IMPORTANT NOTE: Unless mentioned below, all assumptions are open to your interpretation. Please make your own assumptions and build this system. We do not entertain any queries or clarifications associated with the problem statement below. As a team, you are open to drawing your own boundaries. Your grade has no impact on incompleteness of your assumptions.

Problem Code: AucSportz

Problem Statement: The website **pec.iiit.ac.in**, developed in 2024 by a TA as the official sports portal of IIIT-H, serves as a hub for campus sports activities, announcements, leagues, team details, and event registrations. Initially built on **MongoDB Atlas** for scalability, the platform has faced growing issues such as latency from cloud dependency, limited relational querying, difficulty with complex aggregations, and poor local development support. To overcome these challenges and improve performance, the backend is required to be migrated to a **locally hosted MySQL database** that enables faster access and stronger relational capabilities and implement a new sports auction management system.

Skills: Your team will be assessed on Git proficiency, database design (SQL & NoSQL), full-stack development with MERN, and problem-solving in handling complex data relationships and workflows.

Deliverables:

Part 1: Porting from NoSQL to SQL (40%)
The objective is to assist your TA by first reverse-engineering the current data model of the closed-source sports portal through its live features, identifying key entities such as users, players, teams, sports events, tournaments, match results, registrations, and announcements. Using visual inspection of the pec.iiit.ac.in website, you must then design an equivalent relational schema in SQL database with proper data types, primary keys, and foreign key relationships. Finally, normalize the schema to at least 3rd Normal Form (3NF) by removing redundancies, eliminating anomalies, and separating concerns like users, teams, sports, and results into distinct yet related tables to ensure consistency and maintainability.

Part 2: Sports Auction Management - MERN Stack Implementation (60%)
Build a **MERN stack** web platform to manage student-run sports auctions for leagues like basketball, cricket, and frisbee. The system will let **auctioneers** host and configure auctions with custom rules (e.g., budgets, team sizes, categories), while **participants** can

register, view live updates, and track team compositions. Core features include role-based access control, secure login, flexible auction setup, and persistent storage of users, teams, bids etc. **Offline mode**: Auctions can also be run offline, with the host updating bids and rosters in real time, and teams viewing live progress through the system.

Example: An auctioneer can set up a cricket league where each team is allotted **100 attendance tokens** to spend, with rules like a **minimum of 11 players per team** and limits such as "only 3 players with more than 20 bonus attendance points." The system enforces these constraints during registration and bidding, ensuring teams are formed fairly based on attendance-based tokens.

Submission Criteria:

- **Source Code:** Git repo link with all frontend and backend code (commit history reviewed).
- **README.md** with MERN implementation details and set-up steps along with short details on design decisions, solution diagram.
- **Submit Peer Review Form**: https://forms.office.com/Pages/ResponsePage.aspx?id=vDsaA3zPK06W7IZ1VVQKHFzW4INMf2JMjyL9qPnlPbNURjRUSDA0QURKWFBCTzVCNFdDU1pDVTJZSy4u
- **LLM Usage Form**: https://forms.office.com/Pages/ResponsePage.aspx?id=vDsaA3zPK06W7IZ1VVQKHFzW4INMf2JMjyL9qPnlPbNUNFFSOTM3MDE4T0NPWUpNOTZKSFFQSUJTRC4u