

DevFest Challenge - Will Dapp (Hard)

- DevFest Challenge - Will Dapp (Hard)
 - Introduction
 - Problem Statement
 - Requirements
 - Expectations
 - Notes
 - Evaluation
 - Setup

Introduction

The smart contract, or as they call it **__transaction protocol**, is a program that lives in the Blockchain and is intended to automatically execute actions based on the terms written on it in an immutable way. This blockchain-based technology provides a solution for many problems that require trust from the two sides.

Problem Statement

Jumping to the future... you're a grandpa now, you were a big successful crypto trader in your youth days, and you left behind you a fortune of ethers *a big one*.

However, you have big fears that your children and inheritors will have conflicts about your fortune after you pass away; and here, the last task of your Software engineer journey comes to find a solution to this problem as you always did before, can you grandpa? 😊

As you were a Blockchain developer **__or more precisely, a web3 developer**, you recognize the potential of this technology and that it provides a perfect solution for such a problem.

Requirements

1. The solution must be Web3 based.
2. All the functionalities implemented on the smart contract must have valid tests written on `tests/will.js`

Expectations

- We expect a working solution that respects the requirements and is implemented in a user-friendly way that is visually appealing.

Notes

- The fortune is an **amount of Ether**.
- The fortune division process must be executed automatically after the owner's death.
- The solidity compiler is configured under the **0.8.17 version**, feel free to change it to your preferred solidity version on `./hardhat.config.js`
- You may use any CSS framework, or any Javascript framework/library to implement the design.
- The use of Blockchain development frameworks other than hardhat is not allowed.
- A minimalistic design that implements the required functionalities is more than accepted.

Evaluation

This challenge has maximum points of **25** for implementing all the required features.

Setup

To get you started with this challenge, we have provided you with a starting point, so you need first to copy to your local environment the `starter` folder you see next to this challenge and install all the dependencies:

```
cd starter  
npm install
```

You can do it **grandpa!** 🐶

↑ **back to top**