GAMIFICATION + KNOWLEDGE BASE + SESSION PERSISTENCE

Implementation Plan for PCBancard Field Sales Intelligence Suite

brochuretracker.com — Replit Build Prompt

═══

SECTION 0: PRIME DIRECTIVE

═══

⛔ DO NOT BREAK ANYTHING

This is a production application with real users. Before you write a single line of code:

1. **READ first, WRITE second.** Before modifying ANY file, read the entire file and understand what it does, what imports it, and what depends on it.
2. **ADDITIVE changes only.** Create new files, new tables, new functions, new API routes. Do NOT rewrite, restructure, or "improve" existing files unless specifically instructed.
3. **If you are unsure about ANYTHING, STOP and ask me.** Specifically:
    - If you don't know where something is stored
    - If you don't know what depends on a file you want to change
    - If you need to choose between approaches
    - If you're about to modify a shared component
    - If a database migration could affect existing data
    - If you can't find where a feature is implemented
4. **Test after every change.** Verify existing features still work before moving to the next task.
5. **Small surgical edits to existing files.** When you must add event hooks into existing components, make the smallest possible insertion. Do not reorganize, reformat, rename, or refactor surrounding code.

═══

SECTION 1: UNDERSTAND THE CURRENT STATE — ASK THESE QUESTIONS FIRST

═══

Before doing ANY implementation work, investigate the codebase and answer these questions. If you cannot determine the answer by reading the code, ASK ME.

Questions About the Knowledge Base / RAG System

Q1: The older AI Role-Play Coach page uses a 675-line built-in knowledge base (`roleplay-knowledge.ts`) AND syncs content from Google Drive. Where exactly is that knowledge base file? What is the full path? Read it and confirm what's in it.

Q2: How does the Google Drive sync work? Find the code that:

- Connects to Google Drive
- Syncs/imports documents from the connected Drive folder
- Stores those documents in the database

- Injects that content into the AI context for the Coach roleplay

Identify every file involved in this pipeline. I need to understand how it works so we can extend it to the Interactive Training modes WITHOUT breaking the existing Coach functionality.

Q3: The SalesSpark / Prospecting Coach uses Claude AI with the full knowledge base. Find that implementation. What knowledge does it inject? How does it format the context?

Q4: What database table stores the synced Google Drive content? What are the columns? How is content retrieved when the AI needs it?

Questions About the Interactive Training Modes

Q5: Find the Interactive Training component(s). Confirm:

- Where are the 20 merchant persona definitions stored? (File path, format)
- How is Gemini called for the roleplay? (Direct API? Through a server route? Replit's built-in AI integration?)
- What context/system prompt is sent to Gemini for each roleplay conversation?
- Where is the Objection Gauntlet scoring logic? (Confirm it's frontend keyword-matching only)
- Where is the Scenario Trainer logic? (Confirm it's static multiple-choice)
- Where is the Delivery Analyzer's Gemini integration? What prompts does it send?

Q6: Do ANY of the 4 Interactive Training modes call ANY backend API endpoint when a session starts, during a session, or when a session ends? Or is it all client-side?

Questions About the Existing Gamification System

Q7: Find the existing gamification tables and code. Specifically:

- What database tables exist for XP, badges, scores? (I provided a schema in a previous prompt — check if it was implemented, partially implemented, or not at all)
- Where is the XP award logic? What function/endpoint handles it?
- Where are the badge definitions and threshold checks?
- Where is the daily cap (300 XP/day) enforced?
- What events currently trigger XP awards? List each one with the file/function that triggers it.

Q8: The existing gamification hooks connect to: lesson completion, quiz passing, module completion, EquipIQ quiz, roleplay session (old Coach page), and Daily Edge. For EACH of these, find the exact line of code where the XP award is triggered. I need to see the pattern so we can replicate it for the missing hooks.

Q9: Is there an existing API endpoint like `POST /api/gamification/events` or `POST /api/gamification/award-xp` or similar? What does it expect as input?

Questions About Session Persistence

Q10: The old Coach roleplay saves sessions to the `chat_sessions` table. Does the new Interactive Training roleplay save ANYTHING to the database? Confirm that conversations, scores, and completion status are NOT persisted for the 4 new modes.

Questions About the Profile / Dashboard

Q11: What's currently on the agent's Profile page? Is there already a gamification dashboard, progress display, or XP counter visible to agents?

