More DApps

Note:

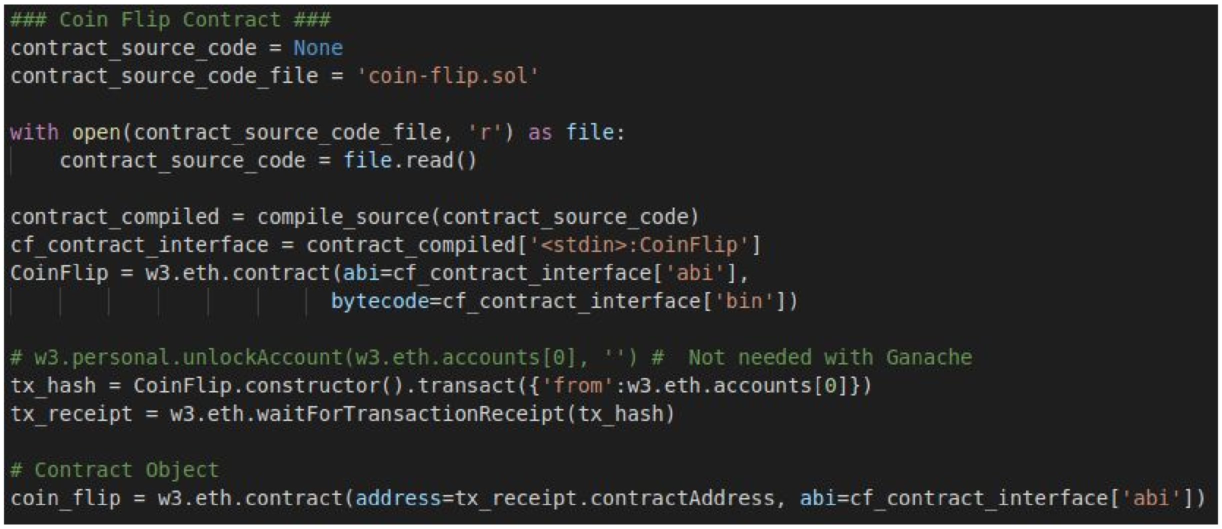
- Source code to the Flip Coin DApp and corresponding crowdsale can be found on https://github.com/vierivincent15/coin-flip-dapp
- All related contracts are compiled and deployed to Ganache (localhost:8545)
    - Meaning that you will need to have Ganache before running the DApp
- Metamask accounts that are used to interact with the DApp and Crowdsale are also retrieved from Ganache.
- To run the source code, execute the following commands:

```
> pip3 install py-solc flask web3
> python3 -m solc.install v0.4.25
> export PATH=$HOME/.py-solc/solc-v0.4.25/bin:$PATH
> python3 server.py
```

Question 1

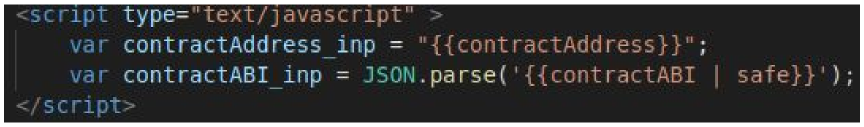
Using Python, Web3py, Web3js, Metamask, and Flask, develop a (DApp) web application able to manage the coin tossing service created previously. It should:

- **Compile and deploy smart contract(s) automatically**
  Upon executing *server.py* the CoinFlip smart contract will be automatically compiled and deployed to the network (in this case Ganache). The required information to interface with the contracts, contractAddress and contractABI, are then passed over to the front end. The code shown below is expected to compile and deploy the smart contract upon execution.

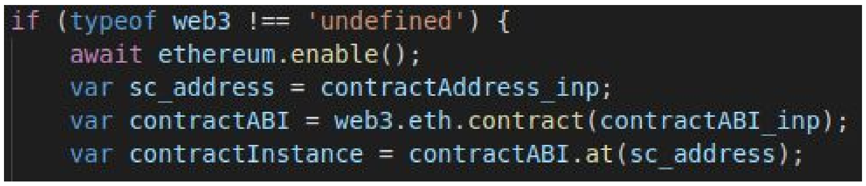


```python
### Coin Flip Contract ###
contract_source_code = None
contract_source_code_file = 'coin-flip.sol'

with open(contract_source_code_file, 'r') as file:
    contract_source_code = file.read()

contract_compiled = compile_source(contract_source_code)
cf_contract_interface = contract_compiled['<stdin>:CoinFlip']
CoinFlip = w3.eth.contract(abi=cf_contract_interface['abi'],
                            bytecode=cf_contract_interface['bin'])

# w3.personal.unlockAccount(w3.eth.accounts[0], '') #  Not needed with Ganache
tx_hash = CoinFlip.constructor().transact({'from':w3.eth.accounts[0]})
tx_receipt = w3.eth.waitForTransactionReceipt(tx_hash)

# Contract Object
coin_flip = w3.eth.contract(address=tx_receipt.contractAddress, abi=cf_contract_interface['abi'])
```

