

PANA SPORTS V2.0 - ARCHITECTURE DOCUMENT

Table of Contents

1. [Project Overview](#)
2. [Database Schema Summary](#)
3. [Feature Implementation Plan](#)
4. [Workstream Breakdown](#)
5. [Technical Specifications](#)
6. [API Endpoints to Create](#)
7. [Component Inventory](#)
8. [Implementation Phases](#)

Project Overview

Version: 2.0

Target: Complete Ethiopian Football Platform

Core Objectives:

- **Season Management:** Centralized season control for all leagues and cups
- **Cup Competitions:** Full knockout/group stage support for Ethiopian Cup
- **Match Control Overhaul:** Time persistence, penalty management, event editing
- **Standings Automation:** Auto-calculate with manual override capability
- **Enhanced UI/UX:** Larger navbar, global search, dynamic ads
- **Player Stats System:** Career stats across seasons

Database Schema Summary

NEW TABLES (Created in migration):

Table	Purpose
<code>seasons</code>	Central season management
<code>season_teams</code>	Team participation per season
<code>season_players</code>	Player participation per season
<code>season_stats</code>	Aggregated season statistics
<code>cups</code>	Cup competition definitions
<code>cup_editions</code>	Yearly cup instances
<code>cup_groups</code>	Group stage groups
<code>cup_group_teams</code>	Teams in groups with standings

Table	Purpose
player_career_stats	Lifetime player statistics
player_season_stats	Per-season player statistics
match_commentary	Live text commentary
head_to_head	Team vs team historical stats
ad_campaigns	Advertising campaigns
ad_images	Ad carousel images
ad_analytics	Ad impression/click tracking
fixture_templates	Auto-generated fixtures
migrations	Migration tracking

MODIFIED TABLES:

Table	Changes
matches	+ season_id, cup_edition_id, is_cup_match, time tracking columns, penalty scores
standings	+ season_id, points_deduction, form[], head_to_head_data, clean_sheets
news_categories	+ is_featured, category_type
news	+ season_id
leagues	+ league_type, tier, promotion/relegation spots, current_season_id
top_scorers	+ season_id, penalties, minutes_played, goals_per_match
knockout_rounds	+ cup_edition_id, is_cup_round
match_events	+ updated_at, new event types (injury, save, post, chance)

NEW DATABASE FUNCTIONS:

1. `calculate_match_elapsed_time(match_row)` - Calculates live match time from timestamps
2. `update_standings_from_match()` - Trigger function for auto-updating standings
3. `update_standings_rankings()` - Trigger function for ranking calculation
4. `update_head_to_head_from_match()` - Trigger function for H2H stats

⌚ Feature Implementation Plan

PHASE 1: Foundation (Database + Core Hooks)

1. Run database migration
2. Create season-related hooks
3. Create cup-related hooks
4. Update match hooks for new fields

PHASE 2: Main Site - UI Enhancements

1. Navbar enlargement (logo + nav)
2. Move Women's League to dropdown
3. Add "Features" (Opinion Articles) nav item
4. Dynamic AdBanner component
5. Global search functionality
6. Full standings table (no limits)
7. "Other News" component (2x3 grid)

PHASE 3: League Pages Overhaul

1. Season toggle in header
2. Fetch data per season
3. Standings color coding fix (green/red only)
4. Team row click → team detail
5. Ad placements on all tabs

PHASE 4: Cup Pages (Ethiopian Cup)

1. New cup page structure
2. Knockout bracket visualization
3. Group stage tables (if applicable)
4. Cup-specific match display

PHASE 5: CMS - Season Management Module

1. Season CRUD pages
2. Team assignment to seasons
3. Player assignment to seasons
4. Season statistics display
5. Archive previous seasons

PHASE 6: CMS - Match Control Panel Overhaul

1. Time persistence implementation
2. Pause vs postpone separation
3. Restart functionality
4. Penalty shootout management
5. Event editing/deletion
6. Auto-timeout feature
7. Lineup management (dropdown-based)

PHASE 7: CMS - Other Modules

1. Dashboard upgrade
2. Ad management module
3. News categories expansion
4. Standings automation display

5. Cup management module

PHASE 8: Automation & Stats

1. Standings auto-calculation testing
 2. Player stats automation
 3. Top scorers automation
 4. Head-to-head stats display
 5. Form calculation display
-

Workstream Breakdown

WS1: Navbar & Logo Enhancement

Files to modify:

- `components/shared/navbar.tsx`

Changes:

```
// Desktop logo: increase height from h-12 to h-16
// Desktop nav items: increase font size from text-sm to text-base
// Move Women's League from standalone to League dropdown
// Add "Features" link for opinion articles
```

WS2: Dynamic AdBanner Component

Files to modify:

- `components/shared/AdBanner.tsx`

New props:

```
interface AdBannerProps {
  variant?: 'full' | 'sidebar' | 'inline';
  height?: number;
  showControls?: boolean;
  className?: string;
}
```

New files to create:

- `lib/hooks/public/useAds.ts`
- `app/api/public/ads/route.ts`

WS3: Global Search

Files to create:

- components/shared/GlobalSearch.tsx
- components/shared/SearchResults.tsx
- app/api/public/search/route.ts

Search across:

- Teams (name_en, name_am)
- Players (name_en, name_am)
- News (title_en, title_am)
- Matches (team names, date)

WS4: Season Management Core

Files to create:

CMS Pages:

- app/cms/seasons/page.tsx (list)
- app/cms/seasons/create/page.tsx
- app/cms/seasons/[id]/page.tsx (detail)
- app/cms/seasons/[id]/edit/page.tsx
- app/cms/seasons/[id]/teams/page.tsx (assign teams)
- app/cms/seasons/[id]/players/page.tsx (assign players)

Components:

- components/cms/seasons/SeasonForm.tsx
- components/cms/seasons/SeasonTeamsList.tsx
- components/cms/seasons/SeasonPlayersList.tsx

Hooks:

- lib/hooks/cms/useSeasons.ts
- lib/hooks/cms/useSeasonTeams.ts
- lib/hooks/cms/useSeasonPlayers.ts
- lib/hooks/public/useSeasons.ts

API Routes:

- app/api/cms/seasons/route.ts
- app/api/cms/seasons/[id]/route.ts
- app/api/cms/seasons/[id]/teams/route.ts
- app/api/cms/seasons/[id]/players/route.ts
- app/api/public/seasons/route.ts

WS5: Cup Management

Files to create:

Main Site Pages:

- app/cups/page.tsx (cups list)

- app/cups/[slug]/page.tsx (cup detail with current edition)

CMS Pages:

- app/cms/cups/page.tsx
- app/cms/cups/create/page.tsx
- app/cms/cups/[id]/page.tsx
- app/cms/cups/[id]/edit/page.tsx
- app/cms/cup-editions/page.tsx
- app/cms/cup-editions/[id]/page.tsx

Components:

- components/cups/CupBracket.tsx
- components/cups/CupGroupTable.tsx
- components/cups/CupPage.tsx

WS6: Match Control Panel Updates

Files to modify:

- components/cms/matches/MatchControlPanel.tsx

New features:

1. Time calculation from `match_started_at`
2. Separate pause/postpone buttons
3. Restart match button
4. Penalty shootout panel
5. Event edit/delete modals

Files to create:

- components/cms/matches/MatchEventEditor.tsx
- components/cms/matches/PenaltyShootoutPanel.tsx
- components/cms/matches/MatchTimeline.tsx

WS7: Featured Articles Page

Files to create:

- app/features/page.tsx (opinionated articles)
- components/news/FeaturedArticleCard.tsx

Filter news by `category_type = 'featured'` or `category_type = 'opinion'`

🔧 Technical Specifications

Match Time Calculation Logic

```
// Frontend calculation for live match time
function calculateMatchMinute(match: Match): number {
  if (!['live', 'second_half', 'extra_time'].includes(match.status)) {
    return match.minute;
  }

  const now = new Date();
  let elapsedSeconds = 0;

  switch (match.status) {
    case 'live':
      // First half
      if (match.match_started_at) {
        elapsedSeconds = (now.getTime() - new
Date(match.match_started_at).getTime()) / 1000;
      }
      break;
    case 'second_half':
      // Second half starts at 45
      if (match.second_half_started_at) {
        elapsedSeconds = 45 * 60 + (now.getTime() - new
Date(match.second_half_started_at).getTime()) / 1000;
      }
      break;
    case 'extra_time':
      // Extra time starts at 90
      if (match.extra_time_started_at) {
        elapsedSeconds = 90 * 60 + (now.getTime() - new
Date(match.extra_time_started_at).getTime()) / 1000;
      }
      break;
  }

  return Math.min(Math.floor(elapsedSeconds / 60), 120);
}
```

Season Toggle Component Specification

```
interface SeasonToggleProps {
  currentSeasonId: string;
  leagueId?: string;
  cupId?: string;
  onSeasonChange: (seasonId: string) => void;
}

// Location: League page header, prominently displayed
// Style: Large, engaging toggle with season name display
// Fetches: Available seasons for the league/cup
```

Standings Color Coding Rules

```
// Only two colors:  
// - Green: Top positions (promotion zone / title race)  
// - Red: Bottom positions (relegation zone)  
  
const getPositionColor = (rank: number, totalTeams: number, promotionSpots: number, relegationSpots: number) => {  
  if (rank <= promotionSpots) return 'bg-emerald-500';  
  if (rank > totalTeams - relegationSpots) return 'bg-red-500';  
  return 'bg-transparent';  
};
```

