



શ્રી સ્વામિનારાયણ ગુરુકુલ રાજકોટ સંસ્થાન

શાસત્રી સ્વામી શ્રી ધર્મજીવનદાસજી

**સાયન્સ & IT ગુરુકુલ કોલેજ**  
ગુરુકુલ કેમ્પસ, કોલેજ રોડ, જૂનાગઢ

# Online Auction

Auction platform where user  
can bid in auction and submit

#### Project Partners:

MR. MADHAV R. GEDIYA, BCA-6<sup>TH</sup>

MR. JENISH N. DUVANI, BCA-6<sup>TH</sup>

:: Submitted to ::

BKNM University,  
Junagadh

:: GUIDED BY ::

Mr. Ripal V. Pandya

Mr. Millind K. Anandpara



શ્રી સ્વામિનારાયણ ગુરુકુલ રાજકોટ સંસ્થાન

શાસ્ત્રી સ્વામી શ્રી ધર્મજીવનદાસજી

# સાયન્સ & IT ગુરુકુલ કોલેજ



ગુરુકુલ કેમ્પસ, કોલેજ રોડ, જૂનાગઢ

(Affiliated to **Bhakta Kavi Narsinh Mehta University, Junagadh**)

# Project Completion Certificate

This certificate is awarded to

GEDIYA MADHAV RAJESHBHAI

BCA-6-2024

DUVANI JENISH NILESHBHAI

BCA-6-2024

in completion of project work

PYTHON (DJANGO)

19/12/2023

SQL LITE

11/03/2024

Mr. Ripal Pandya

Mr. Milind Anandpara

Project Guide

Director

A

PROJECT REPORT ON

## Online Auction

**Submitted in Fulfillment of Requirements**

**For Completion of Semester - 6 in**

**Bachelor of computer application**

**Year 2024**

**To**

**SHASHTRI SWAMI SHREE DHARMAJIVANDASJI**

**INSTITUTE OF INFORMATION TECHNOLOGY**

**JUNAGADH**

**Guided By:**

**Mr. Ripal Pandya**

**Mr. Millind Anandpara**

**Prepared By:**

**Mr. Madhav R. Gediya**

**Mr. Jenish N.Duvani**

## **PREFACE**

The web-enabled system has two different modules.  
They are as under.

**Client-side**

**Admin side**

Each of the two modules has its own functionality. In the client side section user can View Site. User can get information about auction and bidding on it .

In admin side module administrator are the user who manages the whole system and all data of this website. He has the database control.

Hear our website on Our Online Auction website. User can get to know about our Auction and its Bidding Lineup. At the time of visiting our site we can see all categories with good resolution. And we also provides the in depth description of each and every product.

The customers here can register for their products to be auctioned or the user can bid on the auction available on our site.

## **ACKNOWLEDGEMENT**

We are very thankful to all whose have helped in preparing this project. We are feeling a great happiness to present this website project. First of all we would like to thank “**BKNM University**” who give me an opportunity to give a chance to prepare a project.

Before we get in to thick of the things we would to add a few heartfelt words for the people who were part of this project numerous ways, people who give unending support right from the stage project ideas was conceived. In particular we would like to thank **Mr.Ripal Pandya** who has always inspired us and has directed us towards the successful completion of our project. They have been the guided through the project and their encouragement has left me indebted to them.

We are very thankful to the **Director Sadhu RushikeshdasjiSwami** and the **Asst. Director Mr. Rajesh Bharad** of **Shastri Swami Shree Dharmajivandasji Institute of Information Technology – Junagadh.**

We are also thankful to our classmate and few other people who helped us directly or indirectly in solving problem and in making our web development project more efficient and attractive.

Thank you...

**Date:**

Mr. Madhav R. Gediya

**Place:** S.S.S.D.I.I.T - JND

Mr. Jenish N. Duvani

# I N D E X

<b>NO</b>	<b>Particulars</b>	<b>Page No</b>
1	Project Profile	1
2	Use of System Development Life Cycle Model	2
3	Feasibility Study	5
3	Requirement Gathering	8
	<b>Requirement Analysis</b>	10
	1) Hardware and Software Requirement	
	2) Front - End Tools	
	3) Back - End Tools	
	4) Other Tools & Technology Used	
4	Project Abstracts (User Roles & Capabilities)	14
5	Proposed System	16
6	Advantages & Limitations of Proposed System	18
7	Evaluative Report Using Pert Chart and Gantt Chart	20
8	<b>Data Flow Diagram</b>	23
	1) Context level	
	2) 1st Level	
	3) 2nd Level	
9	Use Case Diagram	27
10	Flow Chart	28
11	Cost Estimation	29
12	Data Dictionary	31
13	Screen Layouts	35
14	Special Utilities	53
15	Testing	54
16	Test Case	57
17	Implementation	59
18	Bibliography	61

## PROJECT PROFILE

<b>Project Title</b>	Online Auction Platform
<b>Project Description</b>	Online Auction: Elevate your auction experience. Register to showcase products or bid on treasures in live auctions. Secure success with the highest bid.
<b>Front End</b>	HTML,CSS and Javascript
<b>Back End</b>	Python(Django), SQLite
<b>Other Tools</b>	Bootstrap
<b>Guide</b>	Mr. Ripal Pandya
<b>Submitted To</b>	S.S.S.D.I.I.T College

## **USE OF SYSTEM DEVELOPMENT LIFE CYCLE MODEL**

- Software Development Life Cycle (SDLC) is a process for development of software. There are some steps to follow to create a software application.
- Following are the different Life Cycle Model example.
- Waterfall model
- Iterative waterfall model
- Prototyping model
- Evolutionary model
- Spiral model
- R.A.D. model (Rapid Application Development)

Our project follows the Waterfall model, a well-established software development life cycle (SDLC) approach known for its structured and sequential nature. In the Waterfall model, each phase of development is completed before moving on to the next, making it particularly suitable for projects with stable and well-defined requirements, where changes during development are minimal.

## **Waterfall Model :-**

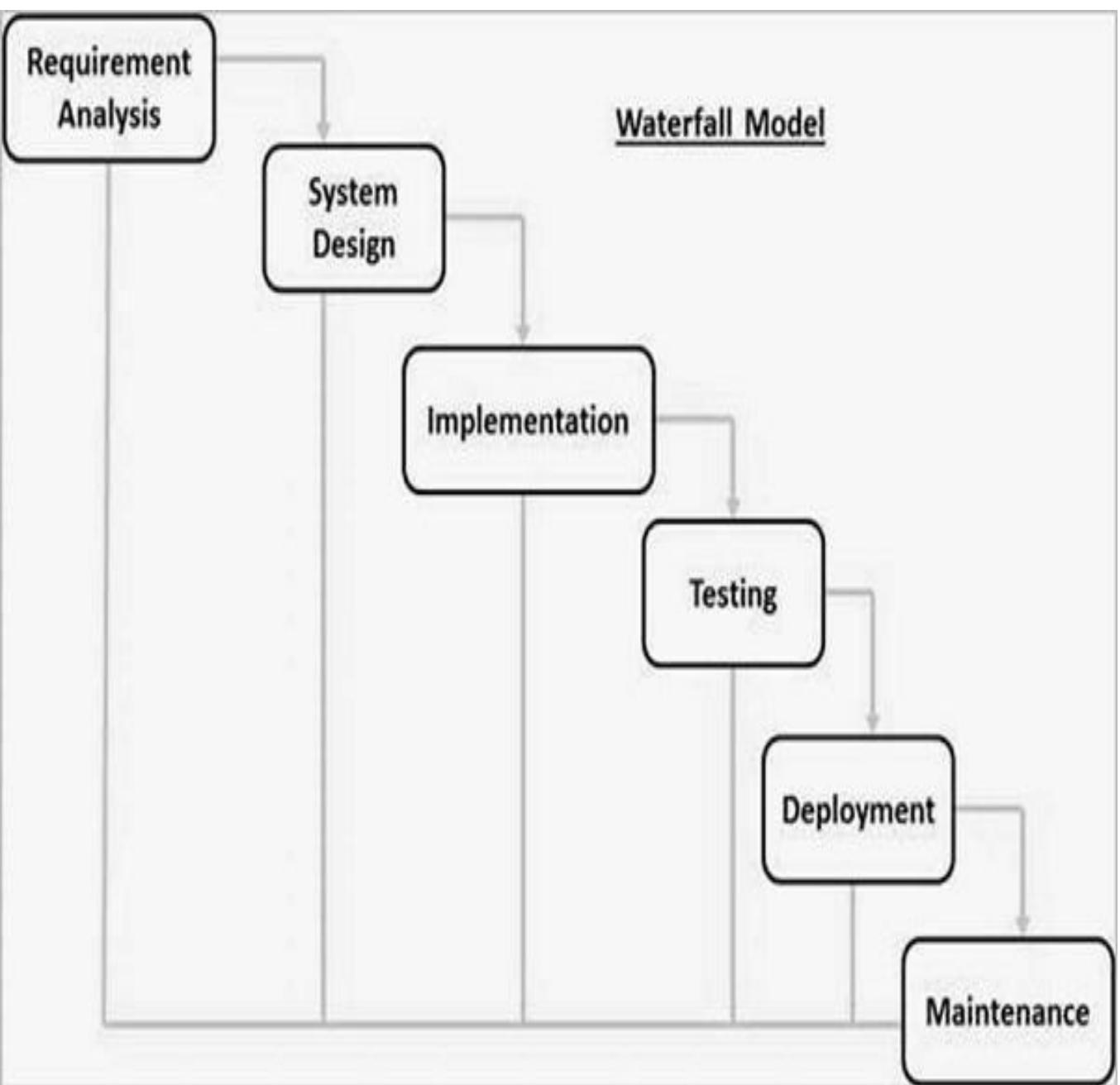
In this project I have used the Waterfall model. waterfall modalworks well for smaller and low budget projects where requirements are very well understood. Clearly defined stages and well understood. It is easy to arrange tasks. Process and results are well documented.

The waterfall model is the first model in software development history. It is also referred to as the (Linear sequential life cycle model) this model is simple to understand and use. In the waterfall model, each phase is completed before moving to the next phase. The waterfall model is the first and earliest approach used in SDLC for the development of software products in this model that do not overlap. Waterfall Model Design The waterfall approach was the first approach in the software development life cycle and was widely used in the development of software products and for the success of the project. Requirement Analysis System Design Implementation Testing Deployment Maintenance The sequential phases of waterfall model. Requirement Analysis. All the key requirements of the system development and captured in this phase and we document them in the software requirement specification document.

