ShipSpeak Phase Documentation - Master Prompt

Overview

This document provides complete specifications for generating detailed development documentation for **ShipSpeak**, an AI-powered Product Manager leadership development platform. Documentation will be generated **one phase at a time** and includes:

1. **Integration Contract** - Detailed feature specifications with UI layouts, data structures, user flows
2. **Slice Prompts** - Granular 4-8 hour implementation tasks with acceptance criteria
3. **Phase Summary** - Overview of what gets built and validated
4. **Master Tracker** - Progress tracking sheet for the phase

---

Product Context

Core Concept

ShipSpeak helps Product Managers develop executive communication skills and product sense by:

1. Analyzing their actual work meetings (via bot that joins Zoom/Google Meet/Teams)
2. Identifying specific communication patterns and gaps
3. Recommending personalized learning modules based on their speaking patterns
4. Providing practice exercises with AI feedback
5. Tracking progress toward career level goals (PO → PM → Senior PM → Group PM → Director)

The Problem Being Solved

Product Managers need to improve their communication and product sense to advance in their careers, but:

- Traditional coaching is expensive and has stigma attached
- Generic courses don't address their specific gaps
- They lack feedback on their actual meeting performance
- There's no clear framework for career progression

The Core Differentiator: Bot Discretion

Having a bot named "ShipSpeak Coach" join your board meeting creates stigma - others think you need help. ShipSpeak solves this by giving users **complete control over bot identity**:

- Custom bot names ("Executive Assistant", "My Notetaker", personal names)
- Stealth mode (silent, no video, minimal presence)
- Smart exit rules (leave when CEO joins, when specific keywords mentioned)
- Meeting-type specific configurations (different identity for board vs team meetings)

This allows PMs to get coaching privately without appearing to need coaching publicly.

The Four Pillars

1. **Meeting Intelligence Assistant**
    - Captures and transcribes real work meetings

   - Identifies speaker-specific patterns
   - Analyzes PM's communication style
   - Maps performance to career level framework
2. **Voice Coach**
   - Practice exercises for specific skills
   - Audio recording and transcription
   - AI feedback on practice sessions
   - Progress tracking
3. **Sense Labs** (P1/P2)
   - Product case studies from real companies
   - Practice product thinking and decision-making
   - Community discussions
   - Expert feedback
4. **Real Work Integration**
   - Learning modules personalized to meeting patterns
   - Career progression framework
   - Skill gap identification
   - Development roadmap

Career Progression Framework

PO (Product Owner) → PM (Product Manager):

- Learning basic product sense
- Developing influence without authority
- Communicating with stakeholders
- Data-backed decision making

PM → Senior PM:

- Strong executive communication (answer-first, concise)
- Trade-off articulation and decision frameworks
- Cross-functional leadership
- Strategic thinking beyond features

Senior PM → Group PM:

- Multi-product strategy
- Organizational influence
- Coaching other PMs
- Vision setting and roadmap planning

Group PM → Director:

- Department-level strategy
- Executive presence with C-suite
- Team building and culture
- Business model thinking

Development Methodology: Frontend-First (Bravax Approach)

Core Philosophy

Build the complete user experience with mock data BEFORE integrating real services. This allows:

- Rapid iteration on UX without backend complexity
- User validation before infrastructure costs
- Faster feedback loops
- Less getting stuck on technical integration issues

Three-Phase Structure

Phase 1: Complete UX with Mock Data (4 weeks, 44-54 hours)

- Build every page, component, and user flow
- Use realistic JSON mock data for everything
- Users can click through entire experience
- 10 beta users can validate the learning flow
- Zero backend infrastructure needed
- Result: Demo-ready product with validated UX

Phase 2: Real Integrations (4 weeks, 36-44 hours)

- Replace mock data with real services
- Same UI, now connected to real backends
- Zoom/Google Meet/Teams integration
- Deepgram transcription
- OpenAI analysis
- Supabase database
- Result: Production-ready MVP with paying users

Phase 3: Polish & Launch (2 weeks, 10-12 hours)

- Error handling and edge cases
- Performance optimization
- Onboarding flow
- Payment integration
- Result: Public launch ready

Why This Matters for Documentation

Each slice must specify:

- **Phase 1 implementation:** What gets built with mock data
- **Phase 2 implementation:** How mock data gets replaced with real services
- **What stays the same:** UI/UX should be identical between phases
- **What changes:** Data source and backend integrations

The Level of Detail Required

Based on successful Bravax methodology and recent refinement discussions, documentation must be **highly specific and actionable**, not high-level overviews.

Each Slice Must Include:

1. Specific Components to Build

List every React component, not just "build the dashboard":

- `MeetingListPage` - Main container
- `MeetingCard` - Individual meeting item
- `MeetingFilters` - Filter controls
- `SearchBar` - Search functionality
- `EmptyState` - No results state

2. Detailed UI Elements

Describe exactly what appears on screen:

- Meeting card shows: title, date/time, duration, participant avatars (max 5 visible), score badge (color-coded), meeting type tag, quick actions
- Score badge: green for 8-10, yellow for 6-8, red below 6
- Participant avatars: circular, 32px, overlap slightly, hover shows full name
- Filters: chips for meeting type, date range picker, participant multi-select

3. Specific Data Structures

Provide actual JSON examples, not just "meeting data":

json

```
{
  "meetings": [
    {
      "id": "mtg-001",
      "title": "Q4 Board Meeting",
      "date": "2025-11-01T14:00:00Z",
      "duration_minutes": 62,
      "meeting_type": "board_meeting",
      "participants": [...],
      "preliminary_scores": {...}
    }
  ]
}
```

4. All User Interactions

List every clickable/interactive element:

- Click meeting card → navigate to detail page
- Hover participant avatar → show name tooltip
- Click score badge → show score breakdown popup
- Use search bar → filter real-time as typing
- Click filter chip → update list to show matches
- Toggle list/grid view → change layout

5. What User Validates

Clear learning objectives for each slice:

- Does the meeting list feel manageable or overwhelming?
- Are scores displayed clearly enough?
- Can they find a specific meeting quickly?
- Does visual hierarchy work?

6. Granular Acceptance Criteria

Specific, testable requirements:

- □ 10 distinct mock meetings with varied data
- □ All filters work correctly (type, date, participant)
- □ Search finds meetings by title or participant name
- □ Score badges show correct colors
- □ Clicking meeting navigates to detail page
- □ Responsive on mobile (cards stack vertically)
- □ No console errors

Phase 1 Detailed Breakdown

Phase 1 Overview

Goal: Build complete learning experience with mock data for beta validation
Duration: 4 weeks (44-54 hours)
Outcome: 10 beta PMs can experience full product flow and provide feedback

Phase Breakdown:

- **Phase 1A: Foundation (Week 1)** - Slices 1-3 (10-12 hours)
  - Project setup, auth, dashboard structure
- **Phase 1B: Meeting Intelligence (Week 2)** - Slices 4-6 (12-15 hours)
  - Meeting list, transcript view, AI feedback panel with Next Steps
- **Phase 1C: Learning Modules & Practice (Week 3)** - Slices 7-10 (12-15 hours)
  - Module library, exercise content, recording interface, practice feedback
- **Phase 1D: Progress & Career Tracking (Week 4)** - Slices 11-15 (10-12 hours)
  - Progress dashboard, settings, mobile polish, testing

Key Features:

- ✅ Two-path onboarding (analyze meetings OR start practicing)
- ✅ Complete meeting analysis with Next Steps
- ✅ Personalized learning modules based on gaps
- ✅ Practice exercises with AI feedback
- ✅ Career progression tracking

Phase 1A: Foundation (Week 1) - 10-12 hours

Slice 1: Project Setup & Design System (4 hours)

What you deliver:

- Next.js 14 project with App Router, TypeScript, Tailwind CSS
- Design system foundations: color palette, typography scale, spacing system
- Base layout structure: sidebar navigation, header, main content area
- Reusable components: Button, Input, Card, Badge, Avatar

Specific components to build:

- `app/layout.tsx` - Root layout with font loading
- `components/ui/Button.tsx` - Primary, secondary, tertiary variants
- `components/ui/Input.tsx` - Text input with label, error, helper text
- `components/ui/Card.tsx` - Container with padding, shadow, border radius
- `components/ui/Badge.tsx` - Status badges (info, success, warning, error)
- `components/ui/Avatar.tsx` - User avatars (circular, with initials fallback)
- `components/layout/Sidebar.tsx` - Navigation sidebar
- `components/layout/Header.tsx` - Top header with user menu

Design system specifications:

- Colors:
  - Primary: Blue (`#3B82F6`)
  - Success: Green (`#10B981`)
  - Warning: Yellow (`#F59E0B`)
  - Error: Red (`#EF4444`)
  - Neutral: Grays (`#F9FAFB` to `#111827`)
- Typography:
  - Font: Inter or similar sans-serif
  - Scale: 12px, 14px, 16px, 18px, 20px, 24px, 30px, 36px, 48px
  - Weights: 400 (regular), 500 (medium), 600 (semibold), 700 (bold)
- Spacing: 4px base unit (4, 8, 12, 16, 24, 32, 48, 64, 96)
- Border radius: 4px (small), 8px (medium), 12px (large), 9999px (full)

Layout structure:

- Sidebar: 240px wide, fixed position, includes logo, navigation links, user profile at bottom
- Header: 64px tall, sticky, includes page title, search (future), notifications (future), user menu
- Main content: Fills remaining space, max-width 1440px, centered, padding 24px

Navigation links (sidebar):

- Dashboard (home icon)
- Meetings (calendar icon)
- Practice (microphone icon)
- Progress (chart icon)
- Settings (gear icon)

What user sees: Empty shell of app with working navigation. Clicking any nav link shows empty state for that section.

Acceptance criteria:

- ☐ Project builds without errors (`npm run dev`)

- □ All UI components render correctly
- □ Design system variables defined in Tailwind config
- □ Sidebar navigation highlights active route
- □ Layout is responsive (sidebar collapses to bottom nav on mobile)
- □ All components have TypeScript types
- □ ESLint and Prettier configured and passing
- □ Dark mode toggle works (if implementing)

Slice 2: Authentication Pages (Mock) (3 hours)

What you deliver:

- Login page with email/password form
- Signup page with registration form
- Onboarding flow for setting career goals
- Mock authentication (stores user in local storage/session)

Specific pages to build:

- `app/login/page.tsx` - Login page
- `app/signup/page.tsx` - Signup page
- `app/onboarding/page.tsx` - Multi-step onboarding
- `app/api/auth/mock/route.ts` - Mock auth endpoint

Specific components to build:

- `components/auth/LoginForm.tsx` - Email/password with validation
- `components/auth/SignupForm.tsx` - Name, email, password, confirm password
- `components/auth/OnboardingSteps.tsx` - Step indicator (1 of 3)
- `components/auth/CareerGoalSelector.tsx` - Current level → Target level selection

Login page UI:

- Centered card (max-width 400px)
- ShipSpeak logo at top
- Heading: "Welcome back"
- Email input (type="email", required, validation)
- Password input (type="password", required, show/hide toggle)
- "Forgot password?" link (non-functional for now)
- "Log in" button (primary, full-width)
- "Don't have an account? Sign up" link
- Shows loading state when submitting
- Shows error message if mock auth fails

Signup page UI:

- Similar layout to login
- Heading: "Create your account"
- Name input (required)
- Email input (validation: must be valid email)
- Password input (validation: min 8 chars, must include number)
- Confirm password input (validation: must match password)
- Terms checkbox: "I agree to Terms of Service and Privacy Policy"
- "Create account" button (disabled until form valid)

- "Already have an account? Log in" link

Onboarding flow (3 steps):

Step 1: About You

- "What's your current role?"
  - Product Owner
  - Product Manager
  - Senior Product Manager
  - Group Product Manager
  - Director of Product
  - Other (text input)
- "Which company?" (optional)
  - Text input with autocomplete (common tech companies)
- "Continue" button

Step 2: Career Goals

- "Where do you want to be?"
  - Same role options as above
  - Must be same or higher than current role
- Visual showing: Current Level → Target Level (with arrow)
- "What's your timeline?"
  - 6 months
  - 1 year
  - 2 years
  - Just exploring
- "Continue" button

Step 3: Choose Your Starting Path

- Header: "How would you like to get started?"
- Two-column layout with clear visual separation

Path A: Analyze My Real Meetings (Left)

- Icon: Calendar with checkmark
- Title: "Analyze My Real Meetings"
- Description: "Connect your calendar and let us analyze your actual conversations. We'll identify patterns and recommend personalized learning."
- Benefits list:
  - ⏱️ First insights in 24 hours
  - 📊 More accurate to your real style
  - 🎯 Personalized to your actual gaps
- Primary button: "Connect Calendar & Set Up Bot"

Path B: Start Practicing Now (Right)

- Icon: Microphone
- Title: "Start Practicing Now"
- Description: "Jump into practice exercises right away. We'll establish your baseline and you can add meetings later."
- Benefits list:
  - ⏱️ Start learning in 2 minutes

    - 🎯 Build skills immediately
    - 📈 See baseline assessment
- Primary button: "Start First Exercise"
- Footer note: "You can always add the other option later!"

