Last updated 16 September 2026, 6:33 PM EDT

VANTAGEPOST · ENGINEERING DELIVERY REPORT

What shipped 14–16 September 2026, and what it means for the product

Every item agreed in the 14 September standup, plus the open items reconciled from the 1 September sync, with the pull request, test run or ticket that proves each one was delivered — and, just as plainly, what was raised on that call and is not delivered yet. This is a living document; it is updated as work lands. Written for a non-engineer first; the code is explained where it matters.

8	16	18	2,583
pull requests open against staging	Linear tickets filed or reassigned	independent review passes	tests passing, whole repository

7
real-app and mockup screenshots

Status: delivered and verified, not yet on the staging site. All eight pull requests are open against the `staging` branch, reviewed and green. They are not built into staging.vantagepost.ai `(https://staging.vantagepost.ai)` yet: seven other merges landed on staging during the same day (PRs #91, #92, #93, #94, #95, #102, #104), which shifted the build base under every open branch several times and forced one branch (#101) to be re-merged against the moving trunk. Rather than build on a base that kept changing, each branch was re-verified against the final state of staging and held for a single, ordered merge. Recommended order, each after the previous one lands and its CI re-runs: #89 → #90 → #96, #97, #98 → #100 → #99 → #101.

Contents

9. Create Article spacing and a real Back control (PR #101)
10. The two keyword mockups
11. Tickets filed and reassigned
12. Items reconciled from the 1 September sync
13. How every item was verified

1. Commitments from the 14 September standup

Each row is one thing the standup agreed would happen. "Delivered" means the work exists in a reviewed pull request or a filed ticket, with the receipt in the last column.

COMMITMENT	STATUS	RECEIPT
Finish the missing article types so every type can be created everywhere	Delivered	PR #89 (https://github.com/deepstation/vantagepost/pull/89) — review found one gap (the copilot's own create tool), fixed on the branch; CI green
Two keyword mockups: before writing and after writing	Delivered	Section 10; interactive version at claude.ai/artifact/AmXVPnfz5BbUkmwViV8aui (https://claude.ai/artifact/AmXVPnfz5BbUkmwViV8aui), light mode with a dark toggle
Fix the remaining generator bugs	Delivered	#96 (https://github.com/deepstation/vantagepost/pull/96) retry data loss · #97 (https://github.com/deepstation/vantagepost/pull/97) silent progress · #98 (https://github.com/deepstation/vantagepost/pull/98) duplicate scheduled articles · #101 (https://github.com/deepstation/vantagepost/pull/101) Create Article spacing/back · VP-242 verified already fixed
Rework the keyword feature, including its fallback when suggestions fail	Delivered	PR #99 (https://github.com/deepstation/vantagepost/pull/99) — tray, Vince prefill, failure message, one-primary rule; browser-verified on localhost
Investigate the share-link/slug bug and file it	Delivered	Traced in code: share links are random tokens and cannot break; the published URL slug is what goes stale. Filed as VP-385 (https://linear.app/deepstation/issue/VP-385)
Update Linear with the full feature-flag design, including the "hide the endpoints" question	Delivered	VP-366 (https://linear.app/deepstation/issue/VP-366) comment; implementation in PR #90 (https://github.com/deepstation/vantagepost/pull/90)
Move billing and auto-refunds to Grant	Delivered	VP-286 (https://linear.app/deepstation/issue/VP-286) reassigned with the agreed rules (token-based charge, auto-refund, stop-proof, no dollar amounts, history placement)

COMMITMENT	STATUS	RECEIPT
Find out why the staging account is missing its articles	Delivered	Cause found: the content copy only ever ran as a dry run (`apply: false`) and the config names a different profile email. Filed as VP-386 (https://linear.app/deepstation/issue/VP-386); the run itself is a production-database write and is held for sign-off
Research the internal MCP tool registry, the Bitwage automation branch and WebMCP; recommend one standard	Delivered	VP-387 (https://linear.app/deepstation/issue/VP-387) — memo with options A/B/C and a recommendation (B)
Put the outstanding standup items into Linear without duplicating Grant's tickets	Delivered	Section 11; no tickets created for read time or fact check (Grant's VP-382/383 already cover them)
Check in with Grant	Scheduled	Material is ready: this report, the mockups and the eight PRs

Raised on the 14 September call and not delivered

Listed so the report reads as a record, not a highlight reel. Each row says where the item stands today.

