Implement an NFT Staking Contract

We are a NFT company that give loans to clients that put the NFT's as collateral in the Protocol. Many NFT projects are looking for ways to bring utility to users and incentivize long-term holding or participation. In essence, a staking contract holds tokens and tracks a few different variables to their respective owners.

We want to simulate the process of staking/unstaking NFTs (ERC721) in our protocol in order to get rewords in ERC20 tokens as incentive for depositing our NFT's in the staking protocol.

The challenge should have three different contracts. One is for the NFT, another for rewards, and finally, one is for the staking / unstacking contract itself.

As first step create a no-frills ERC20 token called RewardsToken. Use it to give users rewords for staking there NFT's in the Protocol. U can use oppenzeppelin or others implementations for the ERC2o token.Minting and rewords given for the stking is up to you.

Create a ERC721 NFT contract and mint some Net's to be able to mint some NFT's tokens to stake them in the protocol.

Staking/Unstaking process:

For every NFT staked in contract, we need to keep track of some information, and to incentive the liquidity provider the lenders will receive ERC20 tokens as rewords. The contract to receive ERC721 tokens, it must inherit IERC721Receiver from OpenZeppelin. Also we want the contract to accept as many different NFT's collections as possible.
Additionally, create a few events that will relay information to the frontend whenever someone adds or removes tokens from the contract.

The Unstaking process it will be
You'll need a staking token, data structures to track stakes, functions to create and remove stakes, and finally a reward. Your rewards system will be very dependent on your project's intended tokenomics (up to you). Calculate and distribute rewards from this demo.
Also in the part of Unstaking we want to simulate a valuation (Mock Json obj) and if the NFT value goes bellow 50% will send an event that will announce us that we are starting the liquidation process.

Implement all use cases needed to test all smart contract Functionality.

Dockerization of the smart contract's will be nice to have.