RegenAI

Recovery commerce — live experience, engineering vision and delivery roadmap

Project by Zahidul Islam. Client review edition, updated Oct 8, 2026.

Live frontend: https://regenai.zahidul-islam.com

Latest review — 14 October 2026

The latest documented release combines the Shopify-backed Hydrogen storefront with the Python Support Studio assistant on the same public site. The assistant adds contextual chat, encrypted saved preferences, image understanding, bounded web retrieval, ticket recommendations and owner-reviewed decisions. Sections 15–19 retain the detailed capability, operations and handoff evidence; the earlier storefront milestones remain below as history.

Fresh read-only checks on 14 October 2026 returned HTTP 200 for the storefront, Support Studio and cart pages. The assistant health endpoint reported status ok and worker running; account sign-in returned HTTP 404. These checks establish those responses at review time, not continuous uptime, a complete browser journey or renewed Shopify/Gmail access.

The local publication source remains at commit 67d9afb, dated 8 October 2026. The recorded release checks include 86 Python tests, 184 JavaScript tests with one existing skip, six workflow regressions, type/lint/security/build checks and seventeen public HTTP checks. These are the dated October 8 results, not a new test run. That checkpoint also records fourteen successful native GitHub checks, three expected opt-in skips and an outstanding required approving review on main.

Public ordering and customer-account sign-in remain closed in the documented release. External refunds and outbound email remain disabled; no real refund or delivered customer reply is claimed. The Gmail consent arrangement can expire after seven days, so the prior successful connection checks must not be read as proof that access is still valid today. Stable client grants, populated authorized ticket/order cases, genuine controlled action receipts and client handoff remain open acceptance items.

This review adds no application feature, deployment or paid inference. Million-user capacity, observed thirty-day availability, full legal compliance and a complete commercial launch remain unproven. The following earlier material is retained; newer dated evidence takes precedence when reading historical descriptions of the former static storefront, pending Hydrogen cutover or earlier test totals.

Current release update

The live main storefront now runs the Shopify-backed Hydrogen application. It preserves the approved 3D design while reading six sample products from the development store. Visitors can browse categories, search, compare products, open product pages, and read help, contact and policy content. The public cart explains that ordering is closed; account sign-in and checkout remain disabled. These are store designs, without verified saleable inventory or a real-payment offer.

At the prior frontend milestone, the source passed 137 storefront unit tests with one existing skip, type checks, lint with zero errors and four prior warnings, Hydrogen and static-preview builds, and scoped security scans. Independent source review found no blocking defect. The same tested image was released on staging and main with 144/144 and 145/145 release-file parity respectively. Public browser checks on each host covered route status, motion and reduced-motion behavior, finder copy, search

requests, mobile/desktop automated accessibility scans, and ten hero widths without overlap. These are bounded checks, not full accessibility or production-readiness proof.

Frontend-first phase status: F1 shopping journey is implemented for the sample catalog. Its enabled controlled development-store cart passed a local desktop/mobile add, quantity and remove sequence of 0 to 1 to 2 to 1 to 0; a real unavailable-variant browser case and fresh full-journey review remain due. F2 frontend completeness has a reviewed live release; a complete manual screen-reader journey and broader visual acceptance remain due. B1 buyer-platform test checkout and account sign-in, B2 merchant installation and Function activation, R1 production controls, and R2 integrated release and handoff remain due.

The development store's test payment gateway and staging account callback settings were verified, but no test order or account login was completed. A separate merchant application has an inactive draft with three Shopify Functions; it has not been installed or activated. No real payment or physical order is open.

The completed frontend source is pushed to the GitHub branch fix/hydrogen-edge-transform. It has not been merged into main. The earlier static release was retained as rollback material, but a live rollback drill has not been run. The remaining sections preserve the original September snapshot and longer-term plan; their descriptions of a former static container and pending Hydrogen cutover are historical.

RegenAI is a Shopify-focused recovery and wellness commerce portfolio project. The vision is to combine a distinctive shopping experience with dependable commerce, evidence-aware content, responsible recommendations and an architecture that can grow with a real merchant. The current release combines a Shopify-backed storefront and a Python AI support assistant. It is not an operating medical business, a clinically validated product range or a store accepting real payments.

This document separates verified delivery from implementation in progress, planned controls and longer-term ideas. It replaces the earlier day-by-day snapshot; historical CI results, product claims and deployment assumptions are not treated as current proof.

1. What clients can review today

The selected new design is live on its own HTTPS subdomain. It uses a warm neutral background, blue accents, editorial typography and original product imagery. The experience includes a moving Three.js product scene, scroll-driven product inspection, a lab-inspired visual section, responsive navigation and a persistent shopping bag.

The six sample products are Pulse One, Form Roller, Range Kit, Restore Wrap, Point Duo and Align Pebble. Collection browsing, search, product details and options, a recovery finder, the bag, informational pages and not-found views are available. Prices, products and specifications are catalog content rather than an offer for sale. The finder supports product discovery; it does not diagnose a condition or establish medical suitability.

Original Blender scenes and web-ready product assets support the design. The public experience uses optimized images and 3D assets with loading and reduced-motion behavior. An approved generated architectural image supports the visual direction. A paid generated video has not been produced; future video work would need a clear creative benefit, accessibility treatment, performance budget and spending approval.

