AUCTION MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted by

SUMAN S (2303811710421161)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOVEMBER- 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **" AUCTION MANAGEMENT SYSTEM"** is the bonafide work of **SUMAN S (2303811710421161)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

CGB1201-JAVA PROGRAMMING
Dr.A.DELPHIN CAROLINA RANI, M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR

SIGNATURE

Dr.A.Delphin Carolina Rani, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

CGB1201-JAVA PROGRAMMING
Mr.MALARMANNAN A, M.E.,
SUPERVISOR
ASSISTANTP ROFESSOR

SIGNATURE

Mr. A. Malarmannan, M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on 06/12/2024.

CGB1201-JAVA PROGRAMMING
Mr. M. SARAVANAN, M.E.,
INTERNAL EXAMINER
ASSISTANT PROFESSOR

INTERNAL EXAMINER

CGB1201-JAVA PROGRAMMING
Mr.R. KARTHIK, M.E.,
EXTERNAL EXAMINER
ASSISTANT PROFESSOR
8138-SCE, TRICHY.

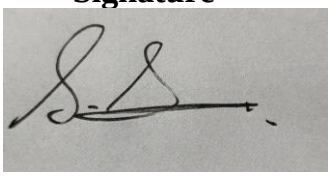
EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **"AUCTION MANAGEMENT SYSTEM"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING.**

.

Signature



SUMAN S

Place: Samayapuram

Date:06/12/2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **MR. A. MALARMANNAN, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

An Auction Management System is a digital platform designed to streamline the process of organizing, conducting, and managing auctions. It replaces traditional manual methods with an efficient, user-friendly, and transparent system that caters to buyers, sellers, and administrators. The system ensures a secure and seamless experience by integrating advanced technologies like real-time bidding, data encryption, and automated notifications. The primary objective of this system is to provide an interactive and intuitive interface that allows users to participate in auctions from any location. The platform supports different types of auctions, including English auctions, Dutch auctions, and sealed-bid auctions, offering flexibility to sellers in choosing the format that best suits their needs. Registered sellers can list their items with detailed descriptions, images, and base prices, while buyers can place bids dynamically and monitor the auction's progress in real time.Key features include user registration and authentication, item listing management, bid tracking, payment integration, and report generation.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
An Auction Management System is a digital platform designed to streamline the process of organizing, conducting, and managing auctions. It replaces traditional manual methods with an efficient, user-friendly, and transparent system that caters to buyers, sellers, and administrators. The system ensures a secure and seamless experience by integrating advanced technologies like real-time bidding, data encryption, and automated notifications. The primary objective of this system is to provide an interactive and intuitive interface that allows users to participate in auctions from any location. .	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO7 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Objective

The primary objective of the Auction Management System is to design and implement a user-friendly, interactive platform that facilitates the management of auctions. This includes functionalities for adding auction items, placing bids, and displaying auction details, ensuring a seamless experience for users. The system leverages Java Swing and AWT to provide a graphical user interface (GUI) that demonstrates efficient event-driven programming, data validation, and real-time updates, fostering practical skills in object-oriented software development.

1.2 Overview

The primary objective of the Auction Management System is to design and implement a user-friendly, interactive platform that facilitates the management of auctions. This includes functionalities for adding auction items, placing bids, and displaying auction details, ensuring a seamless experience for users. The system leverages Java Swing and AWT to provide a graphical user interface (GUI) that demonstrates efficient event-driven programming, data validation, and real-time updates, fostering practical skills in object-oriented software development.

1.3 Java Programming Concepts

The Auction management system utilizes several fundamental Java programming concepts to create a robust and interactive application. The project is based on Object-Oriented Programming (OOP) principles, where the core logic is encapsulated in classes such as auction management, which contains fields and methods responsible for managing auction amounts, expenses, and budget insights.

Object-Oriented Programming (OOPs) is essential for structuring programs effectively. Classes and Objects represent entities like Expenses, Users, and Categories. Polymorphism enables handling different types of expenses, such as fixed vs. variable, or generating various reports. Inheritance allows for reusing and extending functionalities, ensuring a modular and scalable design.

The Auction management system uses Swing to build the graphical user interface (GUI) with components like 'JFrame', 'JPanel', and 'JButton'. Java AWT provides basic components for building user interfaces. Labels (Label) are used to display static text like "Item Name," "Base Price," or "Highest Bidder." Text Fields (TextField) enable users to input data, such as item details, bidder names, or bid amounts. Buttons (Button) allow users to perform actions like adding items, placing bids, or closing the auction, facilitating a functional and interactive interface.

Java Swing offers a comprehensive range of prebuilt components, such as buttons, text fields, labels, combo boxes, and tables, that can be customized for building intuitive auction interfaces. It supports interactive functionality through event listeners and handlers, enabling user interaction. Moreover, Java Swing promotes modularity by separating functionalities into smaller components, making the system easier to manage, debug, and enhance.

CHAPTER 2
PROJECT METHODOLOGY

2.1 Proposed Work

The proposed work outlines a structured approach to designing and developing an Auction Management System with a focus on creating an intuitive and user-friendly interface using Java's AWT (Abstract Window Toolkit) and Swing. These Java frameworks are well-suited for graphical user interface (GUI) development, providing a range of components for building interactive applications. The system aims to streamline the management of auction items, bidders, and bidding processes while ensuring a smooth user experience. By leveraging Java Swing and AWT, the application will provide a robust platform for managing auctions efficiently.

