Advance E-Auction System Using Python Django

Deepali Joshi[1], Vivek Benbanshi[2], Sayali Baviskar[3], Pranesh Jadhav[4]

[1]Asst Prof Department of Computer Engineering & IETE'S Bharat College of Engineering
[2] BE Department of Computer Engineering & IETE'S Bharat College of Engineering
[3] BE Department of Computer Engineering & IETE'S Bharat College of Engineering
[4] BE Department of Computer Engineering & IETE'S Bharat College of Engineering

---***---

Abstract - The Advanced E-Auction System developed in this project using Django offers a comprehensive platform for conducting online auctions, featuring both English and Dutch auction types. Leveraging the robust capabilities of the Django framework, the system ensures scalability, security, and ease of maintenance. The English Auction, characterized by its open ascending price format, allows bidders to place increasingly higher bids until the highest bid wins. Real-time bidding updates, bid management functionalities, and automatic bid increments enhance the transparency and efficiency of this auction type. Conversely, the Dutch Auction, with its reverse bidding process, start with a high price that gradually decreases. The system facilitates this process by managing price decrement intervals, tracking bidder responses, and ensuring fairness in price adjustments. Key features like user registration and authentication, item listing, real-time auction progress updates, secure payment processing, and administrative tools for managing auctions and monitoring user activities. By incorporating both auction types within a single platform, the system provides flexibility and versatility to cater to various auction scenarios, fostering a dynamic marketplace environment for bidder and sellers.

***Key Words*:** Web Application, Django, Html, CSS, javascript, Python, Secure Payment using RayzorPay, SQlite, English Auction, Dutch/tender Auction, Bidder, Seller, model, User authentication.

1.INTRODUCTION

The purpose of this project is to build an "online auction system," providing a platform for bidder/buyer and sellers to engage in trade across various items. The system operates as a web portal, allowing registered users to initiate new auctions and place bids on items. Each auction is characterized by a name, description, and possibly a photo uploaded by users, with a defined end period preventing bids once concluded. Administrators possess authority to approve or reject user-proposed auctions, access user and item information, and manage auction categories. Implemented with a 3-tier architecture, the system comprises a relational database housing item, user, auction, and category data; an application server managing system logic; and a presentation layer accessible via web browsers for user interaction. This architecture ensures database integrity and enables administrators to manipulate data via their browsers without direct database access.

The aim of this research paper is to present the development of an online auction system using the Python Django framework. Django is a powerful and popular web development framework that provides a robust foundation for building web applications. It follows the Model-View-Template (MVT) architectural pattern, which promotes modularity, scalability, and code reusability. By leveraging the capabilities of Django, we aim to create a secure, efficient, and user-friendly online auction system.

2. EXISTING SYSTEM

The existing auction system functions as a traditional marketplace where buyers and sellers convene to trade goods and services. Participants engage in auctions through physical gatherings or designated venues where items are presented for bidding. The process typically involves an auctioneer facilitating the event, overseeing bids, and announcing winning offers. In this conventional setup, prospective buyers physically inspect items before placing bids, fostering a sense of transparency and confidence in the transaction. Auction catalogs or listings may be distributed beforehand to inform attendees about available items. Transactions in the existing system are conducted in real-time, with bids escalating until a final offer is accepted by the seller. Payment and item collection often occur immediately or within a specified timeframe after the auction's conclusion.

While the traditional auction system has proven effective for centuries, it may face limitations such as geographical constraints, limited audience reach, and logistical challenges. Additionally, the lack of digital infrastructure may hinder accessibility and convenience for participants, especially in today's increasingly digital age.
However, despite these drawbacks, traditional auctions continue to thrive in various industries, including art, antiques, real estate, and automotive, where the in-person experience and expertise of auctioneers remain valued aspects of the process.

3. PROBLEM MOTIVATION

In today's digital age, traditional auction systems face challenges related to accessibility, scalability, and efficiency. To address these issues and capitalize on the benefits of online platforms, there is a need to develop an advanced E-Auction System using Python Django web technology. The system aims to revolutionize the auction experience by providing a robust, user-friendly, and feature-rich platform for buyers and sellers to engage in virtual auctions seamlessly.

