

Workshopr App: one workshop, Strategize to Synthesize

Scenario: a signed-in facilitator takes a single workshop through the app shell, from making the case for the session to synthesizing what came out of it, with anonymous participants joining the live session by share code. Columns are steps in time; rows are who or what does the work. Red dashed lines mark the line of interaction, the line of visibility, and the line of internal interaction.

Facilitator lanes Backstage Failure points and gaps

	01 · ENTRY Sign in and land	02 · ENTRY First-run onboarding	03 · STRATEGIZE Make the case	04 · STRATEGIZE Pitch and pressure-test	05 · PLAN Build the agenda	06 · PLAN Agenda Check	07 · FACILITATE Start a live session	08 · FACILITATE Participants join	09 · FACILITATE Capture in the room	10 · FACILITATE End the session	11 · SYNTHESIZE Synthesize	12 · SYNTHESIZE Prove: Impact Report
Physical evidence	- Dashboard at <code>/app/</code>: recent-workshop card, workshop grid, right rail - In-frame sign-in wall when signed out	- Welcome overlay (shown at 0 workshops) - Two coach-mark spotlights: the rail and the <i>More</i> launcher - First-agenda toast - Welcome email sent at registration	- Six-blank sentence workbench with a 0/6 meter and a Saved locally chip - Red document bar: title, Saved pill, save, delete - Three kit cards with live previews painted from the sentence - Right rail: DNA badge, journey checklist, the Strategize film	- The pitch email, rewriting itself in <code>#stEmailPreview</code> - Five objection cards, three verbatim scripts each - Go/no-go verdict card with flags and asks from the sentence - Ready to build? exit strip on every step	- Full-bleed agenda builder (embedded planner) - Saved agendas list - The <i>/pro/</i> upsell page if not entitled	- Agenda Check report: grade, issues, fix suggestions "AI unavailable" error state	- Live console: timer, script, transport bar - Share code and QR for participants - Workshop picker when no plan is chosen	- Participant page: mobile agenda view - Poll voting UI, reaction taps, raise-hand button	- Notes and voice notes - Intervention cards - Poll results, reaction bursts, help queue - Participant feedback form	- Session wrap - Journey strip flips to "delivered" on MyWorkshopr, the saved list, and the Synthesize picker	- Synthesis document: decisions, action items - PDF export and share link - Pushes to Slack, Notion, Jira, Linear	- Printable Impact Report page - Browser print to PDF
Facilitator actions	- Opens <code>/app/</code> - Signs in (cookie or Bearer token) - Scans the dashboard for the next workshop	- Reads or dismisses the welcome - Follows the two coach marks - Sees the toast after saving the first agenda	- Picks a workshop type, fills six dropdown blanks - Types a hypothesis, or presses Draft It For Me - Copies the brief - Picks one of three kit tools	- Picks time unresolved and cost of inaction blanks - Types current approach, expected outputs, proposed date - Picks an objection, copies the scripts, generates a 60-second pitch - Answers three scope selects; names a proxy if sending one	- Picks a path - Drags exercises, icebreakers, and breaks onto the canvas - Sets durations, saves	- Pastes or generates an agenda - Clicks Check - Applies suggested fixes and re-checks	- Picks the workshop - Opens Facilitate, step run - Starts the session and shares the code	- Shows the QR or share link - Watches participants appear	- Advances items - Adds notes and voice notes - Logs interventions - Creates and closes polls - Resolves help requests	- Ends the live session - Closes the console	- Opens Reflect, then Synthesize - Imports the workshop record - Runs synthesis, edits, exports, shares	- Opens Prove, then report - Reads the Impact Report and prints it
Participant actions	Not in this step	Not in this step	Not in this step	Not in this step	Not in this step	Not in this step	Not in this step	- Scans the QR or opens the share link - No account, no login - Reads the live agenda on a phone	- Votes in polls - Taps reactions - Raises a hand for help - Leaves feedback	- Page goes quiet when the session ends - Feedback form stays open	Not in this step	Not in this step
LINE OF INTERACTION												
Frontstage	- Boot veil, then dashboard - Sign-in renders in-frame and keeps <code>?view</code> and <code>?step</code> for the return trip - Unknown <code>?view</code> falls back to home	- Overlay only on home, only when the workshop total is 0 - Coach marks for users who skip the CTA - Toast when the count becomes exactly 1	- Renders <code>/app/?view=strategize&step=sentence kit</code> - Sign-in wall replaces the view; no Pro gate in this phase - Autosaves to <code>localStorage</code> on a 300ms debounce, flushed on pagehide - Continue stays disabled until all six blanks are filled	- Renders <code>step=pitch practice check</code>; unknown steps fall back to sentence - Exit strip flips to ready only when the recommendation level is go - Every export is copy-to-clipboard: no PDF, no download, no share link	- Canvas updates and autosaves - Pro gate in <code>app/index.php</code> redirects to <code>/pro/?from=planner</code>	- Report renders in the check stage - Rate-limit and 502 states surface as inline errors	- Console loads the plan - Missing <code>?plans=</code> redirects to the picker - Pro gate redirects to <code>/pro/?from=facilitator</code>	<code>facilitator/participant.php?</code> <code>code=SHARE_CODE</code> - Serves plans with visibility unlisted or public, joined to <code>plan_items</code>	- Live widgets refresh by polling - Reactions shown from the last 30 seconds - Help queue and poll results update on the console	<code>end_live</code> returns and the console closes - Lifecycle strip shows delivered	- Full-bleed synthesize tool - Deep link <code>?plan=N</code> skips the picker - The old <code>synthesize/app.php?plan_id=N</code> still 301s here	- Server-rendered report - Missing <code>?plan=</code> redirects to the picker
