

I'll analyze your documents and CSV files to help build out the complete framework for your POWLAX app. Before I dive into the detailed research, I have a few high-level clarifying questions to ensure I understand your priorities correctly:

High-Level Strategic Questions:

1. **Migration Strategy & Timeline:** Are you planning to run WordPress and the React app in parallel indefinitely (WordPress as backend/API), or is the goal to eventually migrate completely off WordPress? This affects whether we should design for deep API integration or plan for data migration.
 1. I'm planning on running them in parallel and switching out components as we go. So I would like to get a practice planner built as soon as possible, because I don't need any of the data through API for that. But so the goal is to eventually migrate off of WordPress, but I'm not sure because I don't know the functionality aspects of how hard it will be to build these components. But as AI continues to get so much better, I definitely think that my ability to replace functions and make it my own will become easier and easier. I think each step is going to bring to light more details. So for now, any API integration that isn't solvable in about a month or two, I think we keep the API for for now.
2. **User Experience Priority:** For the initial React app launch, which user type is most critical to get right first - Club Directors, Coaches, Players, or Parents? This will help prioritize which data flows and features to focus on initially.
 1. The most important users are coaches and players. Coaches who are going to be able to see the analytics of their team, build their own play books, build their own practice plan drills with this format, the coaches are really the main one. But for the players, having the online skills Academy look better and not be so cumbersome due to learn dash would be incredible. So, keeping the educational piece of the video playbooks and Master Classes on Learndash, but migrating away to create a skills academy with react that harnesses Gamipress at first would be ideal. The only gamipress point feature I use is "Quiz Type" passed. Essentially, all of the skills Academy workouts, which are in quiz format in Learn dash have a tag. When a player completes the quiz with that tag, they get points. I would love to brought in this to tag each individual drill, and then eventually add in a video element where the player can film themselves doing it, or the AI will analyze their movements as they do it, or even just their drops during wall ball.
3. **Data Ownership & Real-time Requirements:** For the practice planner and skills tracking features, do you need real-time sync between WordPress and Supabase (e.g., points update immediately in both systems), or is it acceptable to have Supabase as the primary source with periodic syncs to WordPress?

1. Supabase primary with periodic sinks with the WordPress will be fine. The key thing I don't know here is how we link actions in the app with gamipress for tracking, or if we just rebuild the points in a new react and supabase system.

Mid-Level Technical Questions:

4. **Authentication Flow:** Do you want to keep WordPress/BuddyBoss as the single source of truth for user authentication (using JWT/SSO), or would you prefer to migrate user auth to Supabase Auth while maintaining user IDs for reference?
 1. Ideally, I just want the two to sync. If that is a tough thing to build, requires a lot of input, keep it for now. But eventually, I'd like to have Supabase be the Auth. If Supabase can be the Auth and I can integrate member press, data and WordPress, and buddy boss then doing the Supabase auth would be preferable. My concern is just that the users need to be able to sign into the site, whether they're coming from the app and or the current WordPress site.
5. **Payment & Subscription Data:** Should MemberPress remain the sole payment processor with webhooks updating Supabase, or are you considering moving payment processing to the React app in the future?
 1. I'm definitely considering moving sole payment processor with web folks to supabase. My current member press structure works, but it's not ideal. I was offering a ton of different options for memberships that included very specific items on by themselves, but now I've moved to a more cohesive system, where I'm not trying to sell individuals, I'm trying to sell clubs that have lots of individuals.

Lower-Level Data Questions:

6. **GamiPress Integration:** You mentioned that ranks, points, and badges are currently in GamiPress. Do you want to replicate this gamification logic in Supabase for better performance, or continue using GamiPress via API?
 1. I would love to duplicate the GamiPress log logic in Supabase. My only concern with that is that my master classes with the quizzes also integrate a little bit. Not enough for it to matter, to not do it. The key for me is that I would like to reanalyze exactly how it works and create new guidelines within it because the current system awards the same point for a five drill workout as a drill workout for the badges. So I'd like to create a system that is more cohesive and actually makes more sense, like getting streaks and other things of that nature.
7. **LearnDash Content:** For the Skills Academy videos, would you prefer to migrate video URLs to Supabase and serve them directly, or keep LearnDash as the LMS and just reference progress/completion data?

