## Assignment No: 3

**Title:** Write a smart contract on a test network, for Bank account of a customer for following

operations:

- Deposit money
- Withdraw Money
- Show balance

## Implementation:

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

contract BankAccount {
    address public owner;
    uint256 public balance;

    constructor() {
        owner = msg.sender; // Set the contract creator as the owner
        balance = 0; // Initialize balance to 0
    }

    // Deposit function: Allows the owner to deposit money into the account
    function deposit(uint256 amount) public {
        require(amount > 0, "Amount must be greater than 0");
        require(msg.sender == owner, "Only the owner can deposit");

        balance += amount;
    }
```

```
    // Withdraw function: Allows the owner to withdraw money from the account
    function withdraw(uint256 amount) public {
        require(amount > 0, "Amount must be greater than 0");
        require(msg.sender == owner, "Only the owner can withdraw");
        require(amount <= balance, "Insufficient balance");

        balance -= amount;
    }

    // Get Balance function: Allows anyone to check the account balance
    function getBalance() public view returns (uint256) {
        return balance;
    }
}
```

**OUTPUT:**