Path A Flow (if selected):

- Redirects to bot configuration wizard
- Step 3a: Bot Identity Setup
    - "What should your bot be called?"
    - Preset options: "Executive Assistant", "Meeting Recorder", "My Notetaker", Custom name
    - Explanation: "This is how it will appear to others in your meetings"
- Step 3b: Meeting Types
    - "What types of meetings should the bot join?"
    - Multi-select with examples:
        - ✓ Board meetings (recommended)
        - ✓ Cross-functional planning
        - ✓ Customer/stakeholder meetings
        - □ Team meetings
        - □ 1:1s (usually too intimate)
- Step 3c: Calendar Connection
    - OAuth buttons for: Zoom, Google Meet, Microsoft Teams
    - Permission explanation
    - "Your bot is ready!" success message
- Redirects to dashboard with message: "Waiting for your first meeting..."

Path B Flow (if selected):

- Redirects to baseline exercise sequence
- Brief intro: "Let's establish your baseline in 3 quick exercises (5 minutes total)"
- Exercise 1: Product Prioritization (60 seconds)
    - Prompt: "You have 3 features to build but only capacity for 1 this quarter. Explain your choice to your CEO."
    - Record response
- Exercise 2: Stakeholder Question Response (90 seconds)
    - Prompt: "Engineering says your timeline is unrealistic. How do you respond in a planning meeting?"
    - Record response
- Exercise 3: Trade-off Explanation (60 seconds)
    - Prompt: "The board asks why you chose speed over quality. Explain your reasoning."
    - Record response
- Processing screen: "Analyzing your baseline..." (mock AI processing)
- Results page: "Here's where you are now" with preliminary scores
- CTA: "Start your personalized learning path" → Dashboard
- Secondary CTA: "Want even more personalized learning? Connect your calendar."

Mock authentication logic:

typescript

```
// On signup:
const user = {
  id: crypto.randomUUID(),
  email: formData.email,
  name: formData.name,
  current_role: onboardingData.current_role,
  target_role: onboardingData.target_role,
  timeline: onboardingData.timeline,
  focus_areas: onboardingData.focus_areas,
  created_at: new Date().toISOString()
};
localStorage.setItem('shipspeak_user', JSON.stringify(user));
localStorage.setItem('shipspeak_auth_token', 'mock_token_' + user.id);

// On login:
// Check if email exists in mock users, return success
// In Phase 2, this becomes real auth with Supabase/NextAuth
```

User interactions:

- Type email/password → validation messages appear in real-time
- Click "Log in" → loading state → redirect to dashboard
- Click "Sign up" → navigate to signup page
- Complete signup → automatically proceeds to onboarding
- Complete onboarding → redirect to dashboard
- Back button works between onboarding steps

What user validates:

- Is the signup flow clear and not overwhelming?
- Does the career goal selection resonate?
- Is the onboarding too long or too short?
- Do they understand the value prop before signing up?

Acceptance criteria:

- □ Login form validates email format
- □ Signup form validates password strength
- □ Can't submit forms with invalid data
- □ Mock auth stores user data in localStorage
- □ Protected routes redirect to login if not authenticated
- □ Onboarding steps show progress indicator
- □ Can go back to previous onboarding steps
- □ Career level selector prevents selecting lower target than current
- □ Form data persists if user refreshes during onboarding
- □ Two-path fork displays both options clearly
- □ Path A button redirects to bot configuration wizard

- □ Path B button redirects to baseline exercise sequence
- □ Bot configuration wizard (Path A):
  - □ Bot name input accepts custom names
  - □ Preset options populate bot name field
  - □ Meeting type multi-select works correctly
  - □ Calendar OAuth buttons present (non-functional in Phase 1)
  - □ Success message shows after configuration
- □ Baseline exercise sequence (Path B):
  - □ All 3 exercises display correct prompts
  - □ Timer counts during each exercise
  - □ Recording interface works (mock recording in Phase 1)
  - □ Processing screen shows with animation
  - □ Baseline results page displays preliminary scores
- □ User choice is stored (Path A vs Path B) for analytics
- □ Responsive on mobile (forms are readable, inputs properly sized, two-path cards stack vertically)

Slice 3: Dashboard Layout & Empty States (3-4 hours)

What you deliver:

- Main dashboard page with overview widgets
- Empty states for each section (Meetings, Practice, Progress)
- Navigation between all main sections working
- Header showing user context and career level

Specific pages to build:

- `app/(dashboard)/page.tsx` - Main dashboard (default route after login)
- `app/(dashboard)/meetings/page.tsx` - Meetings section (empty for now)
- `app/(dashboard)/practice/page.tsx` - Practice section (empty for now)
- `app/(dashboard)/progress/page.tsx` - Progress section (empty for now)
- `app/(dashboard)/settings/page.tsx` - Settings section (empty for now)

Specific components to build:

- `components/dashboard/WelcomeCard.tsx` - Personalized greeting
- `components/dashboard/QuickStats.tsx` - Overview metrics (will populate in later slices)
- `components/dashboard/EmptyState.tsx` - Reusable empty state component
- `components/dashboard/CareerLevelBadge.tsx` - Shows current → target level

Main dashboard page UI:

- Welcome section at top:
  - Personalized greeting: "Welcome back, Sarah"
  - Current level badge: "Product Manager"
  - Target level with progress: "→ Senior PM (65% ready)"
- Quick stats row (3-4 cards):
  - Total meetings analyzed: 0 (empty state)
  - Practice sessions completed: 0 (empty state)
  - Skills improving: 0 (empty state)
  - Days streak: 0 (empty state)
- Two-column layout:
  - Left: Recent meetings (empty state)

    - Right: Recommended modules (empty state)
- CTA section:
    - "Connect your calendar to start analyzing meetings"
    - "Start your first practice exercise"

Empty state component design:

- Icon (relevant to section - calendar for meetings, mic for practice)
- Heading: Clear, friendly message
- Description: 1-2 sentences explaining what will be here
- Primary CTA button: What to do next
- Optional secondary action

Examples of empty states:

Meetings empty state:

- Icon: Calendar
- Heading: "No meetings analyzed yet"
- Description: "Connect your Zoom or Google Calendar to start getting insights on your communication"
- CTA: "Connect calendar"

Practice empty state:

- Icon: Microphone
- Heading: "Ready to practice?"
- Description: "Start with a quick exercise to establish your baseline. It takes just 2 minutes."
- CTA: "Start first exercise"

Progress empty state:

- Icon: Chart
- Heading: "Your progress dashboard"
- Description: "Complete your first meeting or practice session to see your skills and progress here"
- CTA: "View getting started guide"

Header enhancements:

- Left side: Page title (dynamically updates based on route)
- Center: (reserved for search in later phases)
- Right side:
    - Career level badge (clickable → shows framework modal)
    - Notification bell icon (0 for now)
    - User menu dropdown:
        - Profile
        - Settings
        - Help & Support
        - Log out

User interactions:

- Click "Connect calendar" → navigate to settings/integrations (shows coming soon)
- Click "Start first exercise" → navigate to practice page (will populate in Slice 8)
- Click career level badge → shows modal with full framework
- Click nav items → navigate to each section, see empty states
- Click user menu → see dropdown options

What user validates:

- Does the dashboard give a clear sense of what the product does?
- Are empty states motivating or confusing?
- Is the navigation structure logical?
- Does the career level visualization make sense?

Acceptance criteria:

- ☐ Dashboard shows personalized greeting with user's name
- ☐ Career level badge displays current and target role correctly
- ☐ All navigation links work and show appropriate empty states
- ☐ Empty states have clear CTAs
- ☐ Header updates page title based on current route
- ☐ User menu dropdown works correctly
- ☐ Career framework modal shows full PO → PM → Senior → Group → Director path
- ☐ Quick stats show 0 state with proper formatting
- ☐ Responsive layout (2-column becomes single column on mobile)
- ☐ Loading states show briefly when navigating between pages

---

Phase 1B: Meeting Intelligence (Week 2) - 12-15 hours

Slice 4: Meeting List with Mock Data (4 hours)

What you deliver:

- List view of past meetings with realistic mock data
- Filtering by meeting type, date range, participants
- Search functionality
- Sort options
- List vs grid view toggle

Specific components to build:

- `app/(dashboard)/meetings/page.tsx` - Meetings list page (replaces empty state)
- `components/meetings/MeetingCard.tsx` - Individual meeting card
- `components/meetings/MeetingFilters.tsx` - Filter controls
- `components/meetings/MeetingSearch.tsx` - Search bar
- `components/meetings/ViewToggle.tsx` - List/grid toggle
- `components/meetings/SortDropdown.tsx` - Sort options
- `lib/mockData/meetings.ts` - Mock meeting data

Meeting list page layout:

- Header section:
    - Page title: "Meetings"
    - View toggle (list/grid icons)
    - Sort dropdown
- Filter bar below header:
    - Search input (left)
    - Filter chips (right): Meeting type, Date range, Participants
- Main content area:
    - Meeting cards in list or grid layout
    - Pagination or infinite scroll (if >20 meetings)

- Sidebar (optional):
    - Quick filters (All, This week, This month, Board meetings, etc.)
    - Meeting stats summary

Meeting card UI (list view):

- Layout: Horizontal card, full width
- Left section (fixed width ~120px):
    - Date and time
        - Large text: "Nov 1"
        - Small text: "2:00 PM"
        - Relative time below: "3 days ago"
- Middle section (flex-grow):
    - Meeting title (bold, 18px): "Q4 Board Meeting"
    - Meeting type badge: "Board Meeting" (color-coded)
    - Participants row:
        - Avatar stack (max 5 visible, circular, 32px, overlapping)
        - "+3 more" text if >5 participants
        - Hover any avatar → tooltip with full name
    - Duration: "62 minutes" (icon + text)
- Right section (fixed width ~100px):
    - Overall score gauge (circular, shows 7.5/10)
    - Score trend: ↑ +0.8 (green) or ↓ -0.5 (red) compared to previous
    - "View details" link/button