The earlier shopping bag used browser-local state. The live catalog now reads genuine Shopify data; the public cart, customer sign-in and ordering are closed. Account/order-history and controlled checkout acceptance remain due. The public site deliberately carries noindex controls while real content, SEO and commercial readiness remain under development.

2. Delivery status and evidence

HISTORICAL VERIFIED — The earlier static frontend release passed its seven acceptance criteria and independent review. Dedicated frontend TypeScript checking, linting and production build passed, along with 24 focused recovery component and behavior tests. The release is a built static artifact, not a development server.

HISTORICAL VERIFIED — The earlier static frontend HTTP checks covered 22 valid and missing routes, plus rejection of unsupported request methods. Chrome checks exercised 3D initialization, scroll chapters, product navigation, bag persistence, keyboard-focus restoration, reduced motion and layouts at 390, 768 and 1440 pixels. The recorded browser run had no uncaught application, hydration or console errors and no unexpected external service requests. These checks describe the exercised journeys, not an exhaustive security or accessibility audit.

HISTORICAL VERIFIED — Trusted origin HTTPS and hostname validation succeeded; the public site was served through Cloudflare. All 28 deployed release files and the installed site configuration matched the locally tested candidate byte-for-byte. Existing shared site configurations were preserved. An off-server release archive was extracted and compared successfully. The static bundle and deployment tooling passed the scoped secret scans; Python deployment tooling passed the configured security checks.

CURRENT — The selected design now runs on the verified Hydrogen/Node runtime with genuine Shopify catalog reads. Runtime, shutdown, SSR and bounded frontend release checks have passed; account/checkout and complete commercial acceptance remain open. Earlier foundation and design-system checks are useful history, but do not establish that the latest full repository is ready for a commercial launch.

EXISTING SOURCE, REVALIDATION REQUIRED — The repository contains a merchant application, design-system/Storybook work and four Rust/WASM Shopify Function projects: cart validation, B2B tiered pricing, delivery customization and discount interaction rules. On 24 September 2026 the four Functions passed 47/47 native unit tests and built to WebAssembly at 159–181 KiB each, under Shopify's 256 kB limit. Native runner compatibility, store eligibility, activation and real controlled development-store checkout behavior still require fresh verification. Source files and old compiled artifacts alone are not proof of enforcement.

HISTORICAL ISSUE — The older merchant Worker had no verified sign-in or shop isolation when inspected on 24 September 2026. Its known public URLs are now disabled. Local authentication, token encryption, atomic OAuth state and tenant repairs exist, but remote migration and independent merchant acceptance remain due.

PLANNED — Full privacy, legal applicability, security, SEO, accessibility, performance, observability, availability and scale acceptance work. No million-user benchmark, observed 30-day uptime result, legal certification, clinical validation or full-project production-readiness claim is made.

DOCUMENTED — On 24 September 2026 public architecture, scalability, reliability, security, privacy, compliance and accessibility documents were published in the GitHub repository (see section 13). They describe targets and current evidence; they are not certifications.

3. Architecture: current release and intended platform

Current public path: browser → Cloudflare → HTTPS reverse proxy → Shopify-backed Hydrogen container and same-origin Python support container. Both containers run without root privileges, with read-only root filesystems and restricted capabilities; application listeners are private to the host.

Assistant records and jobs use SQLite on a persistent data mount, with sensitive record payloads encrypted. The shared host is one failure domain; multi-host resilience remains planned.

Current storefront: React and TypeScript with Shopify Hydrogen/React Router server rendering on a validated Node runtime. The selected visual design is retained with typed Shopify catalog reads. Cart/account adapters and their complete acceptance remain separate milestones. Public content may use carefully controlled caching; customer-specific cart, account and administrative responses must remain isolated. Shopify remains the commerce system of record and owns supported checkout and payment processing.

Target merchant layer: an authenticated merchant application with shop-scoped access, review and publishing workflows, versioned content and an auditable decision history. The existing Workers/D1 approach remains a candidate for this layer while authentication, token protection, environment separation and tenant boundaries are repaired and tested. Hosting the storefront on a separately managed server does not replace Shopify or automatically require Oxygen. Platform eligibility and provider limits must be checked for each capability.

Target checkout rules: the four existing Shopify Function projects are evaluated through the supported runner and actual eligible development-store journeys. Rules must use validated, versioned input data, handle missing or malformed data and behave correctly when discounts, currency and delivery options interact. No LLM or arbitrary network dependency belongs in a synchronous checkout validation path.

Target data and recommendation layer: begin with transparent deterministic ranking and an honest no-match fallback. Add retrieval or generation only where a measured benefit justifies the complexity, privacy exposure and cost. A vector database, warehouse or transformation stack is introduced for a concrete retrieval/reporting need, not merely because it appears in an architecture diagram.

Configuration is intended to have one owner for each changing value: validated environment settings for deployment/provider behavior, maintained data files for catalog and business copy, and private secret storage for credentials. Reusable components and a documented design system support consistency across the storefront and future interfaces.

4. Shopify commerce and merchant workflow roadmap

Catalog integration will map real development-store products, collections, media, variants, options, currencies and availability into the accepted design. Acceptance includes filtering, sorting, pagination, long and missing content, sold-out variants, timeouts and truthful 404 responses. Local catalog mode remains explicit; a failing live API must not silently become a made-up successful product response.

