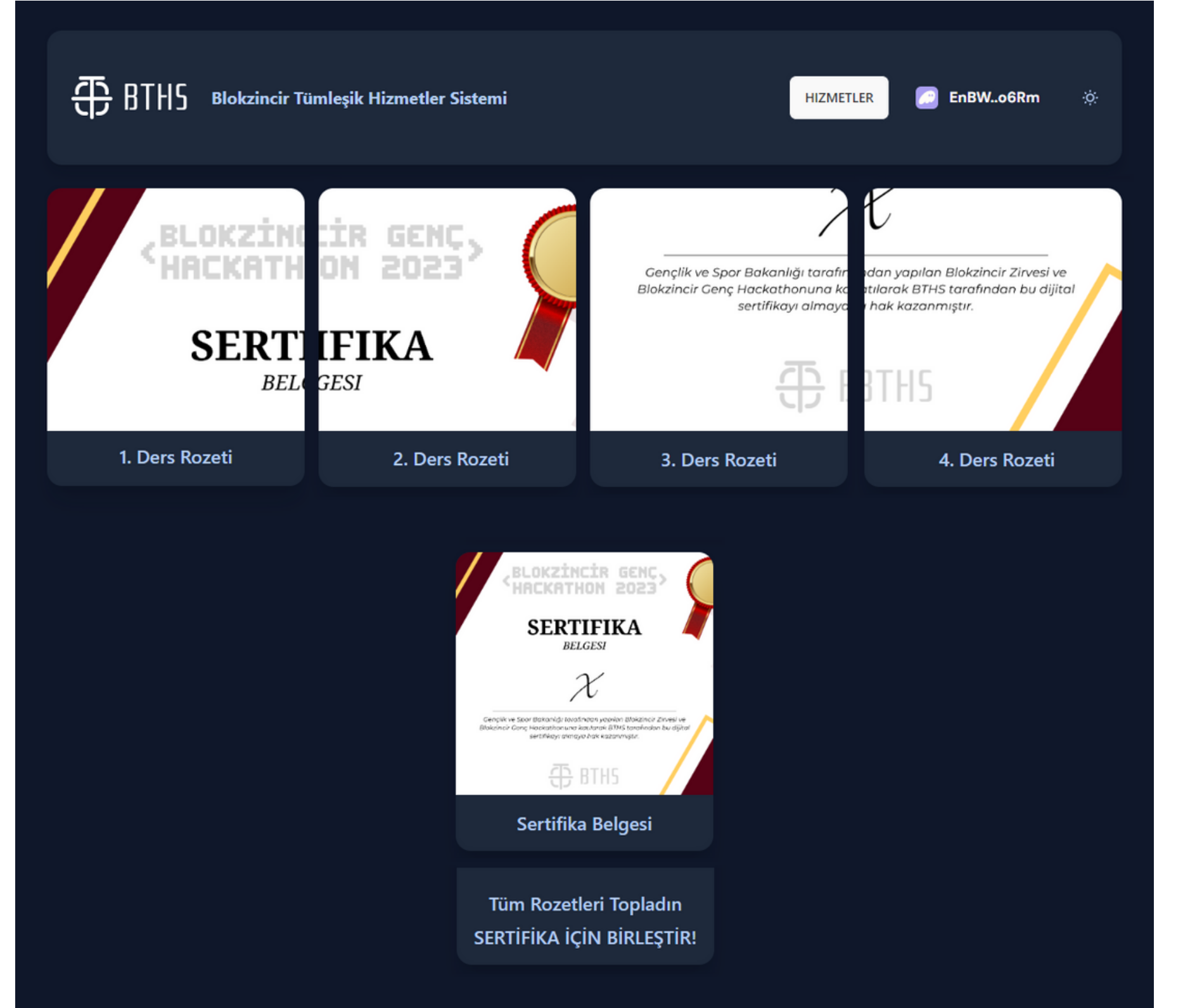
HibeDAO adlı DAO sistemimizle, birçok kurumdan alınan hibeleri kalifiye hale getirerek sertifika sahiplerine başvuru ve ön elededeki projeleri oylama hakkı tanıyoruz, bu şekilde kurumların iş yükünü azaltıyoruz.

Çözüm # 3

Bu NFT'ler, ileri seviye kurslara erişim ve iş dünyasındaki avantajlar sağlarken, işverenler ve eğitim kurumlarının da kursiyerlerin başarılarını doğrulamasına olanak tanıyor.

SERTİFİKA SİSTEMİ

Özel firmaların kullandığı oyunlaştırma kavramını devlet kurumlarına özelleştirerek, kamunun yararına verimli sertifika ve eğitimler sağlayarak ciddi bir adım atıyoruz.



HİBEDAO

HibeDAO adlı bir oluşum kurarak, sertifika sahiplerinin devlet kurumlarının hibe programlarına başvurmasını ve ön elemede projeleri oylamasını sağlıyoruz.

The screenshot displays the Solana Playground interface at `beta.solpg.io`. The left panel shows the program configuration for `hello_world` under the `Instructions` tab. It includes fields for `initVoteBank` and `gibVote`, each with `Accounts` and `Args`. The `gibVote` instruction has `voteType: u64` set to `0`. The right panel shows the Rust code for the program, which includes a `declare_id` for the program ID, a `#[program]` block, and functions `init_vote_bank` and `gib_vote`. The `gib_vote` function handles voting for "OK" or "Not OK" and updates the vote counts. The bottom panel shows the execution results, including the program ID, compute units consumed, and the success status. The output also shows the confirmed transaction, the airdrop amount, and the signature.

```
2 declare_id!("A5zbZUxgGGM4HhQiu6WkRejueTRUkwjyyBReNLRvQnnr");
3
4 #[program]
5 pub mod hello_world {
6     use super::*;
7     pub fn init_vote_bank(ctx: Context<InitVote>) -> Result<()> {
8         // Open vote bank for public to vote on our favorite "GM" or "GN"
9         ctx.accounts.vote_account.is_open_to_vote = true;
10        Ok(())
11    }
12    pub fn gib_vote(ctx: Context<GibVote>, vote_type: u64) -> Result<()> {
13        ctx.accounts.vote_account.is_open_to_vote;
14        match vote_type {
15            0 => {
16                msg!("Voted for OK 🍷");
17                ctx.accounts.vote_account.yes += 1;
18            }
19            1 => {
20                msg!("Voted for Not OK 🍷");
21                ctx.accounts.vote_account.no += 1;
22            }
23            _ => {
24                msg!("Invalid vote type");
25            }
26        }
27        Ok(())
28    }
29 }
30
31 0 implementations
32 #[derive(Accounts)]
33 pub struct InitVote<'info> {
34     // Making a global account for storing votes
35     #[account(
36         init,
```

Program A5zbZUxgGGM4HhQiu6WkRejueTRUkwjyyBReNLRvQnnr consumed 2263 of 200000 compute units
Program A5zbZUxgGGM4HhQiu6WkRejueTRUkwjyyBReNLRvQnnr success

Confirmed
\$ solana airdrop 5
Requesting airdrop of 5 SOL

Signature: 5xZZPG5ZUmGaEchmiLyHZ7AsWDos2rxjJX6WDzECa4PPBAMb18bfyXGQbBJeALkU5tNTsTdR3gGE92Us8aJFiaf

9.53506664 SOL
\$

Connected to Playground Wallet - devnet | 66ZZptgT8Q7a1DMBd33ceo11WA7Kxocr6s7jAEEyWpNf | 9.53506664 SOL

ŞEMATİK HARİTA

