Blockchain based online contracting system

Contracts are the basis for a successful project and require a multi-stakeholder system for satisfactory completion. They contain the essential information and binding agreements between the issuer of the project and the party that implements the project. Your engineering team has been tasked with creating a blockchain based contract issuing and management system. Each contact must have 4 sections, the details of the issuers, list of bidders for the project along with their estimated quote amount (tokens/funds needed to complete the project), brief details of the project along with estimated time required to complete a project. The ledger must contain the information on which project is issued by which party, being implemented (given) by which party along with other project related details including project final price and the agreed timeline.

1 PROBLEM STATEMENT

Create a multi-party blockchain system for recording, issuing and updating contracts for "M" projects amongst "N" parties, each party start with a fixed number of tokens at start and gain/lose tokens as they get more projects or fail to deliver the same on time (duration "t" years). The blockchain is permissioned so the party needs to register and gain valid authority to access the blockchain and issue/create bids on a project. The system also consists of "T" validators who maintain trust and vigilance in the system. A project is approved only if "k" or more validators accept it and contract can only be given to a bidder with the lowest quotation price as submitted to this system, incase of a tie, bidder with the largest "trust" is given the project. Trust in this context is measured in-terms of number of previous projects done and the token assets a party has. A valid project is one that must be **approved** by validators as mentioned above and issued by a party that holds with itself an asset of atleast half of the minimum bid amount of tokens by the highest bidder. The results of a bid are released "y" hours after the bidding time is over.

2 DELIVERABLES

A github repository containing the code for running and maintaining the contracting system along with a project documentation report that list the details on how to use the platform and the features it has. Your team is welcome to make changes to the scheme mentioned above or adapt your own policy for the contracting platform, mention the same in your project documentation report explaining the changes.

3 OUTCOMES

Each operation such as token transfer for project payment, account creation, account deletion, project issuance, bidding for a project, accepting a project offer, updating the timeline for the project etc must be treated as a transaction and audit trail for the same must be recorded in the permissioned blockchain by timestamp.

- Create or delete a party in the platform.
- Update party details on the platform. (User has the option to make the profile anonymous or public)
- Create or update a contract details and timeline for a project. (A news article can never be deleted.)
- Issue rewards in the form of tokens to parties and validators.
- Projects are shown in the platform only when atleast T/2 validators validate a project.

Author's address:

- Rewards are issued to validators and parties involved based on whether the project outcomes are satisfactory and progresses on time. Tokens a dispatched in phases with each milestone the project reaches. You can use your own metric/heuristic to reward the stakeholders involved.

4 MINIMUM DELIVERABLES (60 MARKS)

- Create a new party on the platform.
- A party must be able to create a project on the platform.
- A party must be able to create a bid for a project.
- Select a party based on bids recieved.
- Project report mentioning the features and how to use them.
- Successfully demonstrate the use of blockchain based technology in the platform.

5 QUALITY ATTRIBUTES

- System must atleast scale to a 100 users/parties.
- Word limit for the project description is atmost 200.
- "T" can be atmost 10 at any point of time and a shuffled after some period of time to other users of the forum who meet the "validators" criteria.
- "k" can be atmost 6.
- The minimum bid price can't exceed an amount "x" (Use your discretion to set the value of "x").
- There can be atmost 15 bids for a project. (15 parties can share the quote price, timeline details etc for a particular project)
- Bidding time can be atmost for 24 hours and "y" can be for atmost 6 hours.
- The total project price can't exceed "y" amount of tokens (Use your discretion to set the value of "y").
- A party can be banned or barred from issuing/taking a project if the last reward recieved is less than "z" amount. ($0 \leq z \leq 5$) or last 2 projects have "unsatisfactory" result at the end of the project.

6 GRADING POLICY

- **80 marks** are awarded if the contracting system works without glitch and meets the quality attributes mentioned above.
- **10 marks** are awarded for properly documenting and presenting the features of the system.
- **10 marks** are awarded for showing the contracting system working for 100+ users with 20+ projects.
- **10 marks** are awarded as **bonus** if projects can be filtered by milestones and bids can be filtered by quote price. Remember that bidding parties can't see each other's quote amounts for a particular project. Any other "extra" or "useful" feature can also be worthy of bonus points.
- **10 marks** are awarded as **bonus** for new "useful" or "interesting" features that utilize the underlying blockchain or is privacy-preserving (Bidders must not be able to gain insights from the system in-order to guess a "good" quote price).

7 DISCLAIMER

Before doing anything "extra" (which might fetch bonus marks), first, complete the basic expectations from your implementation.

Software tools are expected to display their results in a user-friendly manner; a user would never like to use a tool that simply spits out a bunch of numbers. So, display the results from your tool suitably possibly in a good web-based UI or the terminal in verbose user-friendly manner.

Discussion is healthy, copying is not. You are encouraged to discuss the projects with your peers, but you must implement the projects by yourself. If any two groups are found with "similar" pieces of code, both of them will be failed (with no concern as to who was the source). Copying from internet sources or open-source github repositories must be refrained from.

TAs may conduct a code-review after every milestone is reached or 15 days (which ever is earlier) so please be careful about plagiarism.