Established – 1961 Subject :-OBJECT ORIENTED PROGRAM

SEVA SADAN’S

R. K. TALREJA COLLEGE

OF

ARTS, SCIENCE & COMMERCE
ULHASNAGAR – 421 003

CERTIFICATE

This is to certify that Mr./Ms. KRISHNA PRASAD
of F.Y.Information Technology (FYIT)
Roll No:-2541032
has satisfactorily completed the Web Designing Mini Project entitled Online Auction System during the academic year 2025 – 2026, as a part of the practical requirement. The project work is found to be satisfactory and is approved for submission.

PROF. INCHARGE **HEAD OF DEPT**

Kumodh Kukreja **Prof. Laxmi Jeswani**

INDEX

TOPIC:- Online Auction System

1.INTRODUCTION :-

An Online Auction System is a program used to manage and simulate the auction process in an organized and systematic way. It allows users to list items for auction, place bids, and track the progress of ongoing auctions. The system helps in automating the bidding process, reducing manual effort, and ensuring fair and transparent auctions.

This project is developed using C++ and follows Object Oriented Programming (OOPS) principles. A base class **AuctionItem** is used to store common item details such as item ID, name, and starting price, while derived classes like **LiveAuction** and **ClosedAuction** handle auction-specific functionality. The project makes use of inheritance, runtime polymorphism, and pointers to manage different auction items efficiently.

The system is menu-driven and allows users to create auction items, place bids, view ongoing auctions, and display auction results. This project is mainly designed for academic learning and helps in understanding the practical application of OOPS concepts in real-world systems like online auctions.

2. TOPIC DESCRIPTION :-

The **Online Auction System** is a software application designed to simulate the functioning of an online auction platform. It provides a structured environment where users can list items for auction, place bids, and monitor the status of ongoing auctions. The system automates the bidding process and ensures that auctions are conducted in a fair and organized manner.

This project is developed using **C++** and is based on **Object Oriented Programming (OOPS)** concepts. It makes use of classes and objects to represent users, auction items, and bids. Core OOPS features such as **inheritance**, **encapsulation**, **polymorphism**, and **data abstraction** are applied to efficiently manage auction operations.

The system follows a **menu-driven approach**, allowing users to perform actions such as creating auction items, placing bids, viewing auction details, and declaring winners. The Online Auction System is intended mainly for **academic purposes**, helping students understand how OOPS concepts can be implemented in real-world applications

3. OOP CONCEPT USED :-

Online Auction System

The **Online Auction System** is developed using **Object Oriented Programming (OOPS)** concepts to make the system modular, reusable, and easy to understand. The following OOPS concepts are used in this project:

1. Class and Object

Classes are used to define auction items and users, and objects are created to perform auction operations**.**

2. Encapsulation

Data members are kept private and accessed using public functions to ensure data security.

3. Inheritance

Derived classes are created from a base class to reuse common auction properties**.**

4. Polymorphism

Same function names are used to perform different actions depending on the object type**.**

5. Abstraction

Only essential auction details are shown while hiding internal implementation.

6. Pointers

Pointers are used to manage objects dynamically and support polymorphism.

7. Menu-Driven Program

A menu-driven interface is used to make the system easy to use**.**

4. IMPLEMENTATION (CODE) :-

```
#include <iostream> #include
<string>
usingnamespace std;

class AuctionItem { public:
    int itemId;     string
itemName;
    float highestBid;

    AuctionItem() {
itemId = 0;         itemName
= "";         highestBid = 0;
    }

    void createItem() {         cout << "Enter Item ID:
";         cin >> itemId;          cin.ignore();          cout
<< "Enter Item Name: ";         getline(cin,
itemName);         cout << "Enter Starting Price: ";
cin >> highestBid;          cout << "Auction item
created successfully!\n";
    }

    void displayItem() {         cout << "\nItem ID: " <<
itemId;         cout << "\nItem Name: " << itemName;
cout << "\nHighest Bid: " << highestBid << endl;
    }

    void placeBid() {          float bid;
cout << "Enter your bid amount: ";
cin >> bid;         if(bid > highestBid) {
highestBid = bid;
            cout << "Bid placed successfully!\n";
```

```
        } else {              cout << "Bid must be higher than current
highest bid.\n";
        }
    }
};

int main() {     const int MAX_ITEMS =
10;      AuctionItem
auction[MAX_ITEMS];
    int itemCount = 0;
int choice;

    do {          cout << "\n--- Online Auction System -
--\n";        cout << "1. Create Auction Item\n";
cout << "2. View Auction Items\n";         cout <<
"3. Place Bid\n";         cout << "4. Exit\n";        cout
<<
"Enter your choice: ";
cin >> choice;

        switch(choice) {
case 1:
                if(itemCount < MAX_ITEMS) {
auction[itemCount].createItem();
itemCount++;
                } else {
                    cout << "Maximum items reached!\n";
                }
break;
              case
2:
                if(itemCount == 0) {
                    cout << "No auction items available.\n";
                } else {                    cout << "\n---
Auction Items ---\n";
for(int i = 0; i < itemCount; i++) {                        cout
<< "\nItem " << i+1 << ":";                    auction[i].displayItem();
                    }
}
break;
              case
3:
                if(itemCount == 0) {                    cout << "No auction
items available. Create item first.\n";
                } else {
int id;
                    cout << "Enter Item ID to bid on: ";
cin >> id;                    bool found = false;
for(int i = 0; i < itemCount; i++) {
if(auction[i].itemId == id) {
auction[i].placeBid();                            found = true;
                            break;
                        }
                    }
                    if(!found) cout << "Item ID not found!\n";
                }
break;
              case
4:
                cout << "Exiting program...\n";
break;              default:
                cout << "Invalid choice! Try again.\n";
        }
    } while(choice != 4);
return 0;
}
```

5. OUTPUT :-

```
--- Online Auction System ---
1. Create Auction Item
2. View Auction Items
3. Place Bid
4. Exit
Enter your choice: 1

Enter Item ID: 101
Enter Item Name: laptop Enter Starting
Price: 62050 Auction item created
successfully!

--- Online Auction System ---
1. Create Auction Item
2. View Auction Items
3. Place Bid
4. Exit
Enter your choice: 2

--- Auction Items ---

Item 1:
Item ID: 101 Item Name: laptop Highest Bid: 62050

--- Online Auction System ---
1. Create Auction Item
2. View Auction Items
3. Place Bid
4. Exit
Enter your choice: 3

Enter Item ID to bid on: 101 Enter your
bid amount: 65000 Bid placed successfully!

--- Online Auction System ---
1. Create Auction Item
2. View Auction Items
```

```
3. Place Bid
4. Exit
Enter your choice: 4

Exiting program...

=== Code Execution Successful ===
```

6. CONCLUSION :-

The Online Auction System is a menu-driven program developed in C++ to simulate a real-world auction platform. This system allows users to create multiple auction items, view item details, and place bids in a fair and organized manner. By implementing Object Oriented Programming (OOPS) concepts such as classes, encapsulation, and functions, the project provides a clear structure and efficient management of auction items.

The project demonstrates how real-world applications like online auctions can be modeled using programming concepts, while also enhancing understanding of data handling, user interaction, and OOPS principles. This system is mainly designed for academic learning and can be further extended to include features like multiple bidders, auction timers, and secure payment handling.