The primary goal of the project is to develop a GUI-based auction management system that facilitates seamless interaction between sellers, bidders, and the auction platform. The system will enable sellers to add items for auction, complete with essential details such as the item name, base price, and description. This functionality ensures that all auction items are well-documented and accessible to potential bidders, fostering transparency and clarity in the auction process.

In addition to seller functionality, the system will provide an interactive platform for bidders to participate in auctions. Bidders will be able to place bids on listed items through the GUI in real-time. The system will dynamically update and display the highest bid for each item, allowing bidders to track the auction's progress and make informed decisions. This real-time interaction enhances user engagement and creates a competitive bidding environment.

The use of Java Swing and AWT ensures a modular design, allowing for the easy addition of new features and functionalities in the future. The application's design will emphasize simplicity and ease of use, making it accessible to users with varying levels of technical expertise. Overall, this Auction Management System aims to provide a streamlined, efficient, and interactive solution for conducting and managing auctions.

2.2 Block Diagram

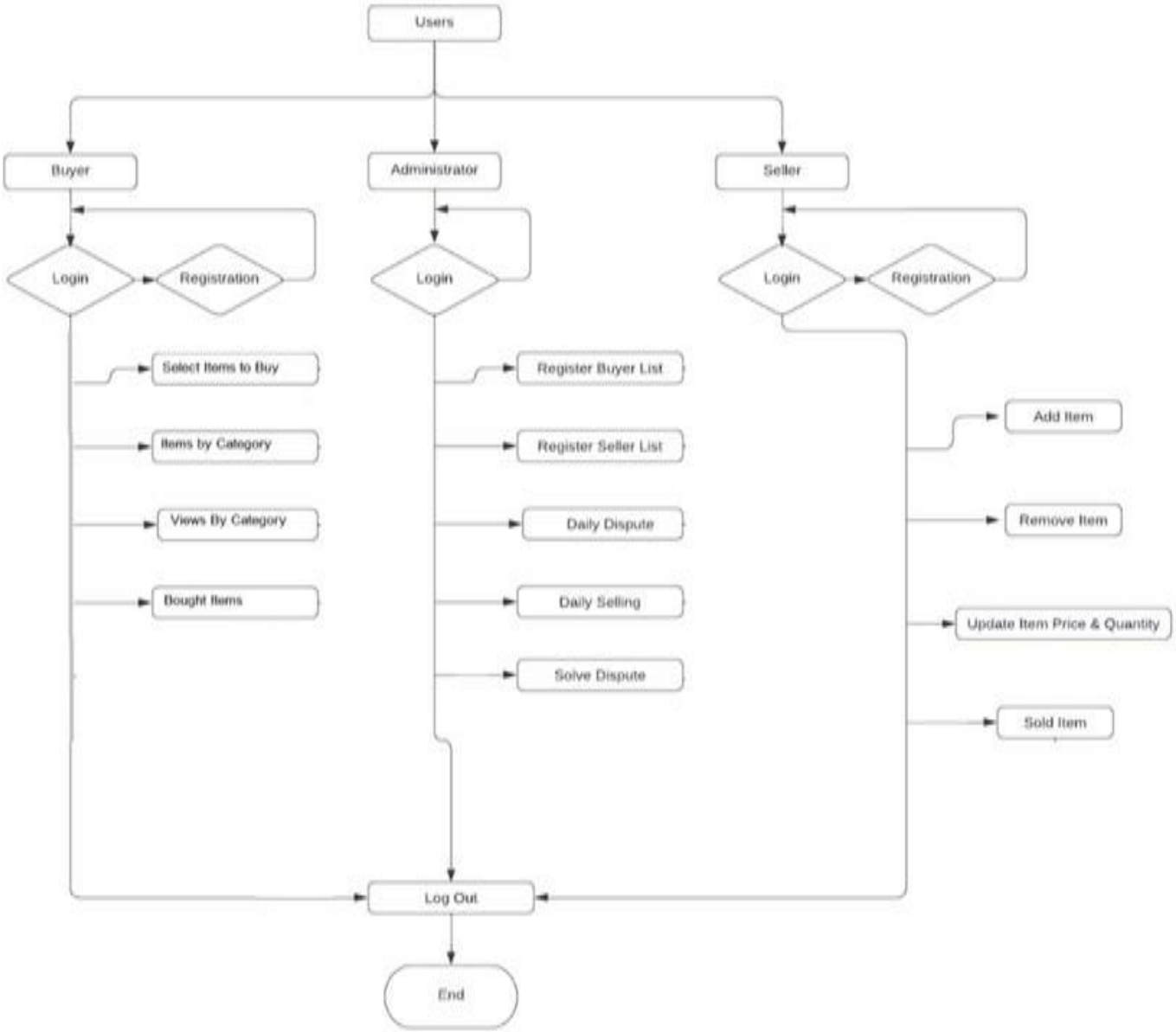


Fig. 2.1 Auction Management System Block Diagram

CHAPTER 3
MODULE DESCRIPTION

3.1 User Interface (UI) Module

The User Interface (UI) Module provides an interactive graphical interface for users to engage with the Auction Management System. This module is designed to offer a seamless and intuitive experience for both sellers and bidders, allowing them to interact with the system efficiently. AWT Components are used to establish the basic layout of the interface. Components like Panels, Buttons, and Labels help in creating simple, functional elements for organizing the interface. These elements enable essential tasks, such as submitting bids, adding items, and displaying basic information. To enhance the look and feel of the system, Swing Components are utilized. Components such as JFrame , JPanel, JButton, JTextField, and JLabel offer a modern and visually appealing design. These components allow for better organization of content and more interactive user actions, such as placing bids or adding auction details. Swing also provides components like JList and JTable, which are used to display auction items and bid history in a structured format. These advanced components help in managing and presenting data clearly, ensuring a smooth and informative experience for users. Overall, the UI module integrates both AWT and Swing components to create an intuitive and user-friendly interface that facilitates efficient interaction between users and the Auction Management System.

