

## Software requirement:

### Web development-

1. Html
2. Css3
3. Reactjs
4. Bootstrap5
5. JavaScript

### ( Backend)

1. Python 3.12.2
2. Django rest framework

### Data management:

1. Mysql

## Physical features -

### 1. User Management

**Registration and Authentication:** Allow users to sign up, log in, and log out (can be integrated with Django's built-in User model or customized).

**User Profiles:** Profile details such as username, email, password, contact info, and roles (e.g., buyer, seller, admin).

**JWT Authentication:** Implement JWT for secure authentication for RESTful APIs.

**Account Activation:** Email-based account activation after registration.

### 2. Auction Listings

**Create Auction:** Sellers can create an auction listing with details like item name, description, starting price, auction start time, auction end time, images, and reserve price.

**Edit/Delete Auction:** Sellers should be able to modify or delete their listings before any bid is placed.

**View Auction Listings:** Public or authorized users can view all active auction listings.

**Auction Status:** Display if the auction is open, closed, or ended.

**Categories:** Option to categorize auction items by types (e.g., electronics, art, vehicles).

### 3. Bidding System

**Place Bid:** Registered users can place bids on active auctions. Ensure each bid is higher than the current bid.

**Auto-bidding (Optional):** Allow users to set a maximum bid and the system automatically bids on their behalf up to that amount.

**Bid History:** Display the history of bids for each auction, showing bidder usernames and amounts (optional).

### 4. Auction Expiry and Winner

**Auction End:** When the auction ends (based on the end time), declare the highest bidder as the winner.

**Reserve Price:** If the reserve price isn't met, the item is not sold.

**Winner Notification:** Notify the winning bidder via email or in-app notification.

### 5. Payment and Transaction

**Payment Gateway Integration:** Integrate a payment service (Stripe, PayPal) to handle payments from winning bidders.

**Payment Flow:** Securely handle transactions once the auction is complete, including payment tracking.

**Commission System:** Implement commission deduction (if the platform takes a percentage of the sale).

### 6. Admin Management

**Admin Panel:** A Django admin interface to manage users, auctions, bids, and monitor platform activity.

**Moderation:** Admin can review, approve, or reject auction listings and have control over banning/suspending users.

**Reports:** Generate reports on sales, bids, and user activity.

### 7. Notifications

**Real-time Updates:** Use WebSockets or Django Channels to notify users about new bids or updates on auctions they are watching.

**Email Notifications:** Notify users about bid placements, auction closures, winning, or payment status.

### 8. Watchlist and Favorites

**Auction Watchlist:** Allow users to add auctions to a watchlist for easy access and monitoring.

**Favorites:** Let users mark certain auction items as favorites.

### 9. Search and Filters

**Search Auctions:** Implement a search bar to allow users to find specific auctions.

**Filters:** Filter by category, price range, auction end time, seller, and more.

## 10. Reviews and Ratings

**Seller Rating System:** Buyers can rate sellers after an auction is completed, providing feedback and building trust.

**Item Reviews:** Buyers can leave reviews on the items they win.

## 11. Security Features

**Data Validation:** Proper validation for inputs like bids, auction details, and payment

### **Functionality -**

**Item Attributes:**

Weight, dimensions, condition, material, and color.

**Images:** Multiple photos showcasing the item.

**Location:** Specify item location for shipping or local pickup.

**Shipping:** Options for shipping methods, cost calculation, and tracking integration.

**Local Pickup:** Option for buyers to collect items locally.

**Return Policy:** Clear return and refund policies, including handling of return shipping.

**Packaging:** Details on packaging materials and shipping insurance for fragile or high-value items.

**Item Variations:** Support for listing different sizes, colors, or versions of the same item.

### **Non - Functionality -**

**Performance:** Fast response times, optimized for high traffic.

**Scalability:** Can grow with user and auction demand.

**Security:** Strong authentication, encryption, and protection against common attacks.

**Usability:** Intuitive and user-friendly interface.

**Availability:** High uptime and fault tolerance.