🌐 API Endpoints to Create

Public API

Endpoint	Method	Description
/api/public/seasons	GET	List all seasons
/api/public/seasons/current	GET	Get current season
/api/public/seasons/[id]	GET	Get season details
/api/public/cups	GET	List all cups
/api/public/cups/[slug]	GET	Get cup with current edition
/api/public/search	GET	Global search
/api/public/ads	GET	Get active ads for page
/api/public/head-to-head	GET	Get H2H between two teams
/api/public/matches/[id]/commentary	GET	Get match commentary

CMS API

Endpoint	Method	Description
/api/cms/seasons	GET, POST	List/create seasons
/api/cms/seasons/[id]	GET, PUT, DELETE	Season CRUD
/api/cms/seasons/[id]/teams	GET, POST, DELETE	Manage season teams
/api/cms/seasons/[id]/players	GET, POST, DELETE	Manage season players
/api/cms/seasons/[id]/archive	POST	Archive a season
/api/cms/cups	GET, POST	List/create cups

Endpoint	Method	Description
/api/cms/cups/[id]	GET, PUT, DELETE	Cup CRUD
/api/cms/cup-editions	GET, POST	List/create editions
/api/cms/cup-editions/[id]	GET, PUT, DELETE	Edition CRUD
/api/cms/match-events/[id]	PUT, DELETE	Edit/delete match events
/api/cms/match-commentary	POST	Add commentary
/api/cms/ads	GET, POST	List/create campaigns
/api/cms/ads/[id]	GET, PUT, DELETE	Campaign CRUD
/api/cms/ads/images	POST	Add ad images

❖ Component Inventory

New Shared Components

Component	Location	Purpose
GlobalSearch	components/shared/GlobalSearch.tsx	Search overlay with results
SeasonToggle	components/shared/SeasonToggle.tsx	Season selector for league pages
OtherNews	components/news/OtherNews.tsx	2x3 grid of additional news
AdBannerDynamic	Enhanced AdBanner.tsx	Configurable ad component

New CMS Components

Component	Location	Purpose
SeasonForm	components/cms/seasons/	Season create/edit form
SeasonTeamManager	components/cms/seasons/	Assign teams to season
CupForm	components/cms/cups/	Cup create/edit form
MatchEventEditor	components/cms/matches/	Edit/delete events modal
PenaltyShootoutPanel	components/cms/matches/	Penalty management
AdCampaignForm	components/cms/ads/	Ad management

New Cup Components

Component	Location	Purpose
CupBracket	components/cups/	Visual knockout bracket
CupGroupTable	components/cups/	Group stage standings

Component	Location	Purpose
CupPage	components/cups/	Main cup page layout
CupMatchList	components/cups/	Cup-specific match display

Implementation Phases

Phase 1: Database & Core Infrastructure

Estimated tokens: ~50K

- Database migration created
- Run migration on Supabase
- Update TypeScript types
- Create base hooks for seasons

Phase 2: Main Site UI Changes

Estimated tokens: ~40K

- Navbar enlargement
- Menu structure changes
- Dynamic AdBanner
- Global search
- Full standings table
- Other news component
- Ad placements

Phase 3: Season Integration

Estimated tokens: ~60K

- Season API endpoints
- Season hooks
- Season toggle component
- Update league pages for season filtering
- Update match fetching for season

Phase 4: Match Control Overhaul

Estimated tokens: ~70K

- Time persistence logic
- New status handling
- Pause/resume functionality
- Penalty management
- Event editing
- Restart functionality
- Auto-timeout

Phase 5: CMS Modules

Estimated tokens: ~60K

- Season management pages
- Dashboard upgrade
- Ad management pages
- Cup management pages
- News category updates

Phase 6: Cup Competition System

Estimated tokens: ~50K

- Cup pages on main site
- Bracket visualization
- Group stage display
- Cup-specific matches view

Phase 7: Automation & Polish

Estimated tokens: ~30K

- Standings automation testing
 - Player stats display
 - H2H display on match detail
 - Form display in standings
 - Final testing & bug fixes
-

Notes for Cross-Chat Continuity

When starting a new chat, reference this document and mention:

1. Which phase you're on
2. Last completed task
3. Any blockers or issues encountered

The migration file is at: [supabase/migrations/20241231_v2_0_migration.sql](#)

Run it on Supabase Dashboard > SQL Editor before starting Phase 1 frontend work.

Deferred to v2.5

The following features are explicitly deferred:

- Visual drag-and-drop formation builder (using dropdowns instead)
- Advanced fixture generation algorithm
- AI-powered match predictions
- Social media integration

- Push notifications