Simple Assessment

It takes no more than 30 minutes

※ You can find another test on the left tab of this document

Blockchain developer assessment:

You can access the demo project here:
https://bitbucket.org/dredsoftlabs/prohouse/src/main/

Task: Run the demo project and develop a simple decentralized voting system using an Ethereum smart contract and integrate it with a Node.js backend using web3.js:

- Requirements:

1) Create a Solidity smart contract called Voting

   Contract Features:

   - Only the contract owner can add candidates.
   - Users can vote for a candidate only once.
   - Provide a function to retrieve all candidates and their vote counts.
   - Provide a function to declare the winner.

   Suggested Structure:

```
struct Candidate {
    string name;
    uint voteCount;
}
```

   Functions to implement:

   - addCandidate(string memory name)
   - vote(uint candidateIndex)

- getCandidates() returns (Candidate[] memory)
- getWinner() returns (string memory name)

Deploy it on a local Hardhat node.

2) Create a Node.js backend with the following features:
   - POST / candidates: Add a candidate (only owner account)
   - GET / candidates: List all candidates and their vote counts
   - POST / vote: Cast a vote for a candidate (pass account address and candidate index)
   - GET / winner: Return the winner's name

- **Evaluation Criteria**:

1. Smart contract:

   Logic correctness, safety (e.g., modifiers), proper use of mappings and structs

2. Integration

   Clean integration using Web3.js, appropriate async handling

3. Code Quality

   Organization, modularity, naming conventions

4. Functionality

   All endpoints working and integrating with the blockchain

5. Extra

   Using middlewares, error handling, clean logs

- **Deliverables:**

1. Voting.sol smart contract

2. Deployed contract ABI + address
3. Video explanation of the result of running the project