The waterfall model is most commonly used in software engineering and product development, less often – in other projects and industries. Employ the waterfall model only if your project meets the following criteria .All the requirements are known, clear, and fixed.

The classical waterfall model divides the life cycle into a set of phases. This model considers that one phase can be started after the completion of the previous phase. That is the output of one phase will be the input to the next phase. Thus the development process can be considered as a sequential flow in the waterfall. Here the phases do not overlap with each other.

## **The diagram of Waterfall Model :-**



## **FEASIBILITY STUDY**

The feasibility study for our online auction platform developed using Django, Pandas for image management, Bootstrap, and JavaScript with an SQLite database, aims to gauge the project's viability and effectiveness in meeting user needs while optimizing resource usage.

### **TECHNICAL FEASIBILITY:**

Technical feasibility revolves around the existing technology infrastructure and its ability to support the proposed system. Online Auction utilizes the following technologies:

- **Web Framework:** Django - A high-level Python web framework for rapid development.
- **Image Management:** Pandas - A data manipulation library for efficient image handling.
- **Front-end:** Bootstrap and JavaScript - Ensuring an intuitive and responsive user interface.
- **Database Management System:** SQLite - A lightweight, embedded database for efficient data storage.

The compatibility and synergy between these technologies contribute to the technical feasibility of Online Auction, offering a robust and dynamic online auction platform.

## **ECONOMIC FEASIBILITY:**

Evaluating the economic feasibility of Online Auction involves more than just estimating project costs. It includes a thorough analysis of potential benefits, conducting a cost-benefit analysis, understanding financial implications, budgeting, and identifying associated risks. While the primary goal may not be monetary gains, a clear understanding of these financial aspects is essential for the project's overall success.

This assessment entails:

- 1. Cost Estimation:** Carefully estimating expenditures for software development, server hosting, personnel, and unforeseen expenses.
- 2. Potential Benefits:** Identifying non-monetary benefits such as enhanced user engagement and increased brand visibility.
- 3. Cost-Benefit Analysis:** Quantifying tangible and intangible benefits to assess the overall value proposition of the project.
- 4. Budgeting:** Developing a robust financial plan that outlines resource allocation throughout the project's lifecycle.
- 5. Risk Assessment:** Identifying and assessing potential risks related to economic factors, ensuring a proactive approach to financial stability.

In summary, the economic feasibility study for Online Auction provides a focused examination of its financial landscape, aligning considerations with the project's broader goals for sustainable success.

## **OPERATIONAL FEASIBILITY:**

Operational feasibility assesses the effectiveness of implementing Online Auction in the current operational context. Key considerations include:

- 1. Resource Availability:** Evaluate the availability of necessary resources, including hardware, software, and personnel, compatible with Django, Pandas, Bootstrap, and JavaScript.
- 2. Skills and Expertise:** Ensure the project team possesses the required skills, addressing any gaps related to Django, Pandas, Bootstrap, and JavaScript.
- 3. Integration with Existing Systems:** Evaluate how Online Auction will integrate with existing systems, ensuring smooth data flow and compatibility with SQLite.
- 4. Scalability:** Analyze the project's ability to scale with potential growth in users or data volume, planning for scalability to avoid constraints.
- 5. Compliance and Security:** Ensure Online Auction complies with relevant regulations and security standards, especially concerning user data and privacy, within the context of Django and SQLite.

**By conducting a thorough feasibility study, Online Auction aims to affirm its practicality, economic viability, and seamless integration within the current operational environment, utilizing Django, Pandas, Bootstrap, JavaScript, and SQLite to deliver a powerful and user-friendly online auction platform.**

## **REQUIREMENT GATHERING**

### **1) What is the primary objective of Online Auction, your online auction platform?**

The primary objective of Online Auction is to provide users with a seamless and secure online marketplace for participating in auctions, allowing them to bid on products, register items for auction, and experience the excitement of winning through competitive bidding.

### **2) Could you specify the types of products that users will be able to bid on?**

Online Auction will facilitate the auction of a diverse range of products, including electronics, fashion items, collectibles, art pieces, and more.

### **3) Will there be specific categories or collections to organize the auctioned products?**

Yes, Online Auction plans to categorize auctioned items into collections, such as electronics, fashion, art, and other relevant categories, to enhance user navigation.

### **4) Will users have the ability to filter and search for auctioned products based on criteria like starting bid, category, and closing time?**

Yes, Online Auction will implement a robust filtering and search system, allowing users to find products based on criteria like starting bid, category, and closing time, enhancing the user experience.

**5) Is user registration and authentication necessary for participating in auctions?**

Yes, user registration and authentication are integral for secure bidding, tracking auction history, and ensuring a trustworthy online auction environment.

**6) How will the bidding process work for users?**

Online Auction aims to provide a straightforward bidding process, allowing users to place bids on their desired items, with the highest bidder winning the auction.

**7) Tell me more about the backend admin panel. What functionalities and features do you envision for the admin panel?**

The admin panel should offer capabilities for managing product listings, overseeing ongoing auctions, monitoring user activity, handling customer support inquiries, and ensuring the overall smooth operation of the platform.

**8) Do you plan to incorporate any communication features, such as a messaging system between buyers and sellers?**

No, Online Auction will primarily focus on the bidding aspect, and communication features beyond essential notifications will not be implemented.

**9) How do you plan to showcase auctioned products, including high-quality images and detailed descriptions?**

Professional images and comprehensive product descriptions will be used to showcase auctioned items, ensuring that users can make informed bidding decisions.

**10) Have you considered implementing any additional features, such as user feedback or promotions, to enhance the user experience and engagement?**

Currently, Online Auction is not considering additional features like user feedback or promotions, but future enhancements will be explored based on user needs and feedback.

## **REQUIREMENT ANALYSIS**

The Requirement Analysis phase for Online Auction, an online auction platform, is a pivotal step in outlining the functional and non-functional needs essential for the successful development and deployment of the platform. This analysis aims to address the requirements of both administrators and users while ensuring a seamless and secure online auction experience.

### **Requirement Prioritization:**

During the requirement prioritization process, we distinguish between essential (must-have) and desirable (nice-to-have) requirements to focus on core functionalities.

### **Essential Requirements (Must-Have):**

#### **1. User Registration and Authentication:**

- **Description:** Users must be able to create accounts, log in, and secure their personal information.
- **Details:** Security and user identity are paramount in the online auction environment.

#### **2. Product Listings and Auctions:**

- **Description:** The platform should display a diverse range of products available for auction.
- **Details:** Product listings must include clear images, descriptions, starting bid, and auction closing time.

#### **3. Bidding Process:**

- **Description:** Users should be able to place bids on auctioned items.
- **Details:** A straightforward and transparent bidding process is crucial for user engagement.

## **Functional Requirements:**

Functional requirements outline specific functions and features that the system must perform to meet user needs. Each requirement is clear, unambiguous, and testable.

### **1. User Registration and Authentication:**

- **Description:** Users can create accounts and sign in.
- **Details:** User data security is a top priority.

### **2. Product Listings and Auctions:**

- **Description:** The platform categorizes products and displays detailed information.
- **Details:** Product images, descriptions, starting bid, and auction closing time should be included.

### **3. Bidding Process:**

- **Description:** Users can place bids, view current bids, and receive notifications on bid status.
- **Details:** Ensuring a transparent and competitive bidding environment.

## **Non-Functional Requirements:**

Non-functional requirements focus on system attributes like performance, security, usability, and scalability.

### **1. Performance:**

- **Requirement:** The website should load auction pages within 2 seconds.
- **Details:** Fast loading times enhance the user experience and engagement.

### **2. Security:**

- **Requirement:** User data and transactions must be encrypted and stored securely.
- **Details:** Ensuring user privacy and financial information protection.

### **3. Usability:**

- **Requirement:** The user interface should be intuitive and responsive across devices.
- **Details:** A user-friendly design enhances user engagement and satisfaction.

### **4. Scalability:**

- **Requirement:** The system must handle a substantial increase in users and auctioned products.
- **Details:** Ensures the platform can accommodate growth in user traffic and product offerings.

### **Use Cases/User Stories:**

Use cases and user stories describe how users will interact with the system, providing scenarios and acceptance criteria.

#### **1. User Registration Use Case:**

- **Description:** A new user wants to create an account on the platform.

- **Actors:** User, System

- **Scenario:**

- User clicks on the "Sign Up" button.

- User fills out the registration form.

- User receives a verification email.

- **Acceptance Criteria:**

- User should be able to log in after registration.

#### **2. Bidding User Story:**

- **Description:** A user wants to place a bid on an auctioned item.

- **Scenario:**

- User logs into their account.
- User selects the desired product.
- User places a bid on the item.
- User receives bid status notifications.

- **Acceptance Criteria:**

- User should receive real-time bid status updates.

**This Requirement Analysis sets the foundation for the development of Online Auction, ensuring a user-centric, secure, and scalable online auction platform.**

## PROJECT ABSTRACT

### **Online Auction: Your Gateway to Dynamic Online Auctions**

Online Auction is a cutting-edge online auction platform designed to revolutionize the bidding experience. With a commitment to providing users with an exhilarating and secure bidding environment, Online Auction stands as the go-to destination for those seeking unique products through competitive bidding.

#### **Key Features:**

- 1. Diverse Auctioned Products:** Explore a wide array of products, from electronics and fashion items to collectibles and art pieces, all available for competitive bidding.
- 2. Intuitive Bidding Interface:** Our user-friendly interface ensures a seamless bidding process, allowing users to place bids effortlessly and engage in the thrill of competitive auctions.
- 3. User Registration and Authentication:** Secure and straightforward user registration and authentication processes, prioritizing user identity protection and ensuring a trustworthy bidding environment.
- 4. Flexible Payment Options:** Currently offering 'cash on delivery,' Online Auction aims to integrate online payment gateways in the future for secure transactions, providing flexibility to users.

#### **Vision:**

Our vision for Online Auction is to redefine the online bidding landscape by creating a platform where every bid is not just a transaction but an experience. We aspire to be the preferred destination for users seeking excitement in competitive bidding, offering a wide range of products and ensuring a secure and user-friendly environment.

Imagine a platform where users can bid on diverse products, from electronics to art pieces, and experience the joy of winning through competitive auctions. Online Auction is not just about products; it's about creating memorable experiences in the world of online auctions, where every bid counts and every win is celebrated. Welcome to Online Auction, where bidding meets excitement!