ITEM FROM THE CALL	WHERE IT STANDS
Written wrap-up of the meeting in Linear	Not written The items were filed as individual tickets instead
Confirm the copilot (Vince) can still do everything it could, including editing an article	Not tested No ticket
Finish article import/export as JSON	Not built Tracked as VP-305 (https://linear.app/deepstation/issue/VP-305); the labelled-sections requirement was added to it
Copy the local articles to staging	Not run Cause found and documented in VP-386 (https://linear.app/deepstation/issue/VP-386); the run writes to the production database and is held for sign-off

ITEM FROM THE CALL	WHERE IT STANDS
Verify the engine still writes its own title when the title is blank	`Not verified` Filed as VP-408 (https://linear.app/deepstation/issue/VP-408)
"Link selected" on the SEO tab, which failed live on the call	`Not fixed` Filed as VP-409 (https://linear.app/deepstation/issue/VP-409)
Per-article cost metering and where the transaction history lives	`Not built` Handed to billing, VP-286 (https://linear.app/deepstation/issue/VP-286)
Performance-monitoring hooks in the engine	`No ticket`
Decide the agent/MCP standard (A, B or C) with Grant	`Not decided` Memo written in VP-387 (https://linear.app/deepstation/issue/VP-387)
Re-point the unpublished local branches onto the new sidebar navigation	`Not done`
Follow-up check-in with Grant	`Not held` This report is the material for it

From the same call on Grant's side, not yet delivered: replacing the staging OpenAI key (VP-361), a delete control and longer brief on proposal cards, the tooltip that sticks during generation, looping the keyword bank into the research phase, and keyword-aware proposal titles.

2. Article types — every type creatable everywhere

PR #89 · VP-240 · feat(articles): restore the four missing article types

`branch micah/vp-240-rebuild-article-types · head e28bef8 · +159 / -28 · 10 files · CI: success`

What it does. The article type list had shrunk to nine. This restores Industry Analysis, Product Review, Case Study and Opinion Piece, so thirteen types are available, and makes every place that offers a type read from one list instead of keeping its own copy.

What users see. The same thirteen choices in the guided Create page, the Traditional Writer, the schedule dialog, the YouTube-to-article dialog and when asking Vince to write — and each choice actually changes how the article is structured, rather than quietly falling back to Thought Leadership.

TASKS IN THIS TICKET

- Four types restored with their own writing guidance (each block is several hundred words of structure instructions the engine prepends to its prompts).
- Every dropdown derives from the one canonical list; the hand-copied list in the guided Create page was removed.
- Both v2 API routes validate against the list, so an unknown type is rejected instead of stored.
- The old test that banned Product Review and Case Study was replaced with a test that each restored type's guidance insists on honest grounding.
- Review gap fixed on the branch: Vince's `generateArticle` tool still carried a hand-written list of seven types, two of which never existed ("How-To Guide", "Industry Trends"). It now uses the canonical list.

Receipts

- Review pass (independent, with shell access) verified all eight surfaces; verdict "needs one change", change made in commit e28bef8.
- `tsc --noEmit` clean; jest for copilot, prompts and article-engine: 11 suites / 253 tests passing; CI "Build & Type Check" green after the push.
- No database migration; migration 0060 (already on staging) had removed the old type check constraint.
- Follow-up filed: VP-391 `(https://linear.app/deepstation/issue/VP-391)` — the classic API still accepts any string for the type.

3. Feature flags — Newsletters and Social can be switched off

PR #90 · VP-366 · feat(flags): switch Newsletters and Social Media on and off from admin

`branch micah/vp-366-feature-flags · head de1893a · +1438 / -86 · 71 files · CI: success`

What it does. Adds a Features section to the admin panel with an on/off switch and an audience (everyone / admins only) for each product area. Turning an area off removes it in three places at once, because a flag that only hides a menu item is not really off.

What users see. Nothing, until an admin flips a switch: both flags default to on for everyone. When an area is off, its sidebar entry and dashboard shortcut disappear, its pages redirect to the dashboard, its public share links return 404, and every one of its API routes answers 404 — the same response a route that never existed gives, so from outside a disabled feature is indistinguishable from a missing one.

TASKS IN THIS TICKET

- Admin UI: Features section with per-area switch and audience.
- Navigation: sidebar, mobile drawer and dashboard quick actions read the flags.
- Pages: gated once at each area's layout, so any new page under the area is covered automatically.
- API: 13 newsletter routes and 24 social routes wrapped; background jobs stop claiming work while the area is off.
- A cross-cutting test walks each area's API directory and fails the build if any handler is not wrapped for the right key.
- Linear: VP-366 updated with the three-layer design and the answer to "can disabled endpoints be hidden, not just blocked" (there is no OpenAPI spec to hide them from; a plain 404 is already indistinguishable).

Receipts

- jest: 194 suites / 2,615 tests passing, 68 new; `tsc` clean; no migration (the flags table already existed).
- Fails open on a database read error so a hiccup cannot black out Newsletters for every customer; the MCP key-issuing gate still fails closed.