Cart and checkout work will implement authoritative Shopify totals, validated quantity/variant mutations, separate buyer sessions, visible mutation errors and a fresh supported checkout URL. Two-browser isolation, inventory contention, retries and error recovery are required. Testing uses a verified development store and supported test-payment flow; no real charge is implied by deployment or a controlled development-store exercise.

Customer-account work covers supported sign-in, sign-out, expiry, callback protection and permission-scoped order/address views. Cross-customer access must be denied. Account tokens and sensitive request data must not appear in logs. Export/deletion utilities need authenticated ownership checks and explicit downstream boundaries.

Merchant administration will validate install/session flows, atomic one-time callback state, token encryption and rotation, least-privilege roles and shop isolation on every data query. Review actions need valid transitions, concurrency protection, required reasoning and an append-only audit history. Content must not acquire a fabricated clinician endorsement because an approval button exists.

Webhook and job processing will verify signatures before processing, associate the correct shop, tolerate duplicate and out-of-order deliveries, use idempotency and bounded retries, and support recovery after a worker failure. Uninstall and required privacy topics must remove or revoke the relevant access and data according to the approved lifecycle.

Expanded commerce remains planned: B2B company/location pricing, supported markets and localization, subscription lifecycle and cancellation, eligible checkout extensions, account utilities and explicitly specified affiliate or consultation workflows. Store plan, distribution method, platform grants, business decisions and legal review determine which features can genuinely ship. No premium Shopify capability or subscription is assumed to be available for free.

5. Security and trustworthy engineering

The current frontend/support release has scoped protections: HTTPS, restrictive response headers, same-origin boundaries, noindex controls, limited container privileges, protected configuration and local-first releases. Assistant controls include encrypted records, server-owned workspace scope, bounded input/storage/provider usage and version-bound approvals. These are bounded checks, not blanket repository security certification.

The public security model, published 24 September 2026, lists nine tracked gaps (SEC-01 to SEC-09). Release blockers cover merchant token encryption at rest, per-request authentication and shop scoping on merchant routes, atomic one-time OAuth state and separated environments; the earlier HSTS gap was resolved on the checked frontend hosts; the wider domain policy has not been audited. Security issues can be reported privately as described in the repository's security policy.

The next security workstream establishes a threat model and a versioned control mapping using OWASP guidance. It covers authentication and authorization, tenant separation, session rotation, supported administrator MFA, CSRF, XSS, injection, SSRF, redirect/upload validation, trusted proxy handling, bounded request sizes, timeouts and abuse controls. Negative tests must prove the boundaries, not merely a successful happy path.

Secrets must stay out of source, generated browser assets, logs, screenshots and public documentation. Recovery records remain private. Dependency findings must be assessed for the deployed path and corrected or explicitly dispositioned; the broader dependency and runtime audit is not yet complete. An automated scanner result is one layer of evidence, not a security certificate.

The development workflow requires acceptance criteria before changes, narrowly scoped implementation, tests/type/lint/security/build gates, real HTTP or UI exercises and independent review. Genuine CI tests remain versioned; private manual tooling and recovery notes stay excluded. Missing scenarios, unavailable credentials and failing checks must be visible rather than converted into artificial success.

Deployment follows local source → verified artifact → versioned release → targeted activation → public behavior checks → file parity. Release manifests, retained rollback artifacts and checks against live drift make changes reviewable. Existing working services are preserved. The final integrated system will also need independent application security testing and an incident-response exercise before real merchant use.

6. USA and EU readiness: an applicability-led plan

The intent is to support clients serving the USA and EU. This requires a maintained, jurisdiction-specific applicability register rather than a promise to satisfy every law. Each applicable obligation needs an accountable owner, current primary source, actual merchant/data/product facts, an implementation

control, test evidence and qualified review where appropriate. The following are launch workstreams, not completed legal opinions.

Privacy and visitor data: inventory forms, cookies, browser storage, logs, account data, processors and cross-border transfers. Establish purposes, applicable lawful bases, notices, retention, deletion/export and effective preference withdrawal. Optional analytics or marketing must obey applicable consent/choice rules. EU GDPR and national electronic-communications rules need review; US state privacy laws, including California requirements where applicable, need separate assessment. Merely hosting in Europe does not establish compliance. [1][2]

Recovery-finder and health-related data: assess each field and possible inference, whether it can be linked to a person and where it travels. The design goal is minimal, ephemeral discovery inputs, without sending sensitive answers to advertising, analytics, AI or customer profiles by default. The relevant review may include GDPR special-category data, Washington consumer-health privacy rules and the FTC Health Breach Notification Rule. HIPAA depends on the actual entities, roles and data flows; wellness branding alone does not settle applicability. [1][3][4]

Product claims and safety: sample products must remain clearly identified. A real catalog requires evidence for benefits and endorsements, review provenance, safety and traceability obligations, and product-specific US/EU classification. FDA device status depends on intended use and claims. EU medical-device or general-product safety duties must be assessed for the actual goods. No unsupported FDA approval, CE conformity, clinical efficacy, certification badge or artificial customer review should be published. Automated content checks can flag problems but cannot replace regulatory judgment. [5][6]

