Pre-Screen Assignment

American Express · Digital Banking · July 2026

Enterprise KB Chatbot
Product Strategy & Roadmap

Udit Sehgal · Pre-Screen Assignment · Technical Product Owner

01 Problem & Users

The problem. Enterprise knowledge exists — in help centers, wikis, policy docs, and runbooks — but employees can't get to the right answer fast. They ask a colleague, escalate a ticket, or act on stale information. The cost shows up as longer handle times, inconsistent answers to customers, slow onboarding, and repeated escalations for questions that have documented answers.

Primary users.

- **Frontline servicing / customer-care agents** — need one correct, current answer mid-interaction, in seconds.
- **Operations and back-office teams** — need procedure and policy answers to execute correctly the first time.
- **New hires** — the heaviest question-askers; today they consume senior colleagues' time.

Secondary users: knowledge managers and content owners (need to see what's missing, stale, or mistrusted), and compliance/risk partners (need confidence the bot won't answer outside its lane).

Key challenges today.

1. **Trust.** One confidently wrong answer and users stop using the bot — the failure mode isn't "no answer," it's an *ungrounded* answer. Basic retrieve-and-generate systems hallucinate when retrieval misses.
2. **Coverage vs. quality.** Knowledge is scattered and uneven; ingesting everything imports noise and contradictions.
3. **Freshness.** Policies change; a bot serving last quarter's policy is worse than no bot.
4. **No learning loop.** Today's system can't tell us which answers failed, which questions have no source content, or where users gave up.

02 Product Strategy

Vision: trusted answers at the speed of conversation. The product's job is not to answer every question — it's to be *right when it answers, honest when it isn't sure, and measurably better every week.*

Strategic pillars.

Grounded, cited, or silent.

Every answer is generated strictly from retrieved source passages and shows its citations. Below a confidence threshold, the bot degrades gracefully to "here are the top source documents" rather than guessing. This is the single biggest driver of sustained adoption.

The feedback loop is the product.

Every conversation ends by asking the user to rate the answer; that stream — ratings, escalations, unanswered questions — drives content gap-mining, retrieval tuning, and the roadmap itself.

Guardrails as a feature, not a constraint.

A hard, code-level firewall keeps the bot out of topics it must not free-lance on (HR/legal/compliance determinations) — it recognizes them and routes to the right human channel with context attached. In a regulated enterprise this is what makes broad deployment possible at all.

How this improves on what exists. Search returns documents and makes the user do the reading. Basic chatbots answer fluently but unaccountably. This product closes the gap: direct answers with receipts, an explicit "I don't know" behavior, and instrumentation that turns usage into a prioritized content and retrieval backlog. A large share of *general* servicing questions can be answered this way from the public and internal knowledge base alone — no core-system integration — which is why the roadmap below can start cheap, safe, and fast.

03 Roadmap (Phased)

0 Agent-assist (optional on-ramp)

The same grounded assistant, deployed first in front of care professionals rather than customers — the answer and its citations appear beside the agent, who stays the human in the loop. Proves quality on real traffic with zero customer-facing risk, and mirrors how regulated enterprises typically graduate GenAI.

1 Trust Foundation

Capabilities: strict grounding with inline citations; confidence-threshold fallback to source search; end-of-conversation rating capture; an offline evaluation harness — a golden set of real questions with verified answers, scored on every retrieval or prompt change; sensitive-topic firewall with routed handoffs.

Why first: trust is the adoption bottleneck. Nothing else matters if week-one users catch the bot hallucinating. The evaluation harness comes first, not later — you cannot safely improve what you can't measure.

2 Coverage & Freshness

Capabilities: onboarding of additional knowledge sources, each gated by an evaluation pass (a source ships only if it raises answer quality on the golden set); freshness pipeline — content sync SLAs, stale-content detection, change-triggered re-indexing; analytics dashboard for knowledge owners (top unanswered questions, lowest-rated answers, stalest high-traffic content); gap-mining that converts unanswered questions directly into content-owner tickets.

Why second: with trust mechanics in place, coverage growth compounds value instead of compounding noise.

3 Scale & Action

Capabilities: role-aware answers (an agent and a new hire get differently scoped responses); authenticated, account-aware answers under full model-risk validation; action-taking for the top validated intents — open the ticket, start the form, not just describe it; additional channels (agent desktop, Teams/Slack); latency and cost optimization (answer caching for the frequent head, model right-sizing per query class).