Q12: In the Admin / Team Management views, is there already any gamification data displayed per agent?

SECTION 2: FIX THE KNOWLEDGE BASE GAP

Problem Statement

The 4 Interactive Training modes (Live Roleplay, Objection Gauntlet, Scenario Trainer, Delivery Analyzer) are running WITHOUT access to:

- The 675-line training knowledge base (`roleplay-knowledge.ts`)
- The Google Drive synced training materials
- The NEPQ framework, objection handling formulas, sales scripts, psychological principles, dual pricing details, or PCBancard-specific content

This means the AI is “winging it” based on short persona descriptions and general knowledge. That’s a significant quality gap compared to the older Coach roleplay which has full context.

What Needs to Happen

2A: Give the Interactive Roleplay Access to the Knowledge Base

The Live Roleplay Simulator currently sends Gemini a short persona system prompt. We need to AUGMENT that prompt with relevant training knowledge — without making the prompts so long they hit token limits or slow down responses.

Approach (choose the best fit based on how the existing Coach does it):

1. **Find how the Coach roleplay injects knowledge.** It already works. Study that pattern.
2. **Create a knowledge injection layer for Interactive Training.** This should:
    - Pull relevant content from the same sources the Coach uses (built-in knowledge base + Google Drive synced content)
    - Select RELEVANT portions based on the current context (e.g., if the persona is an objection-heavy merchant, pull objection handling content; if it’s a restaurant owner, pull restaurant-specific selling points)
    - Inject it into the Gemini system prompt ALONGSIDE the existing persona description
    - NOT replace or modify the persona descriptions — ADD to them
3. **The system prompt for each roleplay should now include:**
    - The existing persona description (personality, business, objection style, weak points) — KEEP AS-IS
    - ADDED: Relevant PCBancard product knowledge (dual pricing details, how the program works, competitive advantages)
    - ADDED: The NEPQ framework summary (so the AI can recognize and reward good questioning technique)
    - ADDED: Relevant objection handling frameworks (so the AI knows which objections this persona type would realistically raise, based on our actual training materials)
    - ADDED: The sales process stages (so the AI can track where in the sales process the trainee is)
4. **Keep response quality high:**
    - The AI should still stay in character as the merchant
    - The knowledge is for the AI to USE when evaluating and responding, not to dump on the trainee
    - Responses should still be 1-4 sentences (don’t let the added context make the AI verbose)

⚠️ **ASK ME before implementing this.** Show me how you plan to structure the augmented system prompt and where the knowledge will come from. I want to verify you’re pulling from the right sources.

2B: Give the Objection Gauntlet AI-Powered Scoring

Currently the Gauntlet uses frontend keyword matching. This is the weakest part of the training system.

Enhancement:

1. After the trainee submits their response to each objection, send the response to Gemini (or Claude, whichever is more appropriate — ASK ME which to use) along with:
    - The objection text
    - The “best response approach” from the existing static data
    - Key principles from the training knowledge base for handling this specific objection
    - The NEPQ framework context
2. Have the AI score the response on a 1-10 scale based on:
    - Did they acknowledge the concern? (not dismiss it)
    - Did they use a question-based approach? (NEPQ style)
    - Did they reframe rather than argue?

    - Did they move toward a next step?
3. Return the AI's score + specific feedback on what was good and what to improve
4. **Keep the existing keyword scoring as a fallback** in case the AI call fails (network error, rate limit, etc.)
5. The existing static "best response" and "key principles" display should remain — ADD the AI feedback alongside it

⚠️ **This adds AI API calls to a mode that currently has none.** That means latency. Consider:

- Showing a brief loading state ("Analyzing your response…")
- Caching/batching if possible
- Making the AI scoring async so the UI doesn't freeze

2C: Give the Delivery Analyzer Training Context

The Delivery Analyzer already uses Gemini for analysis, but its prompts don't include the actual PCBancard presentation content. This means it's doing generic analysis rather than checking against OUR specific presentation stages and content.

