full-stack project

* Real-time bidding logic
* User roles (admin, team owners, auctioneer)
* Timer control (for bids)
* Player database management
* WebSocket-based live updates (if going real-time)

Core Features You Can Include:

|  |  |
| --- | --- |
| Feature | Description |
| User Roles | Admin (organizes), Team Owners (bid), Viewer (spectator mode) |
| Player Pool | Upload player list with base price, roles (batsman, bowler, etc.) |
| Auction Mechanism | Timer per player, live bidding, highest bidder wins |
| Wallet/Budget System | Team owners have a fixed budget; can't bid beyond it |
| Team Composition | Team owners can’t exceed a player limit (e.g., 15 players, 4 overseas) |
| Live Auction Updates | Use **WebSockets** for real-time bid updates across all clients |
| Admin Panel | Add players, control auction start/stop, timer reset, etc. |
| Past Bid History | Show player price, who won, etc. |
|  |  |

**Tech Stack**

* **Frontend**: React (with Redux or Context API for state management)
* **Backend**: Spring Boot (REST APIs + WebSocket support)
* **Database**: MySQL or PostgreSQL (store players, teams, bids)
* **WebSocket (for real-time)**: spring-boot-starter-websocket + Socket.IO or STOMP with SockJS on frontend

**Frontend Features (React)**

* Login/Register page
* Team dashboard (see your team, remaining balance)
* Live auction screen with:
  + Current player
  + Timer countdown
  + Bidding buttons
  + Real-time updates of highest bid

**Backend Responsibilities (Spring Boot)**

* REST APIs:
  + Login / Register / Get players / Place bid / Get team details
* WebSocket:
  + Broadcast current player and live bids
* Auction logic (timers, bid validations, winner handling)
* Admin APIs (start auction, pause, next player, etc.)

**Strategy**

You should **build a working MVP in 2 weeks** first:

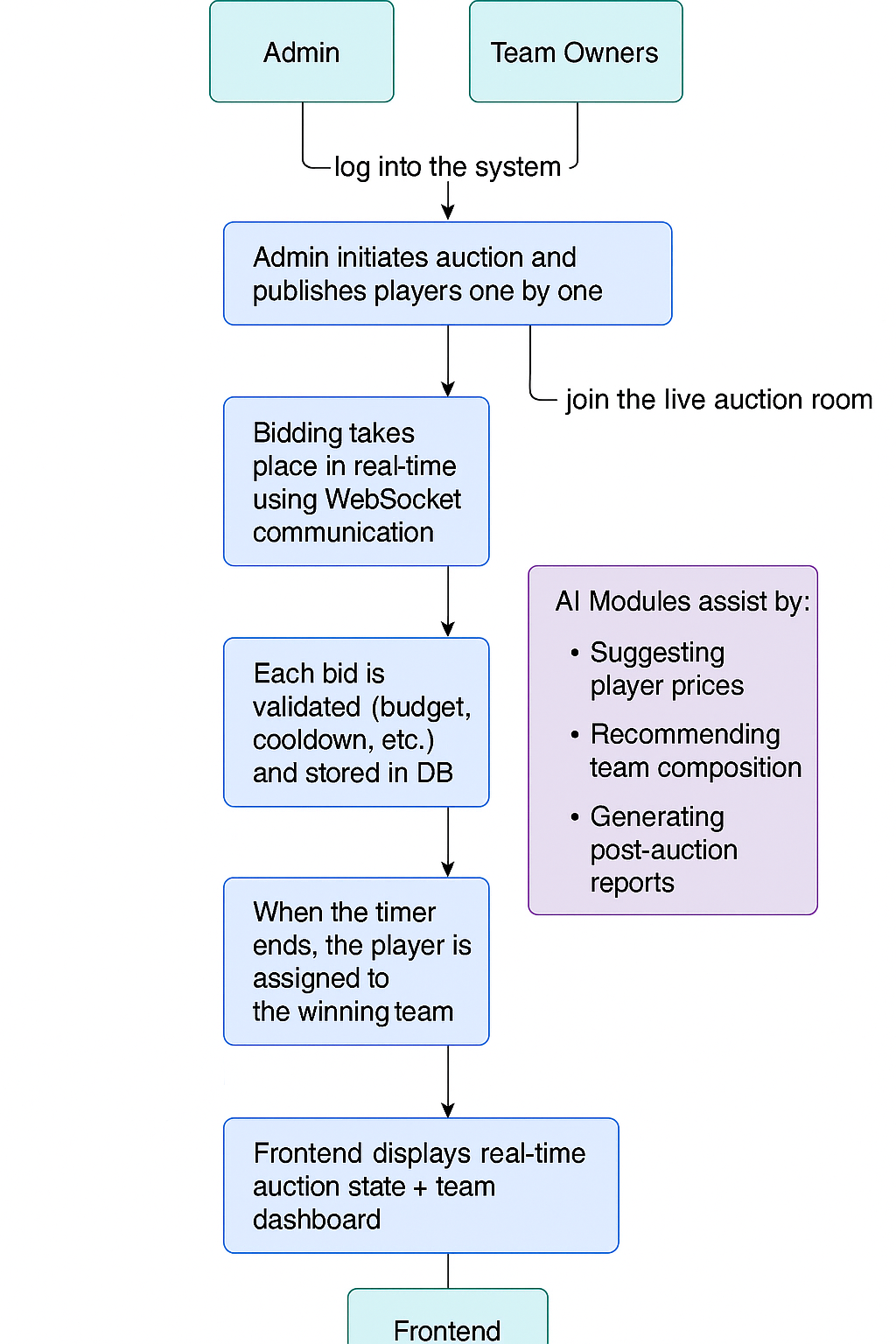
* Static player list
* Bid button with bid increase logic
* Team budget check
* Basic React UI + REST APIs

Then add fancy things like:

* WebSocket live updates
* Admin controls
* Fancy charts/team pages



Flowchart



System Architecture