Homework: Charity Platform

Your task is to create a Solidity smart contract named "CharityPlatform" that implements a decentralized charity platform.

1. Project Structure

Create a Hardhat project as an environment for the involved contracts, following the recommended structure and guidelines provided by Hardhat.

2. Smart Contract Functionality

Charity Creation

- Functionality
  The Charity Platform allows users to create charitable campaigns with a unique identifier, name, description, funding goal, and deadline.
- Execution Process
  - Users call the createCampaign() function, providing the necessary parameters such as the campaign name, description, and funding goal.
  - The smart contract creates a new campaign with a unique identifier and stores the campaign details, including the funding goal.

Donation

- Functionality
  - Users can donate funds to support the charitable campaigns.
- Execution Process
  - Users call the donate() function, specifying the campaign identifier and the amount of funds they want to donate in ETH.
  - The smart contract records the donation and updates the total funds raised for the specified campaign.
  - No more funds can be collected than the specified funding goal.

Successful Campaign and Funds Release

- Functionality
  A campaign can finish in two ways - the funding goal is reached, or the deadline has passed.
  The campaign creator can manually collect the donated funds by providing an address where the funds should be sent in case the funding is successful.
- Execution Process:
  - The campaign creator calls the collectFunds() function, specifying the campaign identifier and the address to receive the funds.
  - The smart contract verifies if the caller is the campaign creator and transfers the collected funds to the provided address.

Refund for Unsuccessful Campaign

- Functionality:
  For unsuccessful campaigns that do not reach the funding goal, users can get back their donated funds.
- Execution Process:
  - Users call the Refund() function, specifying the campaign identifier.
  - The smart contract verifies if the campaign did not reach the funding goal, if the campaign deadline passed, and if the user has donated funds to the campaign.
  - If the conditions are met, the smart contract transfers the donated funds back to the user's address.

The Charity Platform ensures transparency and accountability by providing public visibility into the campaign details, funds raised, and funds disbursed.

3. General Requirements

- Smart Contracts must be implemented using the **Hardhat Development Environment**
- Smart Contracts must be written in **Solidity**.
- The application should have **Access Control** functionality. (e.g., campaign creator)
- Unit tests for 100% of the **Donation** and **Successful Campaign Funds Release** functionalities logic must be implemented.
- **Deployment** task must be implemented.
- **Interaction** tasks must be implemented for the **Charity Creation** Functionality

4. Other requirements

- Proper **Licensing** information should be added.
- **Alchemy** provider must be used for interaction with decentralized blockchain networks (Sepolia).
- **Deploy and Verify the Smart Contracts** on the Sepolia Ethereum Testnet network.
- Apply **error handling** and **data validation** to avoid crashes when invalid data is entered.
- Demonstrate use of programming concepts - Smart Contracts Security, Gas Optimization, and Design Patterns

5. Bonuses

- Brief **documentation** on the project and project architecture (as .md file)
- Mint an NFT for every user that participates in a specific campaign (OpenZeppelin based ERC-721)

6. Assessment Criteria

General Requirements – 80 %

Other Requirements – 20 %

Bonuses – up to 10 %

Additional functionality or libraries outside the general requirements, with motivated usage.