4. Retry can no longer destroy a finished article

PR #96 · VP-270 (Urgent) · fix(articles): retry never destroys a completed article

`branch micah/vp-270-retry-… · head 3435749 · +481 / -8 · 5 files · CI: success`

The problem. Pressing Retry on a finished article deleted its sections and sources first and only then asked the background queue to rebuild it. If the queue was unreachable at that moment, the content was gone and the article sat on "Pending" forever. Two further holes: an article written by the new Article Engine was reset and handed to the old pipeline, which replaced its content with something different; and the SEO score from the old text stayed on the row.

What users see. Retry on an old-pipeline article works as before. Retry on an Article Engine article is refused with a plain message ("Retry only re-runs the classic pipeline… generate a new article to run the engine again") before anything is touched. The SEO tab no longer shows a grade for text that no longer exists.

TASKS IN THIS TICKET

- Confirmed the enqueue-before-delete ordering and restore-on-failure were already on staging; closed what was still open.
- A guard that recognises an engine-written article (engine version v2 with a document that was not converted from legacy) and answers 409; converted legacy articles still retry.
- A failed engine run with no document still retries and converts to the classic pipeline — nothing to lose, keeps a recovery path.
- Stale SEO score cleared on retry and restored, with every other reset field (18 in total), if the queue refuses.
- New shared `conflict()` (409) helper, matching the shape of the existing error helpers; all five client call sites already show the server's message in a toast.
- Engine doctrine §10 updated to name the route and test as the implementation.

Receipts

- 12-case test that runs the route's real rollback with only the queue send faked; the ordering case asserts `['enqueue','delete','delete']` literally; the 409 case asserts zero database writes.
- jest retry + enqueue + article-engine: 8 suites / 105 tests; full `app/api`: 32 suites / 247 tests; `tsc` clean.
- Independent review: no gaps in the 18-field restore, engine marker verified against the conversion code, verdict ship. Follow-up filed: VP-390 `(https://linear.app/deepstation/issue/VP-390)` (hide the Regenerate menu item for engine articles).

5. Silent generation progress made visible

PR #97 · VP-291 · fix(articles): make missing generation progress visible and loud

`branch micah/vp-291-… · head b0e400d · +664 / -38 · 9 files · CI: success`

The problem. After "Let's write it", the Write step polls for progress. When the background worker is not running, or its progress write fails, the article sits on Pending with no snapshot forever; the screen shows outline placeholders, a 0% bar and a ticking clock and says nothing. The only signal was a console warning nobody reads.

What users see. After 15 seconds with no progress: "No progress received yet — the background worker may not be running." If updates stop arriving mid-run for 150 seconds: "Still working — no update in *n* s." If the worker's own progress write is failing: "Progress updates aren't being saved — the run continues, but this screen can't follow it."

TASKS IN THIS TICKET

- A failed progress write now reports to Sentry (tagged with article id and phase) and leaves a note on the article, written with a guard so it can never overwrite a real error; a completed article clears the note.
- The Write step recognises the note and explains it.
- Stall detection measured on the browser clock from when each snapshot arrived, so a wrong server/client clock cannot trigger it; the 150 s threshold clears a single long section.
- The dead duplicate progress type was removed; the worker's seed snapshot and the test share one `buildSeedProgress()` so they cannot drift.
- Six-line "run the full engine locally" checklist added to the developer docs.

Receipts

- Two review rounds; the first asked for four changes (all made in a second commit: the note surfaced and cleared, threshold 60→150 s, client-clock measurement, the guard clause asserted literally in the test), the second: ship.
- jest vnext + inngest: 10 suites / 116 tests; `tsc` clean.
- Follow-up filed: VP-389 `(https://linear.app/deepstation/issue/VP-389)` — the v2 engine route never writes progress at all.

6. Scheduled generation never creates two articles

PR #98 · VP-268 · fix(scheduler): a retried schedule run never creates a second article

`branch micah/vp-268-… · head b6b3555 · +1023 / -82 · 3 files · CI: success`

The problem. The nightly job that turns planned topics into articles did three things inside one retryable unit: create the article row, link it to the planned topic, hand it to the background writer. If the hand-off failed, the retry redid everything — a second article for one topic, billed twice, with the first stranded on Pending forever and the org's free-first-article entitlement burned.

What users see. One planned topic produces exactly one article and one charge, even when the queue hiccups. A topic already being written by hand is skipped rather than written twice. A deleted planned topic no longer aborts the whole night's run for every other customer.

TASKS IN THIS TICKET

- Create and link are one database transaction with a compare-and-set on the pointer actually read; if the link matches nothing, the insert rolls back. No separate link step, so no window for a retry to insert twice.
- Reuse only a live Pending article; an article already generating or done is skipped as `already_in_flight`.
- Hand-off goes through the shared `enqueueArticleGeneration` helper (never throws); a failed hand-off deletes the empty row and returns the topic to `queued`, and the run summary counts it as failed, not created.
- A deleted item is recorded as `skipped: item_deleted` and the loop continues; transient errors still rethrow so the queue retries them.