Enhancement:

1. Find the Delivery Analyzer's Gemini prompts
2. Inject the actual 8-module presentation structure and key talk tracks from the training knowledge base
3. The AI should now score against OUR specific stages:
    - Module 1: Psychology Foundation (opening hook, credibility)
    - Module 2: Opening & Problem Awareness (visceral opening, fee quantification, story proof, identity activation)
    - Module 3: Solution Positioning (three options, competitor disqualification, dual pricing)
    - Module 4: Objection Prevention (customer reaction fear, social proof, math reframe)
    - Module 5: Story Proof & Transformation (hero journey, profit flywheel, counterfactual fear)
    - Module 6: Process & Risk Reversal (friction removal, 90-day protection, authority)
    - Module 7: Solution Fit (in-store, mobile, online)
    - Module 8: Close & Community (values alignment, referral, CTA)
4. The feedback should reference specific PCBancard techniques, not generic sales advice

2D: Give the Scenario Trainer Smarter Feedback (Optional Enhancement)

Currently fully static. This is lower priority, but if time allows:

1. After the trainee picks an option, send their choice to the AI along with the scenario context and training knowledge
2. Get AI-generated feedback that explains WHY that choice is good/bad based on NEPQ principles and PCBancard methodology
3. Keep the existing static scoring (point values per option) — ADD AI narrative feedback

---

═══

SECTION 3: ADD SESSION PERSISTENCE + COMPLETION

═══

Problem Statement

None of the 4 Interactive Training modes save anything to the database. Conversations disappear when you leave the page. There's no historical record, no performance tracking over time, and no way for the gamification system to award XP because there's no backend event.

What Needs to Happen

3A: Live Roleplay — Session Lifecycle

Session Start:

- When a trainee starts a roleplay (selects persona + clicks start), call a backend endpoint to create a session record
- Store: user_id, persona_id, difficulty, started_at
- Return a session_id to the frontend

During Session:

- Each message exchange should be saved (or batched and saved periodically)
- Store: session_id, role (user/assistant), message content, timestamp
- Consider saving to the existing `chat_messages` table if the schema fits, OR create a new `interactive_training_messages` table — **ASK ME which approach to use** based on what the existing table looks like

Session End:

- Add an "End Session" button to the roleplay UI
- Also trigger session end automatically after: (a) 15+ turns, OR (b) trainee says something like "I need to go" / conversation naturally concludes, OR (c) 10 minutes of inactivity
- When session ends:
  1. Send the FULL conversation to the AI for a final coaching evaluation
  2. AI returns: overall_score (0-100), strengths (array), improvements (array), technique_scores (object with categories), recommended_next_step
  3. Save to the session record: ended_at, turn_count, duration_seconds, ai_score, ai_feedback (JSON)
  4. Display the session summary / scorecard to the trainee
  5. **Trigger gamification event** → award XP based on the scoring rules

Anti-gaming validation before awarding XP:

- Session must have ≥ 6 conversation turns (3 back-and-forth exchanges minimum)
- OR session must be ≥ 3 minutes duration
- If neither threshold is met, still save the session but do NOT award XP (show a message: "Complete at least 3 full exchanges to earn XP")

3B: Objection Gauntlet — Result Persistence

When gauntlet starts: Create a gauntlet_session record (user_id, started_at)

After each objection response:

- Save: session_id, objection_id, user_response, score (keyword score + AI score if implemented), feedback
- This enables historical tracking of which objections the trainee handles well vs. struggles with

When gauntlet completes (all 12 done or trainee exits):

- Save final record: ended_at, objections_attempted, objections_passed, total_score, perfect_run (boolean)
- **Trigger gamification event** → award XP:
  - 15 XP per objection completed
  - +50 XP bonus if all 12 cleared in one run (perfect run)
- Display results summary

3C: Scenario Trainer — Result Persistence

After each scenario:

- Save: user_id, scenario_id, option_chosen, points_earned, best_outcome (boolean), ai_feedback (if AI enhancement is implemented)
- **Trigger gamification event** → award XP:
  - 25 XP per scenario completed (from existing config — verify this matches)
  - +20 XP bonus if "best outcome" path chosen

3D: Delivery Analyzer — Result Persistence

After analysis completes:

- Save: user_id, content_submitted (or hash), stages_detected, coverage_percent, overall_score, ai_feedback (JSON), psychographic_analysis, emotional_arc

- **Trigger gamification event** → award XP:
    - 30 XP base (from existing config — verify)
    - +20 XP bonus if all required presentation stages detected

---

SECTION 4: CONNECT GAMIFICATION EVENT HOOKS

Problem Statement

XP award configs exist for Gauntlet (35 XP), Scenario (25 XP), and Delivery Analyzer (30 XP), but there are no event hooks connecting these modes to the gamification engine. The Interactive Roleplay's XP hook connects to the OLD Coach roleplay page, not the new one.