## **PROPOSED SYSTEM**

### **Online Auction: Elevating the Online Auction Experience**

The proposed system for Online Auction revolves around creating a dynamic and engaging online auction platform. By seamlessly integrating user interactions, robust bidding functionalities, and efficient management tools, Online Auction aims to redefine the online auction landscape.

#### **Key Features:**

##### **1. User-Centric Front-End Auction Platform:**

- Diverse Product Auctions:** Showcase a broad spectrum of products, from electronics and fashion to unique collectibles and art pieces.
- Intuitive User Interface:** Ensure a user-friendly interface for easy navigation, bidding, and participation in auctions.

##### **2. User Accounts and Profiles:**

- User Registration:** Allow users to create accounts securely.
- Profile Management:** Enable users to personalize their profiles, track bidding history, and manage account settings.

##### **3. Efficient Bidding System:**

- Live Bidding Interface:** Implement a live bidding system for real-time interaction during auctions.
- Bidding Controls:** Provide users with easy-to-use bidding controls, allowing them to place bids, track auctions, and manage their bids.

##### **4. Detailed Product Pages:**

- **Comprehensive Product Information:** Showcase each auctioned item with high-quality images, detailed descriptions, starting bids, and current bids.
- **User Reviews:** Enable users to provide feedback and reviews on products.

## **5. Administrator Dashboard:**

- **Product and Auction Management:** Administer product listings, create, edit, or remove auctions, and manage inventory levels.
- **Order Processing:** Efficiently process and manage customer orders, track shipment statuses.
- **User Management:** Administer user accounts, address account-related issues, and monitor user activity.

## **6. Security and Data Management:**

- **Secure Transactions:** Implement robust security measures to ensure encrypted transactions and protect user data.
- **Privacy Compliance:** Adhere to privacy regulations to safeguard user information.

## **Vision:**

**Online Auction** envisions becoming a leading online auction platform, setting new standards for user engagement, product variety, and bidding excitement. By integrating cutting-edge technologies and efficient management tools, we aim to create an immersive online auction experience where users can discover, bid, and win unique products. Welcome to Online Auction, where every bid is an opportunity!

## **ADVANTAGES & LIMITATIONS OF PROPOSED SYSTEM**

### **Advantages of the Proposed System:**

#### **1. Dynamic Auction Experience:**

- The proposed system offers a dynamic and engaging online auction experience, enticing users with a variety of products and the thrill of competitive bidding.

#### **2. Efficient Bidding Controls:**

- Users benefit from an intuitive bidding interface, providing them with efficient controls to place bids, track auctions, and manage their bidding strategies seamlessly.

#### **3. User-Centric Front-End:**

- The user-centric design of the front-end ensures an accessible and enjoyable experience, making it easy for users to register, explore auctions, and participate in the bidding process.

#### **4. Live Bidding Interaction:**

- The live bidding interface allows users to engage in real-time bidding, enhancing the excitement and interactivity during auctions.

#### **5. Detailed Product Information:**

- Comprehensive product pages with high-quality images, detailed descriptions, starting bids, and current bids provide users with all the information needed for informed bidding decisions.

### **Limitations of the Proposed System:**

## **1. Initial Development Costs:**

- Building and implementing the proposed system may involve significant upfront development costs, including the integration of live bidding features and a user-friendly interface.

## **2. Maintenance Complexity:**

- Ongoing efforts and resources will be required to keep the platform up-to-date, manage auction listings, and ensure the overall smooth operation of the system.

## **3. Security and Privacy Challenges:**

- Ensuring data security and protecting against potential cyber threats and breaches is a continuous challenge, requiring constant vigilance to maintain a secure online auction environment.

## **4. User Onboarding Efforts:**

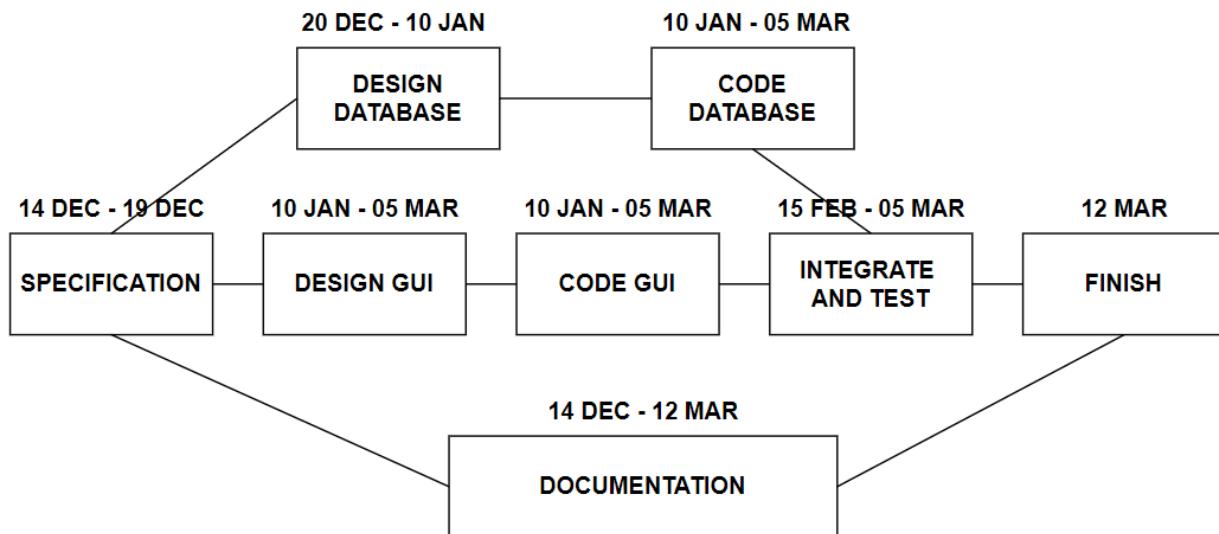
- Encouraging users to register, actively participate in auctions, and adopt the platform may require effective marketing strategies and user onboarding efforts to enhance user adoption.

## **5. Moderation and Content Quality:**

- Without a comprehensive admin panel, moderation of user interactions and content quality may pose challenges. Implementing effective moderation strategies will be essential to ensure a positive user experience.

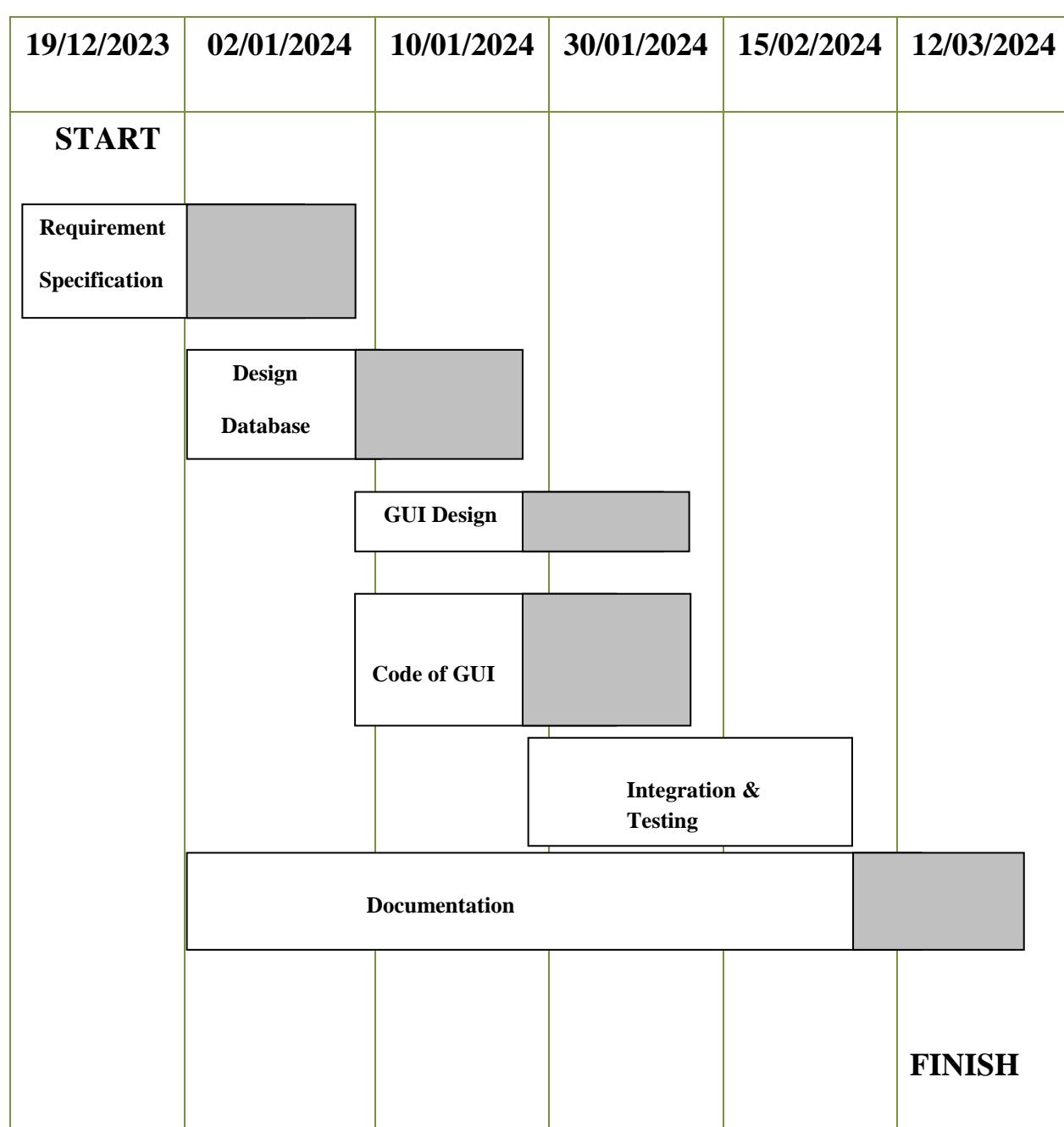
## PERT CHART AND GANTT CHART

- PERT (Project Evaluation and Review Technique) charts consist of a network of boxes and arrows. The boxes represent activities and the arrows represent task dependencies. PERT charts are a more sophisticated form of activity chart. Where instead of making a single estimate for each task, pessimistic, likely and optimistic estimates are made. The boxes of PERT charts are usually annotated with the pessimistic, likely, and optimistic estimates for every task. There are thus not one but many critical paths, depending on the permutations of the estimates for each task. This makes analysis of critical path show by using shaded boxes. The PERT chart representation of the MIS problem of show follows.