Consumer commerce: review seller identity/contact, total pricing, tax/shipping presentation, delivery commitments, returns, withdrawal/cancellation rights, warranties, subscriptions and advertising practices for each market. Terms must match the actual offer and checkout. Cross-border VAT/sales-tax, customs and product obligations require the merchant's business facts and appropriate specialists. The current catalog does not provide approved merchant legal policies. [7]

Accessibility: use WCAG 2.2 AA as an engineering target and separately assess applicable US obligations and the European Accessibility Act, including relevant national implementation and exceptions. An automated scan or accessible component library does not itself prove legal conformance. Product browsing, forms, dialogs and hosted checkout need their own evidence. [8][9]

Payments and operational governance: keep card capture in supported Shopify/payment-provider flows, review payment and PCI responsibilities, and prove test/live environment separation. Define incident notification, processor responsibilities, data-rights handling, evidence ownership and review cadence. If later AI, diagnostics, telehealth, supplements or wearable features materially change the risk profile, reassess the relevant regulatory framework before implementation or public claims.

7. Accessibility, motion and inclusive use

The visual ambition includes accessibility. Current checks exercised keyboard focus restoration, responsive layouts and reduced-motion behavior. Earlier scoped frontend reviews also recorded automated accessibility checks; these do not replace a fresh integrated-store assessment or formal conformance evaluation.

Planned acceptance includes semantic landmarks/headings, meaningful alternative text, visible focus, dialog and validation announcements, contrast, touch targets, keyboard-only journeys, screen-reader exercises and 200%/400% zoom/reflow. Test mobile and desktop through the full shopping and account journey, including failure states and supported checkout.

3D and motion must enhance discovery without becoming the only source of product information. Preserve static fallbacks, readable content without WebGL, reduced-motion controls and the ability to

use the interface without scroll-driven effects. Future video requires captions or other appropriate alternatives, playback controls and a policy on autoplay and data use.

8. Speed, SEO and search-agent visibility

Performance goals cover the real user journey, not only a Lighthouse screenshot. Planned budgets include first-load JavaScript, image/3D/video bytes, request count, main-thread work, server-render latency and cache behavior. The current Three.js bundle remains a material optimization area. Measure representative devices and networks, and distinguish controlled lab results from real-user field data.

The intended field-performance goals are p75 LCP at or below 2.5 seconds, INP at or below 200 milliseconds and CLS at or below 0.1, once there is sufficient representative traffic. These are targets, not measured outcomes of the present release. Use lazy-loaded enhancement, responsive media, careful fonts, bounded third-party code and public-only caching, then verify the tradeoffs. [10]

Traditional SEO work includes meaningful server-rendered HTML, truthful HTTP status codes, canonical URLs, sitemaps, internal links, appropriate localization/hreflang, visible-content-consistent product/offer/breadcrumb metadata and eligible merchant feeds. Do not publish structured-data ratings or stock claims that the visitor cannot substantiate. Private, account, cart and staging content must have appropriate crawl controls.

Search-agent optimization means making accurate product and business information accessible to supported search and assistant systems. Priorities are clear entities, concise factual descriptions, reliable URLs, provenance, machine-readable data and explicit crawler/access policies. Google states that its AI search features do not require special additional optimization beyond established SEO practices; eligibility and inclusion remain under the platform's control. Other agents require their own current documentation and capability review. No ranking, citation, recommendation or agent-driven sale is guaranteed. [11]

The public site currently remains noindex. Search Console validation, indexability decisions, merchant feeds and agent-facing commerce integrations belong to the later approved commercial/content release, not this visual-review milestone.

9. Scalability: a credible path from 10k to 1M+ simultaneous users

The objective is an architecture that can evolve toward 10,000, 100,000 and more than 1,000,000 simultaneous active users for a defined workload. These are design scenarios, not proven capacity. Concurrent open tabs, active sessions, requests per second and simultaneous checkout mutations are different measurements and must never be substituted for one another.

The codebase should make future growth understandable: typed configuration, explicit cache/session/job/telemetry boundaries, stateless request handling where appropriate, public/private cache separation, controlled dependency access, durable job processing and repeatable deployment definitions. Some supported upgrades may eventually require configuration changes; independent hosts, data migrations, provider commitments, bottleneck fixes and operational staffing can require substantial work. A million-user claim cannot honestly rest on a few settings or a diagram.

For the 10k scenario, begin with measured browse-heavy traffic, public CDN caching, safe cache invalidation and bounded dynamic requests. Add replicas or shared state only where the measured workload requires them. Exercise cold caches, hot products, bursts, account traffic and cart mutations separately.

For the 100k scenario, assess independent failure domains, load balancing, durable shared state, queue throughput, database access patterns, upstream API budgets, egress and external monitoring. Two containers on the same host are useful software tests, but are not independent-host redundancy.

For the 1M+ scenario, define distributed edge/origin delivery and regional or failover needs from measurements. Coordinate commerce-provider constraints, buyer identification, checkout throttling, bot behavior and data consistency. Introduce partitioning only for a proven bottleneck. Shopify buyer traffic behavior must not be interpreted as unlimited automated traffic, account operations or checkout creation. [12]

Example workload math: if one million active sessions average one application request every 30 seconds, edge application traffic is about 33,333 requests/second. If 5% is dynamic and 99% of the remaining public traffic is cached, origin application traffic is roughly 1,983 requests/second. A cold public cache can move it toward 33,333 requests/second. This example excludes separate asset and background-job traffic; it is neither a benchmark nor a server-sizing recommendation.

