Blockchain-Powered Voting App

Team members: Calugaritoiu Ion-Victor & Albuica Sergiu-Mihail

---

1. Purpose of the Project:

To develop a decentralized voting application using the MultiversX blockchain. The project will enable secure, transparent, and tamper-proof voting processes for small-scale elections or polls.

---

2. Needs it Tries to Satisfy:

- **Transparency**: Blockchain ensures every vote is recorded and verifiable.
- **Security**: Votes cannot be tampered with or deleted after submission.
- **Ease of Use**: Participants can vote using a simple interface, and results are automatically counted.

---

3. Components of the Project:

a. Smart Contract:

- A Rust smart contract deployed on the MultiversX blockchain that:
    - Manages voter registration.
    - Allows users to cast votes for predefined options.
    - Automatically counts and publishes results.
    - Limits one vote per registered wallet address.

b. Backend:

- A backend service (Rust using MultiversX SDK) to:
    - Deploy the smart contract.
    - Interact with the blockchain for vote submission and result retrieval.
    - Verify voter eligibility (e.g., wallet balance, token ownership, etc.).

c. Frontend:

- A basic CLI or web interface to:
    - Register eligible voters (using wallet addresses).
    - Display voting options.
    - Allow users to cast votes and view results.

---

4. Timeline for Development:

Week	Task Description	Assigned To
1	Research MultiversX SDK (Rust) and finalize design.	Both
2	Develop the Rust smart contract for voting logic.	Victor
3	Create the backend for smart contract interaction.	Sergiu
4	Implement a CLI or web interface for user interaction.	Victor
5	Deploy the system on MultiversX Devnet and test.	Sergiu
6	Debug and add features like wallet verification.	Both
7	Prepare documentation and project report.	Both