## GANTT CHART :-

Gantt chart can be derived automatically from PERT charts. However, PERT charts cannot be automatically derived from Gantt charts because PERT charts incorporate additional information about the time when an engineer doses a task. Also, parallel activities in a project can be easily identified using a PERT chart.



## **DATA FLOW DIAGRAM**

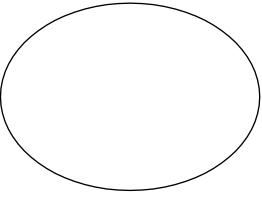
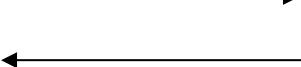
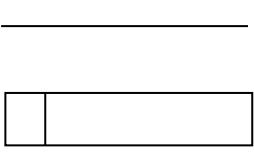
### **Detailed Life Cycle of Project:**

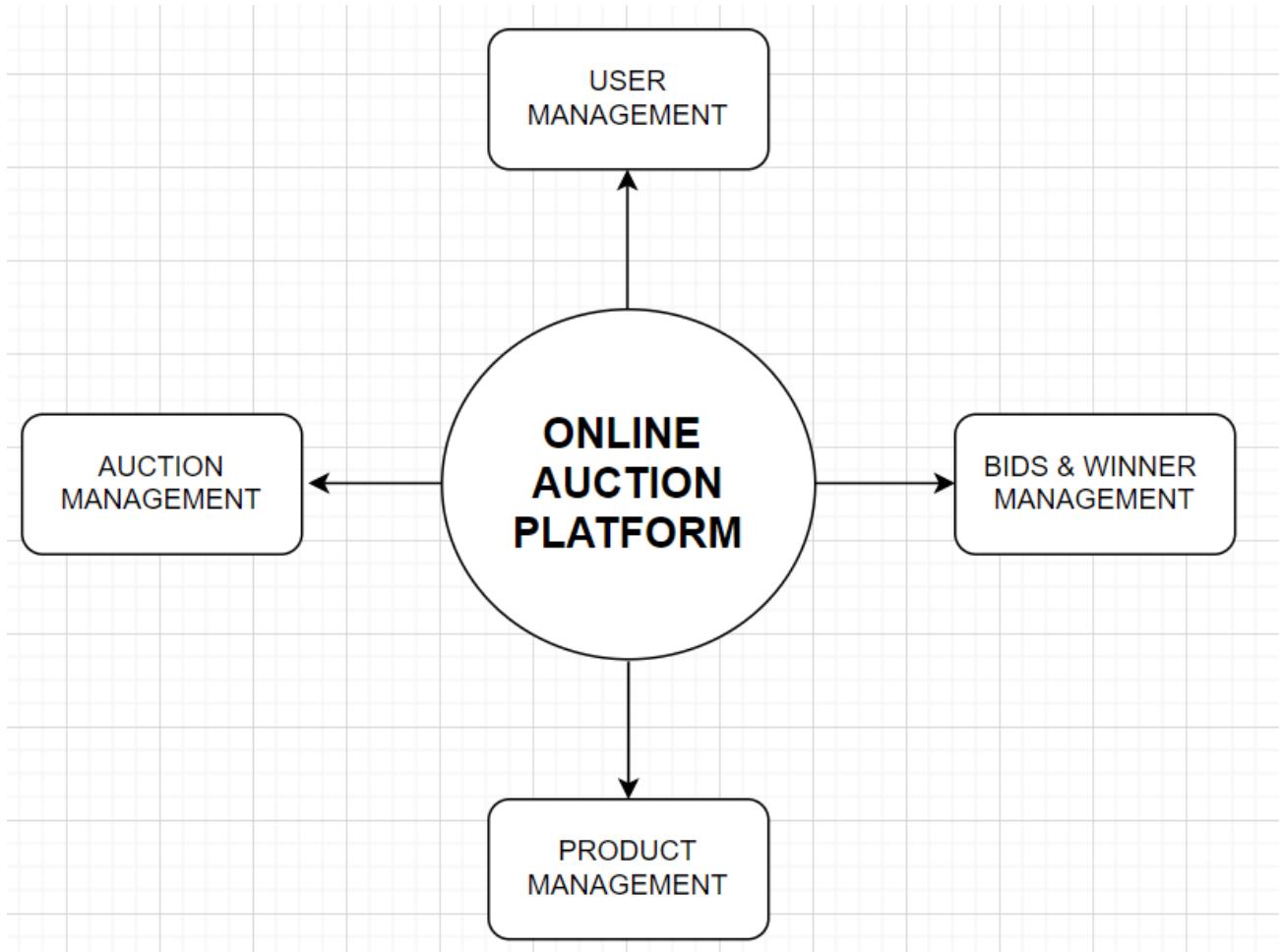
In the discussion of "Detailed Life Cycle of Project" we have to concentrate on DFD (Data Flow Diagram). Here we have work on it while developing this software project.

