NAME – Tanuj Gupta

**COHORT - PCP Blockchain May 2022 Cohort 1**

**E- KYC PROJECT**

As we all know KYC stands for know you customer which is very similar to a social security number in the USA . KYC is required for financial institutions **to establish the legitimacy of a customer's identity and identify risk factors**. KYC procedures help prevent identity theft, money laundering, financial fraud, terrorism financing, and other financial crimes.

Previously a large number of banks were authorized by the central bank to perform KYC for the customers. After detecting suspicious activities some banks were banned from performing KYC. Despite the ban, some banks kept on adding new customers and performing KYC for them.

Performing KYC through BLOCKCHAIN TECHNOLOGY is the perfect solution for this problem. The blockchain network is immutable and untamperable . Moreover, every action made by every bank will get recorded in the blockchain , therefore making it very easy to find the malicious nodes.

I have created a e-KYC blockchain network for the above problems

I have used the following technologies ;-

1. Solidity for coding the smart contract
2. IDE used – REMIX IDE
3. Blockchain network used – Ethereum
4. Server – Ganache (It can also be deployed on Ropsten or Mainnet if required)

Smart Contract specifics

1. Firstly I have used struct to store groups of different types like string, address, unit256, bool
2. Then I used mapping to form key value pairs ( so as to I can refer to multiple banks and bank customers)
3. There are total nine functions in my smart contract ;-
4. Add new bank
5. Allow bank to do kyc
6. Block bank to do kyc
7. Block bank to add customer
8. Allow bank to add customer
9. Add new customer
10. Add request for new customer kyc
11. Check kyc status of existing customers
12. View customer data
13. Only the bank address that deploys the contract gets full access over all the functions( i.e in our problem case should be the Central Bank).
14. Functions a, b, c, d, e are restricted to admin only
15. Used ‘require’ for error handling wherever necessary
16. Used msg.sender=admin to get the restricted access

The ekyc blockchain is fully functioning and scalable with optimum transaction time