Meeting card UI (grid view):

- Layout: Vertical card, ~300px wide
- Top: Score badge (prominent)
- Meeting title (bold)
- Date and time
- Participant avatars (fewer shown)
- Meeting type badge
- "View" button

Filter controls:

- **Meeting type filter:**
    - Dropdown multi-select
    - Options: Board Meeting, Team Meeting, 1:1, Customer Call, Planning Session, All-Hands
    - Shows count next to each option: "Board Meeting (3)"
    - Selected filters appear as removable chips
- **Date range filter:**
    - Date picker or presets dropdown
    - Presets: Today, This week, This month, Last 30 days, Custom range
    - Custom range opens date picker (start date, end date)
- **Participant filter:**
    - Autocomplete text input
    - Shows suggestions as you type (from all participants across meetings)
    - Selected participants appear as chips
    - Can select multiple

Search functionality:

- Searches meeting titles and participant names
- Real-time filtering as user types

- Debounced (waits 300ms after last keystroke)
- Shows "No results for '[query]'" if no matches
- Clear search button (X icon)

Sort options:

- Newest first (default)
- Oldest first
- Highest score
- Lowest score (for finding areas to improve)
- Longest duration
- Most recent (by when meeting occurred)

Mock data structure (10 meetings):

typescript

```
// lib/mockData/meetings.ts

export const mockMeetings = [
  {
    id: "mtg-001",
    title: "Q4 Board Meeting",
    date: "2025-11-01T14:00:00Z",
    duration_minutes: 62,
    meeting_type: "board_meeting",
    participants: [
      {
        id: "p-001",
        name: "Sarah Chen",
        role: "Product Manager",
        is_user: true,
        avatar_url: null // generates initials
      },
      {
        id: "p-002",
        name: "David Martinez",
        role: "CEO",
        is_user: false,
        avatar_url: null
      },
      {
        id: "p-003",
        name: "Lisa Wong",
        role: "CFO",
        is_user: false,
        avatar_url: null
      },
      {
        id: "p-004",
        name: "James Taylor",
        role: "Board Member",
        is_user: false,
        avatar_url: null
      }
    ],
    preliminary_scores: {
      overall: 7.5,
      product_sense: 8.0,
      communication: 7.2,
      stakeholder_management: 7.3,
      technical_translation: 7.8
    },
    score_trend: 0.8, // compared to previous similar meeting
    key_moments_count: 8,
    has_transcript: true,
    has_feedback: true,
    user_notes: "",
    tags: ["strategic", "quarterly"]
  },
  {
    id: "mtg-002",
    title: "Product Roadmap Planning",
    date: "2025-10-30T10:00:00Z",
    duration_minutes: 45,
    meeting_type: "team_meeting",
    participants: [
      { id: "p-001", name: "Sarah Chen", role: "Product Manager", is_user: true },
      { id: "p-005", name: "Alex Kumar", role: "Engineering Lead", is_user: false },
      { id: "p-006", name: "Maria Garcia", role: "Designer", is_user: false },
      { id: "p-007", name: "Tom Wilson", role: "Engineer", is_user: false }
    ],
    preliminary_scores: {
      overall: 8.2,
      product_sense: 8.5,
      communication: 8.0,
      stakeholder_management: 8.0,
      technical_translation: 8.3
    },
    score_trend: 0.3,
    key_moments_count: 12,
    has_transcript: true,
    has_feedback: true,
    user_notes: "",
    tags: ["planning", "eng-collab"]
  },
  // ... 8 more meetings with variety:
  // - Different types: 1:1s, customer calls, planning, all-hands
  // - Different scores: mix of high (8-9), medium (6-7), low (4-5)
```

```
  // - Different participant counts: 2 person 1:1, 15 person all-hands
  // - Different durations: quick 15min, long 90min
  // - Recent to older (past 30 days)
];
```

User interactions:

- View meetings list on load (newest first)
- Click meeting card → navigate to meeting detail page (Slice 5)
- Click filter → dropdown opens, select options
- Selected filters appear as chips below search
- Click chip X → remove that filter
- Type in search → list filters in real-time
- Click sort dropdown → change order
- Toggle list/grid view → layout changes
- Scroll down → lazy load more meetings (if >20)
- Hover participant avatar → see tooltip with name

What user validates:

- Does the meeting list feel manageable or overwhelming?
- Is the most important info (title, score, date) easy to scan?
- Are filters useful for finding specific meetings?
- Do scores give a quick sense of performance?
- Does participant avatar stack work visually?
- Is meeting type color coding intuitive?

Acceptance criteria:

- □ 10 distinct mock meetings with varied types, scores, participants
- □ Meeting cards display all required information
- □ Click meeting card navigates to detail page (can be empty placeholder for now)
- □ Filters work correctly:
  - □ Meeting type filter shows only matching types
  - □ Date range filter shows only meetings in range
  - □ Participant filter shows only meetings with that person
  - □ Multiple filters work together (AND logic)
- □ Search finds meetings by title or participant name (case insensitive)
- □ Sort options correctly reorder the list
- □ List/grid view toggle changes layout
- □ Score badges show correct colors (green 8-10, yellow 6-8, red <6)
- □ Participant avatar stack shows max 5, displays "+X more" correctly
- □ Hover tooltips work on avatars
- □ Empty state shows if no meetings match filters
- □ "Clear all filters" button appears when filters active
- □ Responsive on mobile (cards stack, filters collapse)
- □ No console errors or warnings

Slice 5: Meeting Detail & Transcript View (5-6 hours)

What you deliver:

- Full meeting detail page with transcript
- Speaker identification throughout transcript
- Ability to jump to specific moments
- Highlight system for key moments (strengths, improvement areas)
- Navigation back to meeting list

Specific components to build:

- `app/(dashboard)/meetings/[id]/page.tsx` - Meeting detail page
- `components/meetings/MeetingHeader.tsx` - Meeting metadata at top
- `components/meetings/TranscriptView.tsx` - Main transcript display
- `components/meetings/TranscriptSegment.tsx` - Individual speaker turn
- `components/meetings/TranscriptControls.tsx` - Search, filter, jump controls
- `components/meetings/KeyMomentMarker.tsx` - Highlights in transcript
- `components/meetings/SpeakerIdentifier.tsx` - Speaker label and avatar

Meeting detail page layout:

- **Header section (sticky at top):**
  - Back button to meetings list
  - Meeting title (editable on hover)
  - Meeting metadata in row:
    - Date and time: "Friday, November 1, 2025 at 2:00 PM"
    - Duration: "62 minutes"
    - Meeting type badge: "Board Meeting"
  - Participant list (expanded, shows all):
    - Each participant: avatar, name, role
    - User (you) highlighted/marked
  - Action buttons:
    - "Add notes"
    - "Share"
    - "Export transcript"
- **Two-column layout:**
  - **Left column (60% width): Transcript**
  - **Right column (40% width): AI Feedback Panel** (Slice 6)

Transcript view UI:

- Controls bar above transcript:
  - Search within transcript (input with icon)
  - Speaker filter dropdown (show all speakers, or select specific ones)
  - Jump to timestamp (time input)
  - Playback speed (if audio player added later)
- Transcript segments (conversation flow):
  - Each segment is a speaker turn
  - Layout per segment:

```
[Avatar] Speaker Name          14:32
This is what the speaker said. Could be multiple
sentences or paragraphs depending on how long they spoke.
```

- User's segments highlighted with subtle background color
- Hover any segment → shows "Copy" and "Link to this moment" actions

Speaker identification styling:

- User (Sarah Chen): Bold name, blue background highlight on their segments
- Other participants: Regular name, no background
- Each speaker gets consistent avatar/color throughout
- Speaker name includes role in smaller text below name

Key moment markers in transcript:

- Certain segments have colored left border and icon indicating:
    - 🟢 Green: Strong example ("Great answer-first structure here")
    - 🟡 Yellow: Could improve ("Trade-offs mentioned but not explored")
    - 🔴 Red: Missed opportunity ("Didn't address CEO's constraint concern")
- Clicking marker opens inline feedback explaining the moment
- These moments link to feedback panel on right

Transcript segment data structure:

typescript

```
interface TranscriptSegment {
  id: string;
  meeting_id: string;
  speaker: {
    id: string;
    name: string;
    role: string;
    is_user: boolean;
  };
  text: string;
  timestamp: string; // "14:32" format
  timestamp_seconds: number; // for jumping/searching
  duration_seconds: number;
  key_moment?: {
    type: 'strength' | 'improvement' | 'missed_opportunity';
    feedback: string;
    related_skill: string; // "trade-off-communication"
  };
}
```

Mock transcript data (for Q4 Board Meeting):

- 40-50 segments representing a realistic board meeting conversation
- Multiple back-and-forth exchanges
- User speaks 8-10 times (responding to questions, presenting, discussing)
- Varied segment lengths (short answers, longer explanations)
- Several key moments marked (3 strengths, 2 improvements, 1 missed opportunity)

Example conversation flow:

[David Martinez - CEO] 14:32
"Sarah, let's start with Q4 roadmap. What's your recommendation for our top priorities?"

[Sarah Chen - Product Manager (You)] 14:35
"Based on our user research and business goals, I recommend we focus on three things: mobile checkout flow, API performance, and analytics dashboard. The mobile checkout has the highest impact - we're seeing 60% of traffic on mobile but only 35% conversion rate compared to 52% on desktop."

🟢 Key Moment: Strong answer-first structure with data backing

[Lisa Wong - CFO] 14:58
"What's the revenue impact if we nail the mobile checkout?"

[Sarah Chen - Product Manager (You)] 15:03
"We estimate 15-20% increase in mobile conversions, which translates to roughly $2.3M additional ARR based on current traffic patterns. The engineering lift is about 6 weeks with the current team."

[James Taylor - Board Member] 15:22
"Why mobile checkout over the API work? I thought performance was becoming an issue."

[Sarah Chen - Product Manager (You)] 15:28
"Both are important, but mobile checkout gives us faster ROI. The API work is crucial for long-term scalability, but current performance is acceptable for our usage levels. We can tackle API in Q1 when we expect more enterprise customers."

🟡 Key Moment: Trade-offs mentioned but not fully explored. Could have acknowledged the engineering team's concerns about API technical debt more explicitly.

[David Martinez - CEO] 15:55
"What does the engineering team think?"

[Sarah Chen - Product Manager (You)] 16:00
"Alex's team is comfortable with the mobile work. They did

raise concerns about pushing API work, but we've agreed on some quick wins we can do in parallel."

🔴 Key Moment: Missed opportunity to directly address the board's question about eng team confidence. Could have brought Alex's specific feedback or concerns into the discussion.

Transcript controls functionality:

- **Search within transcript:**
  - Highlights matching text as user types
  - Shows match count: "3 results"
  - Next/previous buttons to jump between matches
- **Speaker filter:**
  - "Show all" (default)
  - "Show only Sarah Chen" (user)
  - Individual names (show only that person)
  - "Show only key moments" (segments with markers)
- **Jump to timestamp:**
  - Text input: "14:32" format
  - Clicking jumps to that segment and highlights it
- **Keyboard shortcuts:**
  - Cmd/Ctrl + F: Focus search
  - Arrow keys: Navigate segments
  - Cmd/Ctrl + Click segment: Copy text

User interactions:

- Read through entire transcript chronologically
- Click speaker name → filter to show only that speaker
- Search for specific words/topics → highlights appear
- Click key moment marker → inline feedback expands
- Click timestamp → copy link to share this specific moment
- Hover segment → shows copy and link actions
- Click user's segment → shows "Practice this response" CTA (future)
- Select text in transcript → "Create practice exercise from this" (future)

What user validates:

- Can they easily follow the conversation?
- Is speaker identification clear?
- Do key moments feel accurately identified?
- Can they find specific topics/moments quickly?
- Does seeing their exact words help them learn?
- Is the transcript overwhelming or digestible?

