

TECHNICAL & DESIGN SPECIFICATION

Retirement Planning Dashboard

Single-file HTML/CSS/JS retirement income planner

Version 7.0

Document	Full technical & design spec
Supersedes	Specification v6.0
Build date	2026-05-15
Architecture	Single HTML file — no build, no server
Status	Current release

Contents

1. Project Overview
2. Data Model (state object `S` + globals)
3. Phase Configuration
4. Calculation Engine
5. Currency & Residence System
6. User Interface & Tabs
7. AI Integration — Prompts, Chat & Rate Updates
 8. 7.A — Plan-with-AI Sidebar (co-pilot) v7
9. Persistence & Data Flow
10. Responsive Design, Theming & Print
11. AI Chat Tab
12. Annuity Calculator Tab
13. What-if Explorer Tab
14. Educational System
15. Plan Health "Why this matters"
16. Longevity System
17. Curated PDF Report v7 expanded
18. Already-Retired User Workflow
19. External Dependencies
20. Version History
21. What's New in v7 (summary)

1. Project Overview

A single-file retirement income planning dashboard. The entire application — HTML structure, CSS theming, and JavaScript calculation engine — lives in one `.html` file with no build tools, frameworks, or server. It runs by opening the file in any modern browser.

The planner models a US-centric retirement across **five sequential life phases**, projecting month-by-month account balances, federal (and where applicable UK / Canadian / Australian) tax, ACA premium subsidies, Medicare/IRMAA, Required Minimum Distributions, Roth conversions, Social Security, pensions and passive income. All monetary values are stored internally in USD and displayed in any of six supported currencies.

1.1 Core Principles

- **Single source of truth:** all plan state lives in one global object `S`, persisted to `localStorage` and round-tripped through JSON export/import.
- **Deterministic engine:** the calculation pipeline is pure and reproducible — the same inputs always yield the same projection. This determinism is later exploited to *verify* AI proposals (Section 7.A).
- **Education in context:** concepts are taught where they are used (inline popovers, smart phase badges, glossary) rather than in a separate manual.
- **Graceful AI degradation:** every AI feature is optional and gated behind a user-supplied API key; the planner is fully functional with no key at all.

1.2 Calculation Flow

```
buildPhaseConfig(startAge, p5End, ...) → calcAllPhases(S, p5EndAge, lumps)
    → calcPhase(phaseOpts) → simPhase(monthly simulation)
```

Each phase is simulated month-by-month with compound growth. Tax brackets, standard/senior deductions, FPL thresholds, IRMAA tiers and Medicare costs are inflation-adjusted per phase via an `inflMult` multiplier.

2. Data Model

All plan state is held in the global object `s`. Defaults live in the `D_USD` constant (all amounts in USD). `autoLoad` and `importJSON` merge saved data over `D_USD` via spread (`{...D_USD, ...d.S}`), so any field added in a later version is back-filled with its default for plans saved before the field existed.

2.1 Field Groups

Group	Representative fields	Notes
Demographics	<code>birthYear</code> , <code>startAge</code> , filing status, spouse fields	MFJ toggles spouse SS, MFJ brackets
Account balances	<code>bal401k</code> , <code>balCash</code> , <code>balEquity</code> , <code>balRoth</code> , <code>superBal</code>	Stored in USD; canonical at replan
Growth assumptions	<code>g401k</code> , <code>gCash</code> , <code>gEquity</code> , <code>inflation</code> , <code>cola</code>	Annual rates; compounded monthly
Per-phase withdrawals	<code>w401k</code> , <code>wcash</code> , <code>wequity</code> , <code>wRoth</code> (per phase)	Monthly draw amounts
Roth conversions	<code>rothConversion</code> (per phase, annual taxable)	All phases (v5+)
Income sources	<code>uss</code> (US SS), <code>ssStartAge</code> , <code>ssBaseAge</code> , UK pension, <code>rentalAnn</code>	<code>rentalAnn</code> also receives annuity income
Part-time work	<code>partTime</code> per phase v7: all phases	Previously phases 0/1/2 only
Tax parameters	federal brackets (single+MFJ), <code>stdDed</code> , <code>seniorDed</code> , FPL 100/250/400, IRMAA tiers, <code>brk12</code> , <code>bracket22</code>	Inflation-adjusted per phase; AI-fetchable
State tax	state name, flat rate, exemptions v7: AI autocomplete	Datalist + AI lookup (\$7.5)
Residence flags	<code>activeCurrency</code> , <code>displayCurrency</code> , <code>foreignResident</code> , <code>ukResident</code>	Four flags — see §5
Phase boundaries	<code>phaseAge1end</code> ... <code>phaseAge4end</code> , <code>p5EndAge</code>	Custom age boundaries (v4+)

2.2 Global Companion State (outside `s`)

- `lumps` / `lumpIdCounter` — cash lump-sum events (filtered by phase number).
- `p5EndAge` — plan horizon end age (default 80; preset buttons for 80/90).
- `_annuityPlanState` — records annuity modifications for exact reversal.
- `_aiHistory` / `_aiSbHistory` — separate chat histories for the AI Chat tab and the Plan-with-AI sidebar.

v7
- `window._lastMCResult` — last Monte Carlo result cache.

3. Phase Configuration

`buildPhaseConfig(startAge, p5End, ...)` derives the boundaries and durations of the five phases from the user's age boundaries and the plan horizon.

