**AUCTION CONTRACT:**

The Practical Work for Module 2 required the creation of a Smart Contract for an auction, the characteristics and considerations of which were outlined in the Campus. Therefore, the steps taken to complete the assigned task will be explained.

1. Introduction

This document describes in detail the operation of an auction smart contract implemented in Solidity (version 0.8.20) for the Ethereum blockchain. The contract manages a transparent auction with advanced features such as time extensions, minimum bid increments, and automatic refunds.

2. Purpose

The contract allows:

• Conducting timed auctions with ETH as the currency

• Ensuring transparency in the bidding process

• Automating refunds for unsuccessful bidders

• Implementing anti-snipe mechanisms (time extensions)

• Managing automatic 2% commissions

3. Key Data Structures:

3.1 struct Bid { address bidder; ( Bidder's address) uint amount;( Amount bid in wei)}

3.2. Mappings Principales

• bidHistory: Historical array of all valid bids

• accumulatedBids: Cumulative total per address

• userBidHistory: Bid history per user

• pendingWithdrawals: Repayable funds per address

4. State Variables

|  |  |  |
| --- | --- | --- |
| **Variable** | **Type** | **Description** |
| **owner** | address | Contract owner |
| **deadLine** | uint | Auction closing date |
| **commissionPercent** | uint(constant) | 2% commission |
| **timeExtension** | uint(constant) | 10-minute extension |
| **ended** | bool | End status |
| **highestBid** | Bid | Current highest bid |

5. Main Features

5.1. Constructor

• Initializes the contract

• Sets the creator as owner

• Calculates the deadline based on \_durationSeconds

5.2. placeBid()

Payable function for bidding:

1. Validations:

• Active auction (onlyBeforeEnd)

• Value > 0

• For non-initial bids: must be 5% higher than the current bid

• The bidder cannot be the current leader

2. Logic:

• First bid sets the minimum

• Subsequent bids require a 5% increase

• Extends the time if bid is made close to the deadline

• Emits the NewBid event

5.3. endAuction()

Auction ends:

1. Requirements:

• Owner only

• Deadline reached

• Auction not previously ended

• At least one valid bid

2. Actions:

• Transfer winning bid (less 2%) to owner

• Automatically refund 98% to losers

• Issue AuctionEnded

5.4. withdrawExcess()

Allows withdrawal of excess funds:

• Only during active auction

• Calculates excess over 105% of the required minimum

• Not available for the current leading bidder

5.5. claimRefund()

Claim refunds:

• Only after the auction ends

• Transfer 98% of accumulated bids

6. Display Functions

|  |  |
| --- | --- |
| **Function** | **Description** |
| **getHighestBid()** | Returns current leading bidder and amount |
| **getBidHistory()** | Complete offer history |
| **timeRemaining()** | Remaining time in seconds |
| **bidsOf(address)** | Bid history by user |
| **totalBidOf(address)** | Total amount offered per user |

7. Events

|  |  |
| --- | --- |
| **Event** | **Description** |
| **NewBid** | New offer registered |
| **ExcessFundsAvailable** | Excess funds detected |
| **AuctionEnded** | Auction ended with winner |
| **PartialRefund** | Partial refund executed |
| **DepositWithdrawn** | Full withdrawal of funds |
| **EmergencyWithdraw** | Emergency withdrawal by owner |
| **DepositRefunded** | Automatic refund to losers |

8. Security Mechanisms

8.1. Modifiers

• onlyBeforeEnd: Only during active auction

• onlyOwner: Restricts the contract owner

8.2. Secure Patterns

• Automatic refunds to avoid "trapped funds"

• Withdraw pattern for withdrawals

• Mappings for fund tracking

9. Financial Considerations

• 2% Commission: Withheld from the winning bid

• 98% Refund: For non-winning bids

• Minimum 5% Increase: For new bids

10. Workflow

1. Setup:

• Owner displays contract duration

• Auction starts immediately

2. Bidding Phase:

• Bidders send ETH via placeBid()

• Each valid bid extends the deadline if it is close to the end

3. Completion:

• Owner executes endAuction()

• Funds distributed automatically

4. Post-Auction:

• Bidders can claim refunds

• Owner can withdraw emergency funds

The design promotes decentralization and minimizes the need for manual intervention, automating critical processes such as reimbursements.