What Needs to Happen

4A: Find the Existing XP Award Pattern

Before writing new hooks, find one that already works (e.g., the lesson_completed or quiz_completed hook) and study exactly how it:

1. Detects the completion event
2. Calls the gamification backend
3. Checks the daily cap
4. Awards XP
5. Checks badge progression
6. Returns feedback to the UI (toast notification, etc.)

Replicate that exact pattern for the new hooks. Do not invent a new approach.

4B: New Event Hooks to Create

Using the pattern from 4A, create hooks for:

Event	Trigger Point	XP Base	XP Bonus Conditions
`interactive_roleplay_completed`	Session end (from 3A above)	30 XP	+50 XP if AI score ≥ 80
`gauntlet_completed`	Gauntlet finish (from 3B above)	15 XP × objections completed	+50 XP if perfect run (all 12)
`scenario_completed`	Scenario finish (from 3C above)	25 XP	+20 XP if best outcome
`delivery_analysis_completed`	Analysis finish (from 3D above)	30 XP	+20 XP if all stages detected

⚠️ **IMPORTANT:** Verify these XP values against what's already in the gamification config. The values I listed come from what was described to me. If the existing config has different values, USE THE EXISTING CONFIG VALUES and tell me about the discrepancy.

4C: Update Badge Progression

When any of these new events fire, the badge progression check should also run. Verify that the existing `checkBadgeProgression` function (or equivalent) will be called. If it's not automatic, add the call.

4D: Update the Daily Cap

The existing daily cap is 300 XP. Verify that the new event hooks respect this cap using the same mechanism the existing hooks use.

4E: Update Skill Score Calculation

The Skill Score has weighted components. The new events should feed into the appropriate components:

- Interactive Roleplay scores → "Roleplay Performance" component (25% weight)
- Objection Gauntlet scores → "Objection Handling" component (25% weight)

- Scenario Trainer scores → "Objection Handling" component (25% weight)
- Delivery Analyzer scores → "Presentation Mastery" component (30% weight)

Find where the Skill Score is calculated and verify these new data sources are included. If the calculation only looks at old data sources, update it to include the new ones.

SECTION 5: XP VALUES — COMPREHENSIVE UPDATE

Here are the INTENDED XP values for the complete system. Compare these against what's currently in the config and reconcile any differences. **If there are conflicts, ASK ME which values to use.**

Teach Me the Presentation

Activity	XP	Condition
Lesson completed	25 XP	Must view for ≥ 30 seconds
Lesson quick-check ≥ 80%	+10 bonus	On top of 25 base
Lesson quick-check ≥ 90%	+25 bonus	Replaces the +10 (not stacked)
Module completed (all lessons done)	+150 bonus	All lessons in 1 of 8 modules
All 8 modules completed	+500 milestone	One-time bonus

Interactive AI Training

Activity	XP	Condition
Live Roleplay completed	60 base	≥ 6 turns OR ≥ 3 min
Roleplay performance bonus	+0 to +40	Based on AI score
Objection Gauntlet per objection	15 each	Response submitted
Gauntlet perfect run (all 12)	+50 bonus	All 12 in one run
Scenario completed	40 per scenario	Option selected
Scenario best outcome	+20 bonus	Best path chosen
Delivery Analyzer full run	80 base	Content submitted + analyzed
Delivery all stages detected	+20 bonus	All 8 stages found

EquipIQ

Activity	XP	Condition
Quiz completed	50 base	Quiz submitted
Quiz score bonus	+(score% ÷ 2)	e.g., 92% = +46 XP
Equipment advisor chat	10	Meaningful interaction

2026 Sales Process

Activity	XP	Condition
Phase reviewed	30 each	Per phase (4 total)

All 4 phases completed	+100 bonus	One-time
Practice session completed	40	Interactive practice

Daily Edge

Activity	XP	Condition
Daily Edge completed	15 per day	Content viewed/listened
3-day streak	+25 bonus	Consecutive days
7-day streak	+100 bonus	Consecutive days
14-day streak	+250 bonus	Consecutive days
30-day streak	+500 bonus	Consecutive days

