CanvasIQ: AI ROI & Roadmap Canvas Agent

Author: Noah Hicks

Introduction

CanvasIQ is a web-based agent that helps organizations turn loose AI ideas into a structured, financially grounded roadmap. It combines an AI-guided interview with ROI calculations, portfolio visualizations, and a printable "AI ROI & Roadmap Canvas." The aim is to give business and technical stakeholders a single, shared view of **what** AI projects to pursue, **why** they matter financially, and **when** to execute them.

This report briefly explains how CanvasIQ works, how it is implemented, what was tested, and where it could go next. The focus is on practical decision support rather than a pure LLM demo.

Product Overview and Flow

CanvasIQ follows a compact, end-to-end flow:

1. **Landing Page** – Introduces the tool with consistent branding and a dark theme. Clear calls to action guide users into the guided interview or a dashboard with demo data.

2. **Guided Interview** – The interview page combines a streaming chat area with a right-hand context sidebar. The agent walks through phases: company context, use-case capture, ROI review, portfolio selection, roadmap generation, and export.

   Prompts ask the model to produce both narrative explanations and JSON describing company context and AI use cases (benefits, costs, effort, risk, impact, dependencies). The client parses the JSON and updates the application state, keeping the conversation and the structured data in sync.

3. **Sidebar & Dashboard** – A scrollable sidebar summarizes **Company Context** and **Use Cases** and shows estimated benefits, ROI, payback, and risk so users can see their inputs reflected immediately.

Once several use cases exist, the dashboard aggregates them into:

- Portfolio-level ROI.
- Three-year NPV at a 10% discount rate.
- Average payback period.
- Count of selected initiatives.

A bar chart shows ROI by use case, and a pie chart shows the share of **selected** initiatives across Q1, 1-Year, and 3-Year horizons.

4. **Roadmap & Export** – The roadmap groups selected initiatives into **Q1**, **1-Year**, and **3-Year**. An auto-assign feature pushes quick wins to Q1 and heavier work to later horizons, with manual overrides available. A short milestone summary turns the roadmap into a few clear commitments.

   The export page renders the AI ROI & Roadmap Canvas as a single page and supports JSON/Markdown downloads plus a print-friendly layout for PDF. Print-specific CSS switches to a light theme, avoids awkward breaks, and adds a footer with credit and generation date.

Architecture and AI Integration

CanvasIQ is built on **Next.js 14 (App Router)** with a clear separation between UI, API routes, and domain logic. Key technologies include:

- **Next.js 14** for routing (`/interview`, `/dashboard`, `/portfolio`, `/roadmap`, `/canvas`) and `/api` endpoints.
- **Tailwind CSS + shadcn/ui** for layout and UI components.
- **Zustand** for global state, persisted to `localStorage` so navigation and refresh do not wipe progress.
- **Recharts** for the bar and pie charts.

State centers on:

- **AIUseCase** objects representing each initiative (problem, benefits, costs, risk, effort, impact, dependencies, timeframe).
- A canvas object summarizing portfolio information for export.
- Chat messages and an interview phase indicator.

Prompts in `lib/prompts.ts` position CanvasIQ as an AI strategy consultant and specify required fields and JSON output. A streaming endpoint (`/api/chat`) uses the OpenAI Node client to send completion deltas to the browser. The client renders markdown as it arrives and scans for `json` blocks; valid JSON is parsed

and merged into state while the user continues chatting. This keeps the interface conversational while financial logic and state updates remain deterministic on the client.

Financial Logic and Canvas Generation

lib/calculations.ts implements simple, transparent formulas:

- **ROI** = (Benefit – Cost) / Cost × 100, where cost includes implementation and annual operating cost.
- **NPV** uses a three-year horizon and 10% discount rate, summing discounted net cash flows and subtracting initial investment.
- **Payback Period** is the implementation cost divided by the monthly net benefit.
- **Risk-Adjusted Value** multiplies NPV by a risk factor (low/medium/high) and an impact/effort ratio. This value drives budget-constrained selection and ranking.

Portfolio metrics are calculated across selected use cases and split into near-term (Q1 + 1-Year) versus long-term (3-Year) impact. lib/export.ts converts company context and selected use cases into a single canvas object, with concise summaries of costs, benefits, timeline entries, risk notes, and a short portfolio-level narrative. That object then powers JSON export, Markdown export, and the on-screen/printable canvas.

Design, Testing, and Limitations

Design & UX. The app uses a dark theme with high-contrast text, a chat interface that streams cleanly without duplicate welcome messages, a scrollable sidebar that scales as more use cases are added, and charts based solely on selected initiatives, so visuals align with portfolio decisions. The print view strips extra visual effects and produces a simple, executive-ready PDF.

Testing. Testing was manual but scenario-driven. Using a TechFlow Solutions example, I: - Captured multiple use cases and reviewed ROI metrics. - Confirmed dashboard aggregates and chart behavior. - Exercised portfolio quadrants and auto-select by budget. - Generated a roadmap and checked timeframe assignments. - Exported JSON, Markdown, and PDF to verify content, footer, and branding. - Confirmed that navigation and refresh do not wipe state.

Limitations and Next Steps. Current constraints include: - Dependence on a working OpenAI API key; no offline mode. - Client-side state only; no multi-user persistence, sharing, or audit trail. - Fixed financial assumptions (discount rate, horizon). - Basic error handling for malformed AI-generated JSON. - Accessibility not fully audited, especially for charts.

Natural extensions would be to add a backend for authentication and saved canvases, make finance settings configurable, model dependencies and team capacity more explicit, validate AI outputs with JSON schemas, offer richer export formats (branded PDFs, slides, shareable links), and complete a deeper accessibility pass.

Conclusion

CanvasIQ shows how an AI-assisted interview can pair with transparent financial logic to support concrete AI investment decisions. It moves users from an open-ended conversation to a structured portfolio and a presentable canvas, while keeping calculations explainable and straightforward.

Although the current version is intentionally scoped to client-side state and fixed assumptions, it provides a solid foundation. With the extensions outlined above, CanvasIQ could grow into a credible, collaborative tool for AI portfolio planning and governance.