3.1 The Five Phases

Phase	Typical span	Character
Phase 1	retirement start → ~62	Pre-SS, pre-Medicare; ACA subsidy window
Phase 2	~62 → 65	SS may begin; still pre-Medicare (ACA)
Phase 3	65 → ~70	Medicare begins; IRMAA matters
Phase 4	~70 → ~72/73	RMD onset; delayed-SS credits stop
Phase 5	~73 → plan horizon (<code>p5EndAge</code>)	Single coarse block to end of plan

3.2 Phase Durations & Horizon

Each phase duration in months is derived from its age boundaries. The Phase 5 duration is computed from the plan horizon `p5EndAge` :

```
const p5mo = p5End===90 ? Math.round((p5End-a4)*12) : 96;
```

For the default horizon of 80 this yields 96 months; for 90 it yields 216 months.

Known limitation carried into v7: the Phase 5 duration is hardcoded to 96 months for any horizon other than 90. A planned "+/- horizon stepper" (80→100 in 5-year steps) will replace this with `Math.max(1, Math.round((p5End-a4)*12))` so 85/95/100 compute correctly while 80/90 stay byte-identical. This stepper is **specified but deferred** and not yet shipped in v7.

3.3 Inflation Multiplier

Each phase carries an `inflMult` that compounds `S.inflation` from the plan start to that phase's midpoint. All tax thresholds, deductions, FPL and IRMAA values are scaled by `inflMult` before use, so the projection reasons in inflation-adjusted future dollars.

4. Calculation Engine

4.1 calcAllPhases → calcPhase → simPhase

`calcAllPhases(S, p5EndAge, lumps)` iterates the five phases in order, threading ending balances into the next phase's opening balances, and returns an array of per-phase result objects. Each result records gross income, taxable income, federal tax, ACA/Medicare cost, net income (nominal and real), Roth conversion effects, RMD, and ending balances per bucket. Every layer that destructures the options object **must** include each field, or it silently drops to `undefined` — a key constraint enforced by the post-change verification checklist.

4.2 Month-by-Month Simulation (simPhase)

`simPhase` walks each month of a phase: applies withdrawals, adds income, compounds each bucket at its growth rate, applies Roth conversions, and tracks depletion. Depletion flags (`k401Depleted` , `cashDepleted` , `equityDepleted`) are surfaced so downstream consumers — including AI proposal verification — can report which buckets ran dry and when.

4.3 Tax, ACA, Medicare

- **Federal tax:** progressive brackets (single or MFJ), standard + age-65 senior deduction, provisional-income computation for SS taxability.
- **SS taxability:** provisional-income thresholds determine the taxable fraction of Social Security (0% / 50% / 85% tiers); the per-phase taxable percentage is surfaced on the phase card.
- **ACA:** premium tax credit and CSR computed from MAGI vs FPL; the 400% FPL cliff is tracked. Only evaluated when `!foreignResident && !ukResident` .
- **Medicare / IRMAA:** from age 65, with IRMAA tiers (single + MFJ) applied on a 2-year MAGI lookback. Excluded for foreign/UK residents.
- **RMD:** estimated from age ~73 on the tax-deferred balance, with a shortfall warning if planned withdrawals fall below the required minimum.

4.4 Lump Sums

Lump-sum events are filtered to a phase by the phase number parsed from their `phaseKey` , and are only displayed/applied when `amtUSD > 0` . They adjust the relevant bucket at the phase in which they occur.

5. Currency & Residence System

The planner separates *what currency the plan is denominated in*, *what currency is shown on screen*, and *where the user is tax-resident*. These are four independent flags.

Flag	Meaning
<code>activeCurrency</code>	Selected country/currency (USD / GBP / CAD / AUD / PHP / THB)
<code>displayCurrency</code>	What is rendered on screen — toggled independently, defaults to USD
<code>foreignResident</code>	<code>true</code> for PHP / THB only — disables US healthcare (ACA/Medicare/IRMAA)
<code>ukResident</code>	<code>true</code> for GBP + "UK Res." toggle — enables UK tax + Foreign Tax Credit, disables US healthcare

5.1 Display vs. Storage

All values are stored in USD. Display conversion uses live FX rates from `open.er-api.com` (USD base) or user-supplied custom rates. **Inline editing is only enabled when** `displayCurrency === 'USD'` to avoid round-trip rounding drift.

5.2 UK Tax Treaty Model

- UK taxes: 401k withdrawals + UK pension + equity gains + part-time income — but **not** US Social Security (US-UK treaty Article 17 keeps SS taxable only in the US).
- US tax is reduced by a Foreign Tax Credit = `min(UK tax, US tax)`.
- Net = total – US tax after FTC – UK tax – healthcare. `const foreign` must be declared before the `isUkRes` block in `calcPhase` (temporal-dead-zone constraint).

5.3 Canada & Australia

CAD activates Canadian federal brackets with CPP/OAS context; AUD activates Australian federal brackets with Superannuation / Age Pension context.

5.4 Custom Exchange Rates v7 refinement

The custom-rates panel can now be **closed while keeping custom rates active**: a "Done" button (`closeCustomRates()`) dismisses the panel without reverting, and the reopen affordance is relabelled "Edit custom rates". A separate "↺ Use live rates" button discards custom values and refetches live rates, resetting the override-button label.

6. User Interface & Tabs

6.1 Tab Bar — Workflow Grouping v7

The 12 content tabs are ordered into a left-to-right workflow, with the Plan-with-AI toggle pinned last. Each tab carries a `data-tab=<name>` attribute; the active highlight is driven by a **name-based** toggle (`t.dataset.tab===name`) rather than the former brittle positional-index array — so the order and the highlight can never desync.