Sales Spark

Activity	XP	Condition
Prompt submitted	10	Max 3 per day

Anti-Gaming Caps

- **Daily maximum:** 400 XP per day across all categories (if existing config says 300, ASK ME which to use)
- **Teach Me lessons:** Max 400 XP per day from lessons/modules specifically
- **Roleplay:** Must meet turn/time thresholds
- **Same quiz:** Cannot award XP more than once per 24 hours
- **Sales Spark:** Max 3 XP-earning prompts per day

⚠️ **NOTE:** These values differ from some of what's currently configured. Specifically:

- Existing config shows roleplay at 30 XP base → New target is 60 XP base
- Existing config shows gauntlet at 35 XP total → New target is 15 XP per objection (up to 180 + 50 bonus)
- Existing config shows scenario at 25 XP → New target is 40 XP
- Existing config shows delivery at 30 XP → New target is 80 XP
- Existing config shows EquipIQ quiz at 20 XP → New target is 50 XP
- Existing config shows daily cap at 300 → New target is 400

ASK ME to confirm before changing any existing values. Show me what's currently configured vs. what's proposed, and I'll tell you which to use.

SECTION 6: THE 5-LEVEL BADGE LADDER

Badge Definitions

Level	Name	Icon Idea	Color	XP Min	Skill Score Min	Completion %
1	Field Scout	Compass 🧭	Slate/Gray	0	0	0–20%
2	Pipeline Builder	Stacked Blocks 🏗️	Blue	500	20	20–40%
3	Objection Breaker	Shield Crack 🛡️	Orange/Amber	1,500	40	40–60%

4	Closer	Handshake 🤝	Green	3,000	55	60–80%
5	Residual Architect	Crown 👑	Purple/Gold	5,000	75	80–100%

Earning rules:

- Agent must meet ALL THREE thresholds (XP, Skill Score, Completion %) to earn the badge
- Once earned, the badge is KEPT even if Skill Score temporarily drops
- If Skill Score drops below the threshold for their current badge, show a visual warning (amber indicator) — but don't remove the badge
- Badge earned = immutable record in `badges_earned` table
- Badge earned = triggers celebration UI (modal with badge icon, name, tagline)
- Badge earned = certificate becomes available for download

Check if there's already a badge system in place. If so, reconcile these definitions with what exists. The 5-level ladder described above is the target state. If the existing system uses a different structure (e.g., Bronze/Silver/Gold per category), ASK ME how to transition.

SECTION 7: CERTIFICATES

Certificate Content

- **Header:** PCBancard Field Sales Intelligence Suite
- **Title:** "Certificate of Achievement" (Levels 1-4) or "Certificate of Mastery" (Level 5)
- **Badge:** Level name + icon
- **Agent Name:** From user profile
- **Date Earned:** Formatted date
- **Verification Code:** Unique alphanumeric (e.g., "PCB-2026-A7X9K2")
- **QR Code (optional):** Links to `brochuretracker.com/verify/[code]`

Implementation

1. Check `package.json` for existing PDF generation libraries
2. If none exist, use `jspdf` (lightweight, client-side) or `pdfkit` (server-side) — pick whichever fits the existing architecture. ASK ME if unsure.
3. Generate PDF when badge is earned
4. Store in `certificates` table with download URL
5. Agent can download from their profile dashboard
6. Admin can view/download from agent management

SECTION 8: UI COMPONENTS

Agent-Facing UI

Progress Dashboard (add to Profile page or create new section)

- **Progress ring:** Circular visualization showing overall completion %
- **Badge display:** Current badge icon + name + level, next badge preview
- **XP counter:** Total XP, XP earned today (with daily cap indicator)
- **Streak display:** Flame icon + consecutive day count
- **Next action card:** Recommended next training activity
- **Badge ladder:** Horizontal stepper showing all 5 levels, earned vs. unearned

XP Toast Notifications

- When XP is earned, show a brief non-blocking toast: "🎉 +25 XP — Lesson Completed"
- Bonus toasts should be slightly more prominent: "⭐ +150 XP — Module 3 Complete!"
- Badge level-up should be a full celebration: modal with badge icon, name, tagline, confetti optional
- Match existing toast/notification patterns in the app (check if there's already a toast component)

Session Summary Cards (for Interactive Training modes)

- After roleplay ends: show score card with AI feedback, strengths, improvements, XP earned
- After gauntlet: show per-objection results, total score, XP earned
- After scenario: show choice feedback, points, XP earned
- After delivery analysis: show stage coverage, overall score, XP earned
- All summary cards should have a "Share" or "Save" option and appear in the agent's training history