Validation must report workload mix, arrival/session model, duration, regions, hardware, cache state, payload/egress, p50/p95/p99 latency, errors, attempted/completed/dropped work and generator saturation. Controlled dependency failures, queue saturation, duplicate events and replica loss need separate tests. Load generators must obey approved targets, limits, provider policies and spending caps; the vision does not authorize flooding Shopify or a shared server.

Current position: the workload contract and scaling roadmap are documented; the distributed runtime, executable load profiles and capacity claims remain unverified. Local Docker is available and was used for this release. Kubernetes is an optional future validation tool, not proof of scale or a requirement imposed before a measured need.

Capacity model published 24 September 2026: with the example assumptions above, origin traffic is about 20, 198 and 1,983 requests per second for 10k, 100k and 1M active sessions with a warm cache, and about 333, 3,333 and 33,333 with a cold cache (double that during a 2× burst). Checkout creation, payments and Shopify Functions run on Shopify infrastructure. Provider limits used in the model — Shopify Storefront, Admin and Functions APIs; Cloudflare Workers and D1 — were checked against official documentation on that date. No load test at these volumes has been run.

10. Reliability, monitoring and recovery

The availability objective, raised on 24 September 2026, is a 99.9% target with 99% as the minimum floor, for each defined core journey and serving region over a rolling 30-day window. A complete 30-day window contains 43,200 minutes: the 99.9% target allows about 43 minutes of downtime, and the 99% floor allows 432 minutes, or 7 hours 12 minutes. This is an engineering objective, not a contractual SLA or a currently observed result.

Error-budget policy: while more than half of the monthly budget remains, releases proceed normally; with 25–50% remaining, each release needs a reviewed rollback plan; below 25%, only reliability and security fixes ship; once the budget is exhausted or availability falls below the 99% floor, releases freeze and reliability work takes priority until recovered. The planned resilience approach uses independent origins behind a health-checked load balancer, external journey probes and tested recovery; the current single-host frontend/support deployment has no measured observation window or independent failure-domain resilience.

Monitoring must distinguish a homepage response from a working product, account or checkout journey. Planned external US/EU probes need explicit intervals, deadlines and expected responses. Missing telemetry is unknown, not successful uptime. Count customer-visible maintenance and dependency

failures while recording their causes separately. Controlled development-store checkout probes must not charge real cards.

Operational telemetry is planned for latency, errors, resource saturation, queue age, failed jobs, API throttling, webhook processing and backup/certificate health. Logs and traces must use redaction and bounded retention/cardinality. Dashboards require appropriate access control; alert delivery and escalation ownership must be tested. No staffed 24/7 on-call service is claimed.

The earlier static release has retained off-server artifact/restore evidence. Current frontend/support releases use immutable image/source/environment manifests, host-only application-consistent snapshots and retained rollback material; a timed full rollback/data-restore drill and independent-host backups remain due. Future persistent commerce/app data requires application-consistent backups, access protection, retention, recovery drills and measured recovery point/time objectives. A container restart cannot recover a failed host; a monitor or sole backup on that same host does not provide independent protection.

Release and incident playbooks should cover dependency outage, bad deployment, compromised credentials, queue backlog, data restoration and rollback/migration compatibility. Error-budget consumption should influence release decisions. Actual achievement of the 99.9% target or the 99% floor can be reported only after the agreed observation window and journey coverage exist.

11. Delivery plan and acceptance milestones

The current master plan has six delivery phases, A–F, with production-readiness and scalability workstreams running across them. It contains 61 task contracts of unequal size. The task count is not a completion percentage. The owner accepted the selected design and authorized source publication, existing-host deployment and this preserving document refresh. Remaining real financial/email execution and client-installation scope is not completed.

Phase A — Preserve and accept the frontend. Retain the chosen design, original asset provenance, accessible fallbacks and visual regression evidence. The bounded public visual-review milestone is delivered; later requested visual changes still require their own review.

Phase B — Foundation and genuine Shopify storefront. Complete dependency/configuration work, prove the Node/Hydrogen runtime, integrate the chosen presentation, connect catalog/cart/accounts and validate controlled development-store checkout. Exit with real SSR and buyer-isolation evidence, not local-catalog-only success.

Phase C — Merchant application and Shopify Functions. Repair install/authentication, protected token storage, tenant separation, environment isolation, review/publishing workflows, webhooks and native Function validation. Exit with authenticated merchant journeys, recovery evidence and actual eligible development-store enforcement.

Phase D — Approved commercial capabilities. Implement the chosen B2B, markets/localization, subscriptions, checkout extensions, account utilities and specified partnership workflows. Feature eligibility, merchant decisions and grants are prerequisites; unsupported behavior must remain explicitly blocked.

Phase E — Recommendations and data. Establish privacy-approved inputs and deterministic ranking first, then evaluate any AI/retrieval provider, bounded budgets, grounded output, catalog allowlists, injection resistance and failure fallback. Add reporting infrastructure only where justified. Recommendations remain discovery assistance, not clinical advice.

Phase F — Integrated release and handoff. Tie CI to the actual candidate artifact, complete applicable production/scale gates, validate staging, prepare rollback and recovery, deploy with drift checks, prove

public behavior and parity, and deliver operations and client evidence. The frontend/support publication closes only its bounded source and deployment subset of this phase.

