

```

// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

/// @title ARZY-G Core – Living Token Prototype
/// @author Arzykul Muratov
/// @notice Tokens are born from verified useful actions, not mining or staking

contract ARZYG {
    /// @notice Address allowed to create tokens (the "Reserve Authority")
    address public reserve;

    /// @notice Tracks how many tokens each user has received
    mapping(address => uint256) public bornTokens;

    /// @notice Stores reasons (work done) for each token issued
    mapping(address => string[]) public workHistory;

    /// @notice Emitted when a new token is born
    event TokenBorn(address indexed user, uint256 amount, string reason);

    /// @param _reserve The address with permission to issue tokens
    constructor(address _reserve) {
        reserve = _reserve;
    }

    /// @notice Births tokens after verifying useful action
    /// @param to The recipient address
    /// @param amount Number of tokens to issue
    /// @param reason Description of the completed useful work
    function birthToken(address to, uint256 amount, string memory reason) public {
        require(msg.sender == reserve, "Only reserve can birth tokens");
        bornTokens[to] += amount;
        workHistory[to].push(reason);
        emit TokenBorn(to, amount, reason);
    }

    /// @notice Returns total usefulness score of a user
    function getUsefulnessScore(address user) public view returns (uint256) {
        return bornTokens[user];
    }

    /// @notice Returns all reasons (tasks) associated with a user's tokens
    function getWorkHistory(address user) public view returns (string[] memory) {
        return workHistory[user];
    }
}

```