1. I definitely want to migrate the skills Academy video URL's and their tags and data and all of that stuff. The Learned dash system had me working for over a week on just getting the quizzes to show up in a way where they were usable. Having react for the skills academy to display the videos correctly was a nightmare. Another thing that I would love to have the ability to do is have the drills all compartmentalize into their features of what they do, so that when a user comes and wants to learn a specific thing, the user is already rated and when they want to improve a specific thing, the right drills are given to them for a workout. The other thing that I want to add to this is that I have done a lot of constructional trick videos and instructional yet very difficult wall videos as challenges, I would love to use those to challenge the player after a workout, or as a random trick that they can try. I think that will be another aspect of this that can generate interest in the app.

Once I understand these priorities, I'll analyze all your CSV files and documents to create a comprehensive data mapping and migration strategy.

!important - one of the things I want you to consider as you're thinking about this is the step-by-step migration, I can get the practice planner built already. But the practice planner is the main focal point of this build. If the practice planner is off, it won't function. But within that, there are a few things that I want to add to it that are not in the technical documents. From the coaches perspective when learning the game and implementing a practice plan, one of the biggest elements that is very hard to include is the exact strategy that we are trying to build with that drill. Another thing that is hard to include is the skills that you can layer on top of that drill. Another aspect is the concepts that are important to that drill because they vary depending on which strategies the coach wants to implement. So, I would like the drills in the practice planner to have the ability to link strategies to them so that it gives a very intuitive feel as far as why they are doing the drill when they are doing it. Because then those strategies go back to the strategy videos, and those concepts relate to what's talked about in those strategy videos. And then at a practical level during practice, when you're running a specific drill and you know that you can integrate a specific set of skills that will help the players get better, that makes it a more cohesive unit, and not only a planning tool, but a learning tool. Because when the coach selects the age bands for their team, and the game state that they want to plan for, having it readily available as far as which strategies their developing, which concepts it's gonna include, and which skills they can teach within it is massive, it would be the thing that makes this different and way better than even I could've imagined at first. So please dedicate a single section in this to outline exactly why we are transferring the data into this specific place, what components it includes in references, and then how it is going to be displayed or linked to other things within the overall system.

Thank you for those detailed answers! Based on your responses, I understand you're looking for a phased migration with parallel systems, prioritizing coaches and players, and wanting to enhance the practice planner with strategic depth. Let me ask three specific clarifying questions before I dive into the full analysis:

1. **Practice Planner Strategy Layer:** When you mention linking "strategies" to drills (like "2-3-1 Motion Offense" or "Adjacent Slide Package"), do you already have a defined taxonomy of these strategies, or would we need to build this classification system? Also, should strategies be age-band specific (e.g., simplified strategies for 8U vs complex for Varsity)?
 1. Yes, I have defined taxonomies of the strategies. That is in the taxonomy CSV, of course you don't have the whole thing, but the most recent build that I made was taking every strategy video, drill video and rating them by three items, one is when a player at a specific age band should "see and do it", when a coach should "Coach it" meaning that is when the player can comprehend the elements of it and take feedback. And finally, when a player who has been playing for 20% of their life should own it. I have a complete system of how to use the age or division or graduation year or grade of a player can be manipulated based on how user's current club defines it. So they go across each age group. Certain ones are age band specific, but within a another recognition of it is a continuum, so an advanced 8U team is looking for drills, they're gonna edge up into the 10 U drills. So I have an entire thing on that the main idea is that the build out of the react component will analyze based on that, once the data is there..
2. **Skills Academy Video Migration:** You mentioned wanting to compartmentalize drills by "features of what they do" - could you give me an example of how you'd like drills categorized? For instance, would it be by skill type (shooting, dodging, ground balls), by position (attack, midfield, defense), or by development outcome (speed, accuracy, decision-making)?
 1. All the above. They're all ready to find by shooting, dodging, ground ball, the position one is not really an issue. I go by with ball and without ball in terms of the actual state that determines a skill. Because in lacrosse you have to do everything, at some point you're the quarterback at some point you're the receiver. Certain drills will be defined by position, but ultimately the skill itself is the important part and players have to use skills regardless of their position. Development outcome can be one of them, I would have to look into it a bit more to see which things I feel would be acceptable to put on each drill, or how I would define them differently, depending on the outcome. For the most part, improving the skill itself is the outcome.
3. **Parent-Child Account Structure:** I see in the CSVs there are parent/child relationships. For the React app, do you need parents to have full visibility into their children's progress/points/badges in real-time, or is this more of an administrative link for registration purposes?
 1. The parent should have visibility into their kids, progress, points, badges in real time, that also includes any messages, they essentially have complete over status of the kids account. Like they should be able to click view as kids account and see all of their data. Because there is community aspects of this, especially for kids who are younger than 13, the parent has to have full access and full control over the kids profile. Another prospect of this as I would like to parent to get a report at some point of like what the kid did the previous week. So it should also integrate into like the WordPress MailChimp part too. Finally, because of the structure of everything, I would like to use WordPress roles as a way of compartmentalizing data come on unless it makes more sense to just have it go to Supabase. But one of the problems that I have with the