LINE OF VISIBILITY												
Backstage	<code>validateAuthToken()</code> in <code>includes/shared-auth.php</code> <code>appDashboardData()</code> in <code>app/includes/dashboard.php</code> - GET <code>/api/workshop-record.php?list=1</code> - GET <code>/api/user.php</code>	<code>app/js/app-onboarding.js</code>, client only <code>localStorage</code> keys <code>workshopr_app_welcome_seen</code>, <code>_tour_done</code>, <code>_first_celebrated</code> <code>sendWelcomeEmail()</code> in <code>api/auth.php</code> at registration	- POST <code>/strategist/api/preplan.php</code> writes <code>plans.preplanning_data</code>; 401 without a bearer token - GET <code>/strategist/api/preplan.php?plan_id=</code> reads name and <code>preplanning_data</code> <code>strategist/includes/workbench.php</code>, shared verbatim with the legacy <code>app.php</code> <code>st_url()</code>, <code>st_recap_card()</code>, <code>st_hypothesis_card()</code> from <code>strategist/helpers.php</code>	- POST <code>/strategist/api/practice-mode.php</code> action: 'pitch' — the only AI the UI reaches <code>api/ai_provider.php</code> via <code>api/config.php</code>: <code>gemini-2.5-flash</code> <code>strategist/api/ai-common.php</code> disables the response cache - POST <code>/strategist/api/documents.php</code>: four generators, no caller	- POST <code>/api/plans.php</code> writes plans and <code>plan_collaborators</code> <code>/api/plan-items.php</code> writes <code>plan_items</code> and touches <code>plans.updated_at</code> <code>/planner/api/*</code> serves library data	<code>/api/agenda-lint-parse.php</code>: <code>requireProUser()</code>, rate limiter, AIProvider call <code>/api/agenda-lint-fix.php</code>: deterministic or AI fix, re-lint, persist	<code>coach.php</code> action=<code>create_session</code> inserts <code>coaching_sessions</code> with <code>linked_plan_id</code> - action=<code>start_live</code> updates status <code>facilitator-live.php</code> action=<code>push</code> writes <code>live_sessions</code> keyed by <code>share_code</code>	<code>participant.php</code> loads plan and <code>plan_items</code> by <code>share_code</code> <code>share_lookup</code> rate bucket: 120 per IP per minute <code>ensureTable()</code> runs four times per anonymous request	<code>coach.php</code> <code>add_note</code> inserts <code>coaching_session_notes</code> (dedupe on <code>session_id + client_id</code>) <code>log_event</code> inserts <code>session_events</code> <code>log_intervention</code> inserts <code>intervention_logs</code> <code>facilitator-live</code> <code>create_poll</code> and <code>close_poll</code> write <code>live_polls</code> <code>participant-live</code> <code>vote</code>, <code>react</code>, <code>help</code> write <code>live_polls</code>, <code>live_reactions</code>, <code>live_help_requests</code> <code>upload-voice-note.php</code> and <code>workshop-feedback.php</code>	<code>coach.php</code> action=<code>end_live</code> sets <code>coaching_sessions.status</code> to completed <code>live_sessions</code> row ended	<code>synthesize.php</code> <code>import_record</code> sets <code>synthesis_sessions.source_session_id</code> and writes an artifact of kind <code>session</code> (replace-atomic) <code>synthesize</code> and <code>synthesize.grouped</code> call the AI and write <code>synthesis_artifacts</code> <code>save</code> writes <code>synthesis_sessions</code> <code>output_generate</code> and <code>output_share</code> mint share tokens <code>synthesize/api/export-pdf.php</code> <code>event</code> writes <code>synthesis_events</code>	<code>includes/impact-report.php</code> reads the Workshop Record <code>getWorkshopRecord()</code> merges notes, interventions, events, reactions, polls, and help - No write path
LINE OF INTERNAL INTERACTION												
