Solidity Interview Exercise

**Please Read the entire document before starting implementation.

Test duration: 1 hour.

Problem statement

Alice has identified a gap in the DeFi market and has devised a solution to provide better and sustainable yields to his users while balancing risk. Alice needed time and money to build this solution, so he decided to do a token funding round to help solve this problem.

Alice's token with the symbol DEFI is nearing its investor cliff but his tech team needs a few more months to deliver the product.

He comes up with the idea of a **staking** contract where the investors could stake their DEFI tokens in exchange for DEFI as yield which would be deducted from the team segment of the token supply as they are taking longer to deliver.

His tech team is busy and he reaches out to you with the following requirements for a Solidity smart contract:

1. Write a staking contract which accepts DEFI as the staking token and gives out DEFI as the reward.
2. The user should be rewarded a total of 1 DEFI token per day for every 1000 DEFI tokens staked.
3. The rewards must be emitted to the user every block.
4. (Stretch Goal) The user should be able to stake multiple times.
5. The user should be able to withdraw at any time.
6. The user should be able to view their rewards at any given time.
7. Upon withdrawal, the entire stake amount and the rewards should be sent to the user.
8. There is no partial withdrawal of the stake amount

Note: Alice intends to deploy the token on Ethereum (block time =6s)
Example scenarios:

1. User A stakes 1000 DEFI on 1st January(BlockNumber1) and withdraws on 11th January(BlockNumber2).
    a. The user should be paid:
        i. Reward + Original
        ii. Reward + 1000
        iii. (BlockNumber2 - BlockNumber1)*1000 + 1000
2. User A stakes 100 DEFI tokens on 1st February(BlockNumber1), and stakes 900 DEFI

tokens more on 11th February(BlockNumber2) then withdraws funds on 21st February(BlockNumber3).

   a. The user should be paid:
      i. Reward + Original
      ii. Reward on stake 1 + Reward on stake 2 + Original
      iii. Reward on stake 1 + Reward on stake 2 + 1000
      iv. (BlockNumber3 - BlockNumber1)*100 + (BlockNumber3 - BlockNumber2)*900 + 1000

Submission

1. This test is for 1 hour and you are required to zip and share the solution after the completion of an hour.
2. If you are familiar with Hardhat, please use it.

Judging metrics and Expectations

**The criteria mentioned are in preference order.

1. A working solution which solves some of the problem requirements.
   a. Your solution should at least solve for the case where one user can only stake once.
2. Correctness of the solution.
3. The volume of data stored on the contract which in turn impacts the gas price.
4. The execution efficiency of state mutating operations also impacts the gas price.
5. Basic security best practices.
   a. You are not required to spend too much time on this.