Le Vote smart contract

```
pragma solidity ^0.4.24;

contract LeVote {

    // Defines the poll
    struct PollData {
        address owner;
        string question;
        string options;
        uint votelimit;
        uint deadline;
        bool inProgress;
        uint totalVotes;
    }

    // Event tracking of all votes
    event NewVote(string votechoice);

    // Declare a public poll called p
    PollData public poll;

    // Initiator function that stores the necessary poll information
    constructor(string _options, string _question, uint _votelimit, uint
_deadline) public {
        poll.owner = msg.sender;
        poll.question = _question;
        poll.options = _options;
        poll.votelimit = _votelimit;
        poll.deadline = _deadline;
        poll.inProgress = true;
        poll.totalVotes = 0;
    }
```

```
    // Function for voting. input is a string choice
    function vote(string choice) public {
        require(msg.sender == poll.owner);
        require(poll.inProgress);

        // If time reached, end poll
        if (poll.deadline != 0 && now >= poll.deadline) {
            endPoll();
            return;
        }

        poll.totalVotes += 1;
        emit NewVote(choice);

        // If votelimit reached, end poll
        if (poll.votelimit != 0 && poll.totalVotes >= poll.votelimit) {
            endPoll();
        }
    }

    // When time or vote limit is reached, set the poll inProgress to
false
    function endPoll() public returns (bool) {
        if (msg.sender != poll.owner) {
            return false;
        }
        poll.inProgress = false;
        return true;
}
```