Receipts

- 18 unit tests, including the compare-and-set predicate asserted literally for both the fresh and the "dangling pointer" case — mutation-checked: reverting the fix fails exactly one test — and a test that drives the real job handler with a fake step runner and asserts the exact step-id sequence.
- Three review rounds; the last found one bug on the heal path (the guard checked for a null pointer that was not null and would have aborted the whole run); fixed in the final commit.
- jest inngest + content-schedule + article-generation: 105 passing; `tsc` clean. Integration test written and gated (it writes to the shared production database, so it was not executed here).

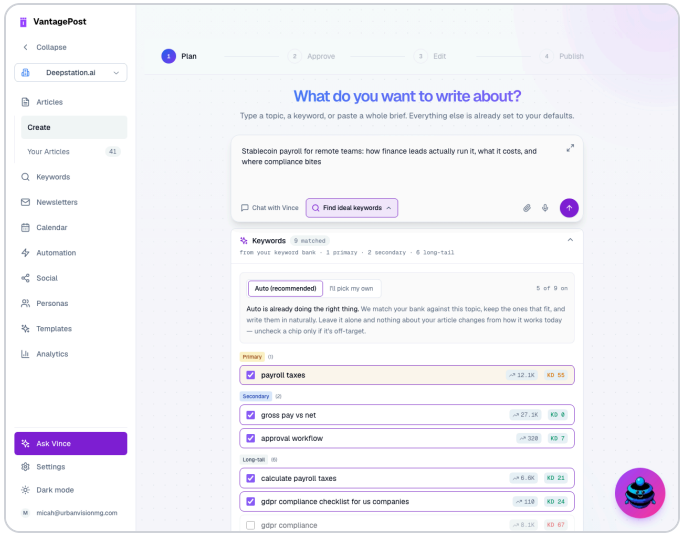
7. Choose keywords before writing — the tray on Create Article

PR #99 · VP-388 · feat(articles): choose keywords before writing — tray on Create Article, Vince can fill it

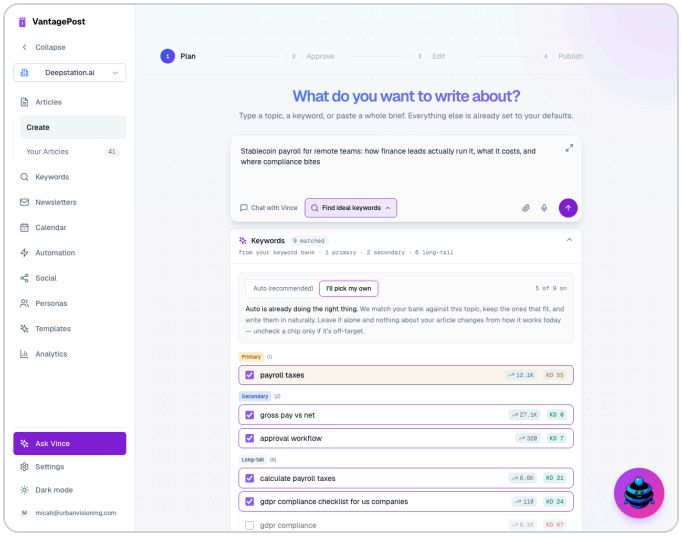
`branch micah/vp-388-… · head a5bf4c1 · +1894 / -34 · 14 files · CI: success`

The problem. The Create page sent no keyword signal at all: both outline calls hard-coded an empty keyword list and the create request never carried target keywords. The only way to steer an article's keywords was to type them into the topic as prose. The engine already accepted primary, secondary and long-tail keywords; nothing in the interface filled them.

What users see. A **Find ideal keywords** button next to "Chat with Vince". It opens a tray under the topic box showing keywords from the organisation's bank that match the topic, grouped Primary / Secondary / Long-tail with search volume and difficulty. *Auto (recommended)* keeps today's behaviour for anyone who never opens it; *I'll pick my own* unlocks the checkboxes. **Search for more keywords** runs a paid DataForSEO discovery, only when clicked. When Vince suggests keywords in chat, a "Vince suggested N" row shows them already selected, with Undo.



Real app, localhost, light mode: the tray after matching the bank against the topic — 9 keywords, Auto pre-selected 5.



"I'll pick my own" unlocks the checkboxes; volume and difficulty pills on every row.

TASKS IN THIS TICKET

- Tray component and grouped rows, lifted from the existing proof-of-concept at `/content/v2` rather than rebuilt; pure state helpers (`groupKeywords` , `toggleKeyword` , `applySuggestedKeywords` , `dropStaleMatches` , `withSinglePrimary` , `extraPrimaryNote`) so every rule is unit-tested.
- Picks flow into both outline calls and into the create body as `targetKeywords` and `recommendedKeywordIds` ; the v2-engine shape is built by the same rules and ready for the day the flow targets it.
- Vince integration: the copilot's `prefillArticleForm` tool gained a `keywords` field; the mascot passes it through; the page applies them into the tray.
- One primary per article, enforced in the payload and explained in the tray ("An article has one primary keyword. X will be the primary; the other will be sent as secondary") — without this the server silently stopped linking every keyword after a second primary.
- Bank matches from a previous topic are dropped when the topic changes; Vince's picks survive.