- **Interact with the contract when required; for example, get winner status or trigger an arbitrage**
  Interaction with the contract is enabled from the front-end by obtaining relevant parameters such as contractAddress and contractABI. The code shown below allows the front-end to establish interaction with the contract that has been deployed to the network.



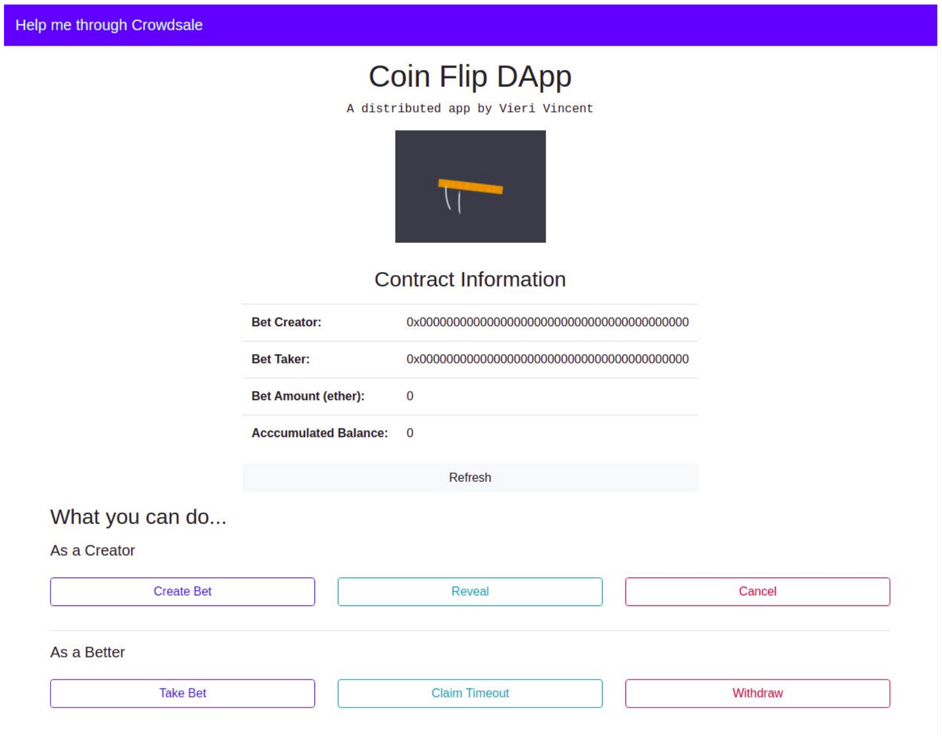
```
<script type="text/javascript" >
    var contractAddress_inp = "{{contractAddress}}";
    var contractABI_inp = JSON.parse('{{contractABI | safe}}');
</script>
```



```
if (typeof web3 !== 'undefined') {
    await ethereum.enable();
    var sc_address = contractAddress_inp;
    var contractABI = web3.eth.contract(contractABI_inp);
    var contractInstance = contractABI.at(sc_address);
```

  Interactions to the contract can be made automatically or through user interaction. For my particular case of the DApp, it will first interact with the contract to obtain relevant information and then wait for user input for any further interactions.

- **Trigger interactions manually from a web GUI**
  Through the GUI, the user will be able to interact with the contract through 6 different actions as shown.