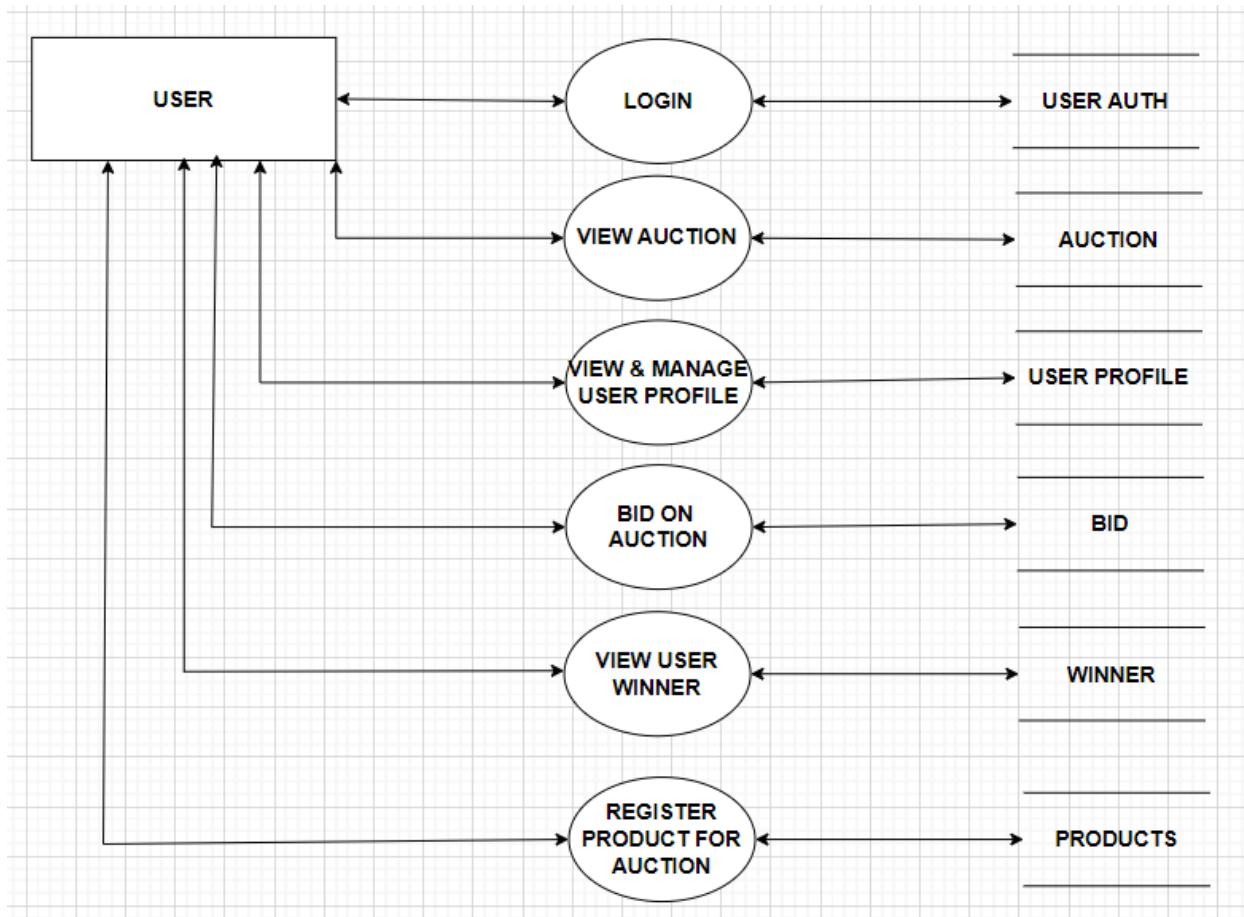
### **DFD (Data Flow Diagram):**

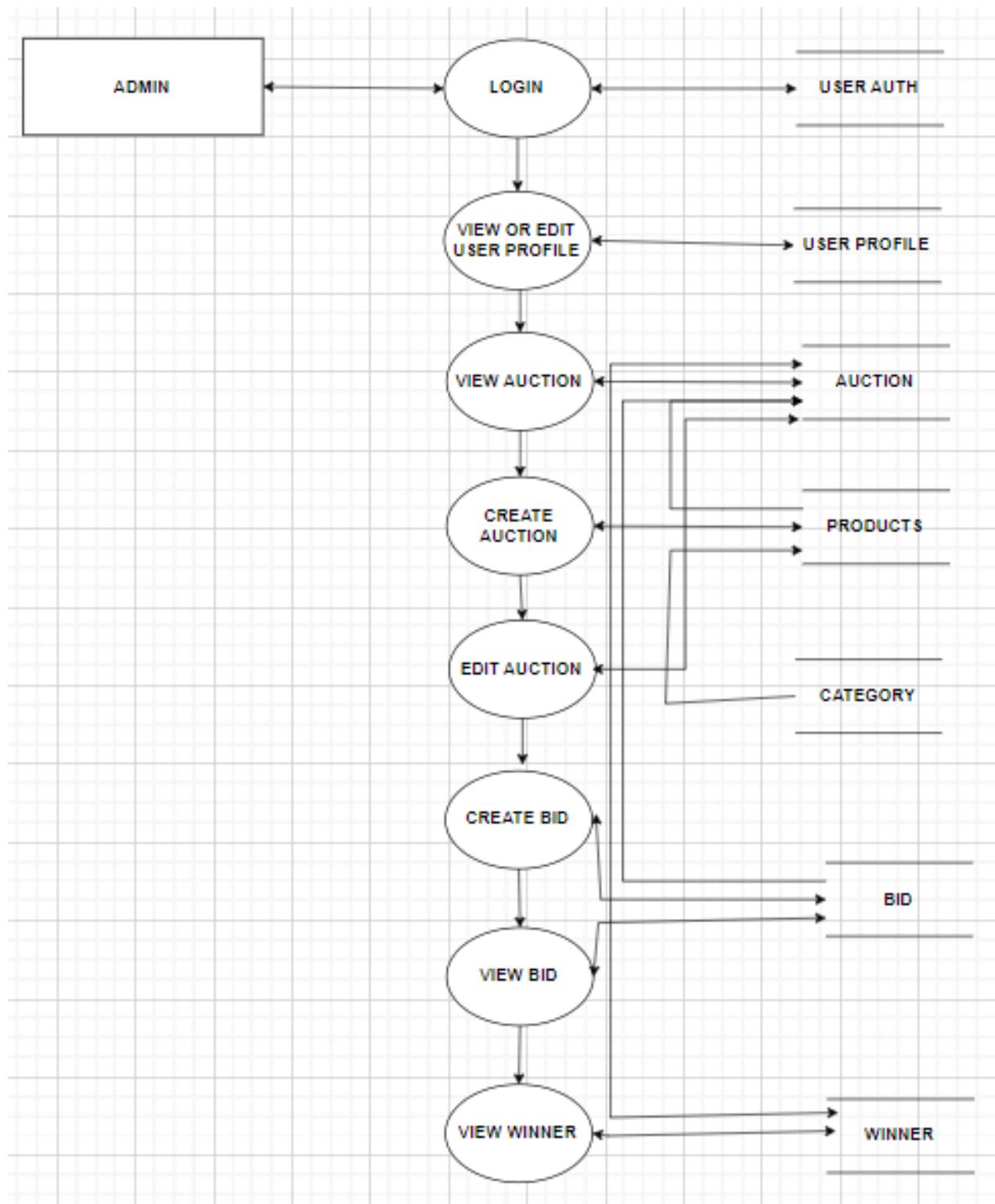
DFD is a graphical view of all system processes and transactions. With the DFD an End-User also can easily understand the system in a short time period. Also it is useful to find out problems or any complications with the system we are going to develop. We can easily get that whether we have understood the system as per the requirements of the customer or not by showing them this diagram. Thus DFD is a necessary phase while developing software.

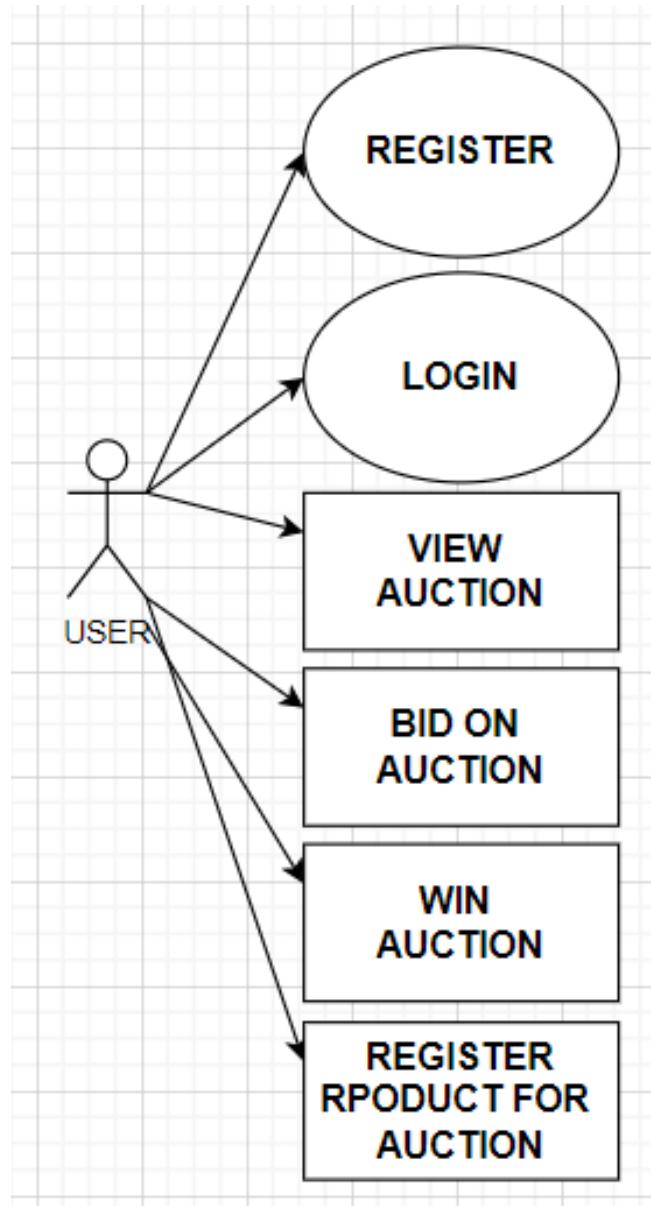
**For Understanding :-**

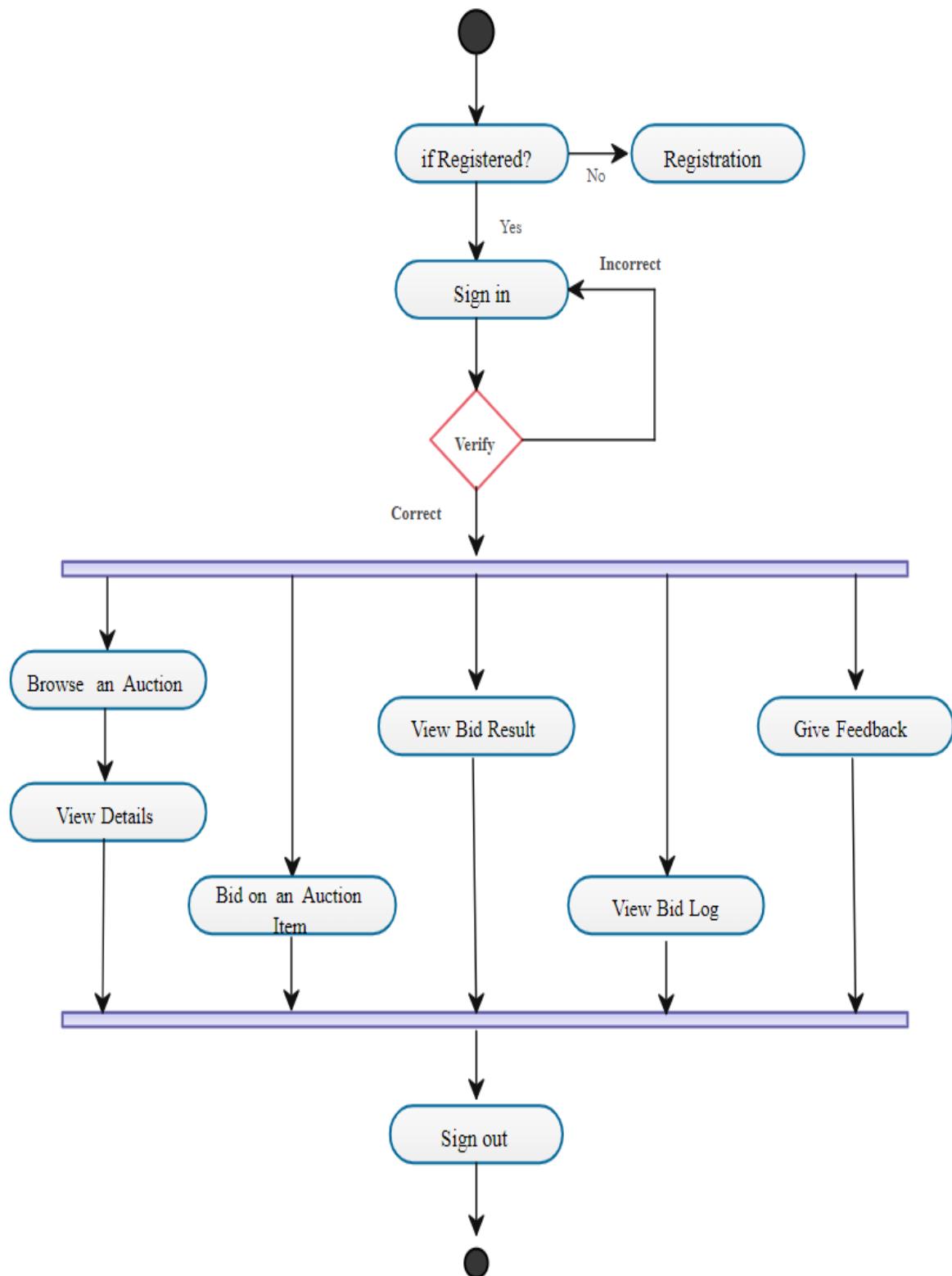
Symbol	Name	Use
	External Entity	Rectangle source and / sink destination data.
	Process / Function	Transformed, Store, or Distribute. Annotated with number and name of function.
	Data Flow	Direction of data flow single piece of data or logical collection of data.
	Data Store	Open Rectangle Parallellines Data Structure, File, Table, Database.

**Context Diagram (0 LEVEL) :-**

**1 Level Diagram :-**

**2 Level Diagram :-**

**USE CASE DIAGRAM**

**FLOW CHART**

## COST ESTIMATION

For the Online Auction Auction Platform, we'll adopt a comprehensive cost estimation approach, considering various components of the project. The estimation will be based on module-wise assessment, ensuring an accurate allocation of costs.

### **Module Wise:**

#### **- User-Centric Front-End Auction Platform:**

- Design and Development: Rs 15,000
- User Interface Optimization: Rs 8,000
- Integration of Live Bidding Features: Rs 12,000

#### **- User Accounts and Profiles:**

- User Registration and Authentication: Rs 10,000
- Profile Management: Rs 6,000

#### **- Efficient Bidding System:**

- Live Bidding Interface: Rs 15,000
- Bidding Controls: Rs 10,000

#### **- Detailed Product Pages:**

- Comprehensive Product Information Display: Rs 12,000
- User Reviews Integration: Rs 8,000

#### **- Administrator Dashboard:**

- Product and Auction Management: Rs 14,000
- Order Processing: Rs 10,000
- User Management: Rs 8,000

#### **- Security and Data Management:**

- Implementation of Secure Transactions: Rs 12,000
- Privacy Compliance Measures: Rs 8,000

**Total Estimated Cost:**

**- Development Costs: Rs 141,000**

This estimation encompasses the major components of the Online Auction Auction Platform. Keep in mind that these figures are approximate and may vary based on specific project requirements, developer rates, and other factors. The modular breakdown allows for a more detailed and accurate assessment of the overall project cost.