- Billing safety: discovery fires only from its button, guarded against double-click and remount, with the server's active-run check as a backstop; the bank match cannot fire per keystroke.
- Failure handling: if suggestions cannot load, the tray says so and the article can still be written.
- Accessibility: `aria-expanded` on the toggle, labelled checkboxes, visible focus ring, panel id only referenced while open. Light-mode theming checked (semantic tokens only).

Receipts

- Two review rounds; the first found the second-primary bug and the stale-match bug (fixed in commit a5bf4c1), the second: ship.
- jest blog-generator + copilot: 29 suites / 371 tests (54 new or extended, including an exact-object assertion that an untouched request is byte-for-byte today's); whole repository 2,583 passing; `tsc` clean.
- Browser-verified on localhost, signed in through the new dev login: typed a topic, opened the tray, bank match returned 9 keywords (1 primary, 2 secondary, 6 long-tail); the only console error on the page is the pre-existing Brief Settings hydration warning (VP-304). The match took ~80 s because the Anthropic key is capped until 1 October and the call fell back to GPT; the tray is unaffected by which provider answers.

8. Dev-only sign-in for localhost

PR #100 · VP-403 · feat(auth): dev-only auto sign-in for localhost

`branch micah/vp-403-… · head ed18417 · +423 / -28 · 8 files · CI: success`

The problem. Opening the local app meant the emailed one-time code every time. Every screenshot, demo rehearsal and automated check that needed a signed-in page went through an inbox. There was no bypass anywhere in the auth configuration.

What developers see. `http://localhost:3050/api/auth/dev-login` signs the local browser in as the profile named in `AUTH_DEV_AUTOLOGIN_EMAIL` and lands on the dashboard; the login page shows a small "Dev sign-in (local only)" link while the feature is on. **Customers see nothing**: every deploy path pins `NODE_ENV=production` at build time, so the route compiles to a 404 in the staging and production images regardless of what is in the environment file.

TASKS IN THIS TICKET

- Guard: only when `NODE_ENV ≠ 'production'` *and* the email variable is set; never creates a user (unknown email → 404), a disabled profile → 403.
- The session is a real database row written through the same adapter Auth.js uses, with the same cookie and 60-day expiry — indistinguishable from a Google or email sign-in. (A standard Credentials provider cannot do this here: the app uses database sessions and Credentials only supports JWT.)
- The email never reaches the browser; the login page receives a boolean.
- Documented in `.env.example` and the local-dev workarounds guide.

Receipts

- 7 route tests: production → 404 with the database never touched; env unset/blank → 404; unknown email → 404 and no insert; disabled → 403; happy path creates the session and sets the dev cookie name with a 307. Login and auth suites: 20 passing. `tsc` clean.
- Independent security review traced the Dockerfile, compose file and all three deploy workflows: verdict ship.
- Used live for every real-app screenshot in this report.

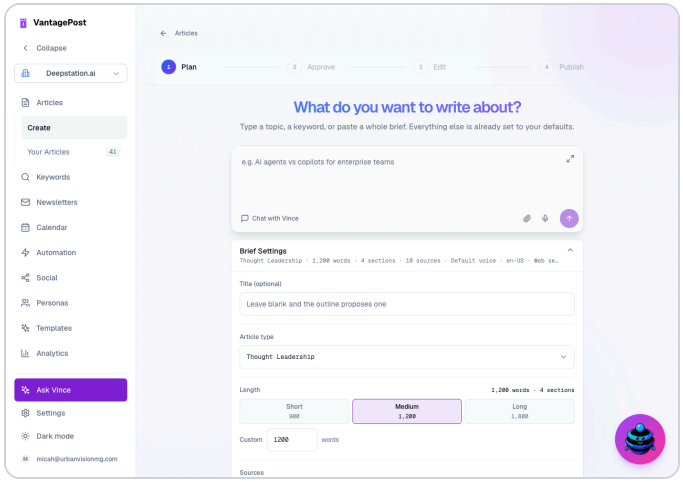
9. Create Article spacing and a real Back control

PR #101 · VP-294 + VP-242 · fix(articles): Create Article sits at the app's top offset with a real back control