#	Tab	data-tab	Group
1	Overview	overview	Inputs
2	Edit values	edit	
3	Cash lump sums	lumps	
4	Balance	chart	Visualize / sensitivity
5	Scenarios	scenarios	
6	Stress test	stress	
7	SS Optimizer	ssopt	Strategy / explore
8	Drawdown	drawdown	
9	Annuities	annuity	
10	What-if?	whatif	Reference
11	Tax & ACA	taxguide	
12	AI Chat	ai	
—	Plan with AI (sidebar toggle)	— (no data-tab)	AI

The Plan-with-AI button deliberately has **no** `data-tab`, so it never receives the content-tab "active" highlight — it is a toggle, not a panel.

6.2 Overview Tab

The Overview presents the summary cards (starting wealth, average net income, projected ending wealth, lifetime tax to horizon), the five phase cards, the Plan Health side panel, and the personalised Concepts-to-learn card.

6.2.1 Grouped Toolbars v7

Two control clusters are now grouped by kind with thin `.ctrl-divider` separators:

- **Currency-bar cluster:** *Dashboard-view* controls (\$ colour, Fit phases, 💡 Concepts) | *Appearance* controls (text-size A·A·A·A, Light/Dark, Style picker).

- **Title-row actions:** status spans · *Data/output* (Export JSON, Import JSON, Create Report) ∷ *Destructive + help* (Reset defaults, Help).

6.2.2 Sticky Header Polish v7

Two stacked sticky bars — `.tabs-sticky-wrap` (top:0) and the Overview `.cmd-strip` (pinned beneath it) — now signal pinning subtly:

- **Lift-on-stick:** when the currency bar scrolls off-screen, an `IntersectionObserver` toggles `body.header-stuck`, fading in a soft downward shadow and a visible hairline border.
- **Frosted translucency:** pinned bars gain a translucent background (`color-mix`) + `backdrop-filter: blur(6px)`; gap-mask pseudo-elements are matched so no opaque seam shows.
- **Sliding active-tab indicator:** a 2px underline (`.tab-indicator`) slides to the active tab via `positionTabIndicator()`, called from `showTab()`, load, resize and text-size changes; wrap-safe across a second tab row. The underline is **hidden** on the soft, fintech and contrast themes (which keep their pill highlight).
- All motion is gated behind `@media (prefers-reduced-motion: reduce)`; the shadow/frost still apply instantly under reduced motion.

6.2.3 Tab-Switch Scroll

Clicking any Overview control tab scrolls the tab row flush to the top of the viewport (`scrollToTabsRow()`) so no currency/style sliver remains visible above it.

6.3 Phase Cards

Each phase card shows income sources, the federal bracket pill (with estimated calendar year), ACA subsidy / CSR rows or Medicare estimate, IRMAA / RMD warnings, and net income (nominal + real). Every badge is clickable and opens a smart popover tied to that phase's actual numbers (§13.2).

v7: the "% of SS taxable" text now always renders on its own line below the bracket badge row (predictable layout regardless of card width). Roth section labels show the per-phase duration, e.g. "Phase 5 · 96mo".

6.4 Edit Values Tab

A dense grid of all editable fields, grouped by phase and category. **v7 additions:**

- **Part-time income in every phase** (previously phases 0/1/2 only); fully wired through `populateInputsFromUSD`, `readInputs`, `updateEditLabels` and the calc engine.
- **State income tax autocomplete:** a `<datalist>` (`#us-states-list`) filters US states as the user types; the label reads simply "State".
- **AI state-tax lookup:** an "Auto-fill rate" button (`aiLookupStateTax()`) populates the flat rate + exemptions for the entered state. The button is disabled until a state is entered and re-disabled if the field is cleared (`syncStateLookupBtn()`).

6.5 Cash Lump Sums Tab

Add/edit/remove one-off cash events tied to a phase. Each row with amount > 0 exposes an "Evaluate as annuity" link that hands the amount to the Annuity Calculator.

6.6 Scenarios, Balance, Stress Test, Monte Carlo

- **Balance:** Chart.js balance + income trajectory across phases.
- **Scenarios:** saved comparison scenarios.
- **Stress test:** a 4×3 grid (inflation × returns) of outcomes — extracted into a pure `buildStressTestGrid(s, p5End, lumps)` so the report can reuse it. v7
- **Monte Carlo:** Box-Muller normal returns, 50–2,000 runs, P10/P25/P50/P75/P90 fan chart — core extracted into a pure `computeMonteCarlo(s, p5End)` returning `{probSolvent, runs, p10end, median, p90end, endAge}`. v7

6.7 Setup Wizard

A multi-step onboarding wizard (Step 1 Welcome → Step 5 Ready), refreshed in v6 with curated feature tiles and the already-retired guidance described in §17.

7. AI Integration — Prompts, Chat & Rate Updates

The planner calls the Anthropic Claude API directly from the browser (`_aiCallApi`). The API key is stored only in `localStorage` (`AI_KEY_LS = 'rp_v6_ai_key'`) and never transmitted anywhere but Anthropic. Every AI feature degrades gracefully to a manual workflow without a key.

7.1 Context Generation

`buildAiContext()` assembles a structured plan summary attached to every message: currency & residency flags, per-phase incomes/taxes/net, Plan Health check results with dollar headroom to each threshold (ACA cliff, IRMAA, 22% bracket, UK rate bands), and — in v7 — a **plan-budget block** (`_aiPlanBudgetText`) summarising depletion status. `_aiFingerprint()` hashes the plan so the context cache busts whenever any relevant field (including `p5EndAge`) changes.

