Blockchain

Final Project

Time Lock Smart Contract

Group Work

**Group: CS-2111**

**Sherkhan Zhunusbay**

**Asanali Tazhigaliev**

**Nurbek Kydyraliyev**

**Introduction**

Blockchain technology, with its decentralized and secure architecture, has revolutionized various industries. Smart contracts, a key feature of blockchain, automate and enforce contractual agreements without intermediaries. However, ensuring the security and responsible governance of smart contracts remains a critical challenge. This report explores the implementation and impact of Time Lock Smart Contracts as a solution to enhance security and governance in blockchain systems. It operates on the principles of decentralization, transparency, immutability, and cryptographic security

How Blockchain Works:

Transaction Initiation: Participants initiate transactions by creating a digital signature using their private keys.

Transaction Verification: The network's nodes validate and verify the transaction's authenticity using cryptographic techniques.

Transaction Broadcast: Validated transactions are broadcasted to the entire network.

Consensus: Consensus mechanisms ensure that the majority of nodes agree on the validity of the transaction and its position in the ledger.

Blockchain's decentralized and transparent nature makes it a powerful tool for enhancing trust and security in a wide range of applications.

Smart contracts, self-executing contracts with coded terms, have emerged as a pivotal feature of blockchain ecosystems. They automate and enforce the execution of contractual agreements without the need for intermediaries, providing transparency and efficiency.

**Literature Review**

A time-locked smart contract is a type of smart contract designed to provide a delay before certain conditions can be met or actions taken. These contracts are commonly used in blockchain and cryptocurrency environments to introduce a time delay for security or management purposes.

A brief description of how a smart contract with a temporary lock usually works:

1)Setting conditions

2)Time delay

3)Blocked funds or actions

4)The unlocking mechanism.

The implementation of time-locked smart contracts may vary depending on the blockchain platform and the specific programming language used. Ethereum, for example, typically uses Solidity for smart contracts, and developers can enable time-based functionality by using functions such as the block.timestamp variable to access the current timestamp in the Ethereum blockchain.

Time Lock Smart Contracts introduce a temporal dimension, enhancing security and governance. They prevent hasty decisions and provide users a grace period to respond to proposed changes. Applications include token vesting and governance structures.

Time Lock Smart Contracts find applications in token vesting and governance. In ICOs and employee compensation, gradual token release aligns incentives. In governance, stakeholders have time to review and potentially veto changes.

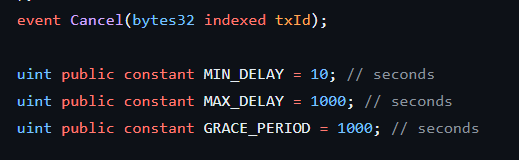
**Results**

First of all we started from smart contract on solidity

here is the declaration of user errors and events that can be triggered during the execution of the contract



time constants for determining delays and warranty period

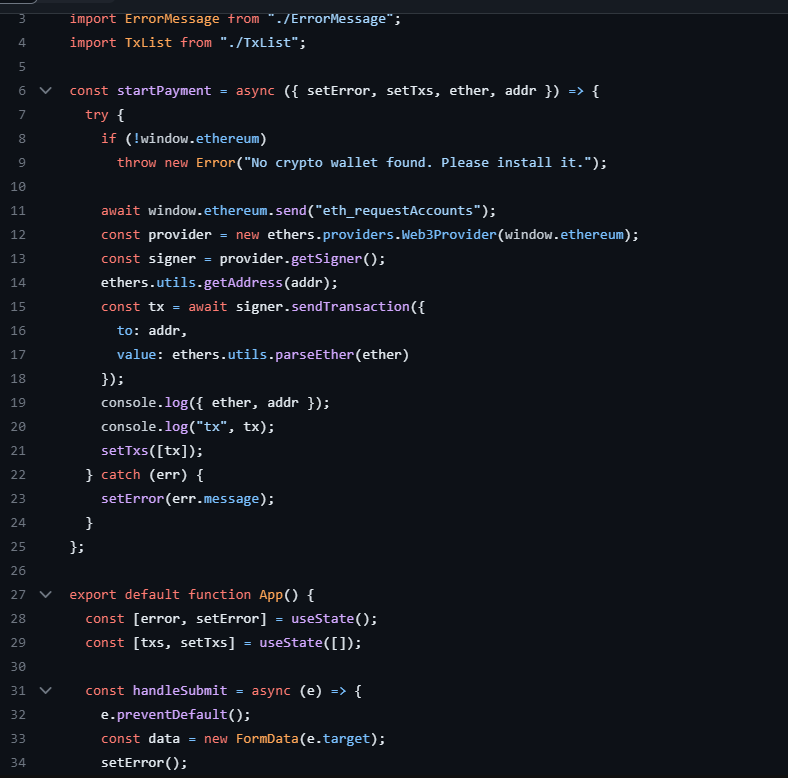
Изображение выглядит как текст, снимок экрана, Шрифт