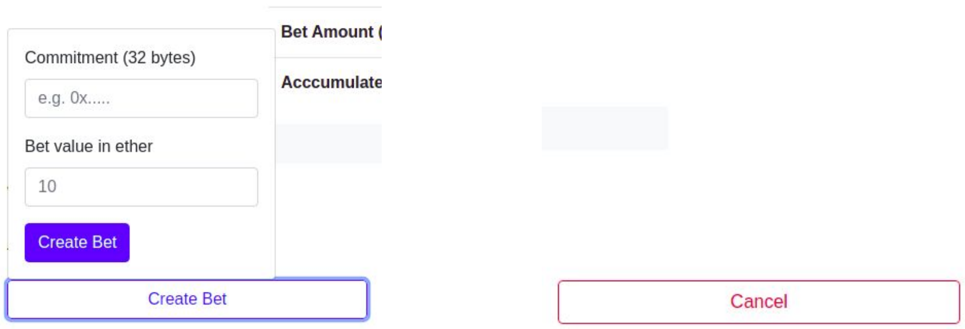
Coin Flip DApp

A distributed app by Vieri Vincent

Contract Information

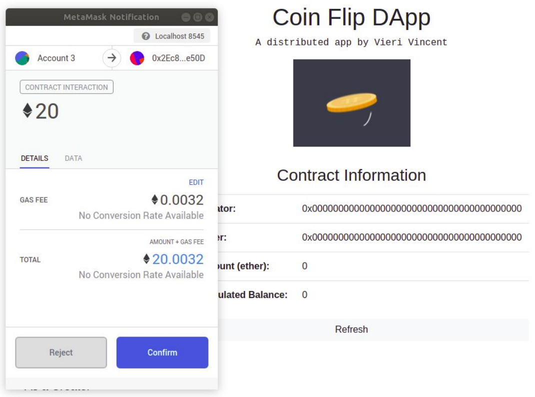
Bet Creator:	0x0000000000000000000000000000000000000000
Bet Taker:	0x0000000000000000000000000000000000000000
Bet Amount (ether):	0
Acccumulated Balance:	0

The user can either be a Creator and perform ‘Create Bet’, ‘Reveal’, ‘Cancel’ or be a Better and perform ‘Take Bet’, ‘Claim Timeout’. ‘Withdraw’. Some actions require user-generated input while other actions will automatically trigger the contract interaction.


Bet Amount (
Accumulate
Commitment (32 bytes)
e.g. 0x.....
Bet value in ether
10
Create Bet

- **Users can interact through the web app using Metamask**
  Users are able to interact with the web app using Metamask as the Metamask extension injects web3 to the browser. However, they will need to give access to the application before their wallet on Metamask can interact with the web page. Shown below on the left is an image when a user is about to ‘Create Bet’ of amount 20 ether with his pre-computed commitment.

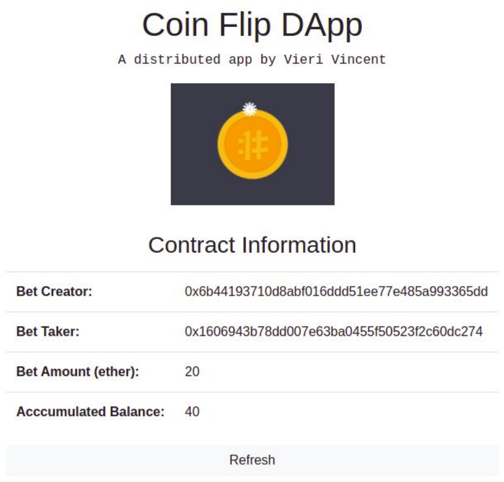
  The web app also supports interaction with multiple users which therefore allow another user to ‘Take Bet’ with the same amount. Shown below on the right is an image of a complete betting procedure where a Creator has created a bet and a Better has taken the bet.


MetaMask Notification
Localhost 8545
Account 3 → 0x2Ec8...e50D
CONTRACT INTERACTION
20

GAS FEE	0.0032 No Conversion Rate Available
TOTAL	AMOUNT + GAS FEE 20.0032 No Conversion Rate Available