current system and how I've been building everything out is that I was about to create four separate registrations for the same team in order to ensure that each user got the correct role assigned to them so that they would see the correct data when they went into the app. Another aspect of this is that the only way that I am currently having people register using member press for things like the team HQ is using their corporate accounts feature. Because it makes the membership, it puts them into the team HQ, but there's no way of compartmentalizing which one is a parent, which one is a coach, which one is a director, which one is a player. So I was gonna have to divide out for separate memberships just for one club organization. So that can be handled and integrated through the API with member press, where we build out a react registration page that goes to member press and puts them in the correct corporate account, but also gives them the correct role based on an answer of which they are, that would be awesome. Another part of this that is tough, but I've been dealing with is I have to limit the number of players to 25 for the Team HQ's. That way Director doesn't create one team HQ and put their entire club in it. This creates a problem when you consider parents joining the club because if they register using the same corporate account registration, it will count against the player total, which is how I monitor how many people are in there. So that was another aspect of it was I would be creating two separate corporate accounts that the director would have to register for, that would create free corporate accounts for the parents, and another for the coaches, just to get them all into the same buddy boss group that acts as the Team HQ. !important - ultimately, having a registration page that links to the coaches corporate account totals for players, but not for coaches and parents, that puts everybody into the correct buddy boss group would be ideal. It would save me so much time. Finally, having this registration that links to the corporate account also be able to push users into not only the correct club OS, which is a parent buddy boss group to the child team HQ groups would be phenomenal.

Once I understand these aspects, I'll create a comprehensive data mapping document that shows exactly what moves to Supabase, what stays as API reference, and how each component fits into your strategic vision for the enhanced practice planner.

Thank you for those comprehensive answers! Based on your detailed responses, I now have a clear understanding of your vision. Just one final clarifying question before I create the complete data mapping and migration strategy:

Registration Flow Complexity: For the enhanced registration system you described (which assigns roles and manages corporate account limits), would you prefer to:

- Build a custom React registration form that uses MemberPress API to create users with proper roles and group assignments in one flow, OR
- Create a simpler interim solution using existing MemberPress forms with some custom PHP hooks to handle the role assignment, while planning for the full React solution later?

This will help me properly sequence the migration steps and determine whether we tackle this registration complexity early (which would benefit all subsequent features) or focus first on getting the practice planner and skills academy migrated.

Once you answer this, I'll create a comprehensive document that includes:

1. Complete data mapping showing what moves to Supabase vs API reference
2. Migration phases with specific timelines
3. A dedicated section on the Practice Planner's strategic enhancement layer
4. Database schema for the new features (strategies, concepts, skills linkages)
5. Solutions for the parent-child visibility and registration challenges
6. Gamification system rebuild specifications

The document will prioritize getting your enhanced practice planner operational first, followed by the skills academy transformation, all while maintaining parallel operation with WordPress.

100% building the react registration. The only caveat with this that I've had to do currently, is build out specific registration pages for specific teams. To give you a little more clarity on how it currently works, when a coach buys a team HQ or a club OS, they pay a one time fee, that includes their first years registration. I then build out all of the separate registrations for their users that has the same price listed for the fee as their annual recurring rate, I then make them a coupon code for that registration for the current year. That way they get into memberpress on a annual recurring model. Finally, one of my guarantees is that they will launch and love it. So when they pay the fee, their evaluation time of the platform itself, and whether I give them a refund doesn't start until they actually launch it to their users. And there's some caveats there for what they have to get their users to do in order to qualify for a refund like 50% of players have to log at least 1 academy workout. If they get the one with coaches training, the coaches have to complete at least 3 modules.

