BUET CSE FEST 2023

***HACKATHON***~ WEB3 & BlOCKCHAIN

     PROBLEM STATEMENT ~

13th July 2023

Location: Department of Computer Science and Engineering, BUET

Supply Chain Application for **authentic artworks & premium collectibles**

**Scenario**

Consider a web platform for selling, auctioning, and buying authentic artworks from artists & premium collectibles in physical form from creators around the globe. This web platform is unique in some ways, such as the artists/creators requiring to register themselves on the platform while the buyers can remain anonymous and still purchase the items. This web platform will also ensure that the artworks/collectibles sold are completely authentic, which is verified by third-party art auditors. The buyer can also view and verify the certificate of authenticity without revealing their identity.

The task is to implement this web platform on web3 rails to enforce anonymity, credentials verification, and authenticity through blockchain and its tools.

**Phase-1**

Assume your team has already built the web dashboard and other components of the platform. Your task is to implement the specific components for implementing 3 stakeholders: Supplier, Buyer, and Verifier. The artists & creators are the suppliers, and the art collectors can be regarded as the buyers and the art auditing firms/companies can be assumed as the verifiers.

It will suffice that all of these 3 stakeholders will interact with your web application with 3 individual wallet addresses (i.e. using Metamask wallet). Each of these stakeholders should be able to perform a set of functionalities:

Supplier:

1. Register their account using their wallet
2. Add an artwork/collectible and upload its details, such as - description, art/collectible image, price, artist/creator credentials, etc.
3. Update artwork/collectible’s quantity/supply
4. Initiate the delivery process & update its delivery/supply status
5. Offer limited-edition premium artworks in an online auction

Buyer:

1. Browse, search & filter available artworks & collectibles from all the artists/creators
2. Verify an artwork’s authenticity certificate
3. Place orders for available artworks/collectibles & confirm the order by making a payment via the platform’s native token
4. View an order’s delivery/supply status
5. Place a bid in the auction to purchase a limited-edition premium artwork

Verifier:

1. Register their account using their wallet
2. [Optional] Complete a short demo KYC process (i.e. upload their auditor id and license, etc.)
3. Issue authenticity/quality certificate against each artwork/collectible in the form of a token. Make sure to handle potential corner cases in this certificate issuance and verification system, e.g. artists shouldn’t be able to transfer their artwork’s unique certificate to other users.

**Phase-2:**

Once you’re done with writing and deploying the smart contract and a very minimal/basic front-end application on top of it to interact with its functions, you should focus on other things that will streamline this supply chain system. Two interesting features you may try to implement are:

- Fee Split: Instead of one-time payment with a platform native token, the buyer can split the payment into multiple stages. For example, the buyer has to pay 10% of the price while confirming the purchase/order and pay the remaining 90% after they confirm receiving the artwork’s delivery.

- Royalty: If an artwork is sold and the buyer then uploads the same artwork on the platform for another sale or lists it for an auction, the platform should be able to detect it through its unique authenticity verification certificate. In that case, the original creator can be paid 2% of all the sale profits from the 2nd sale & onwards.

Feel free to come up with and implement additional features which can further progress your platform e.g. ways to incorporate real-world data to make the system more useful.

Participants will receive additional points for:

* Adding useful and/or creative features to improve your system
* Automating the deployment and/or testing process (e.g. using Hardhat)
* Following engineering best practices (i.e. commit, pull request, programming language best practices) for building the system
* Gas Optimization on smart contract Level
* Security concerns addressed on smart contract
* Incorporating other useful tools and technologies from the Web3 eco-system that they think might be useful

General Instructions

These are some general instructions for participants of any segment:

* Participants will need to start a new GitHub repository onsite and submit it at the end of the competition.
* The project must be developed on the day of the Hackathon. Any project which was developed (even partially) in advance will not be accepted.
* If participants choose to use publicly available code, then they must properly attribute the source.
* Unless otherwise specified, participants can choose their own tech stack.

*Any unfair attempts, misbehaviour or breach of terms will result in disqualification. The organizers reserve the right to make final decisions. Participants will be notified if there is a notable modification in existing rules.*