Admin/Manager-Facing UI

Team Management Enhancements

- In the existing agent list, add: badge icon, Skill Score, total XP, last active date, current streak
- Each agent row should be clickable to drill into a detail view

Agent Training Detail View

- Skill Score breakdown (5 components as bar chart or radar chart)
- XP over time (last 30 days line chart)
- Training completion checklist (what's done, what's not)
- Recent activity feed (last 20 events)
- Badge progression timeline
- Certificate download links

Style Notes

- **Match the existing app.** Look at the current card styles, colors, fonts, spacing, icon usage. Use those same patterns.
- **Mobile-first.** This is a field sales app. Every element must work on a phone.
- **Professional, not childish.** These are adult salespeople. Think LinkedIn achievements, not mobile game rewards.
- Use the same component library/patterns already in the codebase.

SECTION 9: DATABASE SCHEMA

FIRST: Check what gamification tables already exist. The schema below may have been partially or fully implemented from a previous prompt. If tables already exist:

- Do NOT drop and recreate them
- Compare existing columns to what's below
- ADD any missing columns via ALTER TABLE
- ASK ME before making changes to existing column definitions

If no gamification tables exist yet, create these:

```
-- TRAINING SESSION RESULTS (for Interactive Training modes)
-- This is for the NEW interactive training modes, separate from chat_sessions
CREATE TABLE IF NOT EXISTS training_sessions (
  id SERIAL PRIMARY KEY,
  user_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,

  -- Session type
  mode VARCHAR(50) NOT NULL, -- 'roleplay', 'gauntlet', 'scenario', 'delivery_analyzer'

  -- Context
  persona_id VARCHAR(100),        -- for roleplay
  difficulty VARCHAR(20),
  scenario_id VARCHAR(100),       -- for scenario trainer

  -- Results
  score_percent DECIMAL(5,2),     -- overall score 0-100
  score_details JSONB,            -- mode-specific scoring breakdown
  ai_feedback JSONB,              -- AI coaching feedback

  -- Session metrics
  turn_count INTEGER,
  duration_seconds INTEGER,

  -- For gauntlet specifically
  objections_attempted INTEGER,
  objections_passed INTEGER,
  perfect_run BOOLEAN DEFAULT FALSE,

  -- For delivery analyzer
  stages_detected JSONB,          -- array of detected stages
  coverage_percent DECIMAL(5,2),

  -- Timestamps
  started_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
  ended_at TIMESTAMP WITH TIME ZONE,

  -- XP awarded (denormalized for quick display)
  xp_awarded INTEGER DEFAULT 0
);

CREATE INDEX idx_training_sessions_user ON training_sessions(user_id);
CREATE INDEX idx_training_sessions_mode ON training_sessions(user_id, mode);
CREATE INDEX idx_training_sessions_date ON training_sessions(user_id, started_at DESC);


-- TRAINING SESSION MESSAGES (conversation history for roleplays)
CREATE TABLE IF NOT EXISTS training_messages (
  id SERIAL PRIMARY KEY,
  session_id INTEGER NOT NULL REFERENCES training_sessions(id) ON DELETE CASCADE,
  role VARCHAR(20) NOT NULL,      -- 'user', 'assistant', 'system', 'coach'
  content TEXT NOT NULL,
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

CREATE INDEX idx_training_messages_session ON training_messages(session_id);


-- GAUNTLET RESPONSES (per-objection detail)
CREATE TABLE IF NOT EXISTS gauntlet_responses (
  id SERIAL PRIMARY KEY,
  session_id INTEGER NOT NULL REFERENCES training_sessions(id) ON DELETE CASCADE,
  objection_id VARCHAR(100) NOT NULL,
  objection_text TEXT,
  user_response TEXT,
  keyword_score INTEGER,          -- from existing keyword matching
  ai_score INTEGER,               -- from new AI scoring (nullable until implemented)
  ai_feedback TEXT,               -- AI-generated feedback on this specific response
  created_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()
);

CREATE INDEX idx_gauntlet_responses_session ON gauntlet_responses(session_id);
```

For XP ledger, skill scores, badges, certificates, and streaks: Check if the tables from the previous gamification schema prompt already exist:

- `xp_ledger`
- `skill_scores`
- `badges_earned`
- `certificates`
- `streak_log`
- `training_progress`

If they exist, use them. If they don’t, create them using the schema from the previous implementation prompt (it was provided in detail). **ASK ME if you need the full schema again.**

SECTION 10: BUILD ORDER

Execute in this order. Complete each phase before starting the next. ASK ME to verify after each phase.