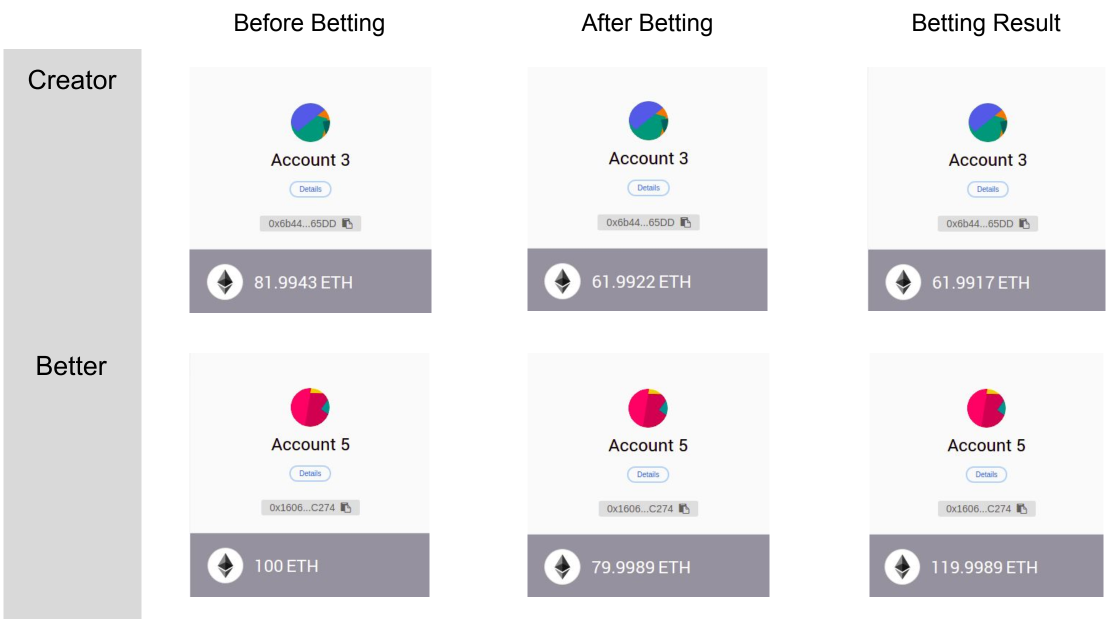
Coin Flip DApp
A distributed app by Vieri Vincent
Contract Information

ator:	0x0000000000000000000000000000000000000000
er:	0x0000000000000000000000000000000000000000
unt (ether):	0
ulated Balance:	0


Coin Flip DApp
A distributed app by Vieri Vincent
Contract Information

Bet Creator:	0x6b44193710d8abf016ddd51ee77e485a993365dd
Bet Taker:	0x1606943b78dd007e63ba0455f50523f2c60dc274
Bet Amount (ether):	20
Acccumulated Balance:	40

Once the result is revealed by the creator, betted ether will automatically be transferred to the winner.



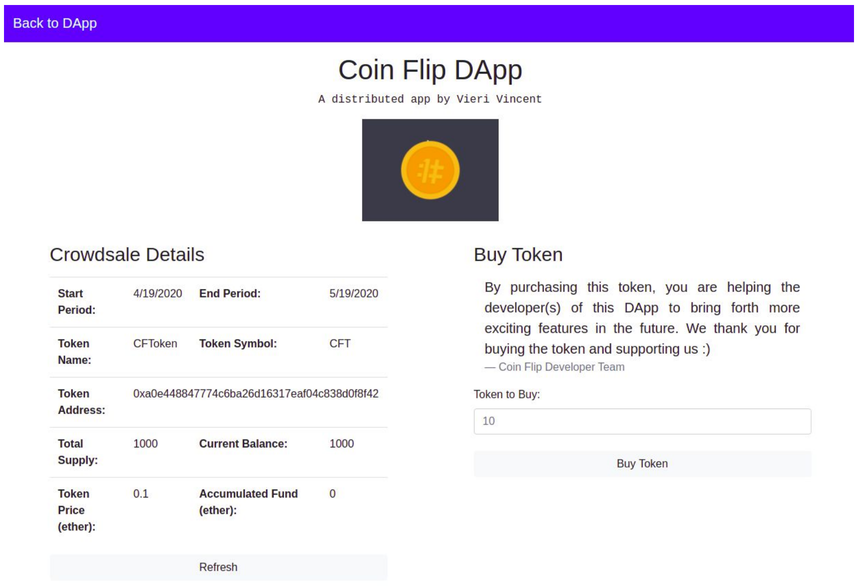
	Before Betting	After Betting	Betting Result
Creator	Account 3 Details 0x6b44...65DD 81.9943 ETH	Account 3 Details 0x6b44...65DD 61.9922 ETH	Account 3 Details 0x6b44...65DD 61.9917 ETH
Better	Account 5 Details 0x1606...C274 100 ETH	Account 5 Details 0x1606...C274 79.9989 ETH	Account 5 Details 0x1606...C274 119.9989 ETH

In the case above, the Better won the game as the result of his flip is the same as the commitment that the Creator has committed at contract creation.

Question 2

Let's launch a crowdsale for your coin tossing service. To do so, you make use of an Initial Coin Offering (ICO), implemented via the ERC20 token standard.

The crowdsale is accessible at the '/crowdsale' route of the DApp. It make use of an ICO, implemented via a ERC20 standard token that is named CFToken (CFT). The GUI of the crowdsale feature is as shown.