Perfect! That registration flow context is extremely helpful. Now I have a complete picture of your needs and requirements. Let me create a comprehensive data mapping and migration strategy document for your POWLAX ecosystem.

I'll analyze all your CSV files, documents, and requirements to build a complete framework that:

- Shows exactly what data moves to Supabase vs stays as API reference
- Creates a phased migration plan prioritizing the practice planner and skills academy
- Designs the enhanced practice planner with strategy/concept/skill linkages
- Solves your complex registration and role management challenges
- Rebuilds gamification with proper point weighting and streaks

Let me dive into the research and analysis now.

[uses the launch_extended_search_task tool]

Based on my comprehensive analysis of your documents and CSV files, I've created a complete framework for your POWLAX React app migration. Here's the strategic roadmap:

POWLAX React Migration Framework & Data Architecture

Executive Summary

Your migration will proceed in 5 phases over approximately 6 months, maintaining WordPress parallel operation while progressively moving features to React/Supabase. The enhanced practice planner with strategic depth will be the cornerstone, followed by a reimagined Skills Academy, all unified by a sophisticated registration system that solves your current limitations.

Phase 1: Enhanced Practice Planner (Weeks 1-4)

Priority: Immediate - No WordPress dependencies

Core Features:

- Drag-and-drop practice builder with timeline view
- Strategic layer linking (drills → strategies → concepts → skills)
- Age-band aware drill filtering with spectrum progression
- Parallel drill support with field visualization
- Live practice execution mode with timer

Data Migration:

```
sql
-- Core tables needed in Supabase
drills (
  id, title, description, duration_minutes,
  category, age_bands[], game_states[],
  video_url, diagram_urls[], equipment[]
)

drill_strategies (
  drill_id, strategy_id,
  primary_focus, age_band_modifier
)

drill_concepts (
  drill_id, concept_id,
  coaching_points[], common_mistakes[]
)
```

```
drill_skills (  
  drill_id, skill_id,  
  progression_level, focus_percentage  
)
```

From your CSVs, migrate these immediately:

- **DrillsExport** → Full migration to Supabase
- **2015 Terminology List** → Partial (drill names/categories only)
- Strategy/concept taxonomy from your defined system.

Questions and clarifications - if my questions are actually things that are solved by this solution, please explain to me how as if you were explaining to a child. I think some of the context may be missing for certain things due to the shortened CSV's. If the data was blank in all of the items, it may not have shown the full picture. The csv's are in https://drive.google.com/drive/folders/1zORyj7YwU0pgpV8TIFmkgRORDDvpErpu?usp=drive_link for you to access with the MCP.

- 1) there are a couple items that need to be fixed and integrated here, one is that there are Lacrosse lab URLs that get embedded into a model that pop up when you click them on the drill. I currently only have one or two locations in my Supabase. This would need to be expanded to include them - I currently have 5 locations on the current CSV. These are the slugs - `_drill_lab_url_1`, `_drill_lab_url_2`, `_drill_lab_url_3`, `_drill_lab_url_4`, `_drill_lab_url_5`
- 2) I would love to add all of the age bands - They are in the full taxonomy list.
- 3) For a direct picture of the practice planner technical specs I am currently wanting, please see the TECHNICAL-STACK-DOCUMENTATION.

Phase 2: React Registration System (Weeks 3-6)

The main addition that I want to have for this is that it will assign Buddy boss groups also to specific team HQ's when there is a Club OS. When the club director registers, they will be taken to a form where they will fill out the information about the team HQ's they want to build, what the names will be etc. I would like users to be placed into these as well. One of the caveat of the system is that they have to be in the club OS before they can get access to the team HQ, because of the parent function.

