CS765 Assignment 3

This assignment can be done in groups of size at most 3.

A Decentralized App (DApp) is an application running on a permissionless blockchain. A DApp is usually implemented as a smart contract. A smart contract is essentially a program whose code is on the blockchain. The code is initially put on the blockchain in a transaction. The smart contract can have many functions. Different functions can be invoked by other transactions later on, provided the person invoking the function(s) has the permissions to do so as specified by the smart contract. When different functions are executed, the state of the smart contract are modified.

In this assignment, you must design a DApp to address the problem of Fake News. Nowadays many news items circulate on social media and elsewhere which are untrue. These are called Fake News. Centralized fact-checkers may have biases and may not be very trustworthy. A DApp is an alternative way of flagging fake news in a decentralized manner.

Your DApp should have the following features.

1. Anyone is allowed to request the DApp for fact-checking a news article or item.
2. Anyone is allowed to register on the DApp as a fact-checker.
3. The fact-checkers can vote to say whether the news item is fake or not. The vote could be binary (0 or 1) or it could be a number over a range, say 1- 10, to indicate how truthful the news is (a higher number could imply that the voter thinks the news is more truthful).
4. The DApp considers all votes and outputs a single number indicating the fakeness or truthfulness of the news.

Some issues you should consider are:

(1) Sybil attack: A malicious person can create multiple identities and vote to skew the result in any direction
(2) Method to evaluate or re-evaluate the trustworthiness of voters: The Dapp should evaluate how trustworthy different voters are based on how they vote. Note that someone might game the system to get a higher trustworthy rating. A method that is more robust to such gaming of the system, is preferable.
(3) The opinions of more trustworthy voters should be given more weight. However, we must keep in mind that someone may be more trustworthy for certain types of news and not others. For example, someone may give excellent opinions about news related to Physics but is not so trustworthy on topics related to Politics or Economics.
(4) Rational voters are to be incentivised to participate and vote truthfully to the best of their ability.
(5) Uploading a news item: Some efficient method should be used to identify a news item (which is to be evaluated) in the Dapp.
(6) Bootstrapping: If the Dapp does not have any trustworthy rating of different initial voters, then how to get started with fact-checking news?

A. (3+6+2+1+3=15 marks) Describe how your Dapp will handle the above issues. Explain in your own words. You can use diagrams, figures, equations etc. to explain. You may also want to ponder about what is asked below before framing your answer.
B. (15 marks) Suppose your Dapp uses a smart contract on a blockchain. Write the pseudo-code of each function of the smart contract using syntax of the Solidity language. You should say what each function accomplishes and if there are any restrictions on who can invoke the function. (See the Subcurrency example in the following link to see Solidity examples of how coins can be minted and transferred, and how some functions are restricted to be used by

only the creator of the smart-contract: https://docs.soliditylang.org/en/latest/introduction-to-smart-contracts.html)

C. (10 marks) Do simulations to study how well your algorithm to evaluate the trustworthiness of voters works. Consider that every news item is either fake (0) or true (1) and that voters need to cast a vote to say if each news item is true or fake. They hence cast a 0 or 1 vote. Consider the case where you have a total of “N” voters, out of which “q” fraction are malicious. Of the honest voters, suppose “p” fraction are very trustworthy and give the correct vote with probability 0.9. The rest of the “1-p” fraction gives the correct answer only with probability 0.7. Malicious users deliberately choose the exact wrong answer. Study whether you algorithm can estimate the trustworthiness of different users well. For example, the trustworthiness of the “p” fraction of honest users should ideally be 0.9, the “1-p” fraction of honest users should be 0.7, and the trustworthiness of the malicious users should be 0. Study the performance of your algorithm for different values of N, p, and q. If your algorithm updates estimates of trustworthiness over time, then evaluate how this estimate varies over time.
You may also try out alternative malicious strategies.

In your submission, upload a single zip file (filename format: RollNo1 RollNo2 RollNo3.zip) containing (a) a report answering A and summarizing results of C, (b) a file containing the pseudo-code in Solidity syntax (part B) which should be well commented, (c) code for your simulations, (d) README file on how to run the simulation code.