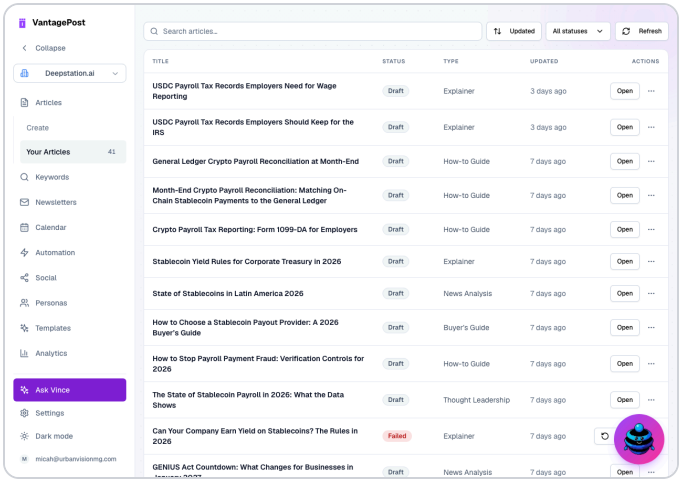
`branch micah/vp-294-… · head 913cbfc (merged with staging after PRs #102/#104 landed) · +406 / -34 · 8 files`

The problem. The Create page stacked four nested paddings before its heading, so it started ~80 px lower than the Articles tab in the same page. The step rail was pinned 65 px from the top for a header that no longer exists. And there was no way out of the writer except the sidebar.

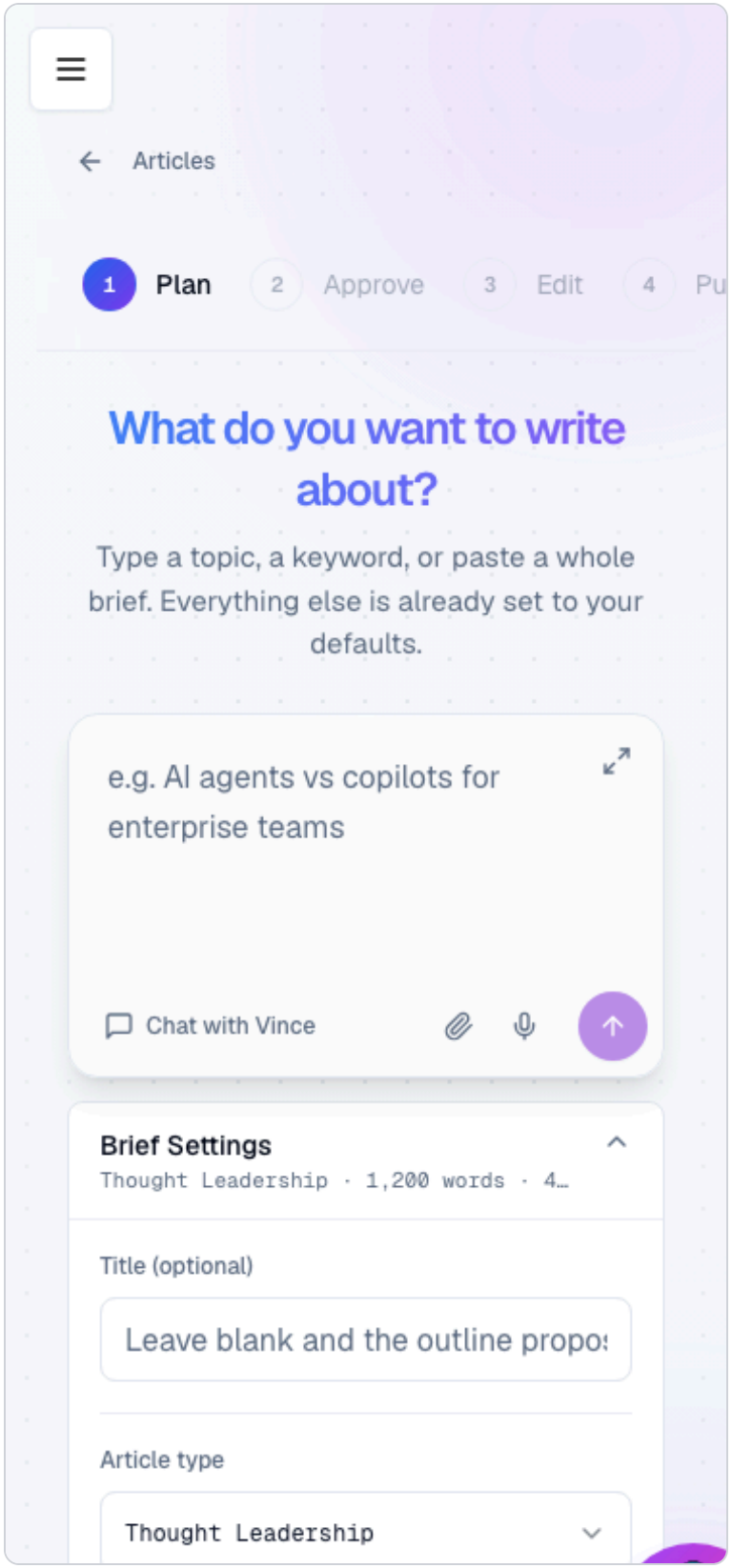
What users see. Create starts at exactly the height the Articles tab does. A top-left **← Articles** link on step one (a real link: cmd-click works) and **Start over** on later steps, disabled while an outline is streaming or an article is being created so it cannot strand a charged article. On phones the control and the rail clear the fixed menu button.



