

Phase 1: Requirement Analysis

Project Name: Reverse Auction Platform for Industrial and IT Services

1. Project Vision

To develop a user-friendly and scalable reverse auction web platform that allows buyers (individuals or companies) to post service or product requirements and enables vendors (suppliers or service providers) to submit competitive bids using the English Reverse Auction method, with the goal of reducing procurement costs and increasing bidding transparency.

2. Objectives

- Simplify B2B(Business to Business) and B2C(Business to Customer) procurement processes through English reverse auctions.
 - Allow buyers to select the best price from multiple vendor bids.
 - Empower vendors with equal bidding opportunities in a competitive environment.
 - Ensure secure, efficient, and transparent bid handling.
-

3. Roles

Buyers : Individuals, CXOs, or companies who post requirements to procure goods or services.

Vendors: Sellers, suppliers, or service providers who compete by placing lower bids.

Admin: System administrator who monitors activity, manages users, and handles issues.

Auction Type

- **English Reverse Auction:** Vendors place progressively lower bids to win the deal.
 - **Auction Close:** Can be time-bound (deadline-based) or manually closed by the buyer.
 - **Bid Visibility:** Can be configured as open (visible to others) or sealed (hidden from other bidders).
-

5. Business Use Cases

- Buyer posts a product or service requirement with a budget and timeline.
 - Multiple vendors are notified and place competitive bids to win the order.
 - Buyer monitors bids in real-time and selects the most suitable vendor.
 - Winning vendor is notified and proceeds with fulfillment (external to the platform).
 - Admin oversees platform activity and resolves disputes as necessary.
-

6. Functional Requirements

Buyer:

- Register and log in
- Post service/product requirements (title, description, budget, deadline)
- View live auction activity (vendor bids)

- Select winning bid
- Rate and review vendors

Vendor:

- Register and log in
- View posted buyer requirements
- Submit competitive bids
- Track auction status (live bidding)
- Get notified of results (win/loss)

Admin:

- Manage user accounts (buyers and vendors)
- Moderate auctions and bids
- Access site analytics and reports
- Handle platform issues and dispute resolution

7. Non-Functional Requirements

- **Security:** Encrypted authentication, role-based access
- **Scalability:** Support growing number of users and bids
- **Performance:** Low-latency for real-time bid updates
- **Availability:** Uptime of 99.9% for core functionality
- **Usability:** Intuitive, responsive user interface
- **Maintainability:** Modular and well-documented codebase

8. Technology Stack

Frontend

- HTML5, CSS3, JavaScript – Core technologies for structuring and styling web pages
- React.js – For building responsive, component-based user interfaces
- Bootstrap / Tailwind CSS – For fast and mobile-friendly UI design
- Axios – For handling HTTP requests to the backend APIs

Backend

- Python 3.x – Programming language used for backend logic
- Django – Web framework for building secure, scalable applications
- Django REST Framework (DRF) – For creating RESTful APIs to handle frontend–backend communication

Database

- PostgreSQL – Relational database management system (preferred for scalability and advanced queries)
- *(Alternatively: MySQL or SQLite for development or lightweight deployment)*

Authentication & Security

- JWT (JSON Web Tokens) or Session-based Authentication – For secure login/session management
- Django's built-in user model & permissions – For access control

DevOps & Deployment

- Docker – For containerization and consistent environment setup
- Gunicorn + Nginx – For serving Django app in production
- Git & GitHub – Version control and collaboration
- CI/CD tools (GitHub Actions or Jenkins) – For automated testing and deployment
- Cloud Hosting (AWS / Heroku / DigitalOcean) – For deploying the production version

Optional / Future Enhancements

- Redis + Celery – For handling background tasks (e.g., email notifications, auction expiry jobs)
- WebSockets (via Django Channels) – For real-time bid updates
- Admin Dashboard (React Admin or Django Admin) – For admin user management

9. Constraints and Assumptions

- Each auction must have a defined closing date or manual close trigger
- MVP version does not include integrated payment processing
- One buyer requirement can receive multiple bids but only one can be selected
- Web platform will be developed first