```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.7;

import "@openzeppelin/contracts/token/ERC721/IERC721.sol";
import "@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol";


contract NFTstaker {



    mapping(address => uint) public stakes;

    function stake(address token,uint _tokenid)public{
        stakes[msg.sender] = _tokenid;
        IERC721(token).transferFrom(msg.sender,address(this),_tokenid);
    }

    function unstake(address token,uint _tokenid) public{
        stakes[msg.sender] = 0 ;
        IERC721(token).transferFrom(address(this),msg.sender,_tokenid);
    }


    // function unstake(address token,uint _tokenid,uint _amount)public{
    //     stakes[msg.sender] = Stake(_tokenid,_amount);
    //     IERC721(token).transferFrom(msg.sender,address(this),_amount);
    // }
    function onERC721Received(
        address operator,
        address from,
        uint256 tokenId,
        bytes calldata data
    ) external returns (bytes4){
        return
bytes4(keccak256("onERC721Received(address,address,uint256,uint256,bytes)"));
    }


}
```