



Crowdfunding Your Billion-Dollar Idea: A Solidity Skills Assessment

Background

Imagine that you, or your company, want to create the next billion-dollar idea. Way to go!

One problem: you need more funds to support the development, marketing, and launch.

To source the initial funds, you use a crowdfunding platform and share your idea for fundraising.

A good crowdfunding platform can help your community and your users become shareholders in your project. Web3 is a perfect technology to help build this crowdfunding platform and campaign.

Your Challenge

Create a crowdfunding campaign where users can pledge and claim funds to, and claim funds from, the contract.

Your contract(s) should be written such that:

- Funds take the form of a custom ERC20 token - create your own token for an added challenge
- Each crowdfunded project should have a funding goal
- When a funding goal is not met, customers can get a refund of their pledged funds

Your project must provide the following to be completed:

Functionality

- Project owners can create a new crowdfunding project
- Every new crowdfunded project has a timeline and a funding goal
- Users can fund different projects within the timeline
- If the funds are not successfully raised by the time the campaign ends, users should be able to withdraw their funds

Use the following quality checks along the way to ensure your submission is working as expected:

- The code compiles on Remix/Hardhat
- The code accomplishes the task described in the prompt
- The code has no glaring security issues - you can run through Slither to confirm
- The code is readable and organized
- The smart contract can quickly and easily run on a local network
- The project demonstrates an understanding of common EVM developer tooling, including tools like Truffle, Ganache, Hardhat, etc.
- The code is optimized for gas ([here](#) is a resource to calculate your gas fees)
- The contract is upgradeable (optional)

Project Submission Requirements

1. Submit your project on GitHub as a public repository that we can run locally, preferably using Truffle, Ganache, or Hardhat.

If you prefer to use different tools, provide explicit instructions to run the contract locally. Specifically, we should be able to open the provided repository in VSCode and have the contract running locally within a few minutes.

2. Record a video of 5 min or less. [Loom](#) is a free online tool you could use. In the video, walk through your code by sharing your screen and explaining the code. In the code walk-through, run your code and present the output.

Measurable outcomes

- Demonstrates ability to create and use modifiers appropriately
- Demonstrates ability to create and emit events appropriately
- Demonstrates ability to use contract inheritance appropriately
- Demonstrates ability to validate conditions and throw sensible errors
- Demonstrates ability to properly use global functions to access information about the transaction, block, address, etc.
- Demonstrates ability to choose appropriate memory types for parameters, variables, etc.

Once finished, visit [HERE](#) to submit your GitHub and video links. Please make sure your GitHub and code walk-through video are publicly accessible.

If you need any support, please email us at academy@metacrafters.io or join our [Discord](#)