Phase 1: Investigation (NO CODE CHANGES)

- Answer ALL questions from Section 1
- Document what you find (file paths, table schemas, function names)
- Present findings to me for review
- Identify any conflicts between existing code and this plan

Phase 2: Knowledge Base Integration

- Identify the knowledge injection pattern from the existing Coach
- Propose how to extend it to Interactive Training (show me the plan)
- After my approval: implement knowledge injection for Live Roleplay
- Implement knowledge injection for Delivery Analyzer
- Implement AI scoring for Objection Gauntlet (with keyword fallback)
- Test that existing Coach still works correctly
- Test that Interactive Training quality improves with the added knowledge

Phase 3: Session Persistence

- Create new database tables (or verify existing ones)
- Build backend API endpoints for session create/update/complete
- Add session lifecycle to Live Roleplay (start → save messages → end → score → save)
- Add “End Session” button + automatic end triggers to roleplay UI
- Add session scoring (AI evaluation at session end)
- Add result persistence for Objection Gauntlet
- Add result persistence for Scenario Trainer

- Add result persistence for Delivery Analyzer
- Test that all 4 modes save results correctly
- Test that conversations persist and can be retrieved

Phase 4: Gamification Hook-Up

- Study existing XP award pattern (from a working hook)
- Show me the XP values comparison (existing config vs. proposed) — get my approval
- Implement `interactive_roleplay_completed` event hook
- Implement `gauntlet_completed` event hook
- Implement `scenario_completed` event hook
- Implement `delivery_analysis_completed` event hook
- Verify daily cap enforcement for new hooks
- Verify badge progression includes new events
- Update Skill Score calculation to include new data sources
- Test XP awards for each mode
- Test badge progression with new events

Phase 5: Badge Ladder + XP Value Update

- Compare existing badge system to 5-level ladder — show me differences
- After my approval: implement/update to 5-level ladder
- Update XP values (after my confirmation of which values to use)
- Test badge thresholds
- Test Skill Score → badge level mapping

Phase 6: Agent UI

- Build progress dashboard component (match existing app style)
- Add XP toast notifications (match existing toast pattern)
- Add badge celebration UI
- Add session summary cards for each Interactive Training mode
- Integrate into Profile page
- Test on mobile

Phase 7: Admin/Manager UI

- Add gamification columns to agent list in Team Management
- Build agent training detail view
- Add Skill Score breakdown visualization
- Add recent activity feed
- Test permission levels (agents see own, managers see team, admin sees all)

Phase 8: Certificates

- Determine PDF library to use
- Build certificate template
- Build generation endpoint
- Add download UI for agents and admins

- Test generation and verification codes

Phase 9: Final Testing

- All existing features still work (Coach roleplay, Teach Me, EquipIQ, Sales Process, Daily Edge, Sales Spark, Prospecting)
- All 4 Interactive Training modes save results
- All modes award correct XP
- Daily caps work
- Streaks calculate correctly
- Badges award at correct thresholds
- Skill Scores calculate correctly
- Admin views show correct data
- Certificates generate
- Mobile responsive
- No console errors

SECTION 11: FINAL REMINDERS

1. **PRIME DIRECTIVE: DO NOT BREAK EXISTING FUNCTIONALITY.**
2. **ASK questions. Lots of questions. I'd rather answer 50 questions than debug a broken system.**
3. **Additive changes only.** New files, new tables, new functions. Surgical edits to existing files only when necessary.
4. **Match existing patterns.** Don't introduce new libraries, new architectures, or new patterns unless absolutely necessary.
5. **Mobile-first.** Everything must work on a phone.
6. **Professional, not childish.** LinkedIn achievements, not Candy Crush.
7. **Show me your plan before executing.** Especially for: knowledge base integration, XP value changes, badge system updates, and any modifications to existing files.
8. **The knowledge base gap is the highest-impact fix.** Getting the training materials into the Interactive Training modes will dramatically improve the quality of every roleplay, every objection response, and every delivery analysis. Prioritize this.