Key Objectives:

Accessibility: Develop a user-friendly web interface accessible from any device with an internet connection, ensuring broad participation regardless of geographical location or time constraints.

Scalability: Design a scalable architecture capable of handling a large volume of concurrent users and auctions, ensuring smooth operation even during peak traffic periods.

Efficiency: Implement advanced bidding algorithms and automation features to streamline the auction process, reducing manual intervention and enhancing transaction efficiency.

Security: Implement robust security measures, including encryption, authentication, and authorization mechanisms, to safeguard user data, transactions, and sensitive information.

Customization: Provide flexibility for users to customize their auction preferences, including auction types, bidding rules, and notification settings, enhancing the overall user experience.

Analytics: Incorporate analytics and reporting tools to provide insights into auction performance, bidder behavior, and market trends, empowering users to make informed decisions.

Integration**:** Seamlessly integrate with external payment gateways, shipping providers, and third-party services to facilitate secure transactions and streamline post-auction logistics.

Administration: Develop comprehensive administrative tools for managing auctions, users, payments, and system settings, enabling efficient oversight and control of the platform.

4. METHODOLOGY

HTML, or Hypertext Markup Language, is the standard language for creating and structuring content on the web ,HTML helps organize information through elements like headings, paragraphs, and lists. It allows for the inclusion of hyperlinks to navigate between sections or external sources. HTML also supports multimedia integration, ensures accessibility, aids discoverability through metadata, and enables interactivity for engaging readers. Overall, HTML serves as a fundamental tool for presenting research effectively online.

CSS for Cascading Style Sheets stands. It is a sheet-style language used to define the appearance and format of a markup document. It gives HTML a supplementary function. Used with HTML, the style of user interfaces and webpage is changed. It may also be used in XML documents of any form, including simple XUL, SVG and XML documents. In most websites, CSS is used with HTML and JavaScript to develop web-based user interfaces and user interfaces for a variety of mobile applications. What CSS accomplishes is: you can add new appearances to your old HTML pages, modify the style and feel of your website with just a few modifications to CSS code. C.S.S. is used in the creation of HTML Tags. C.S.S. is used widely used as a web language, to create a web page, we usually use H.T.M.L., C.S.S. and JavaScript. CSS is also a commonly used language in Cascading Style sheet. It allows web developers to use HTML tags for styling.

JavaScript or JS is an object oriented light weight language used for web page scripting by various online sites. The HTML document is a fully interpreted computer language allowing interactivity dynamically on web pages. In 1995, it was launched to add software to Netscape Navigator's web pages. All other graphical web browsers have been embraced since then. Users may construct contemporary web applications with JavaScript so that they can interact without refreshing the page at all times. Js is used in the conventional website for various sorts of easiness and interaction

Bootstrap is a very popular framework for constructing a responsive and mobile-friendly website for H.T.M.L., JavaScript and CSS. You may download and use it completely free of charge. A front end framework used to make web development easier and quicker. The design template for font, shapes, buttons, table, browsing, modalities, picture carousel and much more are included. The JavaScript plug-ins can also be used. It makes it easier for you to design responsively

Backend Technologies

Django is a web framework from Python which fosters speedy and clean, pragmatic development. Django helps you rapidly and with less code to develop better web apps. The following design philosophies are provided to Django. Loosely Coupled − Django strives to distinguish one part of its stack from the others. Less coding − There is less code, hence a fast development in turn. Do not repeat yourself (DRY) – All should be programmed in only one spot rather than again. The principle of rapid development Of Django is to do everything it makes development hyperfast. Django rigorously keeps its own code for clean design. Django supports the style of MVC like the most contemporary frameworks. Let's first explore what the Model View Controller (MVC) is, and then look at the Model View-Template specificity (MVT) pattern of Django. MVC Models, We commonly talk about MVC model while talking about apps that provide UIs (web or desktop) and as the name implies, the MVC design consists of three components: model, display and controller.

Model - it is class representing the table or relation in our DB and in which each class attribute is a table or collection field. App/models.py (in our case: Ecommerce/models.py) defines models.

