Problem : A Simple Lottery Contract

Objective Learn how to manage a list of participants and select a random winner by building a simple lottery smart contract.

Learn These Elements First

- `address[] public players;`
    - **What it is:** An array is a dynamic list. This specific type, `address[]`, will store the wallet addresses of everyone who enters the lottery. We can add new players to this list.
- **Restricting Function Access**
    - **What it is:** You can create rules for who is allowed to call certain functions. For this project, you'll want to make sure only a designated `manager` (the person who deployed the contract) can start the lottery and pick a winner. You will use a simple `require(msg.sender == manager, "Only the manager can call this.");` check.
- **Pseudo-Randomness on the Blockchain**
    - **What it is:** Generating truly random numbers on a deterministic system like a blockchain is very difficult. For this simple exercise, we will use a common but insecure method for learning purposes: hashing `block.timestamp`, `block.difficulty`, and the list of `players`. This produces a "random-looking" number.
    - **The Modulo Operator (%):** This math operator gives you the remainder of a division. It's perfect for picking an item from a list. If you have 10 players, `randomNumber % 10` will give you a result between 0 and 9, which corresponds to an index in your `players` array.
    - **Important Note:** This method is **not secure** for real-money lotteries because miners can manipulate the inputs to influence the outcome.

Your Task

You must develop a smart contract named **`SimpleLottery`** that facilitates a lottery game.

- The contract must designate the deployer's address as the `manager`, who will have exclusive administrative rights.
- It must expose a `payable` function that allows any user to enter the lottery by submitting a fixed entry fee of **0.01 ETH**. The addresses of all participants must be recorded.
- It must include a `manager`-only function to conclude the lottery. This function must select a single winner from the list of participants using a pseudo-random selection method. The winner must receive the entire prize pool (all collected entry fees), and the contract's state must be reset to allow for a new lottery round.

Evaluation Criteria:

1. **Functionality**: Does the lottery correctly handle entries, prize distribution, and state resets?
2. **Access Control**: Is the function to pick a winner properly restricted to the `manager`?
3. **State Management**: Does the contract correctly track the list of players and the prize pool?
4. **Code Quality**: Is the code clean, well-structured, and secure within the known constraints of on-chain randomness?