3.2 Item Management Module

The Auction Item Management Module is designed to handle the addition, updating, and management of auction items within the Auction Management System. This module is essential for ensuring that items are accurately listed, and all necessary information is captured, allowing both sellers and bidders to engage effectively with the auction platform.

At the heart of this module is a form for adding items. This form includes essential fields such as item name, description, and base price, which sellers need to fill out when listing an item for auction. The item name and description provide necessary details, helping bidders understand what is being auctioned. The base price serves as the starting point for bidding, establishing a minimum value for the item. The form is designed to be user-friendly, ensuring sellers can easily input their data without confusion. Data validation is a crucial aspect of this module, ensuring that all fields are correctly filled out before the item is added to the auction list. Validation checks are implemented to verify that the item name and description are not empty and meet predefined length or character requirements. Additionally, the base price field is validated to ensure that a positive numeric value is entered, preventing errors such as entering negative numbers or non-numeric values. This validation process is essential for maintaining data integrity and preventing incomplete or incorrect listings from being posted. Once the data is validated, the item is added to the auction system, making it visible to potential bidders. This module also allows for the modification of item details and removal of items from the auction if needed, ensuring that the auction data remains up-to-date and accurate. Overall, the Auction Item Management Module ensures that auction items are properly added, validated, and maintained, providing a smooth experience for both sellers and bidders.

3.3 Bid Management Module

The Bid Management Module is a critical component of the Auction Management System, responsible for managing the bidding process. This module allows users to browse auction items and place bids in an easy-to-use interface. It ensures that the process is smooth, interactive, and accessible to all participants, providing a user-friendly platform for placing bids. One of the main features of the module is the ability to select an item and place a bid. The system provides an organized view of auction items, and bidders can select the item they wish to bid on. The interface is designed to allow easy input of bid amounts, making it convenient for users to participate in the

auction without difficulty. Bid validation is a key aspect of this module. When a bid is placed, the system checks if the new bid exceeds the current highest bid. If the new bid is lower than or equal to the highest bid, it is rejected. This ensures that the auction remains competitive, and only higher bids are accepted, maintaining fairness throughout the bidding process. Once a valid bid is placed, the system dynamically updates the item details with the new highest bid and the identity of the bidder. These updates occur in real-time, ensuring that all participants can track the current bid status and are always informed of the latest developments in the auction. This feature keeps the auction process transparent and engaging for bidders.

3.4 Real-Time Update Module

The Real-Time Update Module ensures that all bid information is consistently updated across the Auction Management System. Whenever a new bid is placed, the module dynamically refreshes the interface, immediately displaying the latest highest bid and bidder details. This real-time update feature ensures that all users have access to the most current information without needing to manually refresh the page, creating a seamless and engaging experience for both bidders and sellers .In addition to updating bid information, the module also notifies users whether their bid is successful or not. If a bidder's offer surpasses the current highest bid, the system notifies them that their bid has been accepted and updates the auction details.

If the bid is invalid or does not meet the required criteria, users are promptly notified, prompting them to place a higher offer. This ensures the auction remains fair, competitive, and transparent, keeping participants informed and engaged throughout the process.

3.5 Auction Summary Module

The Auction Summary Module serves as a comprehensive tool for displaying and summarizing the results of the auction. It provides an overview of each item auctioned, showing critical details such as the highest bidder and the final bid amount. This module is essential for both sellers and bidders as it allows them to quickly assess the outcome of the auction, offering transparency and clarity on which bid was successful for each item listed. In addition to presenting the highest bid, the module also allows users to view the complete history of all bids placed for an item. This feature displays a detailed table of each bid, including the bid amounts and corresponding bidder information, enabling users to track how the auction evolved over time. This transparency helps foster a sense of fairness, as all participants can see the progression of bids for each item, understanding how the final bid was reached.The Auction Summary Module also offers an option to save auction results to a file. This feature provides users with the ability to store auction data for future reference or analysis. Sellers may want to keep a record of their items' final results, while bidders may wish to track their participation. Saving the results also provides an archival benefit, allowing users to revisit past auctions or generate reports for business or personal use. By offering these features, the Auction Summary Module enhances the overall auction experience.

It not only presents key auction results in a clear and organized manner but also allows for detailed tracking and reporting. This functionality ensures that all participants are informed about the auction outcomes and can retain the information for later review or analysis, making the auction process more efficient and transparent.

CHAPTER 4
CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

The Auction Management System implemented in Java using AWT and Swing demonstrates the power and flexibility of Java for creating interactive and user-friendly desktop applications. By leveraging the robust GUI components provided by Swing and the layout management capabilities of AWT, the system successfully manages auction operations such as adding items, placing bids, and displaying results in real time.

This project highlights the importance of combining core programming principles with intuitive design to deliver efficient, scalable, and maintainable solutions. The system ensures transparency, accuracy, and ease of use, making it suitable for real-world applications. Furthermore, it serves as a foundational step for more advanced auction platforms that can integrate features like database storage, network-based bidding, or analytics.