Create Article, localhost, 1280 px: "← Articles" sits where the Articles tab's first row sits.



The Articles tab at the same width — the comparison the spacing fix targets.



375 px: the back control and step rail clear the menu button (top-left).

TASKS IN THIS TICKET

- One top offset owned by the page; the writer root, the rail and all four step bodies lost their own top padding.
- Stale `sticky top-[65px]` and its comment removed; rail at `top-0` on desktop, `top-14` on phones.

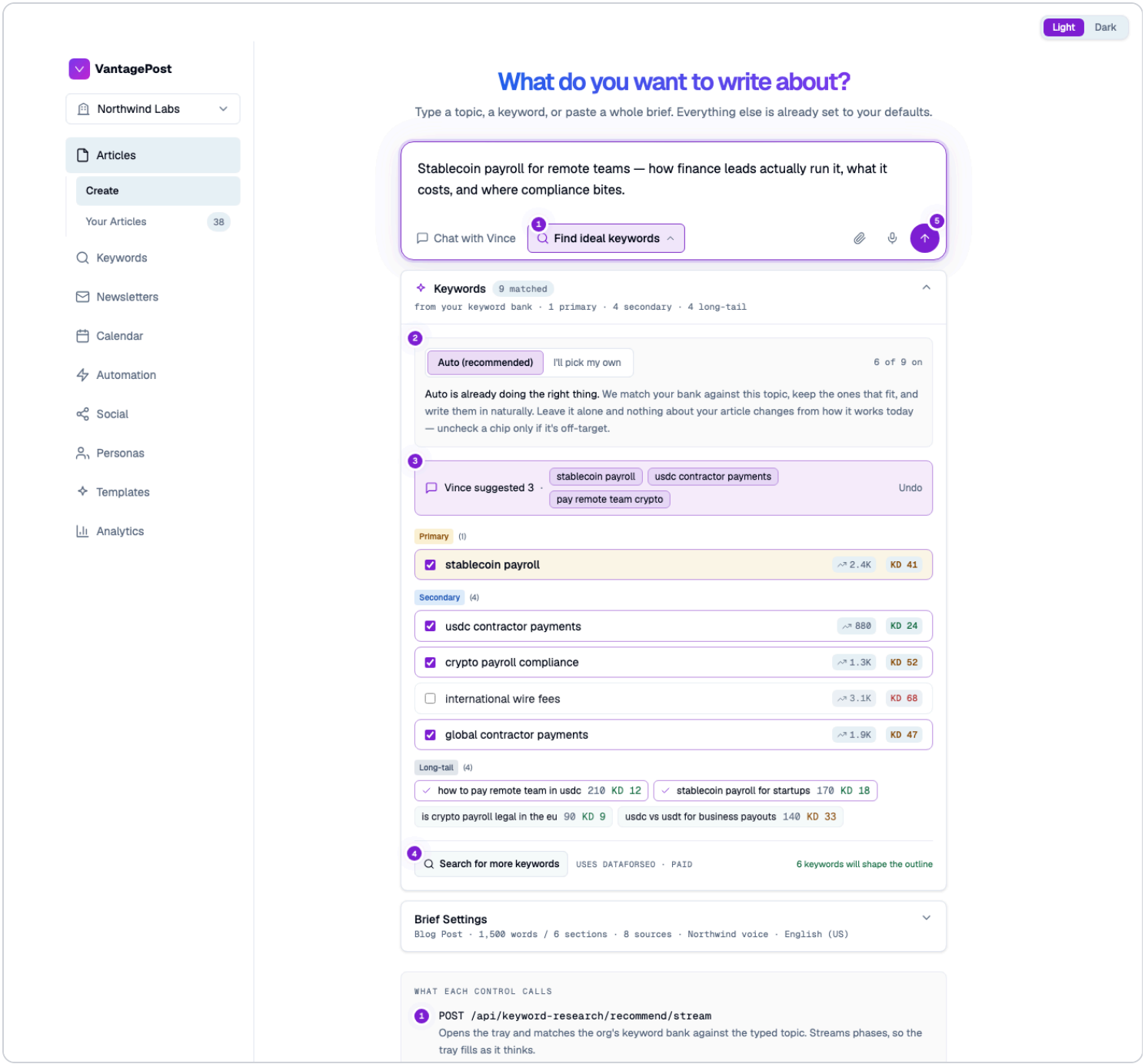
- New `StepBackControl` (a breadcrumb `<nav>`, first in tab order, 44 px touch target, tokens only so light and dark both follow); the old "Back to start options" link removed from the hidden chrome row.
- VP-242 (Start Blank buttons too big) verified already fixed by an earlier commit: tiles are below the app's standard button scale at every breakpoint. No change needed; ticket closable.
- After staging moved: merged with Grant's composer-width change (#102), keeping his shared column-width constant and our zero top padding; one of his new string-match tests updated to read the merged class.