**DATA DICTIONARY & NORMALIZATION****Database Name: online\_auction\_platform****Table 1:** auction

Column Name	DataType (Size)	Constraints	Remarks
Id	integer	A.I, P.K	
start_time	datetime		
end_time	datetime		
created_by_id	integer	F.K.	
product_id	bigint	F.K.	

**Table 2:** bid

Column Name	DataType (Size)	Constraints	Remarks
Id	integer	A.I, P.K	
bid_amount	decimal		
auction_id	Bigint	F.K.	
user_id	integer	F.K.	
bid_time	datetime		

**Table 3:** Category

<b>Column Name</b>	<b>DataType (Size)</b>	<b>Constraints</b>	<b>Remarks</b>
Id	integer	A.I, P.K	
category_name	Varchar(255)		
category_image	Varchar(100)		

**Table 4:** contextus

<b>Column Name</b>	<b>DataType (Size)</b>	<b>Constraints</b>	<b>Remarks</b>
Id	integer	A.I, P.K	
Name	Varchar(150)		
Email	Varchar(254)		
Subject	Varchar(150)		
message	Varchar(255)		

**Table 5:** product

<b>Column Name</b>	<b>DataType (Size)</b>	<b>Constraints</b>	<b>Remarks</b>
Id	integer	A.I, P.K	
products_name	Varchar(255)		
products_base_price	integer		
products_category_id	Bigint	F.K.	
products_description	Varchar(255)		
products_image	Varchar(100)		
Created_by_id	Integer	F..K	

**Table 6:** userprofile

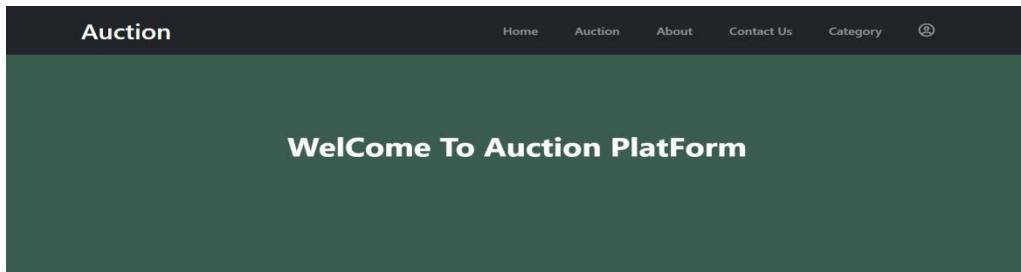
<b>Column Name</b>	<b>DataType (Size)</b>	<b>Constraints</b>	<b>Remarks</b>
Id	integer	A.I, P.K	
Address	Varchar(255)		
City	Varchar(100)		
State	Varchar(100)		
Postal_code	Varchar(20)		
Country	Varchar(100)		
User_id	Integer	F..K	

**Table 7:** winner

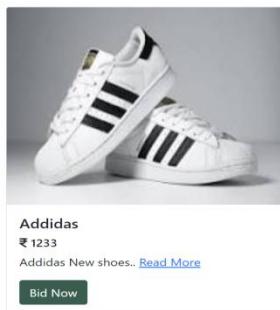
Column Name	DataType (Size)	Constraints	Remarks
Id	integer	A.I, P.K	
bid_amount	Decimal		
Auction_id	Bigint	F.K.	
User_id	Integer	F.K.	

## SCREEN LAYOUTS

### Index Page



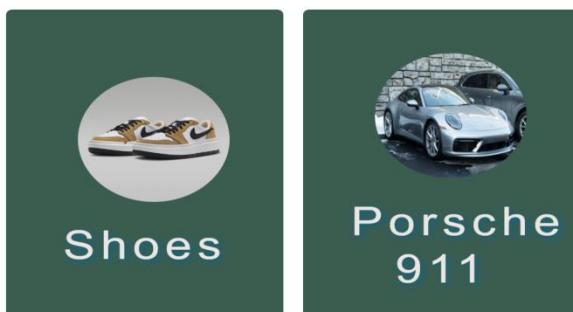
### Live Auction



### Recent Completed Auctions



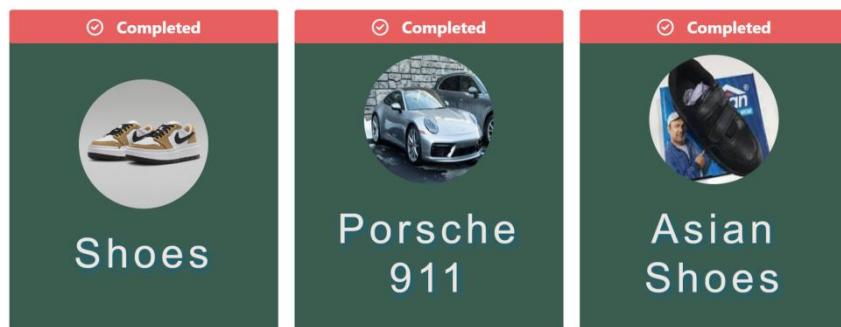
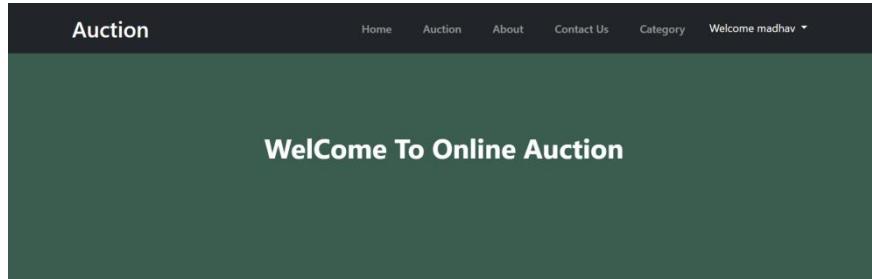
### Recent Winners



**Code :-**

```
def home(request):
    user_profile = None
    if request.user.is_authenticated and not isinstance(request.user,
AnonymousUser):
        user_profile = UserProfile.objects.filter(user=request.user).first()
    # Fetch all products
    product_data = Product.objects.all()
    # Fetch all auctions
    all_auctions = Auction.objects.all()
    # Filter the last three active auctions
    active_auctions = [auction for auction in reversed(all_auctions) if
auction.is_active()][:3]
    completed_auctions = [auction for auction in reversed(all_auctions) if not
auction.is_active()][:3]
    # Fetch winner data for completed auctions and filter in reverse order
    winners_data =
    Winner.objects.filter(auction__in=completed_auctions).select_related('user',
'auction__product')
    winners_data = winners_data.order_by('-id')[:3]
    data = {
        'user_profile': user_profile, # Added this line to include user_profile in the
context
        'product_data': product_data,
        'active_auctions': active_auctions,
        'completed_auctions': completed_auctions,
        'winners_data': winners_data,
    }
    return render(request, "home/index.html", data)
```

## Auction Page



## Online Auction Platform

This Platform is useful for all



[Home](#)  
[Auction](#)  
[About us](#)  
[Contact us](#)

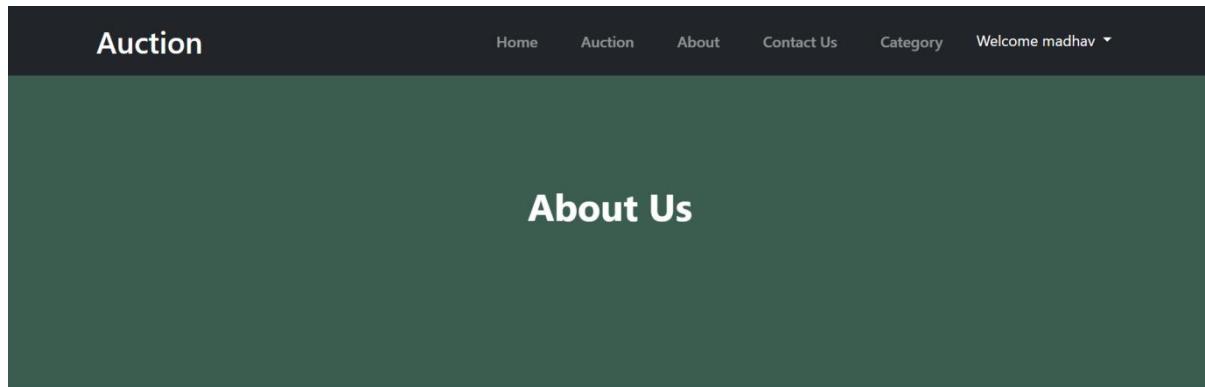
[Ornaments](#)  
[Old Telephones](#)  
[Cars](#)  
[Bikes](#)



Code :-

```
def auctions(request):
    auctions = Auction.objects.all()
    # Determine the active status for each auction
    auction_data = []
    for auction in auctions:
        auction_data.append({
            'auction': auction,
            'is_active': auction.is_active(),
        })
    template_name = 'home/auction.html'
    context = {'auction_data': auction_data}
    return render(request, template_name, context)
```

## About Page



### Our Team / Partner



**Madhav Gediya**

Partner - 1



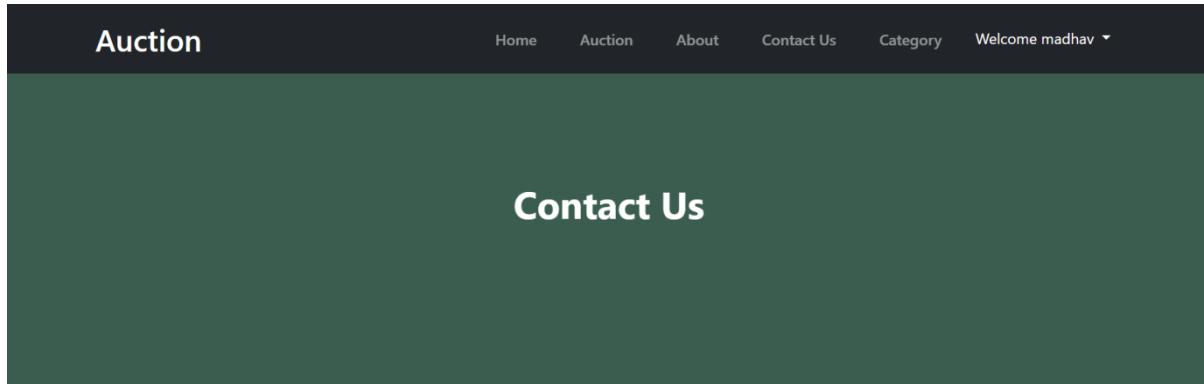
**Jenish Duvani**

Partner - 2

### Code :-

```
def about(request):
    about = {'page': 'About'}
    return render(request, "home/about.html", about)
```