View - it is the place where programmer place our application business logic. The view same as function used for the execution of certain business logic and the return of a user response. The HTML content of a webpage, redirect or 404 error might be this answer. Within the Django project's view.py file, all view functions are built.

Template - Django offers a practical approach to create dynamic H.T.M.L. pages using their template system. There are static components of the intended H.T.M.L. output and a unique syntax outlining how dynamic material is introduced.

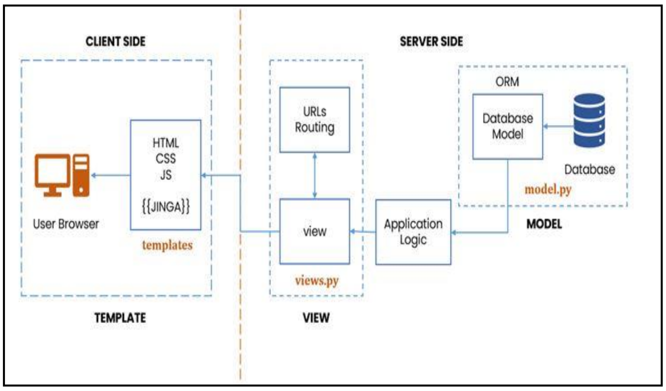


Fig-1: Working Of Django

5. PARTICIPANTS

There are three main stakeholders in an online auction:

- **Seller:** The individual or entity offering an item or set of items for sale. Their primary objective is to maximize or minimize the selling price of the item(s) they are offering**.**
- **Bidder** – A person or entity participating in the auction by submitting bids for the item(s) being sold. Bidders aim to secure the item at the lowest possible price that allows them to win the auction
- **Administrator** – The administrator is responsible for hosting the auction, managing the user, items, categories and report generation of payment. Admin also verifies the user and the items uploaded by the seller for the auction. The user which are verified from the admin then only they can participate in auction.

6. AUCTION TYPES

There are different types of auctions such as English, Dutch, etc. In this project we are have implemented both the type English auction as well as Dutch auction.

- **English Auction** – The English auction is an open bid, ascending-price auction in which bidders place competing bids against other bidders in order to purchase the auctioned item. When a given time expires, the highest bidder wins the auction and must pay an amount equal to the winning bid. This type of auction is often used in real estate. Many online auctions are modelled on the English auction, except that an auction finishes at a predetermined closing time.

- **Dutch Auction** – In a Dutch auction, the selling process is reversed compared to English auctions. Instead of starting with a low price and gradually increasing, the auctioneer begins with a high asking price, which then decreases until a buyer is willing to accept the price and purchase the item. This method is particularly useful in scenarios where the seller wants to quickly sell large quantities of items or securities. As the price decreases, potential buyers must decide whether to accept the current price or wait for further reductions. This creates a sense of urgency and strategic decision-making among participants. Dutch auctions are commonly used in financial markets for initial public offerings (IPOs) of securities and government bond auctions. They facilitate price discovery by allowing market demand to determine the final price, offering an efficient mechanism for sellers to offload their goods or securities.

7. MODELLING

The Data Base which we are using is SQLite. In this project's database following tables/models/relations are used:

- ➢ Users
- ➢ Categories
- ➢ Subcategories
- ➢ Product
- ➢ Participants History
- ➢ Winner Report
- ➢ Order History
- ➢ Payments Reports

8. SYSTEM ANALYSIS

Processor: At least later than Intel Atom processor.
Ram: At least $GB
Space: At least 2GB space is ideal to run.
Operating System: MacOS, Windows (Win 7 or later), or Linux.
Frontend: Html, CSS and JavaScript.
Backend: Django Python.
Data Base setup: SQLite is by default comes with Django.
Web Server: Django have inbuild run-server environment.