Acceptance criteria:

- □ Meeting header displays all metadata correctly
- □ Back button navigates to meetings list
- □ Transcript displays all segments in chronological order
- □ User's segments have distinct visual styling
- □ Each segment shows speaker name, role, avatar, timestamp

- □ Search within transcript works and highlights matches
- □ Speaker filter correctly shows only selected speaker(s)
- □ Jump to timestamp scrolls to correct segment
- □ Key moment markers appear on correct segments
- □ Clicking key moment expands inline feedback
- □ Copy segment functionality works
- □ Link to timestamp generates shareable URL
- □ Responsive on mobile (transcript + feedback panel stack vertically)
- □ Long transcript scrolls smoothly (virtualized if >100 segments)
- □ Loading state shows while mock data loads
- □ 404 state if meeting ID doesn't exist

Slice 6: AI Feedback Panel (5-6 hours)

What you deliver:

- Comprehensive AI feedback sidebar showing analysis of user's communication
- Score breakdowns with visualizations
- Pattern identification with examples
- Key moments linked to transcript
- Recommended learning modules
- Career level assessment

Specific components to build:

- `components/meetings/FeedbackPanel.tsx` - Main sidebar container
- `components/meetings/ScoreCard.tsx` - Individual metric with gauge
- `components/meetings/PatternList.tsx` - Communication patterns identified
- `components/meetings/InsightCard.tsx` - Individual insight with severity
- `components/meetings/KeyMomentsList.tsx` - Interactive list of key moments
- `components/meetings/ModuleRecommendations.tsx` - Suggested learning paths
- `components/meetings/CareerAssessment.tsx` - Level analysis
- `components/ui/GaugeChart.tsx` - Circular gauge for scores
- `components/ui/TrendIndicator.tsx` - Up/down with percentage

Feedback panel layout (sticky sidebar, scrollable):

Section 1: Overall Assessment (always visible at top)

- Large circular gauge (180px diameter) showing overall score:
  - Center: Score "7.5" (large, bold)
  - Ring: Color gradient from red (0) to green (10)
  - Label below: "Overall Communication"
- Trend indicator below gauge:
  - Icon: ↑ or ↓
  - Text: "+0.8 from last meeting" (green if up, red if down)
  - Subtle background color matching direction
- Quick summary (2-3 sentences):
  - "You demonstrated strong product thinking and data-backed decision making. Your answer-first structure kept the conversation focused. Main opportunity: exploring trade-offs more thoroughly with stakeholders."
- Career level badge:
  - "Speaking at Senior PM level in 65% of responses"
  - Progress bar showing 65% filled

    - Tooltip: "To consistently speak at Senior PM level, focus on trade-off articulation"

Section 2: Dimension Scores (expandable)

- Header: "Score Breakdown" with expand/collapse icon
- Four dimension cards (when collapsed, shows just score):
    - **Product Sense: 8.0/10**
        - Mini gauge (60px)
        - Trend: +0.5
    - **Communication: 7.2/10**
        - Mini gauge
        - Trend: +0.3
    - **Stakeholder Management: 7.3/10**
        - Mini gauge
        - Trend: -0.2 (shows in red)
    - **Technical Translation: 7.8/10**
        - Mini gauge
        - Trend: +1.0
- When expanded, each dimension shows subcomponents:
    - **Product Sense (8.0):**
        - Problem Framing: 8.5/10 ✓ (green check)
        - User Focus: 7.8/10 ✓
        - Market Awareness: 7.7/10 ✓
        - Data-Driven Decisions: 8.3/10 ✓
    - **Communication (7.2):**
        - Clarity: 8.0/10 ✓
        - Conciseness: 7.5/10 ✓
        - Structure: 6.5/10 ⚠️ (yellow warning)
        - Confidence: 7.0/10 ✓
    - **Stakeholder Management (7.3):**
        - Active Listening: 7.8/10 ✓
        - Addressing Concerns: 6.5/10 ⚠️
        - Building Consensus: 7.5/10 ✓
    - **Technical Translation (7.8):**
        - Simplifying Complexity: 8.0/10 ✓
        - Appropriate Depth: 7.5/10 ✓
        - Bridging Tech/Business: 8.0/10 ✓

Section 3: Patterns Identified (always expanded)

- Header: "What We Noticed"
- Three subsections:

Strengths (green section):

- Icon: ✓ (checkmark)
- List of positive patterns:

```
    ✓ Answer-first structure (4 of 5 responses)
      You consistently led with your recommendation before
      explaining rationale. This kept the board focused.
      Examples: 14:35, 18:45, 26:10

    ✓ Data-backed claims (all recommendations)
      Every recommendation included specific metrics or
      research findings. This builds credibility.
      Examples: 14:35 (conversion rates), 22:15 (user research)

    ✓ Confident delivery
      Minimal filler words (um, uh) and strong word choices
      throughout. Projected Senior PM-level confidence.
```

Areas to Improve (yellow section):

- Icon: ⚠️
- List of improvement opportunities:

```
    ⚠️ Incomplete trade-off exploration (2 instances)
      You mentioned alternative options but didn't fully
      explore the downsides or constraints. Board members
      asked follow-up questions that could have been addressed.
      Examples: 22:15, 31:40
      → Recommended module: "Communicating Trade-offs to Executives"

    ⚠️ Delayed acknowledgment of concerns
      When James raised API performance, you addressed it but
      after a pause. Acknowledging concerns immediately shows
      you're tracking all perspectives.
      Example: 15:28
```

Missed Opportunities (red section - only if critical):

- Icon: ⚠️ (red triangle)
- List of significant gaps:

⚠️ Didn't directly answer the CEO's question about eng confidence
David asked about engineering team buy-in. You mentioned Alex's comfort but didn't share his specific feedback or concerns. Bringing that detail would have shown stronger stakeholder alignment.
Example: 16:00
→ Practice: "Answer the question directly, then provide context"

Section 4: Key Moments (interactive list)

- Header: "8 Key Moments from This Meeting"
- Sortable: "Strengths first" or "Areas to improve first"
- Each moment card:

🟢 14:35 - Opening recommendation

Context: CEO asked for Q4 priorities

Your response: "Based on our user research and business goals, I recommend we focus on three things..."

What went well:
• Clear recommendation upfront (answer-first)
• Specific data to back up mobile focus (60% traffic, 35% conversion)
• Framed in terms of business impact

→ This is Senior PM level communication. Great example to reference.

[View in transcript →]

🟡 22:15 - Trade-off discussion

Context: James asked about mobile vs API priority

Your response: "Both are important, but mobile checkout
gives us faster ROI. The API work is crucial for long-term..."

What went well:
• Acknowledged both options
• Explained reasoning (ROI vs long-term)

What to improve:
• Could have been more explicit about API risks
• Engineering team concerns weren't surfaced
• Board might think API is less urgent than it is

Better approach:
"Mobile gives faster ROI, but I want to be transparent about
the API trade-off: we're building tech debt that will slow us
down in Q2. Alex's team is comfortable managing it short-term,
but we need to tackle it in Q1."

→ Practice this: "Communicating Trade-offs" module

[View in transcript →]

- Each card:
  - Color-coded left border (green/yellow/red)
  - Timestamp and context
  - Your response (truncated, expandable)
  - What went well / What to improve
  - Suggested action or module
  - Link that scrolls to transcript location

Section 5: Recommended Modules (bottom section)

- Header: "Personalized Learning Path"
- Subheader: "Based on this meeting, we recommend:"
- 2-3 module cards:

```
[Module Card 1]
🎯 Communicating Trade-offs to Executives

Why this matters:
You mentioned trade-offs in this meeting but didn't fully
explore downsides. Board members asked follow-up questions
that could have been preempted.

What you'll learn:
• The "Option-Impact-Risk" framework
• How to surface constraints proactively
• Practice with realistic scenarios

Time: 15 minutes | Difficulty: Intermediate
Relevance: 92% match to your gaps

[Start module →]
```

```
[Module Card 2]
🎯 Stakeholder Concern Acknowledgment

Why this matters:
James raised API concerns and you addressed them, but after
a pause. Immediate acknowledgment shows you're tracking all
perspectives in real-time.

What you'll learn:
• Active listening signals in meetings
• Techniques for instant concern validation
• Balancing multiple stakeholder views

Time: 12 minutes | Difficulty: Beginner
Relevance: 85% match

[Start module →]
```

Section 6: Career Level Assessment (expandable)

- Header: "Your Career Level Assessment"

- Visual breakdown:

```
Current Role: Product Manager
Speaking at:  Senior PM level in 65% of responses

[Progress bar: 65% filled, green]

To reach Senior PM consistently (80%+ of responses):

✓ Product Sense        Already at Senior PM level (8.0)
✓ Technical Translation  Already at Senior PM level (7.8)
⚠️ Communication        Near Senior PM (7.2) - Focus on structure
⚠️ Stakeholder Mgmt     Near Senior PM (7.3) - Address concerns faster

Next milestone: Complete "Trade-off Communication" module
and apply in your next board meeting to reach 75%.
```

Section 7: Next Steps (bottom of feedback panel)

- Header: "Your Development Plan from This Meeting"
- Subheader: "Here's what to do next based on what we learned:"
- Actionable next steps list (2-4 items, prioritized by impact)

Example Next Steps Layout:

📝 Next Steps

1. Practice Trade-off Communication (15 min) ⭐ High Impact

   Why: Based on moments at 22:15 and 31:40 where trade-offs weren't fully explored, practice the "Option-Impact-Risk" framework.

   Your gap: You mentioned options but didn't explore downsides, causing board members to ask follow-up questions.

   [Start Practice Exercise →]

2. Apply This Framework in Your Next Board Meeting

   Before your next board meeting, prepare trade-offs for each recommendation using this template:

   - Option A: [Benefit] but [Constraint/Risk]
   - Option B: [Benefit] but [Constraint/Risk]
   - My recommendation: [Choice] because [Reasoning]

   This Senior PM-level skill will prevent follow-up questions and demonstrate strategic thinking.

   [Save Framework Template →]
   [Set Reminder →]

3. Review Your Strong Example (2 min)

   Your opening at 14:35 was Senior PM level communication. Save this as a reference for answer-first structure.

