Decentralized Mutual Funds

A mutual fund is a company that pools money from many investors and invests the money in securities such as stocks, bonds, and short-term debt. The combined holdings of the mutual fund are known as its portfolio. Investors buy shares in mutual funds. Each share represents an investor’s part ownership in the fund and the income it generates.

In this project, you will be implementing a mutual fund company on the Ethereum blockchain. Below are detailed requirements of the project.

1. Any Ethereum account user should be able to register as a Portfolio manager.
2. The user should deposit some Ether to register as a portfolio manager and start the fund.
3. When the fund is started, generate X fund tokens that will be sold in the market
4. The manager should be able to purchase individual tokens to build her portfolio
5. The frontend should show the breakdown of tokens and remaining Ether in each manager’s portfolio
6. The investor should be able to deposit Ether and receive fund tokens in return. The fund tokens should be minted and sent to the investor
7. The manager should be able to issue dividends to the investors at any time

Questions: How do you determine the value of each token in the fund?

Abhinav: Value of each Token is determined as per APIs given from cryptocompare.

Additional Notes:

1. Use https://github.com/WandXDapp/PortfolioProtocol

Pending questions

1. How to show value of token in etherscan etc. Where to store it.
2. Are there any mock currency exchanges?
3. How to determine fund performance?

**Project Phases**

**A phase is considered complete only if I can checkout your code in github from master branch and am able to test the functionality listed in that phase.**

**If something in the description below is not clear, make an assumption and implement it without waiting till next day for clarification.**

**Phase 1**

1. I should be able to npm install, truffle migrate and execute npm run dev and see your home page with your project name
2. When your webpage loads, if web3 object is not found, show an alert saying install Metmask or use web3js supported browser

**Phase 2**

1. We will only have one portfolio manager in our project to start with
2. Create an ERC20 token which will be used as fund token by portfolio manager. Set the price of the fund token to 0.001 Ether.
3. Create a smart contract which will issue the fund tokens to the portfolio manager when they stake the deposit into this smart contract
4. Implement the frontend to display the Ether balance in the contract and the total tokens available to portfolio manager

**Phase 3**

1. Implement the functionality in the smart contract for any user to buy these fund tokens from the portfolio manager
2. Implement the frontend where user enters the amount of tokens they want to buy and hit purchase
3. The fund tokens should be transferred to the user’s wallet

**Phase 4**

1. Implement the functionality to display all the investors and how many tokens they own
2. Display these details in the web page

**Phase 5**

1. Implement the functionality to issue dividends to the investors. When dividend is issued, new fund tokens should be minted and sent to all the investors.
2. To start with, implement the functionality to issue 100 tokens as dividend. It should be proportionally distributed to all investors based on how many fund tokens they currently own
3. Implement the frontend functionality where portfolio manager can click a button called “Issue Dividends” which will trigger the smart contract function

*\* Don’t worry about how this 100 tokens came etc for now*

**Phase 6**

1. Implement the functionality for user to sell their fund tokens
2. Implement the frontend where user can enter the tokens they want to sell and press the “Sell” button

**Phase 7**

1. Deploy 20 different ERC20 tokens with 100,000 supply and different prices
2. In the frontend, display all 20 tokens to the portfolio manager

**Phase 8**

1. Implement the functionality for fund manager to purchase these tokens and build his portfolio
2. Implement the frontend functionality for manager to purchase these tokens
3. Display all the tokens in fund manager’s portfolio