Priority: Critical - Enables all subsequent features

Key Solutions:

1. **Unified Registration Form**
 - Single React form with role selection (Coach/Player/Parent/Director)
 - MemberPress API integration for payment processing

- Automatic BuddyBoss group assignment based on role
 - Corporate account limit management (25 players, unlimited coaches/parents)
2. **Smart Account Hierarchy**

```

javascript
// Registration flow logic
3.  if (role === 'player') {
4.    checkCorporateLimit(teamId, 'player', 25)
5.    assignToBuddyBossGroup(teamHQ)
6.  } else if (role === 'parent') {
7.    linkToChildAccount(childEmail)
8.    assignToBuddyBossGroup(teamHQ, 'view_only')
9.    // Don't count against player limit
10. }

```

11. **Coupon Integration**

- Auto-apply first-year coupon for new Team HQ purchases
- Track evaluation period start from first player registration

Phase 3: Skills Academy 2.0 (Weeks 5-10)

Another element I would like to add to this is the following. I would love to add the number of reps that the user should do, and I would love to add a challenge element to know that the drill was successful for the player to see. Basically in certain drill situations, you don't just wanna do the three, you wanna do it until you get three perfect having that would be cool as well.

Also, this would need the integration into Gamipress, please research a bit to see what their API structure looks like for tracking the completions of specific quizzes for this.

Priority: High - Major UX improvement for players

Data Migration Strategy:

From your LearnDash exports, we need:

- QuizzesWorkoutsExport → Full migration (restructured)
- Video URLs from quiz content → Direct Supabase storage
- Tags and categorization → Enhanced with new attributes.

New Gamification System:

```

sql
-- Replace GamiPress with native Supabase
points_system (
  drill_id, base_points,
  duration_multiplier, -- 5-drill workout = 5x points
  streak_bonus_percentage
)

player_progress (
  player_id, drill_id, completions,
  best_streak, last_completed,
  video_submission_url -- Future AI analysis
)

```

Skill Categorization Enhancement:

- **Primary:** Shooting, Dodging, Ground Balls, Passing, Defense
- **State-based:** With Ball, Without Ball, Transition
- **Development Focus:** Speed, Accuracy, Decision-Making, Endurance
- **Challenge Level:** Fundamental, Intermediate, Advanced, Elite

Phase 4: Team Analytics & Coach Tools (Weeks 9-14)

The way I've been currently linking these is through the groups learn dash function. The coach is given the group leader role and that gives him access to the analytics. So I've been building snippets that produce some of these analytics. They are in the following folder

Priority: Medium - Builds on Academy data

WordPress API References:

- BuddyBoss activity feed (read-only embed)
- MemberPress subscription status (periodic sync)
- LearnDash Master Class progress (reference only)

New Supabase Analytics:

- Real-time team leaderboards
- Individual player progress tracking
- Practice attendance and performance
- Parent visibility portal with weekly reports

Phase 5: Complete Ecosystem Integration (Weeks 13-18)

Priority: Lower - Polish and optimization

Final Migrations:

- Club OS management tools
- Director oversight dashboards
- Advanced reporting and exports
- Mobile app optimization

Critical Data Mapping Decisions

Migrate to Supabase NOW:

1. All drill data and relationships
2. User profiles with role management
3. Skills Academy videos and categorization
4. Points/badges/gamification logic
5. Team rosters and hierarchies

Keep as WordPress API Reference:

1. Payment processing (MemberPress)
2. Community forums (BuddyBoss)
3. Master Class content (LearnDash)
4. Marketing pages and blog

Strategic Enhancement Layer (Practice Planner)

Your vision for linking strategies, concepts, and skills to drills is the key differentiator. Here's how it works:

```
typescript
// When coach selects a drill
interface DrillEnhancement {
  drill: {
    id: string
    title: string
    baseInfo: DrillData
  }
  strategies: [
    {
      name: "2-3-1 Motion Offense"
      relevance: "primary" | "secondary"
      ageAdaptation: "Full complexity for 14U+"
    }
  ]
  concepts: [
```

```

{
  name: "Off-ball movement"
  coachingPoints: ["Create space", "Time your cuts"]
  videoReference: "strategy_video_id"
}
]
skillsToLayer: [
  {
    name: "Dodging"
    integrationTips: "Add defender after 5 min"
    progressionPath: "Static → Dynamic → Live"
  }
]
}

```

This creates a complete learning ecosystem where:

- Coaches understand WHY they're running each drill
- Drills connect to broader strategic goals
- Skills development is intentional, not accidental
- Age-appropriate progressions are built-in

