Aayush Shah
19BCE245
18 September 2022

BlockChain Technology

Practical 9

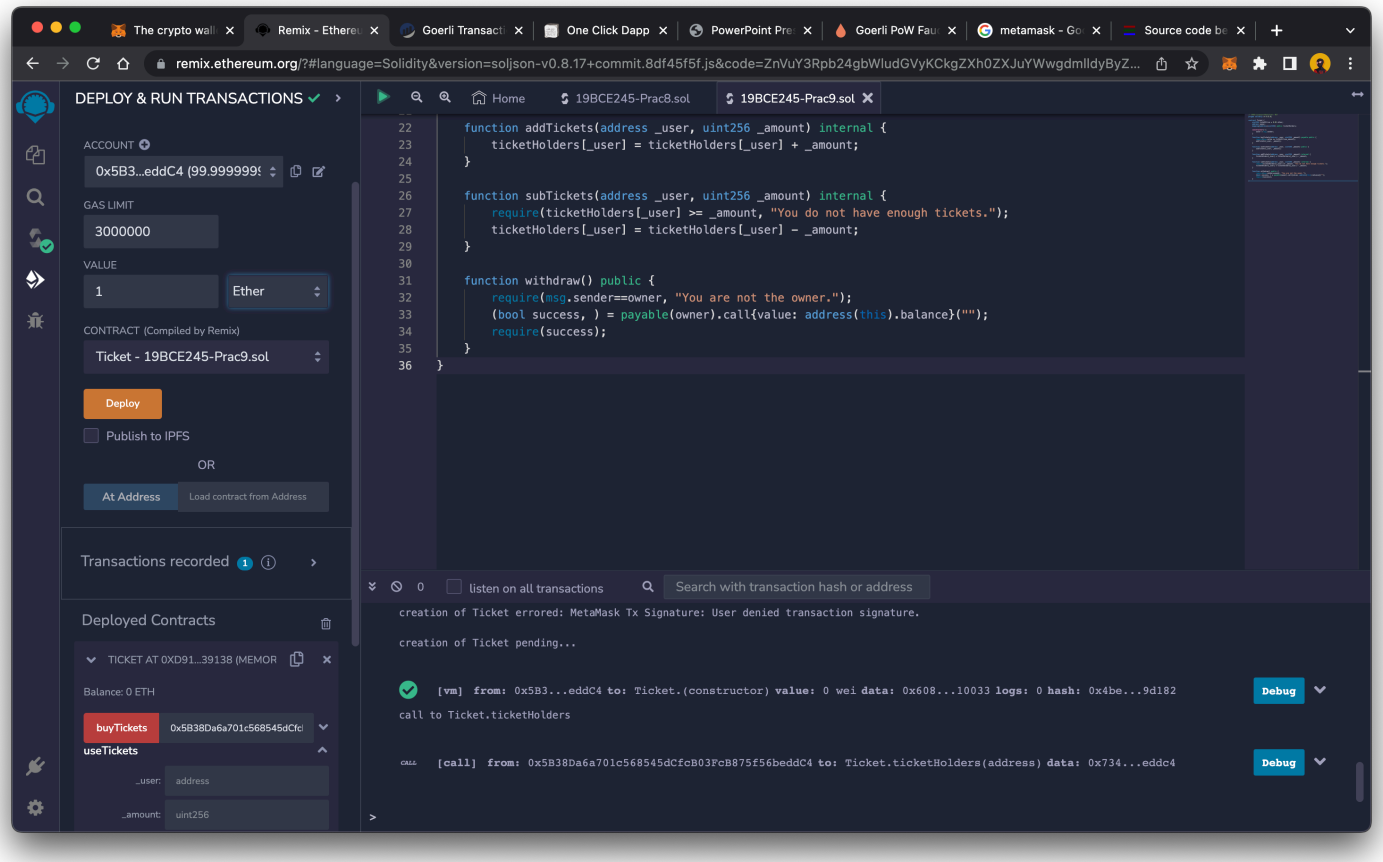
Implements a distributed ticket sales system