Support processes	- MySQL: <code>auth_tokens</code>, <code>plans</code>, <code>coaching_sessions</code>, <code>synthesis_sessions</code> <code>mw_cache_remember</code> short-TTL cache - Service worker <code>app/sw.js</code> asset cache	- Browser <code>localStorage</code> - Email provider, best-effort and non-blocking - No server-side onboarding state	<code>validateAuthToken()</code> sign-in check; <code>isProUser()</code> is never called in this phase - MySQL: <code>plans.preplanning_data</code> JSON — one column, no strategize tables <code>localStorage</code> <code>workshopr_strategist_{uid}</code> and <code>workshopr_strategist_last</code> - Draft It For Me cycles three canned drafts; no AI call	- Google Gemini, for the 60-second pitch only <code>checkRateLimit()</code> on the <code>ai_suggest</code> bucket - Cookie <code>workshopr_workshop_settings</code> (30 days) and <code>sessionStorage</code> <code>workshopr_strategist_prefill</code> carry the handoff - Prompt grounded in <code>academy/ebook/buy-in-playbook.md</code>	<code>isProUser()</code> tier check - Stripe subscription state: <code>account_type</code>, <code>subscription_ends_at</code> - MySQL: <code>plans</code>, <code>plan_items</code>, <code>exercises</code>, <code>icebreakers</code>, <code>breaks</code> <code>sessionStorage</code> agenda buffer	- AI provider chosen in config: Gemini, Anthropic, or OpenAI <code>includes/rate-limiter.php</code> - MySQL	- MySQL: <code>coaching_sessions</code>, <code>live_sessions</code> <code>localStorage</code> snapshot under <code>FACILITATOR_STATE_KEY</code> for resume on reload	<code>includes/rate-limiter.php</code> <code>includes/db-migrations.php</code> <code>ensureTable()</code> - MySQL read of plans and <code>plan_items</code>	<code>plan_id</code> is derived on the server from the owned session's <code>linked_plan_id</code>, never trusted from the client <code>share_code</code> resolves to <code>live_sessions.current_index</code>, then to the active <code>plan_items.id</code> - MySQL, plus file uploads for voice notes	- Lifecycle is derived, never stored: delivered means a completed session with a <code>linked_plan_id</code> - MySQL	- AI provider <code>ensureTable()</code> auto-creates the <code>synthesis_*</code> tables on first touch - OAuth apps for Slack, Notion, Jira, Linear <code>cron/synthesize-digest.php</code> - MySQL	<code>/api/workshop-record.php</code> federated read - MySQL, read-only - synthesized means a <code>synthesis_sessions</code> row exists for this <code>plan_id</code>
Failure points and gaps	Silent global Pro bypass: while Stripe is unconfigured every account is Pro, with no signal to the user or admin (<code>shared-auth.php</code>)	- Clearing browser storage replays onboarding - A failed welcome email is invisible to everyone	- Nothing reaches the server until an explicit Save to account: until then the whole strategy lives in one browser's <code>localStorage</code> <code>?id=</code> is the only thing carrying state across the five steps	- The handoff to Plan never sends <code>state.planId</code>: a saved pre-plan arrives at the builder as a cookie, cut off from its own plans row - The AI role-play coach is unreachable: <code>objection-coach.js</code> binds <code>#stChatForm</code> and <code>#stChatLog</code>, which no view emits	<code>plan_id</code> is the workshop's identity across all three phases: deleting a plan orphans the record downstream	- No API key on the active provider returns a 502 "AI unavailable" - Swapping providers changes output quality with no visible signal	- A missing plan redirects silently to the picker - The <code>localStorage</code> snapshot resumes a reload; it is not an offline queue	<code>ensureTable()</code> issues a SELECT per table per request on the hottest anonymous path - An unknown table name fails as a silent no-op	- Participant stamps that fail to resolve insert NULL <code>plan_item_id</code> rather than dropping the action <code>coaching_session_notes</code> and <code>intervention_logs</code> exist only in prod, so readers must degrade on 42S02	None specific: lifecycle is derived, so there is no write to fail here	- A bookmarked URL without params degrades to "start over" with no message - OAuth app registrations are pending, so pushes are dead until keys land	- No export beyond browser print - The report is thin if the session was never ended
Reading the blueprint	Five things the lanes surface when read together											
	1 The paywall is open by design and nothing says so While Stripe stays unconfigured, <code>isProUser()</code> returns true for every account. The frontstage promises a Pro tier the backstage never checks. Flipping keys into prod .env is the paywall switch, and no lane will tell a user or admin the day it flips.	2 Three silent redirects swallow bad links Facilitate run, Prove report, and Synthesize make all fall back to a picker when <code>?plan=</code> is missing. A stale bookmark or a mistyped share reads as "start over" with no message. Long horizontal stretch: the user is being moved and not told.	3 One AI provider is a single point of failure for two phases Agenda Check (Plan) and Synthesize (Reflect) both route through AIProvider. A missing key returns 502; a provider swap changes the output quality silently. Two frontstage promises, one support dependency.	4 The anonymous path pays a schema check four times per request <code>participant.php</code> calls <code>ensureTable()</code> four times for every participant hit, each a SELECT. It is the hottest unauthenticated route in the product and the only one with no cache in front of it. Candidate for the <code>ws_schema_ensured</code> flag pattern.	5 Export pushes depend on four registrations that do not exist yet Slack, Notion, Jira, and Linear pushes in Synthesize are wired to OAuth apps that are not registered. The physical evidence lane shows a button; the support lane has nothing behind it.							