## Contact Page

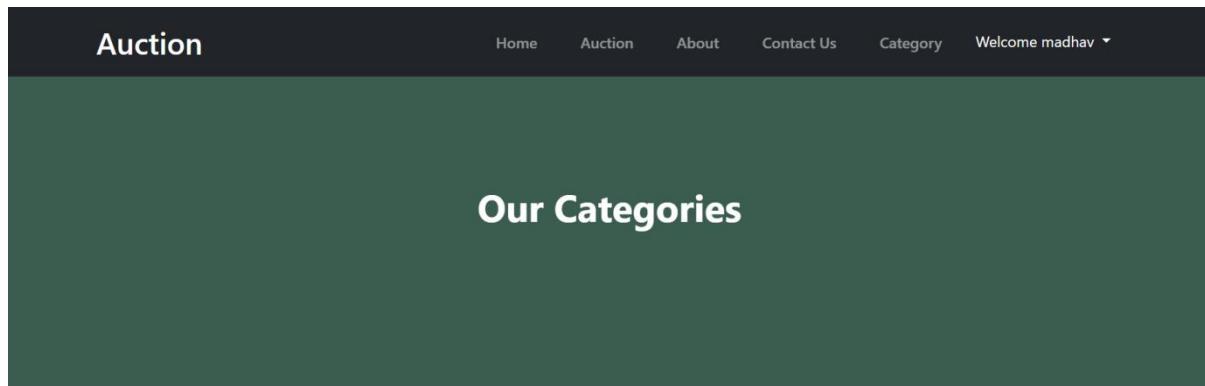


The diagram shows a contact form with a green rounded rectangular border. Inside, there are four input fields: 'Name' (horizontal), 'Email' (horizontal), 'Subject' (horizontal), and 'Message' (vertical). Below these fields is a green button labeled 'Send Message'.

**Code :-**

```
def contact(request):
    if request.method == 'POST':
        form = ContactForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('home')
    else:
        form = ContactForm()
    return render(request, 'home/contact.html', {'form': form})
```

## Categories Page



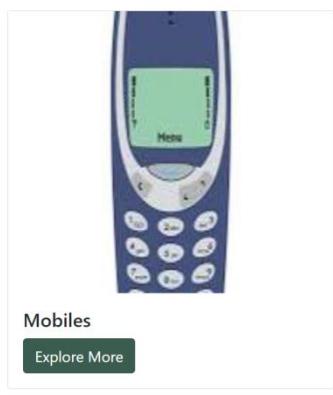
Shoes

[Explore More](#)

Cars

[Explore More](#)

Gold

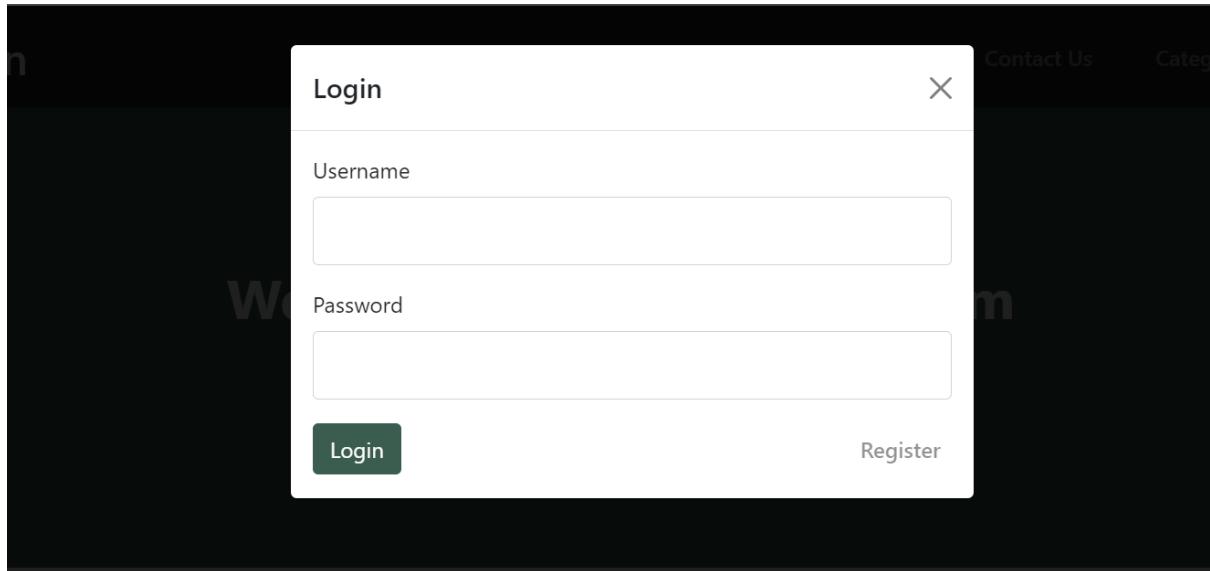
[Explore More](#)

Mobiles

[Explore More](#)**Code :-**

```
def category(request):
    context = {'page': 'category'}
    category_data = Categorie.objects.all()
    data = {
        'category_data':category_data,
    }
    return render(request, "home/category.html", data)
```

## Login User



**Code :-**

```
def my_login_view(request):
    if request.method == "POST":
        username = request.POST.get('username')
        password = request.POST.get('password')

        user = authenticate(username=username, password=password)

        if user is not None:
            login(request, user)
            messages.success(request, "Login")
            return redirect('home')
        else:
            messages.error(request, "Not Login")
            return redirect('home')

    return HttpResponse('404 page')
```

## Register Page

## User Registration

Username:

Email:

Password:

Confirm Password:

**Register**

Already Hava an Account? [Sign In](#)

**Code :-**

```
def register(request):
    if request.method == "POST":
        username = request.POST.get('username')
        email = request.POST.get('email')
        password = request.POST.get('password')
        confirm_password = request.POST.get('confirm_password')
        if password != confirm_password:
            messages.error(request, "Password Do not match")
            return redirect('home')

        myuser = User.objects.create_user(username, email, password)
        myuser.save()
        messages.success(request, "Success")
        return redirect('home')
```

## Products Details Page

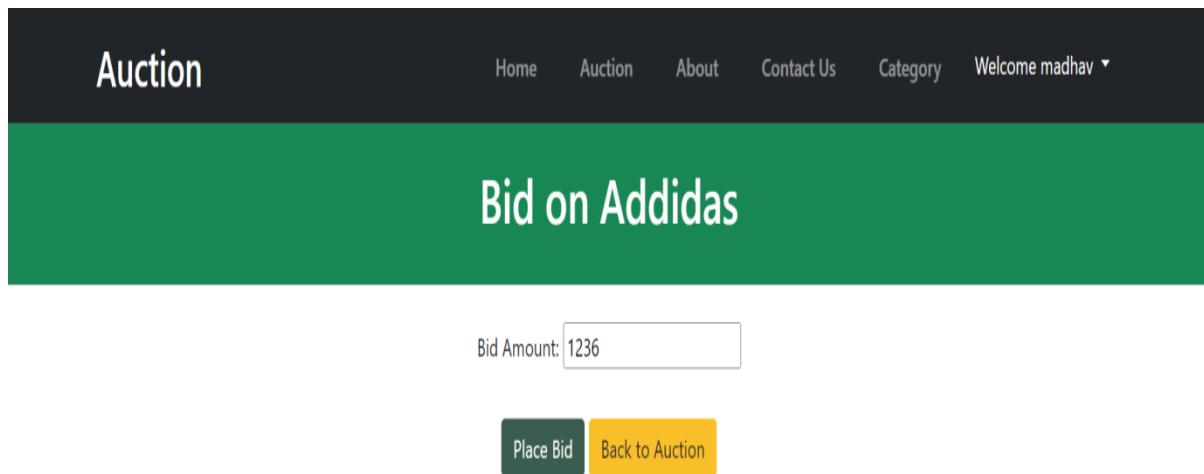
The screenshot shows a website for an auction platform. At the top, there's a dark header bar with the word "Auction" on the left and navigation links for "Home", "Auction", "About", "Contact Us", "Category", and a welcome message "Welcome madhav". Below the header is a green banner with the text "Addidas Auction Details". The main content area features a large image of two white Adidas Superstar sneakers with black stripes. To the right of the image, there's a section titled "Place Your Own Bid" with a "Place Bid" button. Below this, a green box indicates the "Start Time: March 11, 2024, 10:18 a.m." and a red box indicates the "End Time: March 12, 2024, 10:18 a.m.". Underneath, a section titled "Bids:" shows the message "No bids yet."