Overall, the Auction Management System is a practical demonstration of Java’s capabilities in solving real-world problems through interactive and dynamic software applications. It is a valuable project for enhancing programming and software design skills.

Moreover, the integration of modern technologies like REST APIs and cloud computing can help extend the system's capabilities, supporting cross-platform accessibility and large-scale operations.

4.2 FUTURE SCOPE

The future scope of the Auction Management System (AMS) lies in enhancing its capabilities to cater to a broader range of users, improve its functionality, and integrate with advanced technologies. One of the key areas for development is integration with machine learning algorithms to predict bidding patterns and suggest optimal bidding strategies for users. By analyzing historical data, the system could provide insights on bidding trends, helping bidders make more informed decisions and enhancing their chances of success in future auctions. Another potential enhancement is the integration with blockchain technology for increased security and transparency. Blockchain can ensure that the auction process is tamper-proof, offering an immutable ledger of all bids and transactions. This would reduce the chances of fraud and provide a more secure and trustworthy environment for both bidders and sellers. Blockchain could also streamline payment processes, making transactions faster and more secure.

The system could also be extended to support multi-platform access, enabling users to participate in auctions from various devices, including mobile phones, tablets, and desktops. A mobile app version of the Auction Management System could be developed to ensure that users can participate in auctions on the go, increasing accessibility and user engagement. Additionally, features like push notifications can be added to alert users about bid status, auction results, and important deadlines.

Finally, expanding the global reach of the Auction Management System by incorporating multi-language support and currency conversion would make the system more adaptable to international users. By enabling bidding across different languages and currencies, the system can cater to a global audience, making it suitable for international auctions and expanding its user base.

APPENDIX A

(SOURCE CODE)

```
import javax.swing.*;
import java.lang.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
public class AuctionManagementSystem {
private JFrame frame;
private JTextField sellerNameField, itemNameField, basePriceField,
bidderNameField,
bidAmountField;
private JTextArea itemsArea;
private JComboBox<String> itemDropdown;
private String sellerName;
private ArrayList<AuctionItem> auctionItems = new ArrayList<>();
public AuctionManagementSystem() {
// Step 1: Ask for seller name, item name, and base price
frame = new JFrame("Auction Management System");
frame.setSize(500, 600);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setLayout(null);
JLabel sellerNameLabel = new JLabel("Seller Name:");
sellerNameLabel.setBounds(20, 20, 100, 25);
sellerNameField = new JTextField();
sellerNameField.setBounds(140, 20, 200, 25);
JLabel itemNameLabel = new JLabel("Item Name:");
itemNameLabel.setBounds(20, 60, 100, 25);
```

```
itemNameField = new JTextField();
itemNameField.setBounds(140, 60, 200, 25);
15
JLabel basePriceLabel = new JLabel("Base Price:");
basePriceLabel.setBounds(20, 100, 100, 25);
basePriceField = new JTextField();
basePriceField.setBounds(140, 100, 200, 25);
JButton addItemButton = new JButton("Add Item");
addItemButton.setBounds(140, 140, 100, 30);
JButton finishButton = new JButton("Finish");
finishButton.setBounds(250, 140, 100, 30);
frame.add(sellerNameLabel);
frame.add(sellerNameField);
frame.add(itemNameLabel);
frame.add(itemNameField);
frame.add(basePriceLabel);
frame.add(basePriceField);
frame.add(addItemButton);
frame.add(finishButton);
addItemButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
addItem();
}
});
finishButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
if (auctionItems.isEmpty()) {
```

```
JOptionPane.showMessageDialog(frame, "Please add at least one item before
finishing.");
16
} else {
showAuctionScreen();
}
}
});
frame.setVisible(true);
}
private void addItem() {
try {
if (sellerName == null) {
sellerName = sellerNameField.getText();
if (sellerName.isEmpty()) {
JOptionPane.showMessageDialog(frame, "Please enter the seller's name.");
return;
}
}
String itemName = itemNameField.getText();
double basePrice = Double.parseDouble(basePriceField.getText());
if (itemName.isEmpty()) {
JOptionPane.showMessageDialog(frame, "Item name cannot be empty.");
return;
}
// Add item to the list
AuctionItem newItem = new AuctionItem(itemName, basePrice);
auctionItems.add(newItem);
// Clear input fields for the next item
```

```
itemNameField.setText("");
basePriceField.setText("");
17
JOptionPane.showMessageDialog(frame, "Item added successfully!");
} catch (NumberFormatException ex) {
JOptionPane.showMessageDialog(frame, "Invalid base price. Please enter a numeric
value.");
}
}
private void showAuctionScreen() {
frame.getContentPane().removeAll(); // Clear previous components
frame.setSize(500, 700);
frame.setLayout(null);
JLabel itemsLabel = new JLabel("Auction Items:");
itemsLabel.setBounds(20, 20, 100, 25);
itemsArea = new JTextArea(10, 30);
itemsArea.setEditable(false);
JScrollPane scrollPane = new JScrollPane(itemsArea);
scrollPane.setBounds(20, 50, 440, 150);
// Display all added items
for (AuctionItem item : auctionItems) {
itemsArea.append("Item: " + item.getName() + ", Base Price: $" +
item.getBasePrice()
+ "\n");
}
frame.add(itemsLabel);
frame.add(scrollPane);
// Dropdown to select an item
JLabel selectItemLabel = new JLabel("Select Item:");
```

```
selectItemLabel.setBounds(20, 220, 100, 25);
itemDropdown = new JComboBox<>();
for (AuctionItem item : auctionItems) {
18
itemDropdown.addItem(item.getName());
}
itemDropdown.setBounds(140, 220, 200, 25);
JLabel bidderNameLabel = new JLabel("Bidder Name:");
bidderNameLabel.setBounds(20, 260, 100, 25);
bidderNameField = new JTextField();
bidderNameField.setBounds(140, 260, 200, 25);
JLabel bidAmountLabel = new JLabel("Bid Amount:");
bidAmountLabel.setBounds(20, 300, 100, 25);
bidAmountField = new JTextField();
bidAmountField.setBounds(140, 300, 200, 25);
JButton placeBidButton = new JButton("Place Bid");
placeBidButton.setBounds(140, 340, 100, 30);
JButton checkWinnerButton = new JButton("Check Winner");
checkWinnerButton.setBounds(250, 340, 140, 30);
frame.add(selectItemLabel);
frame.add(itemDropdown);
frame.add(bidderNameLabel);
frame.add(bidderNameField);
frame.add(bidAmountLabel);
frame.add(bidAmountField);
frame.add(placeBidButton);
frame.add(checkWinnerButton);
placeBidButton.addActionListener(new ActionListener() {
@Override
```

```
public void actionPerformed(ActionEvent e) {
placeBid();
}
19
});
checkWinnerButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
checkWinner();
}
});
frame.revalidate();
frame.repaint();
}
private void placeBid() {
try {
String selectedItemName = (String) itemDropdown.getSelectedItem();
String bidderName = bidderNameField.getText();
double bidAmount = Double.parseDouble(bidAmountField.getText());
if (bidderName.isEmpty()) {
JOptionPane.showMessageDialog(frame, "Bidder name cannot be empty.");
return;
}
// Find the selected item and update its highest bid
for (AuctionItem item : auctionItems) {
if (item.getName().equals(selectedItemName)) {
if (bidAmount > item.getHighestBid()) {
item.setHighestBid(bidAmount);
item.setHighestBidder(bidderName);
```

```
JOptionPane.showMessageDialog(frame, "Bid placed successfully!");
} else {
JOptionPane.showMessageDialog(frame, "Bid amount must be higher than the
current highest bid.");
20
}
break;
}
}
itemsArea.setText("");
for (AuctionItem item : auctionItems) {
itemsArea.append("Item: " + item.getName() + ", Base Price: $" +
item.getBasePrice() + ", Highest Bid: $" + item.getHighestBid() + ", Highest Bidder:
" +
item.getHighestBidder() + "\n");
}
} catch (NumberFormatException ex) {
JOptionPane.showMessageDialog(frame, "Invalid bid amount. Please enter a
numeric
value.");
}
}
private void checkWinner() {
StringBuilder result = new StringBuilder();
for (AuctionItem item : auctionItems) {
result.append("Item: ").append(item.getName()).append(", Highest Bidder:
").append(item.getHighestBidder()).append(", Bid:
$").append(item.getHighestBid()).append("\n");
}
```

