Decentralized Lottery

By Jogeshwari Biradar

- **Objective:**

  Create a basic decentralized lottery.

- **Required Features:**
  - Users can enter by sending 0.01 ETH.
  - After 10 participants, the lottery ends.
  - Contract picks a random winner.
  - Winner receives 90%, contract owner gets 10%.
- **Language used:**

  Solidity, version ^0.8.20

- **Working of the contract:**

  Once the contract is deployed, users can check the entry fee and the number of participants that can enter in the lottery. In order to participate, the participant must call the function with exact value of 0.01 ETH (10^{16} Wei). Also, the lottery must not be full. If these two conditions are satisfied, only then the participant is allowed to enter.

  Every time a participant participates, the contract checks whether lottery is full or not. As soon as the lottery gets full (10 participants participated), a random participant is chosen as the winner. The transfer of Winner prize and Owner charge from the lottery prize (90% to winner and 10% to owner) takes place. The winner gets announced and the participants are deleted and contract resets for next round.

- **Functions used:**
  - participate() :
    - This is a payable function.
    - Users participate in the lottery using this function.
    - Allows user entry only when he/she sends 0.01 ETH.
    - Automatically selects winner when the lottery is full.
  - declareWinner() :
    - This is a private function.
    - Selects a random winner.
    - Splits ETH between Winner and Owner.
    - Resets the lottery
    - **NOTE:** Here, I have simulated randomness using ‘block.timestamp % participants.length’ for simplicity. In real-world applications, secure randomness like Chainlink VRF is used.
  - getParticipants() :
    - This is public view function.

    - Returns the list of the addresses participated in the lottery.
  - resetLottery() :
    - Deletes the participants.
    - Makes it ready for a new set of 10 participants.
  - event AnnounceWinner :
    - Announces the winner with emit to track the winner
- **Conclusion:**

  This contract is a simple lottery system built using Solidity. It handles user participation, selects a random winner after 10 entries, and splits the prize automatically. The logic is kept clear while making sure all required features work as expected.