Workstream P — Legal applicability; application security; privacy; honest product/commerce content; accessibility; SEO/agent visibility; performance; observability; reliability; and an evidence index linking each claim to source, tests, results and limitations.

Workstream S — Workload definition; portable runtime/state boundaries; bounded dependencies and durable jobs; executable load tests; local multi-instance/recovery proof; a costed scaling decision register; and separately authorized distributed validation at increasing tiers.

The runtime/design/catalog integration is delivered. Next priorities are stable client authorization, rulebook acceptance, genuine approved refund/reply verification, protected merchant installation and controlled buyer account/checkout acceptance. Dates and fixed delivery promises should be set after the remaining scope, access and acceptance dependencies are confirmed.

12. Cost model and longer-term vision

The baseline favors existing infrastructure, local tools and suitable open-source or free-tier components. The current release reused existing hosting; it did not purchase a new server or domain. Existing hosting itself has an ongoing cost. Free tools do not eliminate bandwidth, storage, provider usage, specialist review or operator time.

Growth decisions need a dated bill of materials: measured trigger, selected service, configuration or code affected, migration/recovery work, unit cost, provider limits and approval. Likely paid areas include independent compute, balancing/failover, durable shared storage/queues, external monitoring, log retention, backup storage, egress, distributed test generation and legal/security reviews. Optional image/video/AI usage and premium Shopify capabilities need separate budget decisions; no current price quote or unlimited free scale is promised.

The longer-term product vision includes a richer recovery catalog, additional markets and languages, mobile experiences, supported wearable integrations, progress-oriented interfaces, white-label tenancy and partner workflows. Earlier ideas involving biomarker data, supplements, women's health, camera-based posture features or telehealth are exploratory directions rather than delivered modules or committed near-term scope. They require distinct privacy, evidence, product-safety, licensing and clinical/regulatory assessment before adoption.

The intended client value is a distinctive frontend backed by inspectable engineering: a clear delivery status, maintainable boundaries, measured quality, a defensible growth plan and honest limits. Success is shown through working customer and merchant journeys and retained evidence, not the number of services or technologies listed.

13. Public documentation on GitHub

On 24 September 2026 a public engineering documentation set was added to the project repository. It was initially on the build/b02-foundation branch, submitted to the main branch as draft pull request #19, preserved as earlier delivery history. The integrated source and updated public guides are now published through pull request #20 on fix/hydrogen-edge-transform; main still requires an approving review. Each document labels items as live, built, partial, planned or not measured, and none of them is a certification.

Architecture — components, frontend/backend split, trust boundaries, hosting and repository map.

Scalability — capacity model and tier-by-tier path from 10k to 1M+ concurrent users, using provider limits checked against official Shopify and Cloudflare documentation.

Reliability — 99.9% availability target with 99% floor, journey-level indicators, error-budget policy, backup and disaster-recovery objectives, incident response and observability.

Security model and security policy — threat model, verified controls, nine openly tracked gaps (SEC-01 to SEC-09) and how to report a vulnerability privately.

Privacy — current storefront and assistant privacy notice, including chat, memory, images, tickets, orders, processor access and retention and the planned data map for the full platform.

Compliance — overview of PCI DSS, GDPR, US state privacy, consumer-health data, health-product claims, consumer protection and accessibility obligations for a real launch; not legal advice.

Accessibility — WCAG 2.2 AA engineering target, tested scope and known limitations.

Package guides — all ten applicable READMEs for the storefront, design system, merchant application, Function packages, frontend deployments and Python support service, plus contributing guidelines, a code of conduct and issue/pull-request templates.

14. Primary references

These sources support the planned review framework; they do not certify this project or replace merchant-specific advice. Recheck applicability when products, markets, data use or platform capabilities change.

[1] EU General Data Protection Regulation

[2] California Department of Justice, CCPA

[3] Washington Attorney General, consumer health privacy

[4] FTC health privacy guidance

[5] FTC Health Products Compliance Guidance

[6] FDA product/device determination

[7] European Commission, Consumer Rights Directive

[8] W3C WCAG 2.2

[9] European Commission, European Accessibility Act

[10] Google web.dev, Web Vitals

[11] Google Search Central, AI features and websites

[12] Shopify API limits

[13] Shopify Functions API and resource limits

[14] Cloudflare Workers platform limits

[15] Cloudflare D1 limits

[16] Google SRE Workbook, implementing SLOs

15. AI support assistant: current capability and boundaries

The same public site includes Support Studio and a storefront question panel in the approved cream, navy and cobalt design. The Python service provides contextual text conversations, encrypted saved memory, image understanding, allowlisted live-web retrieval, durable jobs, setup logs, rulebooks and a versioned support review workflow. The storefront continues to read the six Shopify-backed sample products; checkout and customer-account sign-in remain closed.

Shopify support uses a separate installed app with limited order-read/order-write scopes on the owned development store. Its client-credentials grant renews without assuming a refresh token. Gmail uses the

selected account and a dedicated RegenAI support label; the granted credential is encrypted in the assistant. Actual account/label reads, token refresh and restart persistence passed. The current label had no support messages when checked, so this is not proof of processing a populated real-client inbox.

