3327

Assignment

Thank you once again for sharing your interest in joining the 3327 crew.
We created this task to test your motivation and to make sure that you are a fast learner and self-driven individual capable of comprehending the problems and offering potential solutions on your own.
Hence, below you will find more information about the challenge that the Kingdom of Wakanda is facing and how you can help King T'Challa and his nation by utilizing the benefits of blockchain technology.

Once you finish with the task, please share the GitHub repository with your solution and deploy smart contracts on the testnet of your choice.

Wakanda Forever

The Kingdom of Wakanda is organizing an election for officials who will represent King T'Challa at future United Nation summits. Being very modern and forward-thinking, King T'Challa has decided to leverage the power of blockchain technology to make sure these elections are unhindered, and more importantly, **secure**.

In order to make sure all 6 million residents have access to voting, multiple voting centres have been opened across the country. These centres are places where Wakandans can come and cast their ballot for officials of their choice.

As the team lead for the voting committee, your task is to implement the following features in these voting centres:

- A simple UI that allows Wakandans to register for voting, displays the candidates and allows Wakandans to vote
- A Smart Contract that keeps track of who voted for which candidates
- An ERC20 token required for voting (called WKND)
- A local NodeJS Server that collects data from the blockchain

The User Interface (UI)

Registration page

Each Wakandan has a **unique Ethereum address** that they use to vote.
In order for them to vote in the election, they must have a balance of **at least 1 WKND** (ERC20 token described later). To get this token, they must register to vote on a special page.

The registration page needs to contain only 1 input field and a **REGISTER** button. The input field is where Wakandans will input their Ethereum address.

When the **REGISTER** button is clicked, **1 WKND** token should be sent to the inputted address.

Voting page

Only 3 candidates can be chosen as UN officials. This means that there are only 3 candidates who will win and represent Wakanda.

To help speed things along, King T'Challa's sister - Shuri, has already implemented a server with the following endpoint:

```
// GET: https://wakanda.zmilos.com/list
// Desc: Returns the list of candidates
// RESPONSE:
interface CandidatesResponse {
    candidates: [
      {
        name: string;
        age: number;
        cult: string;
      }
    ]
}
```

The voting / register pages can be done using regular HTML / CSS / JS, there is no need to use a front-end framework.

The voting page needs to have the following elements:

- List of candidates to pick from (UI element doesn't matter too much, it can be radio buttons, dropdown etc.)
- Input field for the number of **WKND** tokens to be spent for voting
- **VOTE** button that casts the vote
- **Show Leads** button that displays the top 3 candidates (explained later). This button should trigger the **winningCandidates** SC call.

If someone attempts to vote twice, an error message should appear. Likewise, an error message should be displayed if, for whatever reason, voting is not possible at the moment.

Smart Contract

The Wakanda Voting Smart Contract, among other things, needs to implement the following elements:

- ***(function)*** **winningCandidates** - Returns the top 3 candidates by vote count
- ***(Event)*** **NewChallenger** - Emitted whenever a **new** candidate enters one of the top 3 spots. For example, the first votes for 3 different candidates will trigger this event (because the top 3 slots were empty). If candidates **A**, **B** and **C** are holding the top 3 spots, and a candidate **D** surpasses C in *vote count*, this event should be emitted

WKND ERC20 Token

The **WKND** token is an ERC20 token used for casting votes. It serves as proof of voter registration. It is safe to assume that **1 WKND = 1 vote**, and that each Ethereum address (citizen) can vote **only once** for only **one candidate**. Note that the token is transferable and one address can own multiple of them, in other words, one candidate can get multiple votes from one address.

The **WKND** token doesn't exist yet, and its creation is part of the overall task.

The Server

This should be a Node.JS server which provides leaderboard like REST API as well as details of candidates which will later be presented on the Frontend

Bonus: Using the server as intermediary

The server acts as a backup device that assists in voting.

Each vote that is cast needs to be relayed through the server. If the server is unable to contact the Smart Contract (the internet is down), the server needs to cache / store the voting data ***securely***, and attempt to cast the vote once the connection is back online.

Good luck!