7.2 Proposals Format

The system prompt (`getAiSystemPrompt()`) defines a structured `<proposals>` JSON block with field keys like `s.ssStartAge` , `p1.w401k` , `p2.wcash` . Each proposal carries a field id, a proposed value, and reasoning.

7.3 Applying Proposals

`aiApplyProposals()` parses the block, writes selected values to `S` , and re-renders. Human-readable before/after labels come from a `PARAM_LABELS` lookup. Value formatting is type-aware (`_aiPropFmt`): age fields render as integers ("62 → 67"), rate/percent fields get a "%" suffix, and currency fields get a "\$" + thousands separators — so an age never displays as "\$62".

7.4 AI Chat Tab

The full-width AI Chat tab is the original conversational surface (history `_aiHistory` , `AI_HIST_LS = 'rp_v6_ai_hist'`). It offers conversation-starter chips, a Proposed Changes panel, model selection (Opus / Sonnet / Haiku), and now a **copy icon** on each assistant response (`aiCopyResponse` + `_aiPlainText`) that copies a clean plain-text version into the clipboard for pasting into Word, email, etc.

7.4.1 Conversation Starters

One-click starter chips pre-load constrained prompts using the attached context: summarise & flag risks, tax-bracket headroom check, ACA/CSR check, withdrawal strategy, propose optimisations, Roth conversion advice.

7.4.2 Plan-Health Awareness

The context tells the AI which Plan Health checks are passing/warning/failing and the exact headroom to each threshold. Passing checks are treated as constraints: the model must not propose a change that flips a green check amber/red, and every proposal is sized within the relevant headroom.

7.4.3 Currency & Residence Awareness

- **USD:** full US bracket / ACA / IRMAA / RMD reasoning.
- **GBP + UK resident:** dual-system reasoning, Foreign Tax Credit, treaty Article 17 (SS US-only); proposals must respect UK rate-band ceilings *and* US thresholds simultaneously.

- **PHP / THB:** foreign-resident mode; ACA/Medicare/IRMAA excluded; focus on US federal efficiency.
- **CAD / AUD:** Canadian / Australian brackets with CPP-OAS / Super context.

7.5 One-Click Tax-Rate & State Updates

`autoFetchRates()` calls the API with `claude-haiku-4-5` to fetch US federal brackets (single+MFJ), standard + senior deduction, FPL 100/250/400, IRMAA tiers (single+MFJ), UK / Canadian / Australian parameters, then renders a before/after diff for spotlight fields. The manual prompt+paste workflow remains for key-less users. **v7** adds the AI **state-tax lookup** (\$6.4): a per-state flat rate + exemptions fetch driven from the Edit tab.

7.A Plan-with-AI Sidebar (Co-pilot) v7

A first-class, always-accessible AI co-pilot that lets the user update any plan parameter in natural language while browsing any tab — patterned on coding assistants (Cursor / Copilot). It is a new entry point that **coexists** with the AI Chat tab; both share `_aiCallApi` but keep separate histories.

7.A.1 Placement & Toggle

Opened via the "Plan with AI" button in the tab bar (`toggleAiSidebar()`). The sidebar slides in from the left; the input row is always visible (no bottom cut-off). Open/closed state persists to `localStorage` under `AI_SB_LS = 'rp_ai_sb_open'` . On mobile it overlays the content and dismisses on backdrop click.

7.A.2 Separate Session

The sidebar keeps its own conversation history (`_aiSbHistory` , `AI_SB_HIST_LS = 'rp_v6_ai_sb_hist'`) so it does not entangle with the AI Chat tab thread. It has its own clear-chat button, a copy icon on responses, and an info icon linking to the sidebar help content.

7.A.3 Action-Oriented Chips

When the sidebar thread is empty it shows action-oriented, plan-updating chips (e.g. "rebalance so equity can grow", "optimise pre-Medicare phases for ACA"). The most prominent chip is a getting-started prompt: *"Start here: add your monthly withdrawals in Edit values or directly on the dashboard."*

7.A.4 Verified Proposals (closed-loop self-check)

Because the calculation engine runs locally and free, the sidebar verifies the AI's own proposals before the user applies them — the deterministic engine is ground truth the language model cannot compute mentally:

1. **Dry-run** (`_aiDryRunProposals` via `_aiApplyProposalsToState`): clone `S` , apply the proposed changes, run `calcAllPhases` , and read the depletion flags (`k401Depleted` / `cashDepleted` / `equityDepleted`), counting only funded buckets.
2. **Auto-verify badge**: each proposal group renders an outcome badge (`_aiOutcomeBadge` / `_aiOutcomeText`) — green when the plan stays solvent to the horizon, amber/red when a bucket still depletes.
3. **Closed-loop refine**: if the user asked for a non-depleting plan and the dry-run still depletes, `_aiVerifyAndRefine()` automatically tells the AI its proposal failed in the simulator and asks it to refine, up to `_AI_REFINE_MAX = 3` attempts, posting a system note on each retry. Invoked from both `aiSend()` and `aiSbSend()` .

7.A.5 Apply & Undo

Proposals render as inline cards (all pre-checked); "Apply Selected" writes the checked changes and re-renders. A one-click **Undo** button reverses the entire batch of changes just applied, so the user can experiment freely.

