

✅ Admin Dashboard - Backend Required

Feature	Backend Purpose
Add Tournament	Insert new tournament → tournament table
View Tournaments	Fetch list of tournaments
Add Team	Insert new team → team and tournament_team tables
Add Player	Insert player → player, team_player tables
Assign Captain	Insert → match_captain table
Approve Player	Approve pending → update team_player status
Schedule Match	Insert → match_played table
Enter Match Results	Update → match_played, trigger points update
Card Management	Insert → player_booked table
Delete Tournament	Delete → tournament + cascade (if applicable)
Fields	Fetch + assign → update venue_description referencing soccer_fields

✅ ALL admin actions will hit the backend for inserting, updating, or deleting records.

✅ Guest Dashboard - Backend Required

Feature	Backend Purpose
View Tournaments	Fetch list of tournaments
Match Results	Fetch → match_played + match_details
Top Scorers	Fetch aggregated → goal_details (group by player)

Feature	Backend Purpose
Red Cards	Fetch → player_booked with sent_off='Y'
Teams & Players	Fetch → all team, team_player, person, manager, coach, captain info
Join a Team	Insert join request → insert pending record in team_player or another join_requests table

👉 Most guest pages are read-only but still need backend queries to fetch the data.

🎯 Key Differences

- ✅ Admin → mostly insert/update/delete operations
 - ✅ Guest → mostly read-only fetch operations (except join request)
-

💬 Example of interaction:

1. Guest submits join request (frontend → backend) → stores pending request.
2. Admin views & approves player (frontend → backend) → updates status in team_player.





These two pages interact with the same backend table but from different dashboards/roles.

Feature Dependencies (What relies on what)

Feature (Requires)	Depends on	Why?
✓ Add Team	Add Tournament	You must create a tournament first to assign the team to
✓ Approve Player	Add Player → Join a Team	Player must exist (either added by admin or joined as guest) before approval
✓ Assign Captain	Add Player + Add Team	You can't assign a captain if there are no players or team
✓ Schedule Match	Add Team + Add Tournament + Fields	Need teams, tournament, and available venue/field
✓ Enter Match Results	Schedule Match	Can't enter results without a scheduled match
✓ Card Management	Schedule Match + Add Player	Need a played match + registered players to assign cards
✓ View Tournaments (Guest/Admin)	Add Tournament	Needs tournaments to view
✓ Match Results (Guest)	Enter Match Results	Results display comes from recorded match outcomes
✓ Red Cards (Guest)	Card Management	Red cards are only visible if recorded in card management
✓ Top Scorers (Guest)	Enter Match Results (goal details update)	Scorer data comes from entered match goals
✓ Teams & Players (Guest)	Add Team + Add Player + Assign Captain + Approve Player	To see full team roster, manager, coach, captain, players must be created/assigned

Feature (Requires)	Depends on	Why?
 Join a Team (Guest)	Add Team	Player can only request to join an existing team

In plain English:

-  You need to Add a Tournament before you can Add a Team.
 -  You need to Add a Team before you can Assign Captain, Approve Players, or Schedule Matches.
 -  You need a Scheduled Match before you can Enter Results or Manage Cards.
 -  You need Entered Results before you can View Match Results or calculate Top Scorers.
-

Critical Flow Example (Admin):

1. Add Tournament → Add Team → Add Player → Approve Player → Assign Captain → Schedule Match → Enter Match Results → Card Management

Critical Flow Example (Guest):

1. Join a Team → (Admin approves player) → Player visible in Teams & Players
 2. View Tournaments → View Match Results → View Top Scorers/Red Cards
-

Shared Tables / Interaction Points:

- **team_player table** connects:
 - Join a Team (guest)
 - Add Player (admin)
 - Approve Player (admin)
 - Teams & Players (guest view)
- **match_played table** connects:
 - Schedule Match

- Enter Match Results
 - Match Results (guest)
 - **goal_details table** connects:
 - Enter Match Results → Top Scorers
 - **player_booked table** connects:
 - Card Management → Red Cards
-

✓ **Conclusion:**

Some features **cannot function without a prerequisite one.**

Example:

- ✗ You can't assign a captain if the team or players don't exist.
- ✗ You can't see red cards if no cards have been assigned.

Backend task grouping for 3 people:

Person A: Tournament & Team management

✓ Handles everything related to creating tournaments, teams, and assigning players to teams.

- **Add Tournament** → insert into tournament
- **Add Team** → insert into team and tournament_team
- **View Tournaments** → fetch tournament
- **Join a Team (Guest request)** → insert pending request into team_player
- **Approve Player** → update team_player status
- **Teams & Players (fetch data)** → query team, team_player, person, manager, coach, captain

👉 This person owns **team setup logic + team-player relationships**

Person B: Match scheduling & results

✓ Focuses on handling matches and outcomes.

- **Schedule Match** → insert into match_played
- **Enter Match Results** → update match_played, insert into match_details, goal_details, trigger point updates
- **Match Results (Guest)** → fetch from match_played + match_details
- **Top Scorers (Guest)** → fetch aggregated from goal_details
- **Assign Captain** → insert into match_captain

👉 This person owns **match lifecycle + results + goals**

Person C: Cards, penalties, fields

✓ Manages player cards, venue assignments, red card reports.

- **Card Management** → insert into player_booked

- **Red Cards (Guest)** → fetch player_booked where sent_off = 'Y'
- **Fields** → fetch soccer_fields, update venue_description to assign
- **Delete Tournament** → delete tournament + cascade if needed
- **Match Support/official roles** (if implemented)

👉 This person owns **disciplinary actions + venue management**

🚩 Why this split works:

✅ **Feature dependencies stay grouped** (e.g., Person A handles team + approval + roster features).

✅ **Each developer focuses on logically related tables:**

- Person A → tournament, team, tournament_team, team_player, person
- Person B → match_played, match_details, goal_details, match_captain
- Person C → player_booked, soccer_fields, venue

✅ **Balanced workload** (each handles about 4–5 backend endpoints, matching your schema).