Back to DApp

Coin Flip DApp

A distributed app by Vieri Vincent

Crowdsale Details

Start Period:	4/19/2020	**End Period:**	5/19/2020
Token Name:	CFToken	**Token Symbol:**	CFT
Token Address:	0xa0e448847774c6ba26d16317eaf04c838d0f8f42		
Total Supply:	1000	**Current Balance:**	1000
Token Price (ether):	0.1	**Accumulated Fund (ether):**	0

Refresh

Buy Token

By purchasing this token, you are helping the developer(s) of this DApp to bring forth more exciting features in the future. We thank you for buying the token and supporting us :)

— Coin Flip Developer Team

Token to Buy:

10

Buy Token

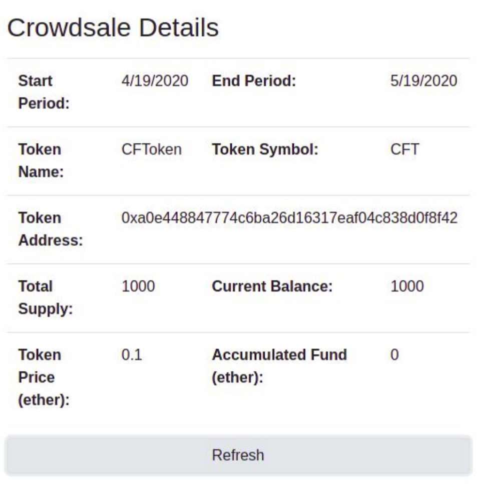
This feature is enabled by compiling and deploying the Crowdsale contract that is implemented under *crowdsale.sol* that requires *token.sol* and *ERC20.sol* (interface). As shown in the image above, there is a total of 1000 CFToken that is priced at 0.1 ether each. Any user can participate in the crowdsale by purchasing any amount of tokens and the corresponding ether will be stored in the Crowdsale contract. This interaction is supported through the GUI through the 'Buy Token' button. The stored ether will then be released only when the ICO period has ended or all the token supplies have been bought. Shown below are different states of the ICO and a participating user's wallet before and after buying the token.

Before Buying Token

After Buying Token

ICO Details

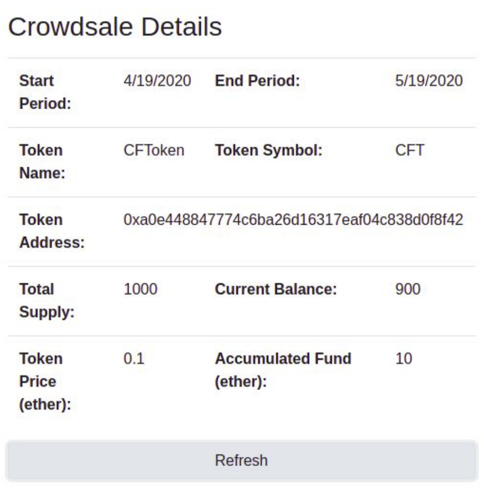
Before Buying Token:



Crowdsale Details

Start Period:	4/19/2020	**End Period:**	5/19/2020
Token Name:	CFToken	**Token Symbol:**	CFT
Token Address:	0xa0e448847774c6ba26d16317eaf04c838d0f8f42		
Total Supply:	1000	**Current Balance:**	1000
Token Price (ether):	0.1	**Accumulated Fund (ether):**	0

After Buying Token:

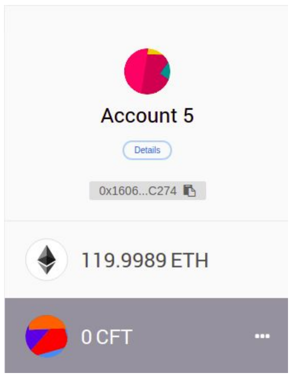


Crowdsale Details

Start Period:	4/19/2020	**End Period:**	5/19/2020
Token Name:	CFToken	**Token Symbol:**	CFT
Token Address:	0xa0e448847774c6ba26d16317eaf04c838d0f8f42		
Total Supply:	1000	**Current Balance:**	900
Token Price (ether):	0.1	**Accumulated Fund (ether):**	10

Participating User

Before Buying Token:



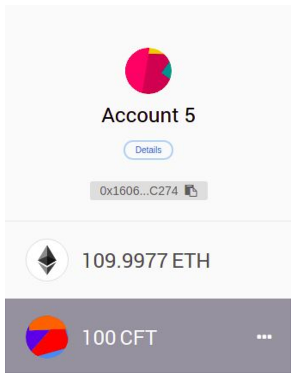
Account 5

0x1606...C274

119.9989 ETH

0 CFT

After Buying Token:



Account 5

0x1606...C274

109.9977 ETH

100 CFT