ZeroDay NFT Smart Contract

Overview

ZeroDay is a robust ERC721-based NFT smart contract designed to facilitate secure and efficient digital asset minting and trading. The contract features phased sales, Merkle tree whitelisting, and built-in royalty management.

Features

- **ERC721 Compliance**: Implements the ERC721 standard for NFTs.
- **Phased Sales**: Supports pre-sale, public sale, and reveal phases with timestamp management.
- **Merkle Tree Whitelisting**: Utilizes Merkle proofs for secure and efficient whitelist-based minting.
- **Royalties**: Implements ERC2981 for secondary sales royalties.
- **Reentrancy Guard**: Protects against reentrancy attacks using OpenZeppelin's `ReentrancyGuard`.
- **Custom Error Handling**: Provides detailed custom error messages for better debugging and user feedback.

Installation

To set up the project, ensure you have Foundry installed. Then, follow these steps:

1. Clone the repository:

```
git clone https://github.com/yourusername/ZeroDay.git
cd ZeroDay
```

PROF

2. Install dependencies:

```
make install
```

Usage

Compile the Contracts

To compile the smart contracts, run:

```
forge build
```

To deploy the smart contract, first of all add these value to your env variables.

- ALCHEMY_ENDPOINT
- ETHERSCAN_API_KEY
- PRIVATE_KEY

To deploy the smart contract, first import a wallet which has sufficient Sepolia Ethereum Testnet

```
cast wallet import <YOUR_NAME> --private-key <YOUR_PRIVATE_KEY>
```

Then run:

```
source .env
forge script script/ZeroDayDeployment.s.sol --rpc-url $ALCHEMY_ENDPOINT
--account <YOUR_NAME> --verify --broadcast
```

To get test cases results:

```
forge test -vvv
```