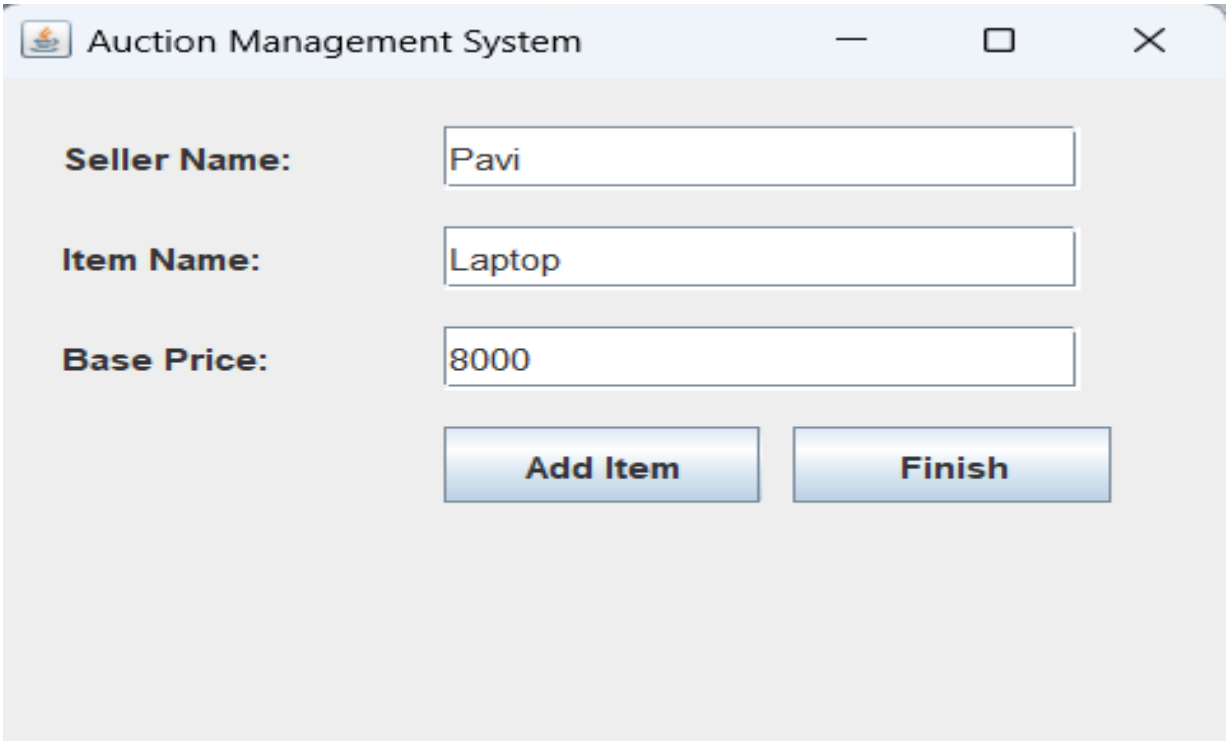
```
JOptionPane.showMessageDialog(frame, result.toString(), "Auction Results",
JOptionPane.INFORMATION_MESSAGE);
}
public static void main(String[] args) {
new AuctionManagementSystem();
}
}
class AuctionItem {
private String name;
21
private double basePrice;
private double highestBid;
private String highestBidder;
public AuctionItem(String name, double basePrice) {
this.name = name;
this.basePrice = basePrice;
this.highestBid = basePrice;
this.highestBidder = "No bids yet";
}
public String getName() {
return name;
}
public double getBasePrice() {
return basePrice;
}
public double getHighestBid() {
return highestBid;
}
public void setHighestBid(double highestBid) {
```

```
this.highestBid = highestBid;
}
public String getHighestBidder() {
return highestBidder;
}
public void setHighestBidder(String highestBidder) {
this.highestBidder = highestBidder;
}
}
```