9. SYSTEM DESIGN

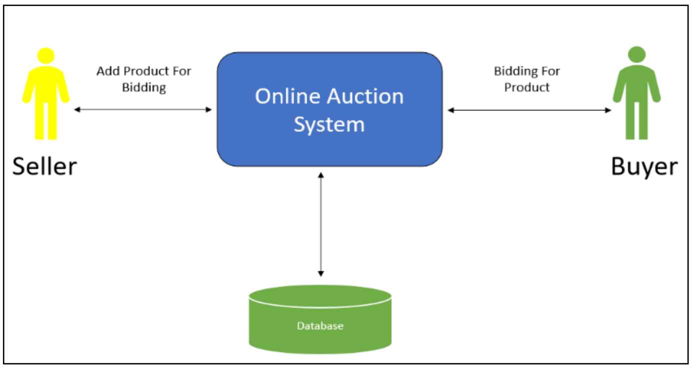


Fig-2: Block Diagram

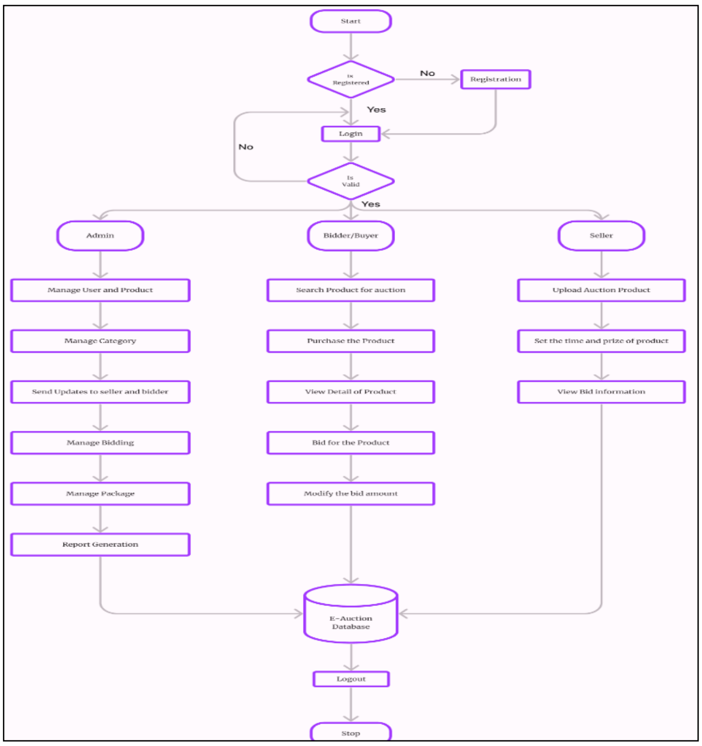


Fig-3: System Architecture

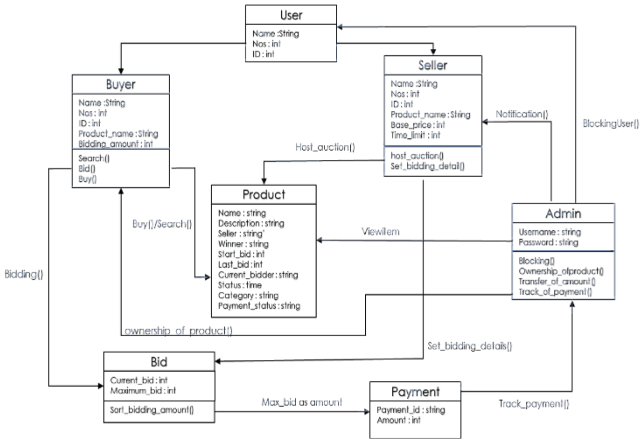


Fig-4: UML Class Diagram

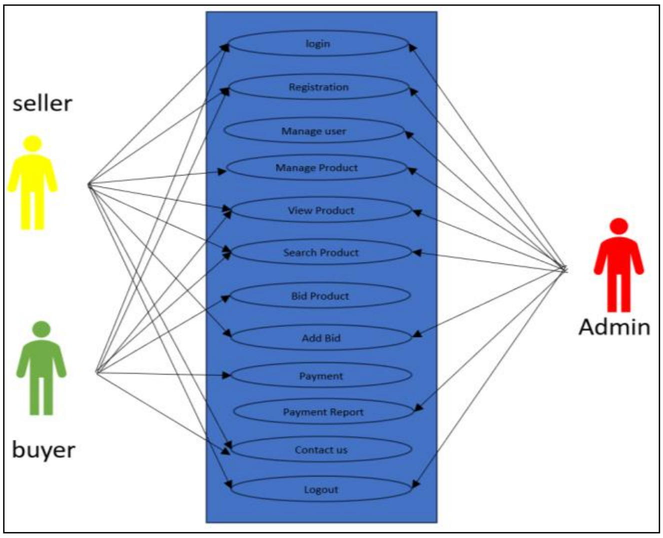


Fig-5: Use Case Diagram