Real AI verification exercised Spanish saved preference recall with image understanding and a damaged-item $89 support recommendation. Exactly two final inference requests cost $0.010510. These results validate those bounded cases, not fluency in every language, unrestricted image understanding or future capacity. No additional paid inference was used for publication.

The support workflow reads a ticket and order context, evaluates the store rulebook, recommends an eligible outcome and drafts the reply. A human reviews the exact current ticket/order/rulebook version and approves the action before execution. Policy changes or stale approvals require a new decision. Financial and email execution are currently disabled; no real refund or sent inbox reply is claimed.

Stored action claims, order locks, receipt validation and ambiguity quarantine are designed to prevent duplicate effects. A timeout after a provider may have acted must enter reconciliation rather than repeat the refund or send blindly. Genuine provider receipts, crash recovery and a controlled approved end-to-end financial/reply flow remain acceptance requirements before activation.

Human escalation retains context and conversation history for review. Actual team routing, staffing, response expectations and retention must be configured per client. The assistant cannot create a staffed overnight team merely by being online. Recurring-job mechanisms exist, but continuous monitoring of this Gmail label has not been activated.

Adapters/configuration paths exist for calendar, email, repositories and messaging services, plus five MCP tools through the HTTP bridge. Only the recorded Shopify/Gmail and bounded web/MCP journeys are verified here. Each additional account requires its own consent, scopes, limits and behavioral evidence. Browser speech-input and read-aloud controls exist and their platform APIs were detected in UI checks; end-to-end microphone/voice quality was not verified. Provider-generated voice/media, expanded channels and comprehensive multilingual evaluation remain future scope.

16. Future assistant scale and provider admission

The one-million-plus active-user roadmap separates storefront sessions, support sessions, request starts, in-flight model generations, queued tickets and external actions. It is not one million simultaneous model calls. The current SQLite deployment encrypts sensitive record payloads and has one authoritative assistant worker on one host; its throughput ceiling has not been measured.

Planning math: one million users with a 1% support share, 0.2 messages per support session per minute and ten seconds of model occupancy yield about 33 model requests per second and 333 in-flight generations. If every user sends one message per minute, the scenario changes to about 16,667 requests per second and 166,667 in-flight generations. These are assumptions, not benchmark results or contracted provider capacity.

Reserve tenant budget and provider request/token capacity before admitting inference. Use bounded queues, waiting/retry guidance, cancellation, deadlines, circuit breakers and controlled degradation. Meter input/output/image/web costs separately; reconcile provider receipts and reject work when a cap is exhausted. Token demand depends on real context and output sizes, so visitor count alone cannot price the service.

Horizontal assistant scale requires authenticated tenant membership, a transactional shared database, durable leased jobs, shared action/order locks, distributed admission control and privacy-approved storage. Migration must retain encrypted records, memory, approvals, audit events and job ownership; prove rollback/restore and cross-tenant denial before promotion. A configuration change alone cannot deliver this migration.

Progression: measure the single-instance ceiling; verify two/four-instance correctness and worker leasing; test 10k and 100k defined workloads with isolated upstream substitutes; then coordinate larger distributed exercises with providers. Report p50/p95/p99, errors, queue age, attempted/completed/dropped work, cold-cache/burst behavior, generator saturation and cost. No load was sent to Shopify or Gmail during publication.

Provider and platform limits remain binding even with more servers. Use quota-aware Shopify Admin jobs, customer-safe Storefront behavior, Gmail batching/backoff and model-provider concurrency/token agreements. External identity, payments and provider availability remain dependencies; their failures count against the defined customer journey. The full capacity and workload contracts are published with the source.

17. Operations, privacy and applicable-law extensions

The availability plan retains a 99.9% target and 99.0% minimum floor per defined journey over a rolling thirty-day window. There is no measured thirty-day uptime result or contractual SLA. Treat missing probes as unknown; include customer-visible maintenance, failed support actions and dependency failures. A responding homepage alone does not establish support availability.

Current releases use locally verified source and immutable images, drift checks, a separate host-only database candidate, exact source/runtime/environment/settings/routing parity and retained rollback material. Cutover preserves one authoritative worker. Rollback must stop the candidate, restart the prior worker, wait for both prior services healthy, then restore routes. Database/schema compatibility is reviewed before rollback.

Planned operations include independent-host resilience, external storefront/support/approved-action probes, tested alert delivery, staffed escalation, redacted telemetry, queue-age/failed-job monitoring, quota/spend alerts and independent encrypted backups. Timed data-restore, full rollback, incident and regional-failure drills remain due. Recovery objectives must be measured, not inferred from a successful container restart.

Google consent currently uses External/Testing status. Gmail refresh access may expire after seven days under that policy; this prevents a claim of indefinitely unattended access. A suitable production-consent/verification arrangement or a documented re-consent process is a client prerequisite. Token-expiry handling checks provider responses, grant type, revocation/consent status, client identity/scopes and clock/expiry metadata without exposing tokens.

The assistant processes conversational text, saved preferences, optional images, tickets, order context and connector grants. Provider inference receives only the approved bounded context needed for the request. Public pages must not invite passwords, payment-card data, medical records or unnecessary identifiers. Configure notices, lawful basis where applicable, processors, transfers, retention, ownership-checked export/deletion and operational logging before each client launch.