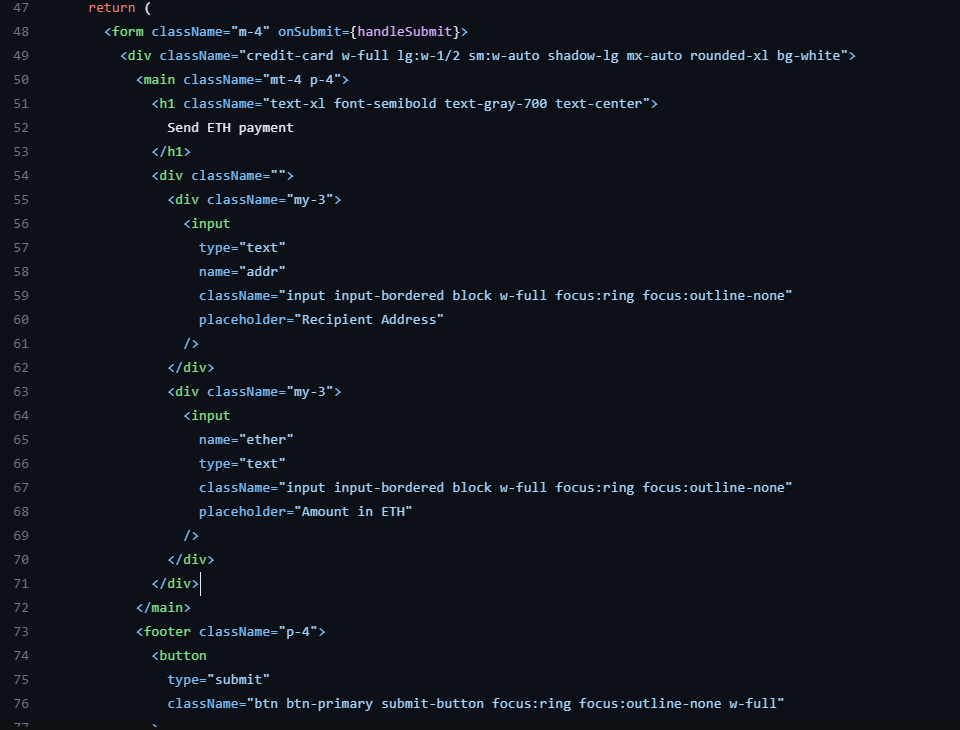
Автоматически созданное описание

Изображение выглядит как текст, снимок экрана

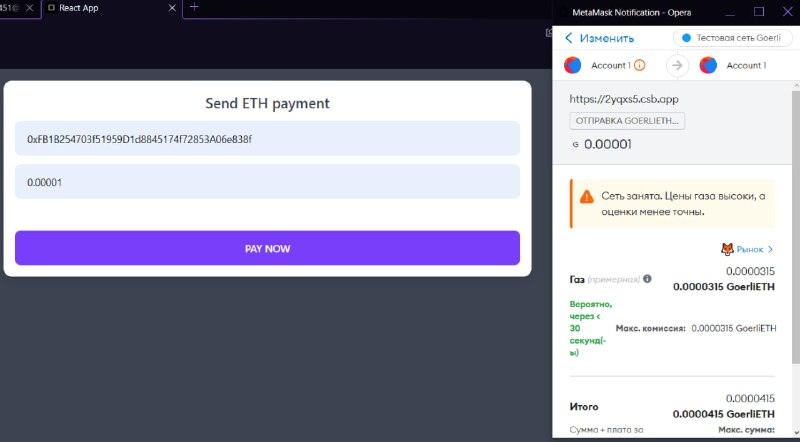
Автоматически созданное описание

Here is the react code that needed to correct work of site and time lock





Here is our site that has a time lock function for transactions

****