   What made it strong:
   - Clear recommendation upfront
   - Data-backed reasoning (60% traffic, 35% conversion)
   - Framed in business impact terms

   [Bookmark This Moment →]
   [See Similar Examples →]

```
4. Track Progress (Ongoing)

   We'll monitor your trade-off communication in future meetings.
   Your goal: Show this skill in 80% of responses within 3 meetings.

   Current: 40% of responses showed trade-off exploration
   Target: 80% (Senior PM level)

   [View Progress Dashboard →]
```

Next Steps Components:

- Each step includes:
  - Number and title (clear action)
  - Impact indicator (⭐ High Impact, ⭐⭐ Medium Impact)
  - "Why" section: Ties to specific meeting moments
  - "Your gap" or "What made it strong": Explains the learning
  - Time estimate (if applicable)
  - Primary CTA button (action to take)
  - Optional secondary action

Next Steps Logic:

- High impact first: Address the biggest gap that's blocking career progression
- Quick wins included: Something they can do in 2-5 minutes to build momentum
- Application step: How to use this in real work (not just practice)
- Progress tracking: Clear metric to measure improvement

Next Steps Data Structure:

typescript

```
interface NextStep {
  id: string;
  order: number; // 1, 2, 3, 4
  title: string;
  impact: 'high' | 'medium' | 'low';
  type: 'practice' | 'apply_framework' | 'review_example' | 'track_progress';
  why: string; // Explanation tied to meeting
  gap_or_strength?: string; // What they need to work on or keep doing
  time_estimate_minutes?: number;
  related_meeting_moments: string[]; // Timestamps
  primary_cta: {
    text: string;
    action: 'start_module' | 'start_exercise' | 'bookmark' | 'view_dashboard' | 'save_template';
    target_id?: string; // module_id, exercise_id, etc.
  };
  secondary_cta?: {
    text: string;
    action: string;
    target_id?: string;
  };
  progress_metric?: {
    current: string;
    target: string;
    deadline?: string; // "within 3 meetings"
  };
}

interface MeetingFeedbackWithNextSteps extends MeetingFeedback {
  next_steps: NextStep[];
}
```

Feedback data structure:

typescript

```
interface MeetingFeedback {
  meeting_id: string;
  overall_score: number; // 0-10
  trend: number; // +/- compared to previous meeting
  summary: string; // 2-3 sentence overview

  career_assessment: {
    current_role: string;
    target_role: string;
    speaking_at_level: string;
    consistency_percentage: number; // % of responses at target level
    breakdown: {
      skill: string;
      current_level: string;
      status: 'achieved' | 'near' | 'gap';
    }[];
    next_milestone: string;
  };

  dimension_scores: {
    product_sense: DimensionScore;
    communication: DimensionScore;
    stakeholder_management: DimensionScore;
    technical_translation: DimensionScore;
  };

  patterns: {
    strengths: Pattern[];
    improvements: Pattern[];
    missed_opportunities: Pattern[];
  };

  key_moments: KeyMoment[];

  recommended_modules: RecommendedModule[];
}

interface DimensionScore {
  score: number; // 0-10
  trend: number; // +/- from previous
  subcomponents: {
    name: string;
    score: number;
    status: 'strong' | 'adequate' | 'needs_work';
  }[];
}

interface Pattern {
  id: string;
  pattern_type: string; // "answer_first", "incomplete_tradeoffs"
  description: string; // What we noticed
  examples: string[]; // Array of timestamps ["14:35", "18:45"]
  impact?: string; // Why this matters
  recommendation?: string; // What to do about it
}

interface KeyMoment {
  id: string;
  timestamp: string;
  segment_id: string; // Links to transcript segment
  type: 'strength' | 'improvement' | 'missed_opportunity';
  context: string; // "CEO asked for Q4 priorities"
  your_response: string; // What you said (truncated)
  what_went_well?: string[];
  what_to_improve?: string[];
  better_approach?: string; // Example of improved response
  related_module_id?: string;
  career_level_relevance: string; // "This is Senior PM level"
}

interface RecommendedModule {
  module_id: string;
  title: string;
  why_recommended: string;
  what_you_learn: string[];
  estimated_minutes: number;
  difficulty: 'beginner' | 'intermediate' | 'advanced';
  relevance_score: number; // 0-1, how well it matches their gaps
}
```

User interactions:

- Scroll through feedback panel independently of transcript

- Hover over score → tooltip explains what goes into that score
- Click dimension card → expands to show subcomponents
- Click pattern → highlights corresponding transcript segments
- Click key moment card → scrolls transcript to that moment
- Click "View in transcript" → jumps to exact timestamp
- Click "Start module" → navigates to module page
- Click "Share feedback" → generates shareable link or PDF
- Toggle between "Show all patterns" and "Show top 3 patterns"
- Collapse/expand sections to focus on what matters
- **Next Steps interactions:**
    - Click "Start Practice Exercise" → navigates to relevant practice module
    - Click "Save Framework Template" → downloads or saves template to library
    - Click "Set Reminder" → creates calendar reminder for next meeting
    - Click "Bookmark This Moment" → saves moment to favorites
    - Click "View Progress Dashboard" → navigates to progress page
    - Each next step can be marked as complete (checkbox)

What user validates:

- Is the feedback specific and actionable?
- Do the scores feel accurate given the transcript?
- Are patterns identified actually present in the conversation?
- Do recommended modules address real gaps?
- Is the career level assessment motivating?
- Does linking feedback to transcript moments help learning?
- Is there too much information or too little?

Acceptance criteria:

- □ Overall score displays correctly with gauge visualization
- □ Trend indicator shows correct direction and amount
- □ Summary text is specific to the meeting
- □ All dimension scores calculate and display correctly
- □ Subcomponents expand/collapse smoothly
- □ Pattern lists show distinct strengths vs improvements
- □ Each pattern links to correct transcript examples
- □ Clicking pattern highlights transcript segments
- □ Key moments list shows all moments in order
- □ Clicking key moment scrolls transcript to correct location
- □ Module recommendations show relevance score
- □ "Start module" button navigates correctly
- □ Career assessment calculates consistency percentage correctly
- □ **Next Steps section displays 2-4 prioritized actions**
- □ **Each next step shows impact indicator**
- □ **Next step CTAs navigate to correct destinations**
- □ **"Why" explanation ties to specific meeting moments**
- □ **Time estimates show for practice/review steps**
- □ **Progress metrics display current vs target**
- □ **Can mark next steps as complete**
- □ Feedback panel is sticky/scrollable independently
- □ Responsive on mobile (becomes bottom drawer or stacked)
- □ All tooltips work on hover
- □ Loading states for each section

- □ No performance issues with large feedback datasets
- □ Can export feedback as PDF or shareable link

Phase 1C: Learning Modules & Practice (Week 3) - 12-15 hours

Slice 7: Module Library & Recommendations (4 hours)

What you deliver:

- Module library page showing personalized learning paths
- Modules organized by skill category
- Priority ranking based on meeting analysis
- Connection between meeting patterns and module recommendations
- Module card design with key information

Specific components to build:

- `app/(dashboard)/learn/page.tsx` - Module library main page
- `components/learn/ModuleCard.tsx` - Individual module card
- `components/learn/ModuleCategoryFilter.tsx` - Filter by skill area
- `components/learn/ModulePriorityIndicator.tsx` - Shows impact/relevance
- `components/learn/RecommendationReason.tsx` - Why this module was recommended
- `lib/mockData/modules.ts` - Mock module library

Module library page layout:

- Header: "Your Learning Path"
- Subheader: "Personalized modules based on your meetings and career goals"
- Filter tabs:
  - All Modules
  - Recommended for You (default)
  - Product Sense
  - Executive Communication
  - Stakeholder Management
  - Technical Translation
  - Trade-off Articulation
- Sort options: Priority, Duration (shortest first), Difficulty

Module card UI:

- Impact badge: ⭐⭐⭐ High Impact (or ⭐⭐ Medium, ⭐ Low)
- Module title: "Communicating Trade-offs to Executives"
- Difficulty badge: Beginner / Intermediate / Advanced
- Duration: "15 minutes"
- Why recommended (personalized):
  - "Based on your Q4 board meeting, you mentioned trade-offs but didn't fully explore downsides."
  - Links to specific meeting: "See example from [Q4 Board Meeting →]"
- What you'll learn (3-4 bullet points):
  - The "Option-Impact-Risk" framework
  - How to surface constraints proactively
  - Practice with realistic scenarios
- Relevance score: "92% match to your gaps"
- Progress indicator: Not Started / In Progress (30% complete) / Completed ✓
- Primary CTA: "Start Module" or "Continue Module"

- Career level tag: "Critical for PM → Senior PM transition"

Module categories with examples:

Product Sense & Strategy:

- User Problem Framing
- Market Opportunity Sizing
- Competitive Positioning
- Product Vision Communication
- Metrics that Matter

Executive Communication:

- Answer-First Structure
- Communicating with C-Suite
- Presenting to Boards
- Executive Updates
- Saying No to Stakeholders

Stakeholder Management:

- Building Consensus
- Managing Conflicting Priorities
- Influence Without Authority
- Stakeholder Mapping
- Concern Acknowledgment

Technical Translation:

- Explaining Technical Concepts to Non-Technical Audiences
- Bridging Engineering and Business
- Architecture Decision Communication
- Technical Feasibility Discussion
- Scope Negotiation with Engineering

Trade-off Articulation:

- The Option-Impact-Risk Framework
- Speed vs Quality Decisions
- Feature Prioritization Communication
- Resource Constraint Discussion
- Strategic Trade-offs

Personalization logic (mock):

- Module priority determined by:
    1. Gap severity from meeting analysis (high weight)
    2. Career level relevance (is this skill needed for target role?)
    3. Frequency of issue (appeared in multiple meetings?)
    4. User's stated focus areas (from onboarding)
- High impact modules: Address skills blocking career progression
- Medium impact: Nice-to-have improvements
- Low impact: General skill development

Module data structure:

typescript

```typescript
interface Module {
  id: string;
  title: string;
  category: 'product_sense' | 'communication' | 'stakeholder_mgmt' | 'technical_translation' | 'tradeoff_articulation';
  difficulty: 'beginner' | 'intermediate' | 'advanced';
  estimated_minutes: number;
  impact_for_user: 'high' | 'medium' | 'low';
  relevance_score: number; // 0-1

  recommendation_reason?: {
    personalized_message: string;
    related_meeting_id?: string;
    related_meeting_moments?: string[]; // timestamps
    gap_identified: string;
  };

  what_you_learn: string[]; // 3-5 bullet points

  career_level_tags: string[]; // "Critical for PM → Senior PM"

  user_progress: {
    status: 'not_started' | 'in_progress' | 'completed';
    completion_percentage: number;
    last_accessed?: string;
    exercises_completed: number;
    exercises_total: number;
  };

  exercises: Exercise[];
}
```

User interactions:

- View all modules or filter by category
- Click module card → navigate to module detail page (Slice 8)
- Click "See example from meeting" → opens meeting detail at that timestamp
- Sort by priority, duration, or difficulty
- Search modules by keyword
- Bookmark/favorite modules for later
- Filter by progress status (Not started, In progress, Completed)

What user validates:

- Are recommended modules actually addressing their gaps?
- Is the connection between meetings and modules clear?
- Does the priority ranking make sense?
- Is the "why recommended" explanation compelling?
- Do they feel motivated to start learning?