Database Schema Highlights

Parent-Child Visibility:

```

sql
-- Full parent access to child data
CREATE VIEW parent_child_dashboard AS
SELECT
  c.*,
  p.points_total,
  p.badges_earned,
  p.last_workout,
  m.recent_messages
FROM users c
JOIN parent_child_links pcl ON c.id = pcl.child_id
JOIN player_progress p ON c.id = p.player_id
LEFT JOIN messages m ON c.id = m.recipient_id
WHERE pcl.parent_id = auth.uid();

```

Registration Intelligence:

```

sql

```

```

-- Smart limits and role assignment
CREATE FUNCTION register_team_member(
  team_id UUID,
  role TEXT,
  corporate_account_id UUID
) RETURNS UUID AS $$
DECLARE
  current_player_count INT;
BEGIN
  IF role = 'player' THEN
    SELECT COUNT(*) INTO current_player_count
    FROM team_members
    WHERE team_id = $1 AND role = 'player';

    IF current_player_count >= 25 THEN
      RAISE EXCEPTION 'Team player limit reached';
    END IF;
  END IF;

  -- Insert logic with proper group assignment
  RETURN new_user_id;
END;
$$ LANGUAGE plpgsql;

```

Implementation Timeline

Month 1: Practice Planner + Registration System **Month 2:** Skills Academy Migration + Gamification **Month 3:** Analytics + Parent Portals **Month 4:** Polish + Mobile Optimization **Month 5:** Advanced Features + AI Integration prep **Month 6:** Full ecosystem running in parallel

Key Success Metrics

1. **Practice Planner:** Coaches save 30+ min per plan with strategic clarity
2. **Skills Academy:** 50%+ increase in player engagement
3. **Registration:** 90% reduction in setup time per team
4. **Parent Portal:** 80% of parents actively checking progress

This framework gives you a clear path from your current WordPress constraints to a modern, scalable React/Supabase ecosystem that truly serves the lacrosse community's needs while maintaining business continuity throughout the transition.

Questions and clarifications - if my questions are actually things that are solved by this solution, please explain to me how as if you were explaining to a child. I think some of the context may be missing for certain things due to the shortened CSV's. If the data was blank in all of the items, it may not have shown the full picture. The csv's are in https://drive.google.com/drive/folders/1zORyj7YwU0pgpV8TIFmkgRORDDvpErpu?usp=drive_link for you to access with the MCP. Phase 1: Enhanced Practice Planner (Weeks 1-4)

- 4) there are a couple items that need to be fixed and integrated here, one is that there are Lacrosse lab URLs that get embedded into a model that pop up when you click them on the drill. I currently only have one or two locations in my Supabase. This would need to be expanded to include them - I currently have 5 locations on the current CSV. These are the slugs - _drill_lab_url_1, _drill_lab_url_2, _drill_lab_url_3, _drill_lab_url_4, _drill_lab_url_5
- 5) I would love to add all of the age bands - They are in the full taxonomy list.
- 6) For a direct picture of the practice planner technical specs I am currently wanting, please see the TECHNICAL-STACK-DOCUMENTATION.

Phase 2: React Registration System (Weeks 3-6)

The main addition that I want to have for this is that it will assign Buddy boss groups also to specific team HQ's when there is a Club OS. When the club director registers, they will be taken to a form where they will fill out the information about the team HQ's they want to build, what the names will be etc. I would like users to be placed into these as well. One of the caveat of the system is that they have to be in the club OS before they can get access to the team HQ, because of the parent function.

Phase 3: Skills Academy 2.0 (Weeks 5-10)

Another element I would like to add to this is the following. I would love to add the number of reps that the user should do, and I would love to add a challenge element to know that the drill was successful for the player to see. Basically in certain drill situations, you don't just wanna do the three, you wanna do it until you get three perfect having that would be cool as well.

Also, this would need the integration into Gamipress, please research a bit to see what their API structure looks like for tracking the completions of specific quizzes for this.

Phase 4: Team Analytics & Coach Tools (Weeks 9-14)

The way I've been currently linking these is through the groups learn dash function. The coach is given the group leader role and that gives him access to the analytics. So I've been building snippets that produce some of these analytics. They are in the following folder