7.A.6 Scope Guardrails

The system prompt includes a "Stay on topic" block: the assistant focuses on retirement planning and politely declines unrelated requests, plus a "Feasibility & self-verification" block that instructs it to respect Plan-Health constraints and headroom when sizing proposals.

8. Persistence & Data Flow

8.1 localStorage

Auto-saves on every change via `autoSave()`. The primary plan blob stores `S`, `lumps`, `lumpIdCounter`, `p5EndAge`, `activeCurrency` and subtitle, with fallback to prior-version keys for migration. Theme is stored separately as `rd_theme`.

Key	Purpose
<code>rp_v6_ai_key</code>	Anthropic API key (browser-only)
<code>rp_v6_ai_model</code>	Selected Claude model
<code>rp_v6_ai_hist</code>	AI Chat tab history
<code>rp_v6_ai_sb_hist</code> v7	Plan-with-AI sidebar history
<code>rp_ai_sb_open</code> v7	Sidebar open/closed state
<code>rp_v6_ai_models_cache</code>	Cached list of available models
<code>retirement_v6_help_audit_date</code>	Last help-audit date (watermark logic)
<code>retirement_v6_pdf_ai_pref</code>	Remembered AI-commentary preference for PDF
<code>retirement_v6_postwiz_nudge_dismissed</code>	Post-wizard nudge dismissal
<code>retirement_v6_learn_dismissed</code> / <code>_learn_collapsed</code>	Concepts-to-learn panel state
<code>retirement_v6_primer_seen</code>	First-visit Concepts Primer flag

8.2 JSON Export / Import

Export bundles all state plus a version number `v` and an ISO `date`. Import merges over `D_USD` via spread (new fields back-filled with defaults) and resets the session baseline (the "blue dots"). The same merge governs `autoLoad`.

8.3 Data Flow

```
User input → readInputs() / inline edit / oninput
  → S updated → calcAllPhases(S, p5EndAge, lumps) → results
  → renderSummary() → renderPhases() → autoSave()
```

Monte Carlo and Stress Test compute on demand (when their tab is opened, and — new in v7 — when the PDF report is built), not on every keystroke.

9. Responsive Design, Theming & Print

9.1 Breakpoints

Width	Phase grid	Other
> 900px	5 columns	Full layout
700–900px	3 columns	Grid-5 → 3 columns
480–700px	2 columns	Net-grid stacks; lump rows wrap; tab row wraps to 2 rows
< 480px	1 column	Single-column; AI sidebar overlays

9.2 Theme System

Theming is driven by `html[data-theme]` with five visual styles: **soft** (default / consumer), **editorial**, **quiet fintech**, **pro dashboard**, and **high contrast** (accessibility). Dark mode is a separate toggle; system preference is detected via `prefers-color-scheme` and the user override persists.

v7: pro + contrast styles, and the tab-indicator underline is suppressed on soft/fintech/contrast.

9.3 Print Styles & Pagination v7

The printable output is the curated PDF report (§16), rendered into a hidden `#pdf-report` container that the `@media print` stylesheet reveals while hiding all dashboard chrome. v7 switches to **native-flow pagination**:

- `@page { size: 8.5in 11in; margin: 15mm 16mm }` — page margins live on `@page`, so every continuation page (not just the first) is inset correctly.
- `.pdf-page { padding: 0; page-break-before: always }` with `.pdf-page:first-child { page-break-before: avoid }`. Fixed page heights and `page-break-after` were removed — they caused Chrome to emit trailing blank sheets and clipped tall AI blocks.
- `_repaginateReport()` is now a no-op; `reportPrint()` simply calls `window.print()`. Per-sheet footers were dropped to avoid blank-page artifacts.

10. AI Chat Tab (UI states)

See §7.4 for mechanics; this section covers on-screen UI.

10.1 First-Run (no key)

A key-entry form (password input for `sk-ant-...`, Save button, privacy note) shows until a key is saved, after which the chat interface appears.

10.2 Active Chat

Header bar (settings cog, model selector, clear-chat), scrollable thread, conversation-starter chips (auto-hide after first message), Proposed Changes panel (only when a response contains `<proposals>`), and the input row. Each assistant message carries a copy icon. 

10.3 Settings Strip

Toggled by the cog: model selector (Opus/Sonnet/Haiku), key-update input, Remove Key. Updating the key refreshes the rate-fetcher and other AI footers automatically.

11. Annuity Calculator Tab

Models fixed annuity purchases and optionally applies them to the plan (`initAnnuityTab()` ; state in `_annuityPlanState`).

11.1 Types

- **SPIA** — income begins at purchase age.
- **DIA / Longevity** — income deferred to a future age (default 80); deferral-age field shown only for DIA.
- **Fixed + COLA** — SPIA-like with a configurable annual income growth rate.

Variable annuities are not modelled. Manual quotes from any insurer can be entered.

11.2 Inputs & Derivation

Premium, payout-rate % (default 6%), buy age, start age (DIA), monthly income (auto-derived from payout rate if not entered: `premium × rate/100/12`), COLA rate, single vs. joint coverage.

11.3 Breakeven

`calcAnnuity()` iterates year-by-year to ~age 100, comparing cumulative annuity income against the premium invested at the 401k growth rate; the first crossover is the breakeven age, with a full comparison table.

11.4 Add / Remove from Plan

`addAnnuityToPlan()` deducts the premium (single cash source or multi-account split with tax-impact disclaimers) and adds COLA-adjusted annual income to each eligible phase's `rentalAnn` , recording every change for exact reversal. `removeAnnuityFromPlan()` restores every field from `_annuityPlanState` .

