Smart Contract Requirements For Simple Projects

1. Esusu Smart Contracts

General Description:

This smart contract can be used to replace one of the popular ways of saving in Africa, known as “AJO” or “ESUSU”. Typically a certain number of persons are involved. Each person contributes a fixed amount of money over a given period and the total money collected from all members combined is paid to one of the participating members. The same process is repeated for other periods of the same length and the contributed funds are handed over to other participating members until every member has received it, then the members can choose to continue or pull out, and the circle starts again.

Current Challenges

- There is a centralized book used to keep track of payments during every period, A single person called the AJO or ESUSU man or woman, keeps track of everyone one's payment by ticking against their names whenever they pay, and everyone has a separate card in which they keep track of their separate payments to verify against what the Esusu Man or woman has written. Many a time there have been disparities between both records without any clear evidence of where the error is coming from because both are prone to manipulations.
- It is required that during a cycle, no member should pull out after receiving his or her share of the collective funds until the entire cycle is completed and everyone has their share, but this is one of the biggest problems with AJO or ESUSU, many members run away after they have received their share.

DEFI can help solve this problem for a given number n of participating members. Leveraging your knowledge of smart contracts, build a DEFI solution that models this AJO system and solves the current problems. Assume there are 4 participating members and the currency being contributed is ETH. The system should be such that if a person runs away after receiving his funds, others will not suffer, one way this could work is that a user can not withdraw all his received funds at once, be careful though not to defeat the aim of the ESUSU in the process, by letting a member who should be receiving the funds have access to the same amount of money he has just contributed.

2. **Register Login and Private Messaging:**

Registration and logging in are very crucial operations in day-to-day life today. Write a smart contract that can allow anyone (wallet) to create an account, (username, email, and password), and log in to his account anytime with email or username and password.

This smart contract should also allow anyone to leave a private message for any other member of the application, such that no other member can access that message except the member it is meant for. This message should not even be readable from the smart contract slots.

3. **ICO smart contract**

Many Web3-based companies have raised millions of dollars from an ICO. Public ICO lets people with ideas sell tokens to the general public at a predefined rate. These tokens are sold at very cheap prices and are projected to grow in value if the founding team can build out the product as proposed in their white paper. Many people buy into ICOs because if the project does well, they can get good returns on their money and many founders use ICO because it easily allows them to raise start-up funds for their project. Purchase of these tokens is done via interaction with a smart contract, people send in some Native coin e.g ETH or SOL, and in turn, immediately paid out a certain amount of tokens by the smart contract.

Your team is building out the next big thing and you are trying to raise funds for it. Create your ICO smart contract letting people crowd-fund your project and better make sure your project does well, or else we will ensure you go to jail 😅 .

4. **Simple P2P Exchange**

A simple P2P exchange allows two or more persons who do not know themselves to make a transaction.

Typically in these exchanges, there are two order books. A buying order book and a selling order book.

The selling book contains a list of sellers, the type and amount of tokens they want to sell, and the price in the currency that they want to sell these tokens for.
The Sellers of these tokens, deposit the tokens into the smart contract waiting for a buyer at the defined rate. The Buyer can view a list of all the sellers and then pay for the tokens of choice at the preferred rate. Once the buyer pays the seller of choice the smart contract transfers the tokens to be purchased automatically to the buyer and pays the seller. The seller can cancel or update the order while no buyer is interacting with it.

The Buying Order book does the same but the opposite thing. It contains a list of people ready to buy a token and the rate at which they want to buy. These Buyers have deposited their funds to the smart contract. The seller can view a list of all the buyers and sellers to anyone of choice.

Task

- Create a currency of your choice, NGN, USD, EUR, INR e.t.c on the blockchain
- Create another token. Let sellers or buyers populate other books to sell or buy the other token at a rate of the currency created
- Provide the smart contract function that allows anyone to get the order books and buy or sell from/to the order book

5. CRUD To-do list

Literally, everyone in this current age uses a To-do list.

The todo list helps us in a very simple way to itemize things we will like to do or buy e.t.c
Create a smart contract that allows a simple user to enjoy the experience of a simple to-do list app. The user can create a new list, add to the list, remove from the list, view the list, update the list, and even make the list private from other persons.

6. Fixed Ratio Token Swap.

P2P exchanges are not so scalable, because you would require someone else who wants the token you have and the token you want and is willing to exchange at a favorable rate. Due to this trilemma, most DEFI projects use the concept of an AMM (Automatic Market Maker) instead of P2P. Typically what happens is that there is a smart contract called the pool smart contract or pair smart contract that holds the two tokens to be exchanged. Some people known as investors are the ones who deposit these two tokens into the smart contract and they receive a fee anytime someone uses the contract to swap their tokens. If anyone has one of the tokens and he wants the other token, he can deposit the one he has into the smart contract and the smart contract will give him some amount of the other token. The amount of the other token he will receive is usually determined by some formula. E.g for Uniswap it is
$X * Y = K$ where X and Y are the amounts of each of the tokens available and K is a constant.

As a smart contract engineer, you will like to create your DEFI exchange using the formula $X = KY$ where X and Y are the amounts of the two tokens and K is a simple fixed number e.g 1 or 2

For example if K = 2, it means X = 2Y. I.e for every single deposit of one token, I get 2 of the other tokens and 2 in our case is always constant.

Task

- Create two tokens.
- Create the pair contract that allows investors to deposit and users to swap these tokens
- Allow anyone to be an investor and deposit these two tokens (following the agreed ratio)
- Allow anyone to swap a token for another at the agreed ratio (no fee is paid to investors in our case)

7. Election Smart contract:

You know this very well, no need to say much about it. 😄

Task

- Allow People to nominate 5 candidates
- Allow people to vote for these candidates
- Present vote results
- Follow all the voting rules (e.g can not submit a candidate when the nomination has ended, can not vote when voting has ended, there is a timeframe for submitting candidate and voting and many others).

- Be creative, and add interesting features.

8. ATM

- Create a smart contract, in which anyone can register and get a card (could be represented by a unique number) and a secret pin

- Once a person has a card, he can slot in the card and pin and can view his balance, withdraw, deposit or exit the machine.

- Try to make the experience similar to one witnessed at a standard ATM

- Be creative

9. Betting Smart Contract

This smart contract would allow two people to bet on the outcome of an event. To validate their BET they would each deposit some tokens and cannot withdraw these tokens. When the BET result is out, the winner gets all the tokens deposited.

- Create this token
- Create this contract
- Be creative

10. PiggyVest

Create a smart contract that allows anyone to create a piggy bank, save for some time, and get their funds at the end of the period. During the saving period, the funds can not be accessed. The smart contract should be reusable, meaning anyone can use this smart contract to create their piggy bank. Use chain link Automation to automate payout when saving time elapses. Read about chain link automation here: https://docs.chain.link/chainlink-automation/introduction