Acceptance criteria:

- □ Module library displays 15-20 mock modules
- □ Modules organized by category with working filters
- □ "Recommended for You" tab shows only high-relevance modules
- □ Each module card displays all required information
- □ Impact badges show correct priority levels
- □ Recommendation reasons are personalized and specific
- □ Links to meeting moments work correctly
- □ Sort options reorder modules correctly
- □ Search finds modules by title or keyword
- □ Progress indicators show correct status
- □ "Start Module" button navigates to module detail
- □ Career level tags display for relevant modules
- □ Responsive on mobile (cards stack vertically)
- □ Loading state shows on initial load

Slice 8: Module Content & Practice Exercises (5-6 hours)

What you deliver:

- Module detail page with learning content
- Framework explanations with examples
- Multiple practice exercises within each module
- Different exercise types (timed, scenario-based, role-specific)
- Exercise selection and preview

Specific components to build:

- `app/(dashboard)/learn/[moduleId]/page.tsx` - Module detail page
- `components/learn/ModuleContent.tsx` - Learning content display
- `components/learn/FrameworkExplanation.tsx` - Framework with examples
- `components/learn/ExerciseList.tsx` - List of practice exercises
- `components/learn/ExerciseCard.tsx` - Individual exercise preview
- `components/learn/ExerciseTypeIndicator.tsx` - Shows exercise format

Module detail page layout:

Header section:

- Back button to module library
- Module title: "Communicating Trade-offs to Executives"
- Difficulty + Duration: "Intermediate • 15 minutes"
- Progress bar: "2 of 5 exercises completed"
- "Why this matters for you" callout:

  - "In your Q4 board meeting, you presented options without fully exploring downsides. This module will help you communicate trade-offs more effectively."

Content section (tabs):

Tab 1: Learn (Framework & Concepts)

- Section 1: The Problem
  - Real example from user's meeting (if available)
  - Why this skill matters for career progression
  - Common mistakes PMs make
- Section 2: The Framework
  - "Option-Impact-Risk" framework explanation
  - Visual diagram of the framework
  - Example breakdown:

```
Option A: Build mobile app
Impact: +$2M ARR, 60% of users on mobile
Risk: 6-month timeline, delays API work

Option B: Focus on API performance
Impact: Better enterprise scalability
Risk: No immediate revenue, mobile gap continues

My recommendation: Mobile app first because...
[Clear reasoning with trade-off acknowledgment]
```

- Section 3: Best Practices
  - Do's and Don'ts
  - Example of strong trade-off communication
  - Example of weak trade-off communication (what to avoid)
- Section 4: Apply to Your Work
  - Template for next board meeting
  - Checklist of what to include
  - Common board questions and how to handle them

Tab 2: Practice (Exercises)

- Exercise list (5-8 exercises per module)
- Each exercise shows:
  - Exercise number and title
  - Format type (Timed response, Scenario-based, Role-specific)
  - Difficulty indicator
  - Time estimate
  - Completion status (Not started / Completed ✓)
  - "Start Exercise" button

Exercise types:

Type 1: Timed Response (60-90 seconds)

- Clear prompt with scenario
- Timer counts down
- Record audio response
- Immediate transcript + AI feedback

Example:

```
Exercise 1: Product Prioritization Trade-off

Scenario: You're presenting to your board. You have capacity
for one major initiative this quarter: either mobile checkout
(high revenue potential) or API performance (technical debt).

Your task: In 60 seconds, present your recommendation and
explicitly explain the trade-offs.

Constraints:
- Board cares about revenue and long-term scalability
- Engineering team is concerned about API stability
- You must acknowledge what you're NOT doing

[Start Recording →]
```

Type 2: Scenario-Based (Multi-turn)

- Interactive scenario with decision points
- User responds to different stakeholders
- Branching based on their choices
- Summary feedback at end

Example:

```
Exercise 3: Handling Stakeholder Objections

You're in a planning meeting. You've recommended focusing
on mobile checkout over API work.

[Engineering Lead]: "I'm concerned about API technical debt.
This will cause problems in Q2."

How do you respond?
A) Acknowledge concern and explain trade-off
B) Defend your decision with data
C) Suggest a compromise

[Record your response or select →]
```

Type 3: Compare and Improve

- Shows two versions of the same communication
- User identifies what makes one better
- Then records their own improved version

Example:

```
Exercise 5: Evaluate Trade-off Communication

Version A (Weak):
"I think we should do mobile. API can wait."

Version B (Strong):
"I recommend mobile checkout first. While API performance
is important, mobile gives us faster ROI ($2M ARR). We'll
build technical debt, but Alex's team can manage it short-term.
We'll tackle API in Q1."

What makes Version B stronger?
[Multiple choice or free response]

Now record your own version:
[Start Recording →]
```

Exercise card UI:

- Exercise number: "Exercise 1 of 5"
- Title: "Product Prioritization Trade-off"
- Format badge: "Timed • 60 seconds"
- Difficulty: Intermediate
- Preview of scenario (first 2 lines)
- Completion status: Not started / In progress / Completed ✓
- "Start Exercise" button

Exercise data structure:

typescript

```
interface Exercise {
  id: string;
  module_id: string;
  order: number; // 1, 2, 3...
  title: string;
  format: 'timed_response' | 'scenario_based' | 'compare_improve' | 'framework_application';
  difficulty: 'beginner' | 'intermediate' | 'advanced';
  estimated_minutes: number;

  prompt: {
    scenario: string;
    context: string;
    task: string;
    constraints?: string[];
  };

  time_limit_seconds?: number; // For timed exercises

  options?: { // For scenario-based
    id: string;
    text: string;
    feedback: string;
  }[];

  evaluation_criteria: string[]; // What AI looks for

  example_responses?: {
    weak: string;
    strong: string;
    explanation: string;
  };

  user_completion: {
    status: 'not_started' | 'in_progress' | 'completed';
    attempts: number;
    best_score?: number;
    last_attempt?: string;
  };
}
```

User interactions:

- Read through framework content
- Toggle between Learn and Practice tabs
- Click exercise card → navigate to exercise page (Slice 9)
- Preview exercise scenario
- Track progress through module (X of Y exercises completed)
- Bookmark framework for reference
- Download template/checklist
- Return to module library

What user validates:

- Is the framework easy to understand?
- Are examples relatable to their work?
- Do practice exercises feel realistic?
- Is the difficulty progression appropriate?
- Do they feel equipped to apply this in real meetings?

Acceptance criteria:

- ☐ Module detail page displays complete learning content
- ☐ Framework explanation includes visual diagram
- ☐ Real examples show both weak and strong communication
- ☐ "Why this matters for you" section is personalized
- ☐ Practice tab shows 5-8 exercises per module
- ☐ Exercise cards display all metadata correctly
- ☐ Different exercise types have distinct visual indicators
- ☐ Progress bar updates based on completed exercises
- ☐ Can switch between Learn and Practice tabs smoothly
- ☐ Template/checklist downloads correctly
- ☐ "Start Exercise" button navigates to exercise interface
- ☐ Responsive on mobile (content readable, exercises accessible)
- ☐ Can bookmark or save framework for reference

Slice 9: Practice Recording Interface (3-4 hours)

What you deliver:

- Professional audio recording interface
- Exercise prompt display during recording
- Timer and recording controls
- Visual feedback (waveform/level meter)
- Ability to review and re-record

Specific components to build:

- `app/(dashboard)/practice/[exerciseId]/page.tsx` - Practice exercise page
- `components/practice/ExercisePrompt.tsx` - Displays exercise scenario
- `components/practice/RecordingInterface.tsx` - Main recording component
- `components/practice/RecordingControls.tsx` - Start, pause, stop, redo
- `components/practice/AudioVisualizer.tsx` - Waveform or level meter
- `components/practice/RecordingTimer.tsx` - Shows elapsed/remaining time
- `components/practice/RecordingPreview.tsx` - Playback after recording

Practice page layout (3 states):

State 1: Pre-Recording

- Exercise header:
    - Module breadcrumb: "Trade-off Communication > Exercise 2"
    - Exercise title: "Present Product Prioritization to Board"
    - Format + Time: "Timed Response • 60 seconds"
- Exercise prompt (large, prominent card):
    - Scenario section:
        - Icon: 👥 Meeting context
        - Text: "You're presenting to your board. You have capacity for one major initiative..."
    - Context section:
        - What the audience cares about
        - Key constraints
        - What you must address
    - Your task (highlighted):
        - Clear instruction
        - Time limit reminder
        - Success criteria
- Tips section (collapsible):
    - Framework to use: "Option-Impact-Risk"
    - Key points to hit:
        - State your recommendation first
        - Explain each option's impact
        - Acknowledge what you're NOT doing
        - Show you understand constraints
    - Avoid: Long preambles, ignoring trade-offs
- Ready section:
    - Microphone test: "Test your mic" button (plays back audio)
    - Large primary button: "Start Recording"
    - Secondary actions:
        - "View example response" (shows strong example)
        - "Skip exercise"

State 2: Recording

- Exercise prompt (condensed at top, always visible)
- Large recording indicator:
    - Pulsing red dot
    - "Recording..." text
    - Large timer:
        - Shows elapsed: "00:34 / 01:00"
        - Turns orange at 10 seconds remaining
        - Turns red when time exceeded
- Audio visualizer (center):
    - Real-time waveform showing sound levels
    - OR circular level meter
    - Confirms mic is picking up audio
- Controls (bottom):
    - Large "Stop" button (primary)
    - "Pause" button (if applicable)
    - "Cancel" link
- Visual cues:
    - Background slightly dim to focus attention

    - Countdown at 10 seconds: "10..." "9..." "8..."
    - Auto-stop at time limit (if set)

State 3: Review

- Audio playback:
    - Play/pause button
    - Scrub timeline
    - Duration displayed: "01:07"
    - Volume control
- Transcript (generated immediately in Phase 1 mock):
    - Full text of what they said
    - Timestamps every 10 seconds
    - Scrollable
- Evaluation preview:
    - Quick assessment: "You hit 3 of 4 key points"
    - What you included:
        - ✓ Clear recommendation
        - ✓ Acknowledged trade-offs
        - ✓ Explained reasoning
        - ✗ Didn't address engineering concerns
    - Estimated score: 7.5/10 (not final)
- Actions:
    - Primary button: "Get Full Feedback" (navigates to Slice 10)
    - Secondary button: "Record Again"
    - Tertiary link: "Save for Later"

Technical considerations (Phase 1 mock):

- Browser MediaRecorder API for audio capture
- Microphone permission handling
- Audio format: WebM or MP4 (browser dependent)
- Mock transcription (instant in Phase 1)
- Visual feedback that mic works
- Prevent navigation during recording
- Auto-save if user closes tab

User interactions:

- Read exercise prompt thoroughly
- Collapse/expand tips section
- Test microphone before starting
- View example response
- Click "Start Recording" → immediately begins
- See real-time feedback (waveform, timer)
- Pause mid-recording (optional)
- Stop when finished
- Listen to playback
- Read transcript
- See quick evaluation
- Decide to get feedback or re-record
- Re-record unlimited times

What user validates:

- Does the recording interface feel professional?

- Is it clear when they're recording vs not?
- Do they feel comfortable practicing out loud?
- Is the timer helpful or anxiety-inducing?
- Do they understand the exercise prompt?
- Is the quick evaluation preview useful?