11.5 Lump-Sum Integration

"Evaluate as annuity" on a lump row (`evalLumpAsAnnuity`) pre-fills the premium; applying it zeroes the source lump (no double-count) and stores it for restoration on Remove.

12. What-if Explorer Tab

A sandbox of live sliders that never write back to **S** (clone-override-recompute-restore via try/finally).

- **SS claim-age slider (62–70):** back-solves the FRA amount, applies SSA actuarial reduction/credits, reports monthly benefit, cumulative to ages 90/95, and breakeven vs. the current plan.
- **Roth conversion-size slider (\$0–\$100k/yr):** per-phase; reports tax-cost delta, MAGI shift, net delta, and live bracket/IRMAA/ACA status chips.
- **Inflation-rate slider (0–8%):** reports final-phase real income and the purchasing power of \$1 in 30 years at the tried rate.

13. Educational System

A layered, click-only (touch-safe) education system.

13.1 Inline ⓘ Popovers

Small ⓘ icons sit beside confusing fields. Content lives in a single `EDU_CONTENT` object keyed by concept slug; each entry is a title + a body that is a static string or a function reading live `S` values (so values stay current after a rate fetch). Each popover links to the Glossary.

v7 additions to `EDU_CONTENT` :

- `getStarted` — the sidebar/dashboard getting-started prompt.
- `bracket22` — a distinct 22% bracket-ceiling popover (previously duplicated the 12% content).
- `ssClaimAge` — a dedicated SS claiming-age popover (Birth year keeps the FRA popover; the two ⓘ icons no longer open the same content).
- `mcSigmaCash` — explains the cash/bond volatility σ (%/yr) input.
- `survivor` — body updated with MFJ / US-tax-obligation setup guidance (see §6.x survivor note).

13.2 Phase-Card Smart Badges

Every clickable badge opens a popover using that phase's computed numbers (e.g. "\$4,200 below the 12% ceiling — you could convert that much to Roth without hitting 22%"), via `openBadgePopover(e1)` reading `data-*` attributes the renderer wrote.

13.3 Searchable Glossary

The Help modal's Glossary pane auto-generates from `EDU_CONTENT` alphabetically with a text filter; "Read more" links jump-and-highlight an entry.

13.4 Concepts Primer & "Concepts to Learn"

A first-visit 5-card primer (MAGI, FRA, ACA cliff, RMD, bracket strategy) and a personalised, rule-driven Concepts-to-learn card on the Overview tab.

13.5 Feature Tour v7

The guided tour (`TOUR_STOPS` + `showTourStep(i)`) is expanded to **9 stops**, adding a "Plan with AI" stop (`#ai-plan-tab-btn`) and a "Create Report" stop. New `_tourLiftAncestors` / `_tourClearLifts` helpers fix sticky-header stacking so highlighted targets are not obscured by the pinned bars.

14. Plan Health "Why this matters"

The 8-check Plan Health panel (`calcPlanConfidence` + `renderConfPanel`) renders an expandable "Why this matters" block per check from `WHY_CONTENT` (string or live-value function, same pattern as `EDU_CONTENT`). The check's `desc` reports personalised current status; the Why block explains the underlying concept and the fix.

15. Longevity System

A three-tier mechanism that keeps help content accurate as tax law changes.

15.1 Tier 1 — Dynamic State References

Five `EDU_CONTENT` popovers + one `WHY_CONTENT` entry use function bodies reading live `S` values (`irmaa` , `fp1400/100/250` , `brk12` , `stdDed` , `seniorDed` , `mfjIrmaa`) via `eduBody(c)` , so they self-update after a rate fetch.

15.2 Tier 2 — AI Help-Content Audit

`auditHelpContent()` uses `claude-haiku-4-5` to fact-check 15 specific claims (`AUDIT_ITEMS`) against today's date, returning colour-coded current/uncertain/changed cards. The audit is read-only and never mutates plan or help data. Date stored to `retirement_v6_help_audit_date` .

15.3 Tier 3 — Verified Watermark

`renderHelpFooters()` appends a four-state verification footer to each Help pane (no key / never audited / verified < 18 months / stale ≥ 18 months). `BUILD_DATE = '2026-05-15'` is the baseline shown without a key; the stale threshold is 545 days.

15.4 Audit Modal Inline Key Entry

Clicking audit without a key opens an inline key-entry form (`_renderAuditKeyEntry()`) that runs the audit immediately on save — no context switch to the AI Chat tab.

16. Curated PDF Report v7 expanded

A purpose-built printable report (`_renderPdfReport(ai)`) — not a dashboard dump. Orchestrated by four execution paths depending on whether the user has a key and their remembered AI-commentary preference (`retirement_v6_pdf_ai_pref`).

16.1 Four-Path Orchestration

1. No key → deterministic-only render + print.
2. Key + remembered "no" → same as path 1, no dialog.
3. Key + remembered "yes" → AI call → preview modal → print.
4. Key + no preference → opt-in dialog (Yes-with-AI / Rule-based / Cancel + "remember" checkbox).