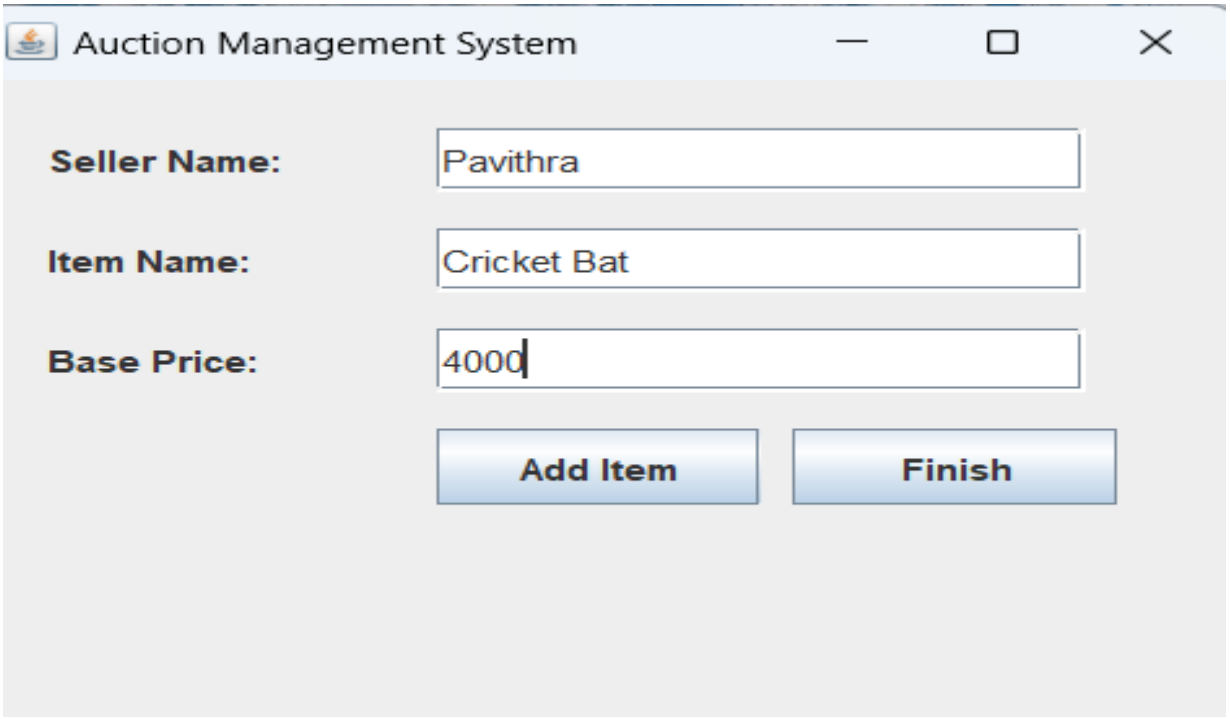
APPENDIX B

(SCREENSHOTS)