Acceptance criteria:

- □ Exercise prompt displays all scenario details clearly
- □ Microphone permission requested appropriately
- □ Test mic function works
- □ Example response shows strong communication
- □ "Start Recording" immediately begins capture
- □ Audio visualizer shows real-time levels
- □ Timer counts correctly (up or down depending on format)
- □ Timer shows warnings at appropriate times
- □ Stop button ends recording immediately
- □ Playback controls work correctly
- □ Transcript generates (mock instant, real 5-10s in Phase 2)
- □ Quick evaluation shows key points hit/missed
- □ Can re-record unlimited times
- □ Each recording is saved (can review attempts)
- □ Confirmation dialog if trying to leave during recording
- □ Works in Chrome, Safari, Firefox
- □ Mobile: Uses device microphone correctly
- □ Audio quality sufficient for transcription

Slice 10: Practice Feedback Display (4-5 hours)

What you deliver:

- Detailed AI feedback on practice session
- Comparison to meeting performance (if available)
- Specific strengths and areas to improve
- Score breakdown by criteria
- Next practice recommendation

Specific components to build:

- `app/(dashboard)/practice/[exerciseId]/feedback/page.tsx` - Feedback page
- `components/practice/FeedbackOverview.tsx` - Score and summary
- `components/practice/TranscriptWithAnnotations.tsx` - Annotated transcript
- `components/practice/CriteriaBreakdown.tsx` - Score by evaluation criteria
- `components/practice/ComparisonToMeetings.tsx` - Progress vs real meetings
- `components/practice/NextPracticeRecommendation.tsx` - What to do next

Practice feedback page layout:

Header section:

- Back to module button
- Exercise title: "Product Prioritization Trade-off"
- Attempt number: "Attempt 2 of 3" (if multiple)

- Date/time: "Today at 2:34 PM"

Section 1: Overall Assessment

- Large score gauge: 7.8/10
- Improvement indicator: "+0.5 from previous attempt" (if applicable)
- Quick summary:
  - "Strong improvement! You clearly stated your recommendation and explained trade-offs. Main opportunity: address stakeholder concerns more proactively."
- Comparison to baseline:
  - "When you started: 6.5/10"
  - "Now: 7.8/10"
  - Progress bar showing improvement

Section 2: Your Response (Annotated Transcript)

- Full transcript of their practice session
- Inline annotations highlighting:
  - 🟢 Strong moments: "Great answer-first structure here"
  - 🟡 Could improve: "Trade-off mentioned but not fully explored"
  - 🔴 Missing: "Didn't acknowledge engineering concerns"

Example:

📋 ✓

[00:00] "I recommend we prioritize mobile checkout this quarter."

🟢 Strong opening: Clear recommendation upfront (answer-first structure)

[00:05] "This will generate approximately $2M in additional ARR based on our current mobile traffic patterns."

🟢 Data-backed reasoning: Specific metrics strengthen credibility

[00:15] "The alternative is API performance work, which is important for long-term scalability."

🟡 Acknowledgment but incomplete: You mentioned the alternative but didn't explain why you're deprioritizing it. Board members would likely ask "Why not API?"

Better approach: "API work is crucial, but mobile gives us faster ROI. We're building technical debt, but the engineering team can manage it short-term."

[00:25] "We'll tackle API work in Q1 next year."

🔴 Missing: No acknowledgment of engineering team's concerns about technical debt. Bringing this up proactively shows stakeholder awareness.

Add: "Alex's team is comfortable with this plan, though they've flagged that we need to address API by Q1."

Section 3: Evaluation Criteria Breakdown

- Score for each criterion (what the exercise was testing)

Example for "Trade-off Communication" exercise:

Answer-First Structure 9.0/10 ✓

You led with your recommendation before explaining.

Data-Backed Reasoning 8.5/10 ✓

Strong use of specific metrics ($2M ARR, traffic patterns).

Trade-off Articulation 7.0/10 ⚠

You mentioned alternatives but didn't fully explore downsides. Board members would likely ask follow-up questions.

Stakeholder Acknowledgment 6.5/10 ⚠

Engineering concerns weren't addressed proactively. Bringing this up shows you've done your homework.

Conciseness 8.0/10 ✓

Response was appropriately brief for a board meeting.

Overall Communication Score 7.8/10

Section 4: Comparison to Real Meetings

- If user has meeting data, compare practice to actual performance
- "How you're improving"

Example:

```
Your Communication Patterns

In your Q4 board meeting:
- Answer-first: Used in 4 of 5 responses ✓
- Trade-off articulation: Mentioned but incomplete (2 instances)
- Stakeholder acknowledgment: Delayed response to concerns

In this practice:
- Answer-first: ✓ Strong (improved)
- Trade-off articulation: ⚠️ Better but still incomplete
- Stakeholder acknowledgment: Missing (focus area)

🎯 Key insight: You're improving on structure and data-backing,
but trade-off exploration remains your growth area. Practice
2 more exercises in this module to build consistency.
```

Section 5: Strong Example (Reference)

- Model response for this exercise
- Annotations explaining what makes it strong
- "Compare to your response" toggle

Example:

📋 ✓

Expert Response: (Senior PM level)

"I recommend we prioritize mobile checkout this quarter.
[Answer-first ✓]

Here's the trade-off: Mobile generates $2M ARR and addresses our highest-traffic channel. API performance is important for scalability, and pushing it means we're building technical debt. [Clear trade-off articulation ✓]

Alex's engineering team is comfortable managing API stability short-term, but we need to tackle it in Q1 before it impacts enterprise customers. [Stakeholder acknowledgment ✓]

The risk is that we delay long-term scalability for short-term revenue, but given our current growth stage, I believe that's the right trade. [Risk acknowledgment ✓]

What questions do you have about this recommendation?"
[Invites discussion ✓]

Section 6: Next Steps

- Immediate actions based on feedback

Example:

📋 ✓

```
🎯 To improve further:

1. Practice Trade-off Exploration (10 min)
   You're mentioning trade-offs but not fully exploring them.
   Try Exercise 3: "Handling Board Questions on Trade-offs"

   [Start Exercise 3 →]

2. Review This Strong Example
   The expert response above shows how to proactively address
   concerns. Study the stakeholder acknowledgment section.

   [Bookmark Example →]

3. Apply in Your Next Meeting
   Before your next board meeting, prepare trade-offs using
   the Option-Impact-Risk template.

   [Download Template →]
```

Feedback data structure:

typescript

```
interface PracticeFeedback {
  exercise_id: string;
  user_id: string;
  attempt_number: number;
  recorded_at: string;

  recording_url: string;
  transcript: string;
  duration_seconds: number;

  overall_score: number; // 0-10
  improvement_from_previous?: number;
  improvement_from_baseline?: number;

  summary: string;

  criteria_scores: {
    criterion_name: string;
    score: number;
    feedback: string;
    status: 'strong' | 'adequate' | 'needs_work';
  }[];

  annotated_moments: {
    timestamp: number;
    text: string;
    annotation_type: 'strength' | 'improvement' | 'missing';
    feedback: string;
    better_approach?: string;
  }[];

  comparison_to_meetings?: {
    pattern: string;
    in_meetings: string; // "Used in 4 of 5 responses"
    in_practice: string; // "Strong (improved)"
    trend: 'improving' | 'consistent' | 'regressing';
  }[];

  expert_example: {
    text: string;
    annotations: {
      moment: string;
      explanation: string;
    }[];
  };

  next_steps: NextStep[];
}
```

User interactions:

- Read through overall assessment
- Scroll through annotated transcript
- Click annotation → see detailed feedback
- Expand/collapse criteria scores
- View comparison to real meetings
- Toggle between their response and expert example
- Click "Start next exercise"
- Download template
- Share feedback (save for reference)
- Compare multiple attempts side-by-side

What user validates:

- Is the feedback specific enough to improve?
- Do annotations help them understand what to do differently?
- Is the expert example helpful as a reference?
- Does comparison to meetings show actual progress?
- Are next steps clear and actionable?
- Do they feel motivated to continue practicing?

Acceptance criteria:

- □ Overall score displays with visual gauge
- □ Improvement indicator shows if score increased
- □ Summary is specific to their response
- □ Full transcript displays correctly
- □ Annotations appear at correct timestamps
- □ Each annotation includes feedback and better approach
- □ Criteria breakdown shows all evaluation dimensions
- □ Scores for each criterion are accurate
- □ Comparison to meetings section appears (if meeting data exists)
- □ Patterns show improving/consistent/regressing trend
- □ Expert example displays with annotations
- □ Toggle between user response and expert works
- □ Next steps are actionable and prioritized
- □ "Start next exercise" button works
- □ Can save/bookmark feedback for reference
- □ Responsive on mobile (readable, scrollable)
- □ Loading state shows while generating feedback

Phase 1D: Progress & Career Tracking (Week 4) - 10-12 hours

Slice 11: Progress Dashboard (5-6 hours)

What you deliver:

- Comprehensive progress dashboard showing skill development
- Career level tracking with visual progress indicators
- Meeting and practice performance trends
- Module completion tracking
- Insights and highlights

Specific components to build:

- `app/(dashboard)/progress/page.tsx` - Progress dashboard main page
- `components/progress/CareerProgressCard.tsx` - Current → target level
- `components/progress/SkillRadarChart.tsx` - Multi-dimensional skill view
- `components/progress/PerformanceTrends.tsx` - Line charts over time
- `components/progress/ModuleCompletionTracker.tsx` - Learning progress
- `components/progress/RecentHighlights.tsx` - Notable improvements
- `components/progress/WeeklyInsights.tsx` - Summary of the week

Progress dashboard layout:

Header section:

- Page title: "Your Progress"
- Date range selector: "Last 30 days" (default), Last 7 days, Last 90 days, All time
- Export button: "Download Report" (PDF)

Section 1: Career Level Progress (Top, prominent)

- Large visual showing career journey:

```
Product Manager ━━━━━━━━●━━━━━━━━━━ Senior PM
                 65%

You're speaking at Senior PM level in 65% of responses

Current level: Product Manager
Target level: Senior Product Manager
Timeline: Reach consistently (80%+) in 6 months

Progress breakdown:
✓ Product Sense        100%  Already at Senior PM level
✓ Technical Translation  95%   Nearly consistent
⚠️ Communication        70%   Improving steadily
⚠️ Stakeholder Mgmt     60%   Focus area

Next milestone: Reach 75% by completing Trade-off module
and applying in next 2 board meetings
```

- Visual progress bar with segments for each skill
- Trend indicator: "↑ +12% this month"

Section 2: Skill Radar Chart

- Multi-dimensional view of skills:
  - Product Sense: 8.2/10
  - Communication: 7.5/10
  - Stakeholder Management: 7.0/10
  - Technical Translation: 8.0/10
  - Trade-off Articulation: 6.8/10
- Overlay comparing:
  - Current level (solid line)
  - 30 days ago (dotted line)
  - Target level (Senior PM benchmark, dashed line)
- Shows which skills improved, which are stable, which need focus

Section 3: Performance Trends

- Two charts side by side:

Chart 1: Meeting Performance Over Time

- Line chart showing overall scores from last 10 meetings
- X-axis: Meeting dates
- Y-axis: Score (0-10)
- Trend line showing improvement direction
- Annotations on key meetings: "Q4 Board Meeting - 7.5"

Chart 2: Practice Performance Over Time

- Similar chart for practice sessions
- Shows learning curve
- Compares to meeting performance (are they practicing and improving?)

Section 4: Module Completion

- List of completed modules with completion date
- In-progress modules with percentage complete
- Recommended modules not started yet

Example:

```
Completed (5 modules)
✓ Answer-First Structure        Completed Oct 28  Score: 8.5
✓ Data-Backed Reasoning         Completed Oct 30  Score: 9.0
✓ Executive Communication       Completed Nov 1   Score: 7.8

In Progress (2 modules)
⏳ Trade-off Communication        60% complete     Current: 7.2
⏳ Stakeholder Acknowledgment     30% complete     Current: 6.8

Recommended (3 modules)
📌 Handling Board Objections      Not started      High impact
📌 Technical Feasibility          Not started      Medium impact
📌 Saying No Gracefully           Not started      Medium impact
```

Section 5: Recent Highlights (This Week)

- Notable improvements or achievements
- Automatically generated based on data

Example:

🎉 This Week's Highlights

✓ First Senior PM-level score on Trade-off Communication (8.5)
In your planning meeting, you proactively addressed engineering concerns. This is a breakthrough for you.
[View meeting →]

✓ Consistency improving: 4 of 5 meetings this week showed answer-first structure (up from 60% last week)

✓ Completed 3 practice exercises
You're building the habit. Keep it up!