16.2 Report Structure

The base report carries the cover, phase-by-phase cards, top risks + recommended actions, key assumptions, and disclaimer. **v7 adds four professional-advisor elements**, all reusing data the app already computes:

- **Asset-allocation mix (%)** — `_buildAllocationPdf(phas)` appended to the balance page: start-vs-end composition by account (401k / Cash / Equity / Roth / Super) and by tax treatment (tax-deferred / taxable / tax-free), with a one-line tax-diversification note.
- **Plan-resilience page** — `_buildResiliencePdf()` : Monte Carlo headline ("succeeded in N% of {runs} simulations to age {endAge}") with P10 / median / P90 ending portfolio, plus the compact 4×3 stress-test grid (inflation × returns) with the base case highlighted and depleted cells flagged.
- **Portfolio charts page** — `_buildReportCharts(phas)` renders balance + income charts on **offscreen** canvases (fixed size, `animation:false` , `toDataURL` , then `.destroy()`), so the report does not depend on the on-screen Charts tab being open or sized.
- An added disclaimer line noting Monte Carlo / stress results are simulation-based scenario analyses, not guarantees.

16.3 AI Commentary Safety

Hybrid by section: facts are always deterministic; only interpretive prose is AI-eligible. The PDF AI prompt receives structured input and returns `planSummary` (60-word cover intro) + `riskNarratives` (one per flagged check, keyed by exact check title). Hard constraint: *"every dollar figure and percentage MUST appear verbatim in the inputs; do not produce new numbers."*

16.4 Preview Modal & Visual Marking

Each AI paragraph appears in a `contenteditable` block (Use / Skip / Regenerate). AI prose in the PDF is wrapped in `.pdf-ai-block` with a blue left border, italic text and an "AI-generated commentary" label, so readers always distinguish facts from AI prose.

16.5 Pagination

See §9.3 — native-flow pagination with `page-break-before`, `@page` margins, and a no-op `_repaginateReport()`, which eliminated the trailing-blank-page and clipped-content artifacts present before v7.

17. Already-Retired User Workflow

Five touchpoints guide users already in retirement to the Replan-from-today anchor: (1) wizard Step 2 inline directive, (2) wizard Step 5 amber "Already retired? Replan from today" tile, (3) a post-wizard floating nudge (`showPostWizardNudge()` , dismissal persisted), (4) a dedicated Help-modal section with a 4-step walkthrough, and (5) a rewritten Replan-panel intro explaining what to enter (actual age, current balances, current SS/UKP with COLA applied), what gets trimmed, and that the original plan stays saved until Apply with a one-click rollback.

17.1 Survivor-Scenario Setup Guidance v7

The What-if survivor-scenario popup now explains that an accurate survivor projection requires the MFJ / US-tax-obligation section to be set up first, and shows a conditional "Take me there →" link (`survivorInfoReady()` / `goToUsTaxSection()` targeting `#us-tax-obligation-card`) only while that section is not yet completed. The same guidance is mirrored in the Help section and Glossary.

18. Dependencies (Self-Contained)

v7 hardened

After v7 hardening the application fetches **no third-party JavaScript or fonts at runtime** — it is a genuinely self-contained single file. The only off-device network paths are two optional, user-initiated services.

Component	Status	Purpose
Chart.js 4.4.1	Inlined into the file (SHA-512 verified against the published CDN hash)	Balance / income / scenario charts; offscreen report charts
Theme web fonts (DM Sans/Mono, Inter, Inter Tight, IBM Plex Sans/Mono, Source Serif 4, JetBrains Mono)	Self-hosted — inlined as base64 woff2 (latin subset)	Typography for the five visual styles
html2canvas	Removed (was unused)	—
open.er-api.com (v6)	Optional runtime fetch — anonymous, 6 s timeout, graceful fallback	Live FX rates (USD base)
Anthropic Claude API	Optional runtime fetch — user's own key, 120 s timeout	AI Chat, Plan-with-AI sidebar, rate/state fetch, help audit, PDF commentary

No build tools, frameworks, or server. The entire application is a single HTML file per version. A [Content-Security-Policy](#) restricts network egress to the two services above.

18.A Privacy & Security v7

The application runs entirely client-side; there is no backend, account system, or database. Plan state persists only in the browser's `localStorage` (with optional manual JSON export/import) and is never transmitted to the vendor. After hardening the file loads no third-party JavaScript or fonts at runtime (§18); the only off-device paths are the **optional, user-initiated** Anthropic API call (authenticated with the user's own key — plan figures are transmitted to generate responses) and an anonymous exchange-rate lookup.

- **API-key handling:** stored client-side only, sent only to Anthropic; an optional **session-only mode** ("don't save on this device" → `sessionStorage`, cleared on browser close) and a Remove-key control.
- **Data-egress disclosure:** the UI states explicitly that plan figures are sent to Anthropic when AI features are used.
- **Injection safety:** no `eval` / `new Function`; all user-supplied text is HTML-escaped before rendering; imported plans are validated (non-finite values coerced to defaults); AI proposals are parsed to numeric fields only and require explicit user approval.
- **Network controls:** a Content-Security-Policy with a restrictive `connect-src` whitelist; HTTPS throughout; request timeouts on all API calls.

A standalone *Security Review* (Rev. 3) accompanies the product; all findings are resolved or accepted as low-risk. The planner is an educational tool — not financial, tax, or legal advice.

19. Version History

Version 7.0 current

Plan-with-AI sidebar (co-pilot): always-accessible left sidebar that updates any parameter from natural language, with a separate persisted session, action-oriented chips, copy icon, info-to-help link, clear-chat, one-click Undo of an applied batch, scope guardrails, and **closed-loop self-verification** — proposals are dry-run through the deterministic engine and auto-refined (≤ 3 attempts) when they still deplete.

