UPSTATE INTERACTIVE

Ethereum Challenge for Smart Contract Developer Position

Objective

- Demonstrate your knowledge of Solidity and your ability to develop smart contracts for the Ethereum Virtual Machine (EVM).
- Demonstrate your familiarity with and ability to use the Hardhat smart contract framework
- Demonstrate your ability to use OpenZeppelin, a library for secure smart contract development.
- Demonstrate your ability to thoroughly document your code using NatSpec.

Specific Deliverables

- Develop a token contract that inherits from OpenZeppelin's ERC20 base contract and extends its functionality so that tokens can only be transferred after a particular `_startTime` and before a particular `_endTime` that are provided in the constructor.
- Develop corresponding unit tests to account for the `_startTime` and `_endTime` constraints.
- Implement a `Contribution` contract that users can donate ETH to. In return for their ETH-based contributions, your `Contribution` contract should issue them tokens from your token contract in return.
- Your `Contribution` contract should store the addresses of users that donate as well as the amount of ETH they've donated.
- Develop a function in your `Contribution` contract that will accept a wallet address and return the amount of ETH that a wallet address has contributed to the `Contribution` contract.
- Develop unit tests for the `Contribution` contract.

Bonus Deliverable

- Develop events that emit when functions in your token and `Contribution` contracts execute.
- Develop corresponding unit tests for the events.

Additional Comments

- Please share a link to your Github repo once you're complete.
- You may use any publicly available documents or libraries to complete this task.
- You may use up to 5 business days to complete this task.
- Please contact us if you have any questions: doug@upstateinteractive.io || kseniya@upstateinteractive.io.