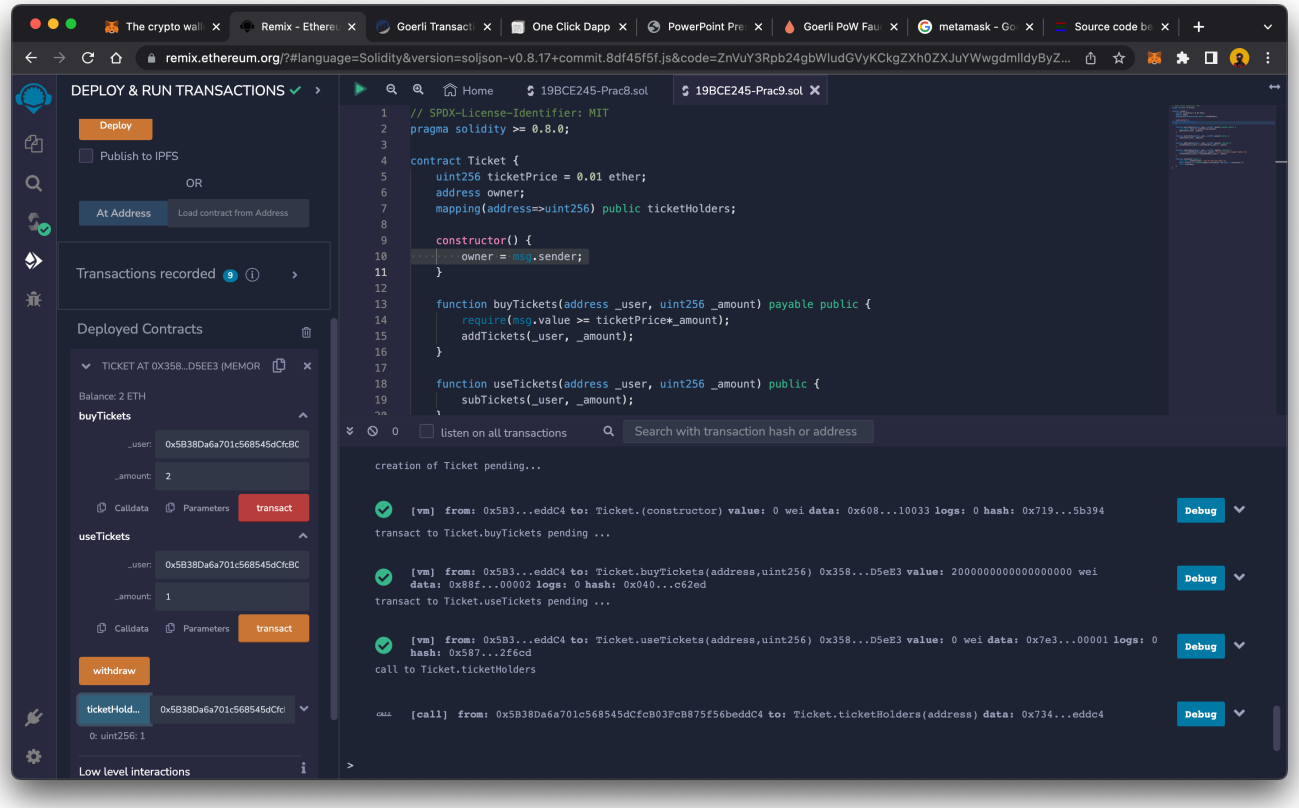
- **Code :**

```
// SPDX-License-Identifier: MIT
pragma solidity >= 0.8.0;

contract Ticket {
    uint256 ticketPrice = 0.01 ether;
    address owner;
    mapping(address=>uint256) public ticketHolders;

    constructor() {
        owner = msg.sender;
    }

    function buyTickets(address _user, uint256 _amount)
payable public {
        require(msg.value >= ticketPrice*_amount);
        addTickets(_user, _amount);
    }

    function useTickets(address _user, uint256 _amount) public
{
        subTickets(_user, _amount);
    }

    function addTickets(address _user, uint256 _amount)
internal {
        ticketHolders[_user] = ticketHolders[_user] + _amount;
    }

    function subTickets(address _user, uint256 _amount)
internal {
```

```
        require(ticketHolders[_user] >= _amount, "You do not
have enough tickets.");
        ticketHolders[_user] = ticketHolders[_user] - _amount;
    }

    function withdraw() public {
        require(msg.sender==owner, "You are not the owner.");
        (bool success, ) = payable(owner).call{value:
address(this).balance}("");
        require(success);
    }
}
```

- **Output Screenshots:**





Conclusion

In this practical, Here we purchased tickets and used tickets. Here all conditions are checked like if user has sufficient ethers, if not it will also give error messages.