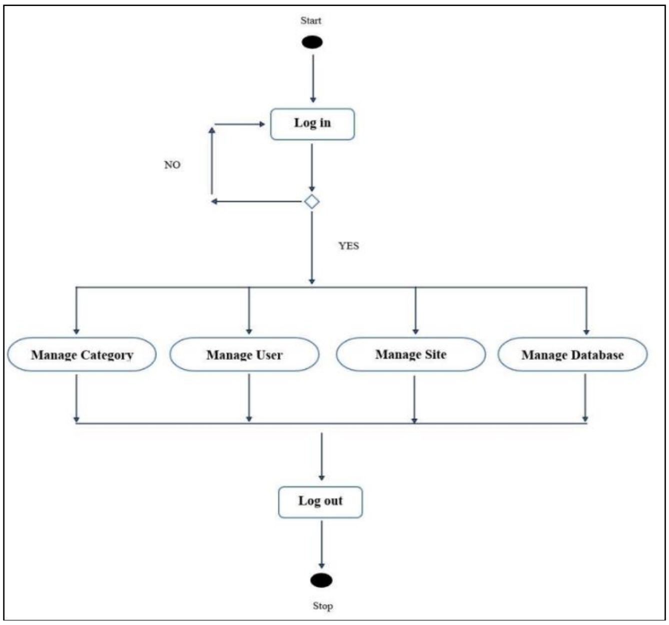


Fig-6: Activity Diagram for Admin

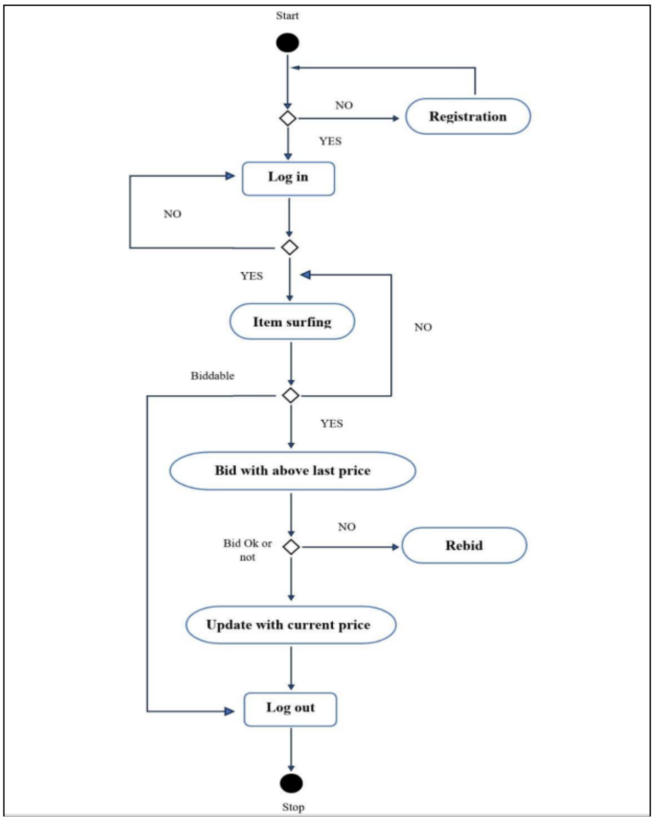


Fig-7: Activity Diagram for Buyer

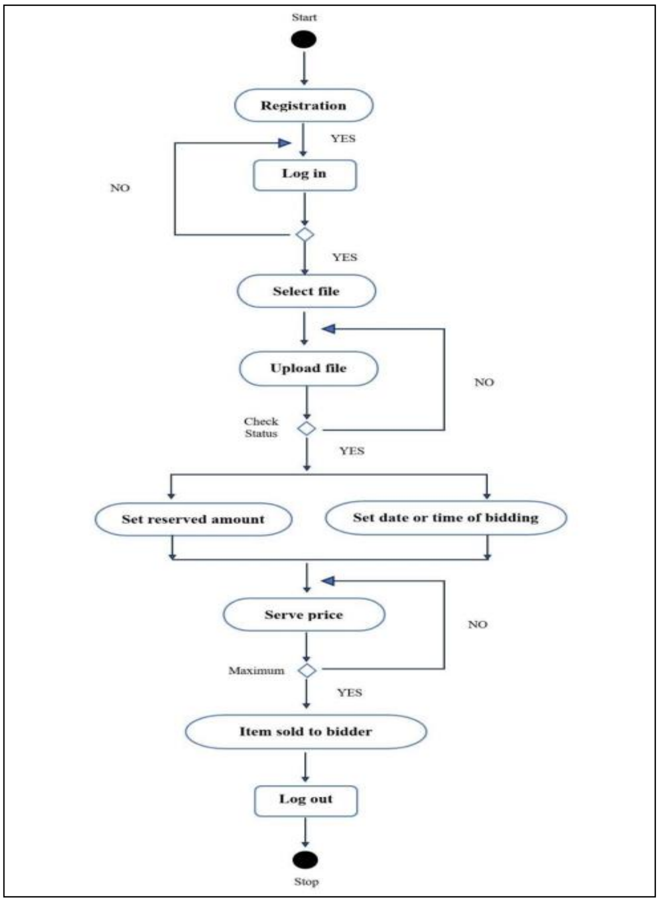
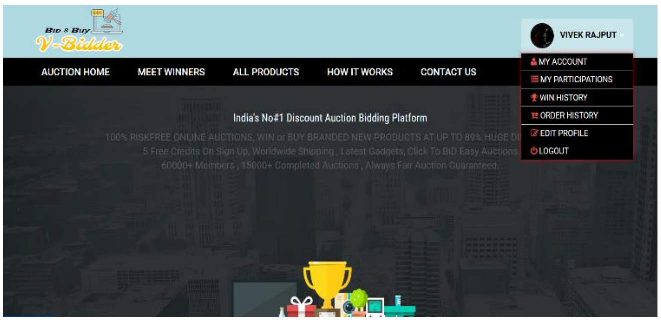


Fig-8: Activity Diagram for Seller

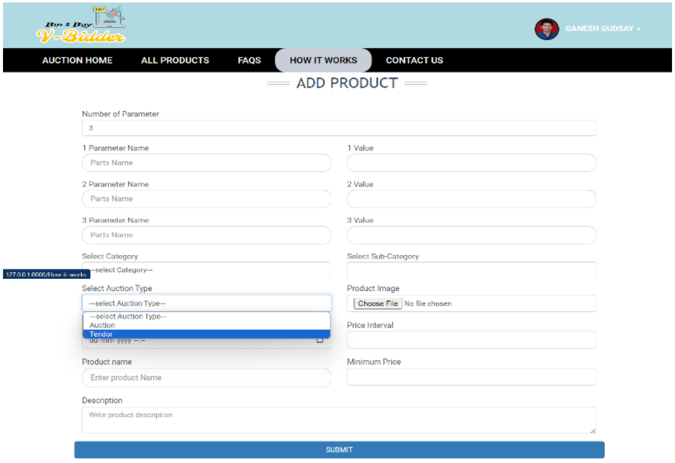
10. RESULT



India's No#1 Discount Auction Bidding Platform

100% RISKFREE ONLINE AUCTIONS, WIN or BUY BRANDED NEW PRODUCTS AT UP TO 89% HUGE DI
5 Free Credits On Sign Up, Worldwide Shipping , Latest Gadgets, Click To BID Easy Auctions
60000+ Members , 15000+ Completed Auctions , Always Fair Auction Guaranteed.

