Homework: Decentralized Auction Platform

Create a decentralized auction platform that allows users to create, bid on, and finalize auctions for various items. Users should be able to participate in multiple auctions simultaneously and retrieve their funds if they are outbid or if an auction ends without their bid being the highest. The smart contract should be efficient and user-friendly, making it easy for users to interact with the platform and manage their auctions.

Functionality

Implement the following functions in the "AuctionPlatform" contract.

1. createAuction

A function to create a new auction with a unique identifier, start time, duration, and item details (name, description, and starting price). The start time must be greater than the current **block.timestamp** to ensure the auction starts in the future. The duration must be greater than zero to ensure the auction has a valid duration. Emit the appropriate event for auction creation.

2. placeBid

A function to place a bid on an auction, which must be payable and check if the bid is higher than the current highest bid for the specified auction. If the bid is valid, update the highest bid details, store the previous highest bidder's bid amount in the **availableToWithdrawal** mapping, and emit the appropriate event.

3. finalizeBid

A function to finalize the auction that can be called by anyone. This function should check if the auction has ended, and if so, transfer the winning bid amount to the auction's creator and set the auction as finalized.

Transfer the winning bid amount to the creator only in case it is bigger than 0.

4. withdraw

A function for users to withdraw their available funds when they have been outbid or when the auction has ended without their bid being the highest. This function should check the **availableToWithdrawal** mapping for the user's address, transfer the specified amount back to the user, and reset the available withdrawal amount for the user to zero.

5. onlyActiveAuction

A custom modifier to check if the auction is still active. Use it with the **placeBid** function giving a change to **placeBid** only when an auction is active.

Hints

- Hints for implementing the "**AuctionPlatform**" contract:
- Define a struct representing an individual auction, including necessary details such as auction creator, start time, end time, item information, highest bid information, and a flag to indicate if the auction has been finalized.
- Set variable to keep track of the id and increment it after new Auction is created.
- Use events to notify users when a new auction is created or when a new highest bid is submitted for an auction.
- Store auctions in a mapping using their unique identifiers.
- Implement a mapping to track the available funds for withdrawal for each user's address.
- Create a function to create a new auction and perform necessary checks for start time and duration before creating it.
- Implement a function to place bids on auctions, ensuring that the bid is higher than the current highest bid and updating the highest bid details accordingly. Store the previous highest bidder's bid amount in the available funds for withdrawal.
- Use custom modifiers to check the status of auctions before allowing users to place bids or finalize auctions.
- Create a function to finalize auctions that can be called by anyone, ensuring that the auction has ended and transferring the winning bid amount to the auction's creator if the highest bid is greater than zero.
- Implement a function for users to withdraw their available funds when they have been outbid or when the auction has ended without their bid being the highest.

```
function createAuction(
    uint256 start,
    uint256 duration,
    string memory itemName,
    string memory itemDescription,
    uint256 startingPrice
) public {
    // Check if start time is in the future
    // Check if duration is greater than zero
    // Create a unique auction ID
    // Store the auction details in the auctions mapping
    // Emit the NewAuction event
}
```

```
function placeBid(uint256 auctionId) public payable onlyActiveAuction {
    // Retrieve the auction details from the auctions mapping
    // Check if the bid amount is higher than the current highest bid
    // Update the available funds for withdrawal for the previous highest bidder (if any)
    // Update the highest bid details
    // Emit the NewHighestBid event
}
```

```
function finalizeAuction(uint256 auctionId) public {
    // Use the auctionEnded modifier
    // Retrieve the auction details from the auctions mapping
    // Check if the auction has not been finalized
    // Set the auction as finalized
    // Transfer the winning bid amount to the auction's creator if the highest bid is greater than zero
}
```

```
function withdraw() public {
    // Check the available funds for withdrawal for the user's address
    // Set the available funds for withdrawal to zero for the user's address
    // Transfer the specified amount back to the user
}
```