Enriched PDF report: added asset-allocation mix (by account and by tax treatment), a plan-resilience page (Monte Carlo % + P10/median/P90 and the 4×3 stress grid), and a portfolio-charts page (offscreen Chart.js). Switched to native-flow pagination — fixed the trailing-blank-page and clipped-content bugs; per-sheet footers dropped.

UI / workflow: tab bar reordered into a workflow grouping with de-fragilized name-based active-tab wiring (`data-tab`); grouped Overview toolbars with dividers; sticky-header polish (lift-on-stick shadow, frosted translucency, sliding tab indicator — suppressed on soft/fintech/contrast); tab-switch scroll lands flush; "Plan with AI" + "Create Report" tour stops (now 9 stops); pro + high-contrast themes.

Inputs & education: part-time income in **all** phases; state-tax autocomplete + AI rate/exemption lookup; new `EDU_CONTENT` popovers (`getStarted` , `bracket22` , `ssClaimAge` , `mcSigmaCash`) and updated survivor guidance; per-phase Roth durations on phase cards; SS-taxable text always on its own line; copy icon on AI Chat responses.

Engine refactors: pure `computeMonteCarlo(s, p5End)` and `buildStressTestGrid(s, p5End, lumps)` extracted so the report reuses them without opening their tabs; custom-rates panel can close while keeping custom rates ("Done" + "Edit custom rates").

Confidence, cash-flow & education (v7.x): historical sequence-of-returns **backtest** (overlapping windows of real US market data since 1928, lump-aware) alongside Monte Carlo; a cross-lens **Confidence summary** (unifies Plan-Health checks + Monte Carlo + backtest, in panel and report); a transparent **Maximum sustainable spending** finder (bisection on a withdrawal multiplier, solvent / Monte-Carlo / historical gates); bidirectional **cash lump sums** (money in / money out with an account cascade) now reflected in Monte Carlo and the backtest; per-account clickable **ending-balance popovers**; a What-if **segmented sub-nav**; and the **retirement spending-smile** concept.

Security hardening (v7.x): Chart.js inlined (SHA-512 integrity-verified) and unused html2canvas removed; theme fonts self-hosted (inlined woff2); a Content-Security-Policy; explicit AI data-egress disclosure; an optional session-only API-key mode; import validation; and request timeouts on all API calls — making the file fully self-contained with no third-party runtime assets (§18, §18.A).

Specified but deferred: Plan-horizon +/- stepper (80→100 in 5-year steps), including the Phase 5 duration fix described in §3.2.

Version 6.0

Released 2026-05-17. Built-in AI Chat advisor (six starters, plan-context attachment, Proposed Changes panel, three model options, browser-only key). One-click tax-rate updates (`autoFetchRates()`). Annuity Calculator

tab (SPIA/DIA/Fixed+COLA, breakeven, add/remove with exact reversibility, multi-account funding, lump-sum integration). What-if Explorer (SS / Roth / inflation sliders). Educational system (22 popovers, glossary, primer, 7-step tour, Concepts-to-learn). Phase-card smart badges. Plan Health "Why this matters". Longevity system (dynamic refs, AI audit, verified watermark). Already-retired workflow (five touchpoints). Curated 5-page PDF report v2 with optional AI commentary + editable preview. Setup-wizard refresh. Tax-notes TOC. Full backward compatibility with v5.

Earlier Versions

Version	New capabilities
1.0	Foundation: 5-phase model, US tax, ACA, Medicare
2.0–3.0	UK pension/tax, multi-currency (USD/GBP/PHP/THB), scenarios, charts, AI prompts
4.0	MFJ, Roth IRA bucket, Roth conversions (Phases 1–2), spouse SS, MFJ brackets; rental/passive income, Roth conversions (all phases), RMD estimator with shortfall warning, onboarding wizard
5.0	Custom phase age boundaries; Canada (CPP/OAS/CAD tax); Australia (Super/Age Pension/AUD tax); CAD & AUD currency; Monte Carlo (Box-Muller, 50–2,000 runs); P10–P90 fan chart; 4×3 stress test

20. What's New in v7 — At a Glance

Area	v6	v7
AI surfaces	AI Chat tab only	+ Plan-with-AI sidebar (separate session, verified + refined proposals, undo)
Proposal trust	Apply as-is	Dry-run through engine + auto-verify badge + ≤ 3 -attempt refine
Part-time income	Phases 0/1/2	All phases
State tax	Manual	Autocomplete + AI rate/exemption lookup
PDF report	5 pages, no allocation/MC/stress/charts	+ allocation %, resilience page (MC + 4×3 stress), charts page; native-flow pagination (no blank pages)
Tabs	Interleaved; positional active logic	Workflow grouping; name-based <code>data-tab</code> wiring
Sticky header	Instant snap	Lift shadow + frost + sliding indicator (reduced-motion safe)
Themes	soft/editorial/fintech	+ pro, + high contrast
Education	22 popovers	+ getStarted / bracket22 / ssClaimAge / mcSigmaCash; survivor setup guidance
Feature tour	7 stops	9 stops (+ Plan with AI, + Create Report)
AI Chat responses	—	Copy-to-clipboard icon
Confidence tools	Monte Carlo + stress	+ historical backtest, Confidence summary, Maximum sustainable spending
Cash lump sums	Money in only	Money in <i>and</i> money out (one-time expenses), reflected in MC + backtest
Dependencies	Chart.js + html2canvas + fonts via CDN	Self-contained: Chart.js inlined (verified), fonts self-hosted, html2canvas removed
Privacy & security	—	CSP, AI data-egress disclosure, session-only key mode, import validation, API timeouts