Fig-9: Bidder and Seller Homepage



ADD PRODUCT

Number of Parameter: 3

1 Parameter Name: Parts Name — 1 Value:

2 Parameter Name: Parts Name — 2 Value:

3 Parameter Name: Parts Name — 3 Value:

Select Category: --select Category-- — Select Sub-Category:

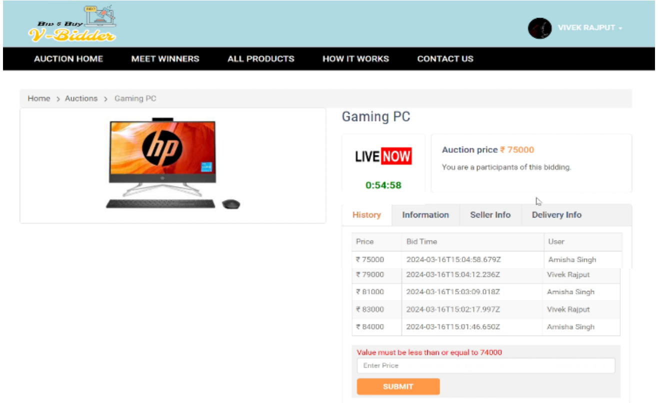
Select Auction Type: ---select Auction Type--- (---select Auction Type--- / Auction / Tendor) — Product Image: Choose File No file chosen

dd-mm-yyyy --:-- — Price Interval:

Product name: Enter product Name — Minimum Price:

Description: Write product description

Fig-10: Seller Adding Product for Bidding



Gaming PC

0:54:58

Auction price ₹ 75000

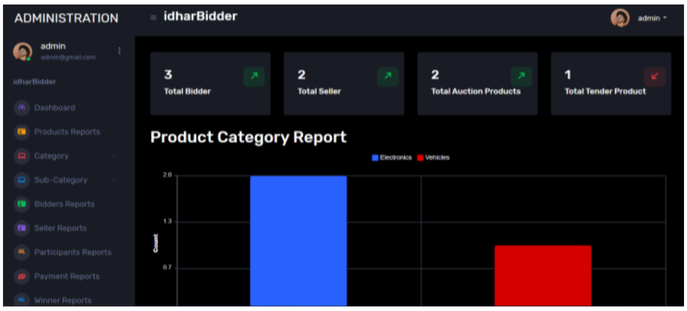
You are a participants of this bidding.

Price	Bid Time	User
₹ 75000	2024-03-16T15:04:58.679Z	Amisha Singh
₹ 79000	2024-03-16T15:04:12.236Z	Vivek Rajput
₹ 81000	2024-03-16T15:03:09.018Z	Amisha Singh
₹ 83000	2024-03-16T15:02:17.997Z	Vivek Rajput
₹ 84000	2024-03-16T15:01:46.650Z	Amisha Singh

