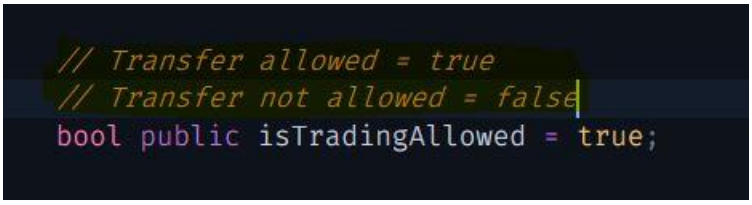
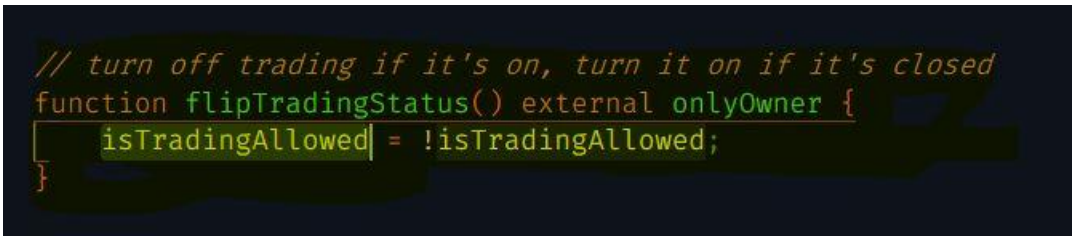
# Audit V2

Attaching snippets of code, I changed in your contract to make it more readable, added few checks, removed comments, and added a chunk of code here and there. You can do these changes in your contract. Highlighted modified code with yellow highlighter.

1. Modified the comment.




2. Changed “setTradingEnabled” function completely. It will just toggle trading status.




3. Added a check, so if by mistake admin is whitelisting already whitelisted address, it will throw an exception, to save gas fee of whitelisting again.

4. Remove these two lines, just in case you forget to.

```
function detectTransferRestriction(
    address _from,
    address _to,
    uint256 value
) public view override returns (uint8 status) {
    // 1 = Transaction is allowed
    // 0 = Transaction not allowed

    // check if trading is allowed
    require(isTradingAllowed == true, "Transfer not allowed");
```

No need

5. Will prefer transferFrom function to be written like this, also notice the if condition, what it does is that if someone has allowed an address of max uint256, then it will not detect any allowance, usually done for exchanges.

```
function transferFrom(
    address sender,
    address recipient,
    uint256 amount
)
    external
    override
    notRestricted(sender, recipient, amount)
    returns (bool)
{
    uint256 allowed = _allowances[sender][msg.sender];
    require(allowed ≥ amount, "Insufficient allowance");
    if (allowed ≠ type(uint256).max) {
        _allowances[sender][msg.sender] = allowed - amount;
    }
    transferSharesBetweenInvestors(sender, recipient, amount);
    return true;
}
```

6. Added two checks, and modified addition/ subtraction for better readability.

```
function mint(address account, uint256 amount)
    external
    onlyOwner
    returns (bool)
{
    require(account ≠ address(0), "Zero address not allowed");

    _totalSupply += amount;
    _balances[account] += amount;
    emit Transfer(address(0), account, amount);
    return true;
}

function burn(address account, uint256 amount)
    external
    onlyOwner
    returns (bool)
{
    require(account ≠ address(0), "Zero address not allowed");
    require(_balances[account] ≥ amount, "Amount greater than balance");

    _totalSupply -= amount;
    _balances[account] -= amount;
    emit Transfer(account, address(0), amount);
    return true;
}
```

7. Modified addition/ subtraction and returning bool.

```
function mint(address account, uint256 amount)
    external
    onlyOwner
    returns (bool)
{
    require(account ≠ address(0), "Zero address not allowed");

    _totalSupply += amount;
    _balances[account] += amount;
    emit Transfer(address(0), account, amount);
    return true;
}

function burn(address account, uint256 amount)
    external
    onlyOwner
    returns (bool)
{
    require(account ≠ address(0), "Zero address not allowed");
    require(_balances[account] ≥ amount, "Amount greater than balance");

    _totalSupply -= amount;
    _balances[account] -= amount;
    emit Transfer(account, address(0), amount);
    return true;
}
```