TD

Build a decentralized lottery

Goal: Build a decentralized lottery on the Ethereum network.

Description: You will build a decentralized lottery open to everyone. The game is simple, at a regulated interval, the lottery opens, and participants can submit their 5 favorite numbers to the lottery. At the end of the period, the lottery closes, and the winner is revealed.

The administrator of the lottery must fill the lottery with an ERC20 token. The winner is reward with the ERC20 token.

Specifications:

- *The lottery has an end date and start date*
- *The reward must be a ERC20 token and is defined by the admin at the creation*
- *Participants must submit 5 numbers (between 0 to 9) to participate to the lottery*
- *We can see the number of participants to the lottery*
- *We can see the reward amount*

Deliverable:

- A working smart contract
- Demo
- Front-end connected to the smart contract.

Build a NFT game

Goal: Build a decentralized Game

Description: Build a NFT game. The game will be composed of different characters with different attributes (Strength, Sex) Characters can fight together. The issue of the fight is based on the strength and some random. The characters can breed together to create a brand-new character. Characters can augment their strength when they win a fight. To generate a new character user, must pay with ERC20 token.

Specifications:

- *Characters have the following attributes: strength, sex, image, name*
- *Character is a ERC721 token*
- *Characters can fight together*
- *The issue of the fight is determined by the strength and some random*
- *After each fight the winner augment their strength*
- *To create a new character's user must pay with an ERC20 token (created)*
- *User can create new characters by breeding two characters*

Deliverable:
- A working smart contract
- Demo
- Front-end connected to the smart contract (bonus)

Decentralize stack overflow

Goal: Build a decentralized stack overflow

Description: Build a stack overflow on the Ethereum blockchain. Like stack overflow user can post a problem. Any other user can respond and find a solution. But in the decentralized stack overflow user who find a solution will be reward by ERC20 tokens. The amount of the reward is defined by the user who post a problem. The user must validate the solution to reward the other users

Specifications:

- Users can post a problem
- Problem has the following attributes: title, mini description, price.
- Users can submit a solution for a problem
- Problem askers must validate the solution do release the money.
- The reward must be an ERC20 token (*is defined by the admin at the creation)*
- User can't submit a solution to a problem already solved
- User can see all problem posted
- User can see all solutions for a problem

Deliverable:
- A working smart contract
- Demo
- Front-end connected to the smart contract

Kickstarter platform

Goal: Kickstarter platform.

Description: Build a decentralized Kickstarter platform. Users want to raise money for his project, after creating an ERC20 token the user want to sell his token to raise money.
Investors can buy the ERC20 token via the platform.
The crowdfunding platform will have the following specification:

Specifications:

- *A user can list their project in the platform*
- *For a defined project we can see the amount collected*
- *Each token has fixed price*
- *For a defined project we can see the number of investors*
- *A project has a start date and an end date, the investor must wait the start date before investing. No investment will be allowed after the end date.*
- *After the end date, project owner can get the money collected (only the owner)*
- *Project have a minimal contribution, if the minimal contribution is not achieved, the project is cancelled, and all the investors get their money back.*

Deliverable:

- A working smart contract
- Demo
- Front-end connected to the smart contract

Note criteria:

- Project respect all specifications
- Originality of the project