Value must be less than or equal to 74000

Enter Price

Fig-11: Live Bidding Page



3 Total Bidder

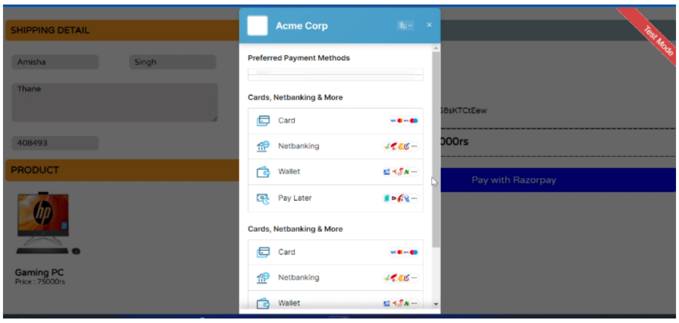
2 Total Seller

2 Total Auction Products

1 Total Tender Product

Product Category Report

Fig-12: Administrator Dashboard



Amisha Singh

Thane

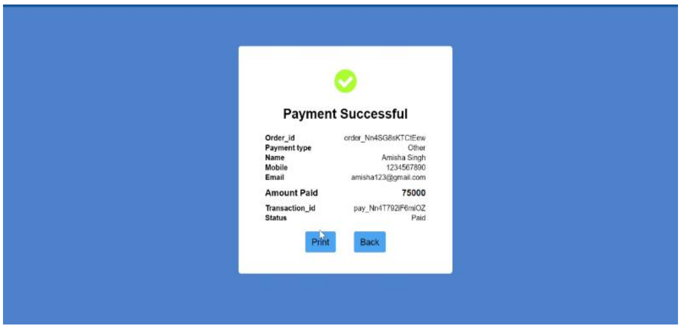
408493

Gaming PC
Price : 75000rs

GBsKTCtEew

000rs

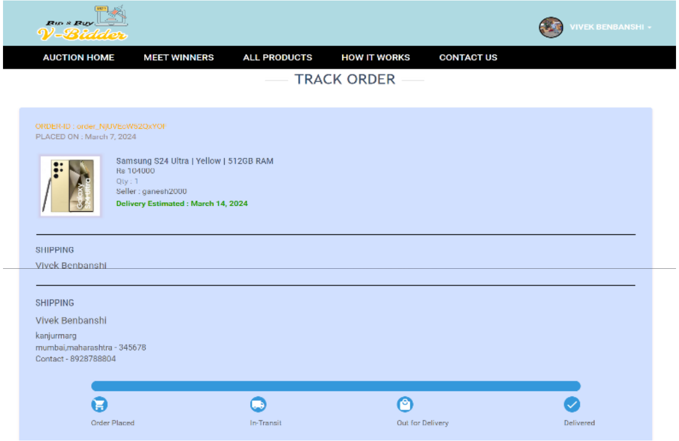
Fig-13: Razor Pay Payment Page



Payment Successful

Order_id	order_Nn4SG8sKTCtEew
Payment type	Other
Name	Amisha Singh
Mobile	1234567890
Email	amisha123@gmail.com
Amount Paid	75000
Transaction_id	pay_Nn4T792iF6miOZ
Status	Paid

Fig-14: Success Payment page



TRACK ORDER

ORDER-ID : order_NjUVEcW52QxYOF
PLACED ON : March 7, 2024

Samsung S24 Ultra | Yellow | 512GB RAM
Rs 104000
Qty : 1
Seller : ganesh2000
Delivery Estimated : March 14, 2024

SHIPPING
Vivek Benbanshi

SHIPPING
Vivek Benbanshi
kanjurmarg
mumbai,maharashtra - 345678
Contact - 8928788804

Fig-15: Tracking Page

11. CONCLUSIONS

In conclusion, the research study on the development of an online auction system using the Python Django framework showcases the effectiveness of Django in constructing a secure and efficient platform. The system delivers a user-friendly interface, robust security measures, and scalability to accommodate a substantial user base and concurrent auctions. Ultimately, the online auction system presents opportunities for individuals and businesses alike to engage in buying and selling activities, fostering a dynamic and thriving online marketplace.

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to our project guide. Mrs. Deepali Joshi Assistant professor, department of Computer Engineering for her valuable suggestions, cooperation, and support in the working of this paper.

REFERENCES

1. Sntosh Mittal, Dev Baghel, Shivam Roy, " Platform For E-Auction" international journal of progressive research in engineering management and science (IJPREMS) vol. 01, issue 02 pp. 6–10, Nov 2021.
2. Vivek Kumar, Vishal Sharma, Shrestha Jaiswal, Faran khanJ, Building a secure and Efficient "Auction System Using Python-Based Django Technology" ,International Research journal of Modernization in engineering technology and science vol. 05, issue 05, May 2023
3. Nazim Majadi, Jarrod Trevathan, Neil Bergmann, uAuction: Analysis, design and implementation of a secure online auction system, 2016 IEEE, 978-1-5090-4065-0/16
4. Auction System Software . Available : https://www.bidderboy.com/.
5. Akshansh Sharma, Firoj Khan, Deepak Sharma, Dr. Sunil Gupta, "Python: The Programming Language of Future", IJIRT, vol. 6 issue 12, paper id 149340, 2020.
6. Python's Documentation: Python_Software_ Foundation (2001). Python documentation. https://docs.python.org/3/.
7. For HTML, CSS and other front end tools: W3Schools (1999). Referred for learning and enhancing front end technologies. https://www.w3schools.com/default.asp
8. Django's Documentation: Django_Software_ Foundation and individual_contributors (2005). Documentation page of Django framework. https://docs.djangoproject.com/en/3.2/.
9. For the Payment Gateway integration, Rayzor Pay API, https://razorpay.com/docs/payments/payment-gateway/web-integration/standard/