Assignment.3: smart contract for bank account

CODE and OUTPUT:

// SPDX-License-Identifier: MIT

pragma solidity >=0.7.0 <0.9.0;

contract Bank {

address public owner;

mapping(address =>uint256) private userbalance;

constructor() {

owner = msg.sender;

}

modifier onlyOwner(){

require (msg.sender==owner, 'You are not the owner of this contract');

\_;

}

function deposit() public payable returns(bool) {

require(msg.value >10 wei, 'Please deposit at least 10 wei');

userbalance[msg.sender] +=msg.value;

return true;

}

function withdraw(uint256 \_amount) public payable returns (bool) {

require(\_amount <=userbalance[msg.sender], 'You dont have sufficient funds');

userbalance[msg.sender] -=\_amount;

payable(msg.sender).transfer(\_amount);

return true;

}

function getbalance() public view returns(uint256){

return userbalance[msg.sender];

}

function getBankBalance() public view onlyOwner returns(uint256){

return address(this).balance;

}

function withdrawBankBalance (uint256 \_amount) public payable onlyOwner returns (bool){

payable(owner).transfer (\_amount);

return true;

}

}