Why third: personalization and actioning multiply value but also multiply blast radius — they're safe to build only on a measured, trusted foundation.

04 Prioritization — Top 5

#	Priority	Rationale
1	**Grounded answers with citations**	Directly attacks the #1 failure mode (hallucination); prerequisite for everything
2	**Evaluation harness + golden question set**	Makes quality measurable; converts every later change from opinion to evidence
3	**Confidence fallback ("cited or silent")**	Converts worst-case from *wrong answer* to *useful search result*
4	**Rating / feedback loop**	Cheapest instrumentation, feeds prioritization forever
5	**Freshness pipeline**	Stale answers erode trust as fast as wrong ones; critical in policy-driven domains

Criteria used: (1) impact on user trust — the adoption currency; (2) reach — % of interactions touched; (3) risk reduction — regulated-environment safety; (4) effort — favoring high-leverage, low-build items first; (5) dependency order — measurement before optimization, trust before scale.

05 Trade-offs

Accuracy vs. speed.

Tier the experience: cache validated answers for the frequent head of the question distribution (fast *and* accurate); stream responses so perceived latency drops without touching retrieval depth; for long-tail queries, spend the extra retrieval passes — a correct answer in 6 seconds beats a wrong one in 2. The confidence threshold is the pressure valve: when deep retrieval still isn't confident, return sources quickly instead of generating slowly and wrongly.

Coverage vs. noise.

Gate every new source behind the evaluation harness: it ships only if golden-set answer quality goes up, not just document count. Weight retrieval by source authority and recency so a definitive policy document outranks a five-year-old slide deck. It's better to be narrow and right, then widen deliberately — users forgive "I don't cover that yet"; they don't forgive confidently wrong.

Feature delivery vs. system reliability.

Treat answer quality as an SLO with an error budget: hallucination and groundedness regressions measured on a continuously audited sample. When the budget is healthy, ship features; when it's burned, the team's capacity pivots to quality until it recovers. New capabilities roll out canary-first (one user segment, one knowledge domain) with automatic rollback triggers on rating-score drops.

06 Success Metrics

North star: self-serve resolution rate — % of questions resolved without escalation to a human or ticket.

Type	Metric	Why it matters
Quality (leading)	Groundedness / citation accuracy on audited sample	The trust number; gates all releases
Quality (leading)	Retrieval hit rate on golden set	Isolates "can't find it" from "can't say it"
Engagement (leading)	End-of-conversation rating score	Fastest user-truth signal
Engagement (leading)	Weekly active users / repeat usage	Adoption is a trust proxy — users return only if it worked last time
Outcome (lagging)	Self-serve resolution rate; escalation deflection	The business case
Outcome (lagging)	Time-to-answer vs. baseline (search / ask-a-colleague)	Productivity value, measurable in agent handle time
Health (guardrail)	Sensitive-topic violations (target: zero); freshness SLA compliance; p95 latency	Safety and operability floors

What matters most: groundedness and self-serve resolution. One is the leading indicator of trust, the other the lagging proof of value — every roadmap item above exists to move one of these two numbers.

Postscript — we built it

By the way: rather than only writing this strategy, we built a **live demo pilot** of it — a running Phase-1 instance, taken from assignment to QA'd product in a single day. Two chat bots are running right now — one answering **Amex banking questions** from a database of 80 indexed public americanexpress.com help-center articles (every answer cites and links to its live source page, declines to guess on weak retrieval, and routes sensitive topics to a human), and one answering questions about **Udit** and how the demo was built. The improvement loop ran for real: a 20-question retrieval test scored 17/20, drove same-day tuning, and re-ran at 20/20. And the build-vs-buy judgment is lived, not theoretical — Udit's own résumé line runs a deliberately *bought* packaged voice agent, while this demo was deliberately *built*, because here the trust mechanics are the product. Build process, architecture, QA, and the release timeline are documented on the site itself.

If you prefer to chat with the strategy instead of reading it:

demo.millionautomations.com

The demo is unofficial, non-commercial, and clearly labeled as such; its database is a July 18, 2026 snapshot of public help-center pages, and every citation links back to the live americanexpress.com source.