For token sale we will require a smart contract that can be owner of the token smart contract and record transaction with addresses

Constructor-
Constructor in this case will take two values , token address and token price
In order to initiate conversation between token sake and token smart contract

```
constructor(Tokens _token⬆, uint256 _tokenPrice⬆) Ownable(msg.sender){
    token = _token⬆;
    tokenPrice = _tokenPrice⬆;
}
```

Sale status is used to define status of the sale
0 = it has not started
1= it is live
2 = it has ended

```
function setSaleStatus(uint256 _saleStatus⬆) public onlyOwner{
    saleStatus = _saleStatus⬆;
}

function viewSaleStatus() public view onlyOwner  {
    if (saleStatus = 0 , "The sale hasn't started yet");
    elif(saleStatus = 1, "The sale is live");
    else ("sale is over");
    return saleStatus;
}
```

Buy tokens is used to take money from the user and register that transaction for vesting contract

```
function buyTokens(uint256 numberOfTokens⬆) external payable {
    require(saleStatus = 1);
    require(msg.value == numberOfTokens⬆.mul(tokenPrice), "Not enough amount");
    require(token.balanceOf(address(this)) >= numberOfTokens⬆, "Not enough balance of smart contract");

    token.transfer(msg.sender, numberOfTokens⬆);
    tokensSold = tokensSold.add(numberOfTokens⬆);

    emit TokensPurchased(msg.sender, numberOfTokens⬆);
}
```