⚠️ Stakeholder Management dropped slightly
Last 2 meetings showed delayed acknowledgment of concerns. This might be stress-related. Try the "Active Listening" module.

Section 6: Weekly Insights

- AI-generated summary of progress
- Pattern analysis across meetings and practice
- Specific recommendations

Example:

📋
✓

```
📊 Insights from Last 7 Days

Strengths:
Your data-backed reasoning continues to be exceptional. Every
recommendation you made this week included specific metrics.
This is Senior PM level and should be maintained.

Improvement Area:
Trade-off articulation is your current growth edge. You're
mentioning alternatives but not fully exploring constraints.
This appeared in 3 of 4 meetings.

Recommendation:
Complete the Trade-off Communication module (60% done) and
practice Exercise 4-5 before your next board meeting. This
skill is critical for your PM → Senior PM transition.

Meeting-to-Practice Connection:
You practiced stakeholder acknowledgment on Tuesday, then
successfully applied it in Wednesday's planning meeting. This
is excellent transfer of learning. Keep practicing before
high-stakes meetings.
```

Progress data structure:

typescript

```
interface UserProgress {
  user_id: string;
  current_role: string;
  target_role: string;
  timeline_months: number;

  career_progress: {
    overall_percentage: number; // 0-100, how close to target
    consistency_at_target_level: number; // % of responses at target level
    trend_last_30_days: number; // +/- percentage points

    skill_breakdown: {
      skill: string;
      current_score: number;
      target_score: number;
      percentage_to_target: number;
      status: 'achieved' | 'nearly_there' | 'improving' | 'focus_area';
    }[];

    next_milestone: string;
    estimated_date_to_reach?: string;
  };

  skill_scores: {
    skill: string;
    current: number;
    thirty_days_ago: number;
    target_benchmark: number;
  }[];

  meeting_performance: {
    meeting_id: string;
    date: string;
    overall_score: number;
    title: string;
  }[];

  practice_performance: {
    exercise_id: string;
    date: string;
    score: number;
    attempt_number: number;
  }[];

  module_progress: {
    module_id: string;
    title: string;
    status: 'completed' | 'in_progress' | 'not_started';
    completion_percentage: number;
    completed_date?: string;
    best_score?: number;
  }[];

  recent_highlights: {
    type: 'achievement' | 'improvement' | 'concern';
    title: string;
    description: string;
    related_item_id?: string;
    date: string;
  }[];

  weekly_insights: {
    generated_at: string;
    strengths: string[];
    improvement_areas: string[];
    recommendations: string[];
    patterns_observed: string[];
  };
}
```

User interactions:

- Change date range to see progress over different timeframes
- Hover over radar chart → see exact scores
- Click meeting in trend chart → navigate to meeting detail
- Click module → navigate to module page
- Click highlight → navigate to related meeting/exercise
- Export progress report as PDF
- Share progress (for transparency with manager - optional)
- Set reminders for next practice session

What user validates:

- Can they see clear progress over time?
- Does the career level tracking feel accurate?
- Are the insights actionable?
- Do they understand which skills to focus on?

- Is the visualization helpful or overwhelming?
- Does this motivate continued usage?

Acceptance criteria:

- □ Career progress card shows current → target level clearly
- □ Overall consistency percentage calculates correctly
- □ Skill breakdown shows all dimensions with accurate status
- □ Trend indicator shows correct direction and amount
- □ Radar chart displays all skills with correct scores
- □ Overlay comparison works (current vs 30 days ago vs target)
- □ Meeting performance chart shows last 10 meetings
- □ Practice performance chart shows all attempts
- □ Trend lines indicate direction correctly
- □ Module completion list shows correct status for each
- □ Recent highlights generate based on actual data
- □ Weekly insights provide specific, actionable recommendations
- □ Date range selector updates all charts/data
- □ Can export progress report as PDF
- □ All navigation links work (to meetings, exercises, modules)
- □ Responsive on mobile (charts adapt, data readable)
- □ Loading states for each section

Slice 12-15: Additional Phase 1 Features

Note: Slices 12-15 cover additional features needed to complete Phase 1:

Slice 12: Settings & Preferences (3 hours)

- User profile editing
- Notification preferences
- Privacy settings
- Account management

Slice 13: Help & Onboarding Tours (2 hours)

- In-app help documentation
- Interactive product tours
- Contextual tooltips
- FAQ section

Slice 14: Mobile Responsive Polish (3 hours)

- All pages optimized for mobile
- Touch-friendly interactions
- Mobile navigation patterns
- Performance on mobile devices

Slice 15: Testing & Bug Fixes (2-3 hours)

- Cross-browser testing
- Edge case handling
- Performance optimization

- Final QA before Phase 2

Instructions for Using This Prompt

When Starting a New Chat for Phase Documentation:

1. **Upload this entire document** to the chat
2. **Specify which phase** you want documentation for:

```
"Please create complete documentation for Phase 1 of ShipSpeak including:
- Integration Contract
- Slice Prompts (Slices 1-15)
- Phase Summary
- Master Tracker

This master prompt includes detailed specifications for Slices 1-11:
- Phase 1A: Foundation (Slices 1-3)
- Phase 1B: Meeting Intelligence (Slices 4-6)
- Phase 1C: Learning Modules & Practice (Slices 7-10)
- Phase 1D: Progress & Career Tracking (Slice 11)

Please generate complete documentation based on these specifications."
```

3. **What you'll receive: Integration Contract** (~15,000 words):
    - Detailed specifications for each feature in the phase
    - UI layouts and component structures
    - Data structures with TypeScript examples
    - User flows and interactions
    - Acceptance criteria
    - Testing requirements

   Slice Prompts (~12,000 words):
    - Each slice (4-8 hours) broken down with:
        - What you deliver (outcomes)
        - Specific components to build
        - Detailed UI elements
        - Data structures
        - User interactions
        - What user validates
        - Granular acceptance criteria

   Phase Summary (~3,000 words):
    - Phase goals and outcomes
    - Week-by-week breakdown
    - What gets validated
    - Success criteria

    - Dependencies and prerequisites
  - **Master Tracker** (spreadsheet format):
    - All slices with time estimates
    - Status tracking (Not Started, In Progress, Complete)
    - Acceptance criteria checkboxes
    - Dependencies
    - Notes section
4. **Iteration**: If you need more detail on specific slices, ask:

```
"Can you expand Slice 5 (Meeting Detail & Transcript View) to include:
- More specific mock data examples
- Additional user interaction scenarios
- Edge cases and error handling"
```

Documentation Quality Standards:

Every slice must include:

- ✅ Specific component names and files
- ✅ Detailed UI element descriptions
- ✅ Data structure examples (JSON/TypeScript)
- ✅ All user interactions enumerated
- ✅ Clear validation objectives
- ✅ Granular acceptance criteria (checkboxes)
- ❌ Not just "build the dashboard" (too vague)
- ❌ Not missing interaction details
- ❌ Not skipping mock data examples

Phase Progression:

Phase 1 (This prompt covers Slices 1-15):

- Complete UX with mock data
- Focus on user validation
- Zero backend integration

Phase 2 (Future prompt needed for Slices 16-26):

- Real integrations replacing mock data
- Same UI, different data sources
- Backend infrastructure

Phase 3 (Future prompt needed for Slices 27-30):

- Polish and production readiness

- Error handling
- Onboarding
- Payment

Document Formatting Requirements

Integration Contract Format:

markdown

# ShipSpeak Integration Contract - Phase 1

## Overview
[Phase goals, outcomes, duration]

## Feature Specifications

### [Feature Name]
Priority: P0
Phase 1 Focus: Mock data implementation
Phase 2 Enhancement: Real integration details

#### Components to Build
- `ComponentName1` - Description
- `ComponentName2` - Description

#### UI Layout
[ASCII diagram or detailed description]

#### Data Structures
```typescript
interface DataStructure {
  // ...
}
```

#### User Interactions
- Action → Result
- Action → Result

#### What User Validates
- Question 1
- Question 2

#### Acceptance Criteria
- [ ] Criterion 1
- [ ] Criterion 2

Slice Prompt Format:

markdown

~~~markdown
# ShipSpeak Slice Prompts - Phase 1

## Phase Overview
[Summary of phase goals]

## Slice [Number]: [Name] ([X] hours)

**What you deliver:**
[Concrete outcomes]

**Specific components to build:**
- `Component1`
- `Component2`

**Detailed UI elements:**
[Exact description of what appears on screen]

**Specific data structures:**
```typescript
// Example structure
```

**User interactions:**
- Interaction 1 → Result
- Interaction 2 → Result

**What user validates:**
- Validation question 1
- Validation question 2

**Acceptance criteria:**
- [ ] Criterion 1
- [ ] Criterion 2

**Phase 2 notes:**
[How this gets enhanced with real data]
~~~

Master Tracker Format:

markdown

```
# ShipSpeak Phase 1 Master Tracker

| Slice | Name | Hours | Status | Dependencies | Acceptance Criteria Met |
|-------|------|-------|--------|--------------|------------------------|
| 1 | Project Setup | 4 | Not Started | None | 0/8 |
| 2 | Auth Pages | 3 | Not Started | Slice 1 | 0/10 |
...

## Progress Summary
- Total Slices: 15
- Total Hours: 44-54
- Completed: 0
- In Progress: 0
- Not Started: 15

## Week-by-Week Plan
**Week 1:** Slices 1-3 (Foundation)
**Week 2:** Slices 4-6 (Meeting Intelligence)
...
```

Ready to Generate Documentation

This prompt contains everything needed to generate comprehensive, actionable documentation for ShipSpeak Phase 1 development.

What's Included:

- ✅ Complete product context and career progression framework
- ✅ Frontend-first (Bravax) development methodology
- ✅ **Detailed Slice 1-11 specifications** (Foundation → Meeting Intelligence → Learning → Progress)
- ✅ **Enhanced Slice 2:** Two-path onboarding (Analyze Meetings OR Start Practicing)
- ✅ **Enhanced Slice 6:** Next Steps section in AI feedback panel
- ✅ **New Slices 7-10:** Complete learning module and practice system
- ✅ **New Slice 11:** Comprehensive progress dashboard with career tracking
- ✅ Placeholders for Slices 12-15 (Settings, Help, Mobile, Testing)

Key Enhancements:

1. **Two-Path Onboarding:** Users choose to either connect their calendar for meeting analysis OR jump into practice exercises immediately

2. **Next Steps Integration:** Every meeting feedback includes actionable next steps (practice, apply framework, review examples)
3. **Complete Learning Loop:** Meeting analysis → Recommended modules → Practice exercises → Feedback → Progress tracking
4. **Career Progression Focus:** All features connect to advancing from current level to target level

To begin, start a new chat and say:

> "Please create Phase 1 documentation for ShipSpeak including Integration Contract, Slice Prompts (Slices 1-15), Phase Summary, and Master Tracker. Use the detailed specifications provided for Slices 1-11."

The generated documentation will include:

- Integration Contract (~20,000 words) - Complete feature specifications
- Slice Prompts (~15,000 words) - Granular implementation tasks with acceptance criteria
- Phase Summary (~3,000 words) - Overview, goals, and validation objectives
- Master Tracker (spreadsheet format) - Progress tracking with checkboxes

Then specify which slices (1-15) you want detailed first, or request all of Phase 1.