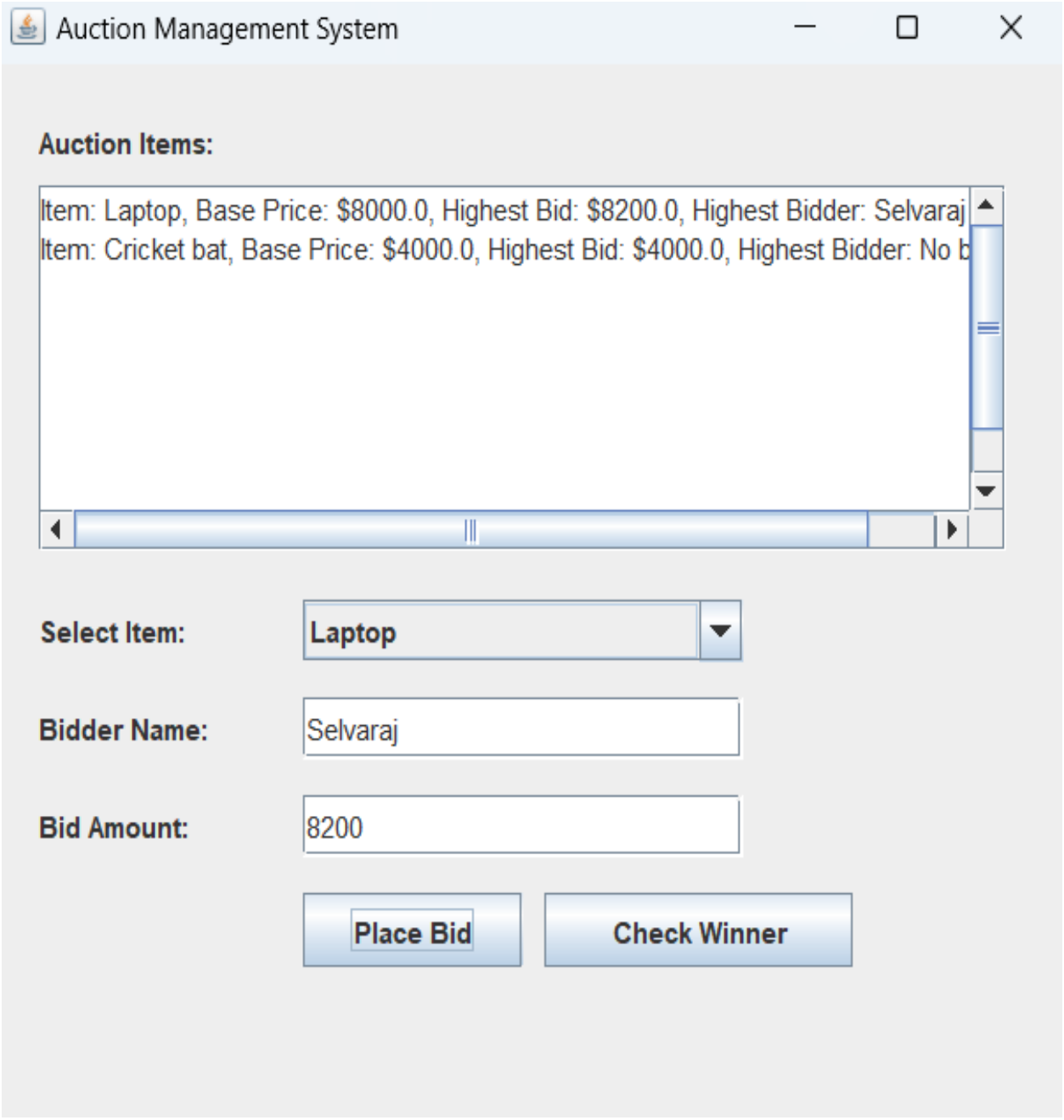
SCREENSHORT 1



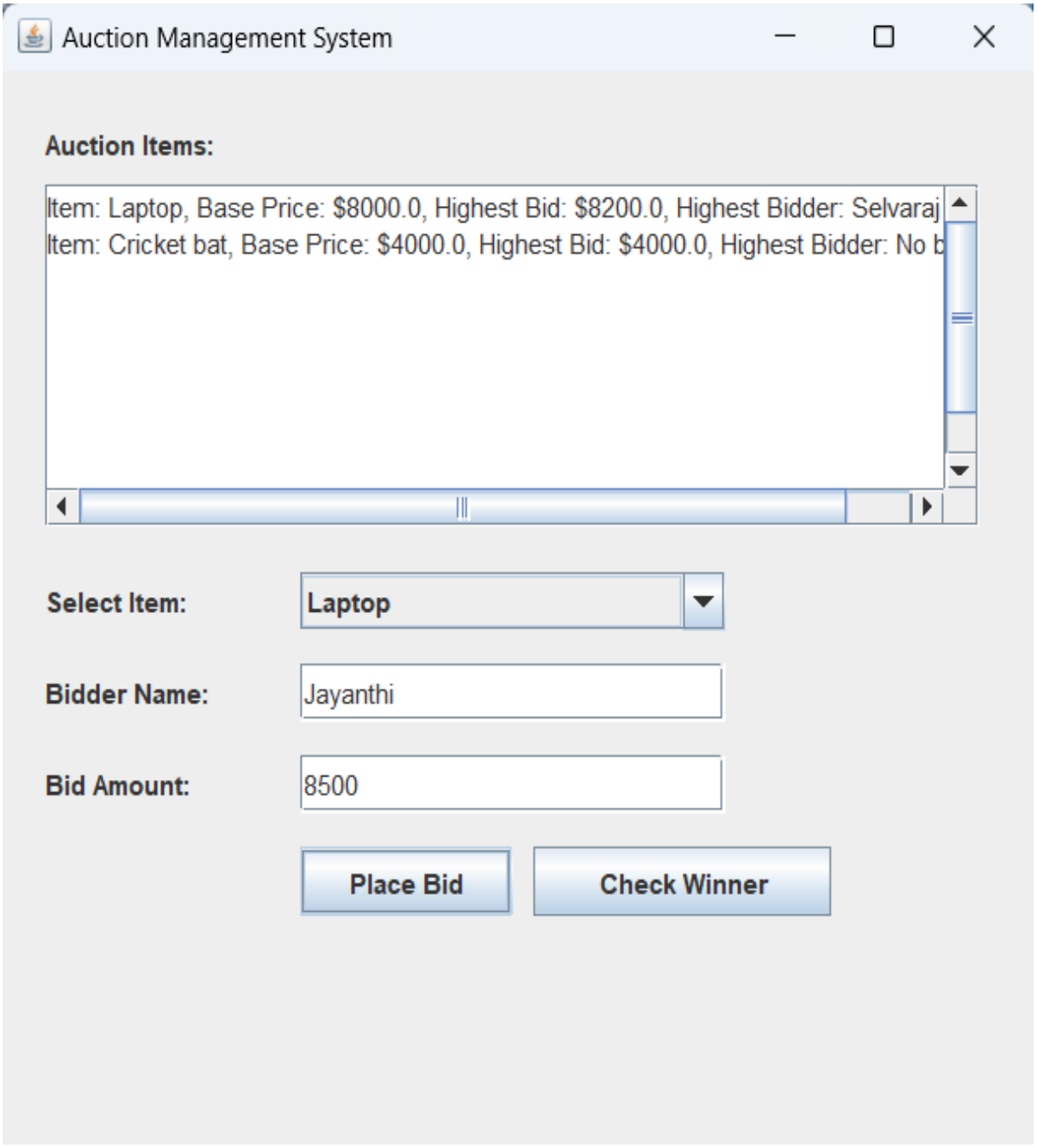
SCREENSHOT 2



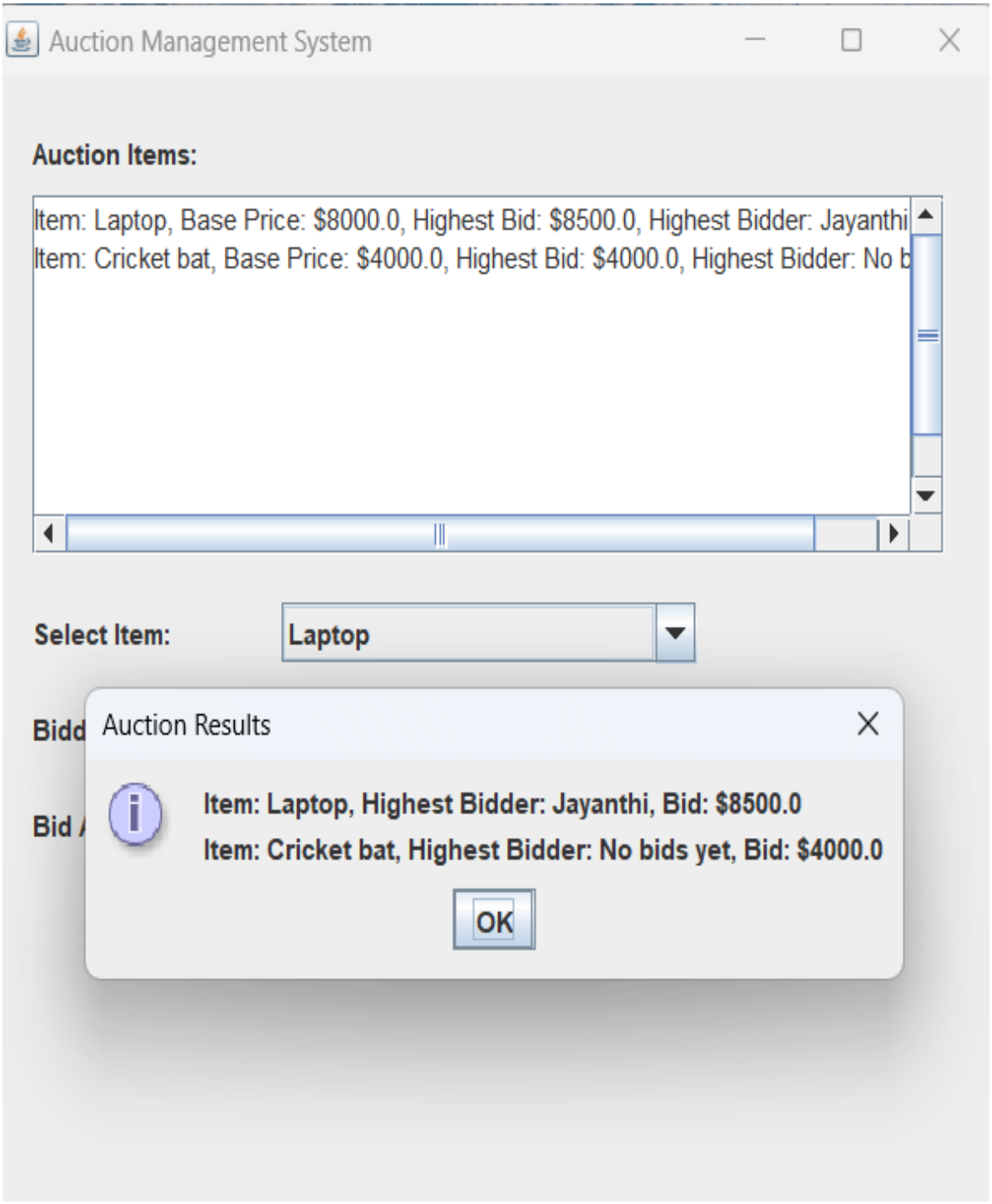
SCREENSHOT 3



SCREENSHOT 4



SCREENSHOT 5



REFERENCES

1. Herbert Schildt, Java: The Complete Reference, McGraw Hill Education, 11th Edition 2019.

2. Cay S. Horstmann, Core Java Volume I Fundamentals, Pearson Education, 11th Edition, 2018.

3. Oracle Java Documentation: https://docs.oracle.com/en/java

4. GeeksforGeeks Java Tutorials: https://www.geeksforgeeks.org/java

5. Telusko Java Programming Tutorials