Use delegated OAuth and limited Shopify/helpdesk access rather than shared client passwords. Keep secrets in approved private configuration/recovery storage; use secure sessions, origin checks, server-owned tenancy and encrypted record payloads. Rotation, recovery/MFA, access revocation and uninstall/privacy lifecycle need a client runbook. SQLite metadata and host backups also require access protection; payload encryption is not full-disk encryption.

AI safety requires adversarial testing of tickets/images/web pages, policy-grounded outputs, catalog allowlists, human review for sensitive issues and exact financial approvals. Instructions embedded in customer content are untrusted. Recommendations assist support and discovery; they do not diagnose health conditions or establish clinical suitability.

The existing USA/EU legal plan remains applicable. Extend merchant-specific review to AI transparency under EU AI Act Article 50 where applicable, provider/platform policies, consumer-facing disclosures, automated-decision/human-review boundaries and complaint escalation. Product/health claims, consumer-health data, GDPR/US state rules, accessibility, returns/subscriptions and payment responsibilities depend on actual merchant/product/data facts. No legal certification is claimed. European Commission guidance states Article 50 transparency obligations apply from Aug 2, 2026; assess actual provider/deployer roles, content, exceptions and applicable national enforcement before client launch.

Use the published applicability register to assign an accountable owner, primary source, data/product facts, control, evidence and qualified reviewer. Reassess when markets, products, support channels or model processing change. GDPR full-text retrieval was unavailable to the publication agent; article-level legal review remains due. Public security and privacy guides distinguish delivered controls, historical evidence and open gaps.

18. Client implementation, handoff and remaining acceptance

A client setup starts with the store/helpdesk inventory, policy/data permissions, contact/escalation ownership and a chosen installation architecture. Create/install the limited Shopify app, obtain delegated inbox consent, configure the model/budget and protected local or hosted connection, then verify scopes, renewal, account isolation and bounded reads.

Generate or refine the rulebook from the client's explicitly approved policy documents and selected past replies. Record refund/return rules, exclusions, escalation triggers and writing voice with traceable sources; the owner approves the result. The current Gmail label connection does not mean the entire mailbox history was imported or that a client's voice has already been learned.

During first live sessions, compare recommendations with owner decisions on normal, ambiguous and denied cases. Record deviations, edit the same versioned rulebook, re-evaluate affected cases and retain a clear setup log. Enable refund/reply execution only after a verified controlled provider flow and owner-approved safeguards; preserve receipts and reconciliation holds.

Handoff covers installation, delegated access, expiry/re-consent, product/policy refresh, cost caps, backup/restore, escalation, security/privacy contacts, troubleshooting and offboarding. A short client walkthrough and environment-specific recovery record accompany the accepted installation. API keys and customer data never belong in public walkthroughs or repository documents.

The two posted roles include personal and service-delivery requirements outside application code: real client calls/install sessions, natural English, Shopify/helpdesk administration, owner-voice editing, the paid-trial setup and recorded Loom, desired rate, scheduled US Central availability and NDA. These are not proven by deployment, and no completed personal application or paid client trial is claimed.

Near-term exit criteria are stable client grants, approved rulebook/voice, populated authorized ticket/order cases, genuine controlled refund-and-reply receipts, escalation delivery and restart/crash recovery. Later exits cover protected merchant/Functions activation, buyer sign-in/checkout, independent security/accessibility reviews, operational observation and measured scale tiers. Dates depend on access, review, budget and provider eligibility.

19. Publication evidence and source references

All ten applicable README files and the public architecture, release/rollback, privacy/security, legal-applicability, scale/reliability and roadmap guides are published under Zahidul Islam's existing GitHub account through pull request #20. The required approving review on main remains intact; publication to the PR branch is not a merge.

Current source verification records 86 Python tests; 184 JavaScript tests plus one existing Windows skip; six workflow regressions; strict service-directory mypy over fourteen source files; type/lint/security/build and actual wheel/install/import checks. JavaScript lint retains four existing warnings and Python reports one upstream deprecation. Independent source and release-helper reviews approved their bounded scopes.

Native GitHub exposed and led to repairs for the strict type-check gap, a descriptor-read race in a test, an unsuitable URL-regex source assertion and two dead-code findings. Two intended catalog-to-Shopify data-transfer CodeQL alerts remain open for owner/security review despite independent false-positive classification. Their dismissal was rejected by automatic approval review. All rules/hooks remain enabled. The corrected source commit's native GitHub checks record fourteen successful checks and three expected opt-in skips; open alert review remains visible.

Both locally rebuilt frontend/backend images are deployed on the existing host. Exact source/runtime/image/environment/settings/routing parity passed, followed by seventeen HTTP checks, desktop/mobile widget and owner-connection UI checks, restart recovery and real Shopify/Gmail reads. The existing Google Doc was updated in place, preserving its earlier content. Its accompanying PDF is exported from this same document and validated before replacing the previous local file. No additional paid inference or new paid resources were used in this publication work. Million-user capacity, measured uptime, complete legal compliance and real financial/reply execution remain unproven.

Support assistant guide; Release and rollback; Workload and scaling contract; Reliability objectives; Privacy and security model; Legal applicability register; Phased roadmap; Source pull request.

Google OAuth token expiration; Shopify API limits; Shopify refundCreate; OWASP ASVS; NIST AI Risk Management Framework; EU AI Act transparency guidance.