### Code :-

```
def auction_detail(request, auction_id):
    auction = get_object_or_404(Auction, pk=auction_id)

    # Get bids for the auction
    bids = auction.bids.all()

    context = {'auction': auction, 'bids': bids}
    return render(request, 'home/auction_detail.html', context)
```

**Bid Page****Code :-**

```
def bid(request, auction_id):
    auction = get_object_or_404(Auction, pk=auction_id)
    if not request.user.is_authenticated:
        return redirect('/#loginModal')
    if request.method == 'POST':
        form = BidForm(request.POST)
        if form.is_valid():
            bid_amount = form.cleaned_data['bid_amount']
            if auction.bids.exists():
                previous_bid = auction.bids.latest('bid_time')
                if bid_amount <= previous_bid.bid_amount:
                    min_bid_amount = previous_bid.bid_amount + 1
                    form.add_error('bid_amount', f'Bid must be higher than {min_bid_amount}.')
                return render(request, 'home/bid.html', {'form': form, 'auction': auction})
```

```
else:  
    if bid_amount <= auction.product.products_base_price:  
        form.add_error('bid_amount', f'Bid must be higher than the base  
price ({auction.product.products_base_price}).')  
        return render(request, 'home/bid.html', {'form': form, 'auction':  
auction})  
  
    user = request.user  
  
    Bid.objects.create(auction=auction, user=user,  
bid_amount=bid_amount)  
  
    if auction.hasEnded():  
        determine_winner(auction)  
        return redirect('winner', auction_id=auction.id)  
  
else:  
    form = BidForm()  
  
return render(request, 'home/bid.html', {'form': form, 'auction': auction})
```

## All Bids View Page

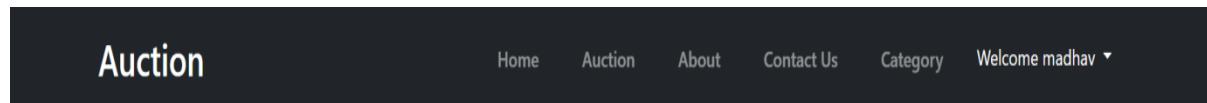
Auction Image	Auction	Bid Amount	Action
	Shoes	1300.00	<button>View Winner</button>
	Shoes	1299.00	<button>View Winner</button>

### Code :-

```
def my_bids(request):
    if not request.user.is_authenticated:
        messages.error(request, 'Please log in to view your bids.')
        return redirect('/#loginModal')

    user_bids = Bid.objects.filter(user=request.user).reverse()
    return render(request, 'home/my_bids.html', {'user_bids': user_bids})
```

## Winning Products Page



### My Winnings

Auction	Winning Bid Amount	Auction Image
Shoes	1300.00	

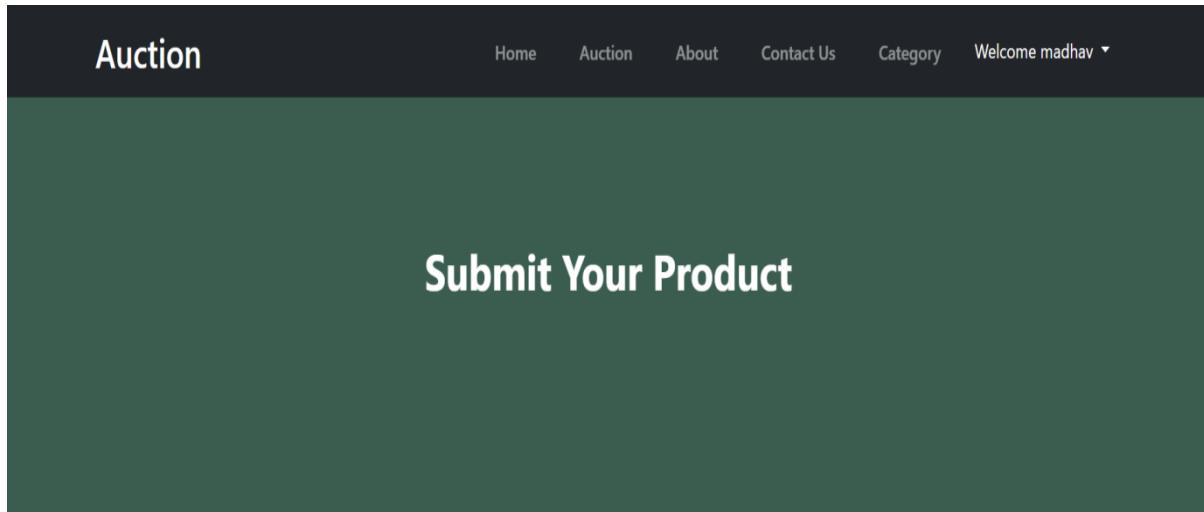
### Code :-

```
def my_winnings(request):
    if not request.user.is_authenticated:
        return redirect('/#loginModal')

    user_winnings = Winner.objects.filter(user=request.user)

    return render(request, 'home/my_winnings.html', {'user_winnings':
user_winnings})
```

## User Register's Products For Auction

A screenshot of a product registration form. The form is contained within a rectangular box with rounded corners and a thick green border. It consists of several input fields: "Product Name" (empty), "Base Price" (empty), "Product Category" (containing the value "Shoes"), "Product Description" (empty), and "Product Image" (containing a file input field with the placeholder "Choose File" and "No file chosen"). At the bottom right of the form is a green button labeled "Submit Product".

**Code :-**

```
def submit_your_product(request):
    if not request.user.is_authenticated:
        return redirect('/#loginModal')
    categories = Categorie.objects.all()
    if request.method == 'POST':
        form = ProductForm(request.POST, request.FILES)
        if form.is_valid():
            product = form.save(commit=False)
            product.created_by = request.user
            product.save()
            messages.success(request, 'Your product is submitted and in review. It will be published for auction once approved by admin.')
            return redirect('submit_your_product')
    else:
        form = ProductForm()
    return render(request, 'home/submit_your_product.html', {'form': form, 'categories': categories})
```

## User's Registered Product

Auction

Home Auction About Contact Us Category Welcome madhav ▾

### My Products



Shoes  
Base Price: 1000  
  
Sale Price: 1300.00  
  
[View Winner](#)

### Code :-

```
def my_products(request):
    if not request.user.is_authenticated:
        return redirect('/#loginModal')
    user_products = Product.objects.filter(created_by=request.user)
    for product in user_products:
        try:
            auction = Auction.objects.filter(product=product).latest('start_time')
            product.auction = auction
            if auction.has_ended():
                winner = Winner.objects.filter(auction=auction).first()
                product.winner = winner
        except Auction.DoesNotExist:
            product.auction = None
            product.winner = None
```

```
return render(request, 'home/my_products.html', {'user_products': user_products})
```

## SPECIAL UTILITIES

**Following are special utilities are being used by Online Auction Platform:**

### Use of Bootstrap:

- In our project we have used Bootstrap so the website can be responsive and can use other tools of bootstrap also.

### Use of Color Palette:

- We have used a color palette by which it shows the whole layout of Website.
- Link : <https://colorhunt.co/palette/fdf7e4faeed1ded0b6bbab8c>

## TESTING

The testing phase for the Online Auction Platform is integral to ensuring the robustness and reliability of the system. Below outlines the testing objectives, types of testing, defect tracking, and key criteria for sign-off.

### **Objectives:**

#### **1. Functional Validation:**

- Verify that each feature and function adheres to specified functional requirements.
- Ensure the seamless execution of individual components.

#### **2. User Experience and Usability:**

- Validate that the platform provides an intuitive and user-friendly experience.
- Assess overall usability and accessibility for diverse user profiles.

#### **3. Defect Identification and Resolution:**

- Identify and document defects or issues during various testing phases.
- Ensure timely resolution of identified defects.

#### **4. Performance, Scalability, and Security Assessment:**

- Evaluate the platform's performance under varying load conditions.
- Assess scalability to accommodate potential growth.
- Identify and address potential security vulnerabilities and threats.

#### **5. Compliance with Data Privacy and Security Standards:**

- Ensure adherence to data privacy and security standards.
- Validate the implementation of secure transaction protocols.

#### **Types of Testing:**

**The testing phase will encompass the following types of testing:**

- Functional Testing:** Validating individual features and functions.
- Integration Testing:** Ensuring seamless integration across different components.
- User Acceptance Testing (UAT):** Confirming alignment with user expectations.
- Performance Testing:** Evaluating speed, responsiveness, and scalability.
- Security Testing:** Identifying and mitigating security vulnerabilities.
- Regression Testing:** Verifying that new changes do not impact existing functionality.
- Usability Testing:** Assessing user-friendliness and accessibility.

**Defect Tracking:**

Defects identified during testing will be meticulously documented, classified, and tracked throughout the testing phase. Regular updates on defect resolution progress will be communicated.

**Test Reports:**

Regular test reports summarizing test results, defect status, and recommendations will be generated. These reports will be shared with project stakeholders to maintain transparency.

**Exit Criteria:**

**The testing phase will be considered complete upon meeting specific exit criteria, including:**

- Minimum defect closure rates.
- Successful test case pass rates.

**Sign-Off and Acceptance:**

Formal sign-off and acceptance by project stakeholders will signify the conclusion of the testing phase, indicating the platform's readiness for deployment.

## Test Case

### **VALIDATION CHECKS TEST CASES:**

A Test case is a set of conditions or variable under which a tester will determine whether a system satisfy requirement or works correctly.

The process of developing test case can also help to find the problem in requirement or design of an application.

A test case can have the following elements, not however that normally a test management tool is used by companies by the tool used.

**Test Suit ID :** The ID of the test suit, to which this test case belongs.

**Test Case ID :** The ID of the test case.

**Expected Result :** The expected result of the test. **Actual Result:** The actual result of the test, to filled after executing the test case.

**Created By :** The name of the author of the test case.

**Executed By :** The name of the person who executing the test cases.

**Pass/Fail :** The result in "Pass" or "Fail" according to the test, when the expected result and the actual result is same then the result is "pass" else result is "fail".

**Remark :** Any comment on the test case or test execution.

I create following some test cases which is executed by me.

Task no	Task	Expected Result	Actual Result	Pass / Fail
1	Enter valid username and password	Login Successfully	Login Success	Pass
2	Enter Invalid username password	Login failed ,Error Occurred	Generating an error message	Pass
3	Click on login button without providing any value	An error message will generate	Generating an error message	Pass

## IMPLEMENTATION

The implementation phase for the Online Auction Auction Platform is focused on seamless integration and user adaptation. Key areas of attention include:

### **1. User-Friendly Environment:**

- Ensure accessibility across devices and browsers for an efficient user experience.

### **2. Efficient Information Handling:**

- Assess the platform's effectiveness in collecting and managing auction-related information.

### **3. Intuitive Design:**

- Evaluate the layout design for visual appeal and user-friendly navigation.

### **4. Functional Integrity:**

- Verify that the platform adheres to specified requirements, conducting thorough testing.

### **5. Security Protocols:**

- Implement robust user authentication and authorization, addressing potential vulnerabilities.

**Human-Centric Approach:****Facilitate user adaptation and satisfaction through:****1. Training and Support:**

- Provide user training sessions and establish a responsive support system.

**2. Change Management:**

- Communicate benefits and improvements, ensuring a positive reception of the new system.

**3. Feedback Loop:**

- Establish a feedback mechanism for continuous improvement based on user insights.

**Continuous Monitoring:**

Regularly monitor performance, user feedback, and analytics for iterative improvements, ensuring ongoing system efficiency.

## BIBLIOGRAPHY

### Websites used:

- <https://docs.djangoproject.com/en/5.0/>
- <https://getbootstrap.com/docs/4.6/getting-started/introduction/>
- <https://www.w3schools.com/django/>
- <https://www.geeksforgeeks.org/django-tutorial/>