Receipts

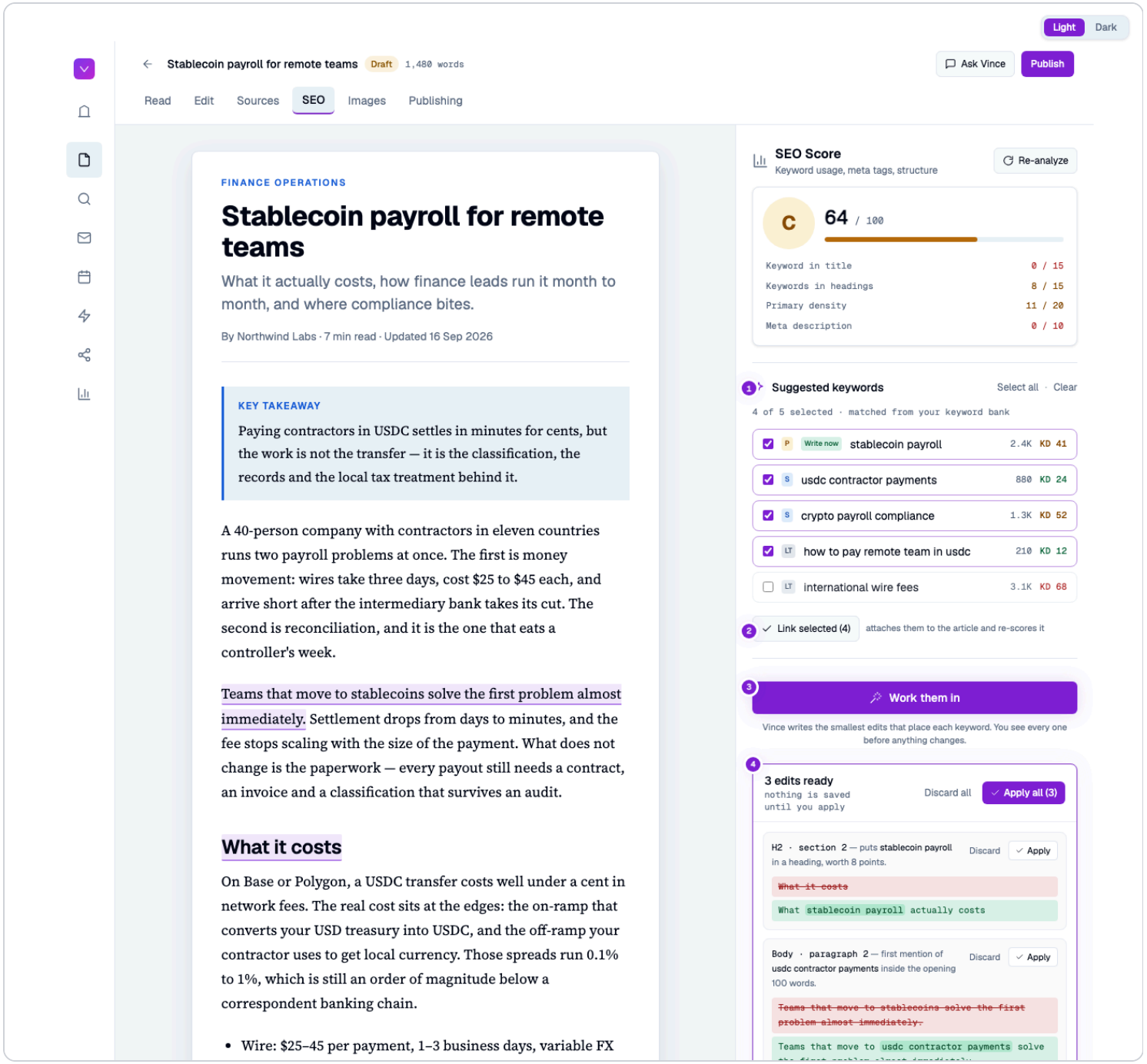
- 32-case test for the back control (link on step 1 with the right href, button on 2–4, disabled while creating, the disabled link refuses to activate). jest blog-generator after the merge: 27 suites / 293 tests; `tsc` clean.
- Two review rounds; the first found the mid-create race and the mobile overlap (fixed), the second: ship. Screenshots above taken at 1280 px and 375 px in light mode.

10. The two keyword mockups

Built from the app's own design tokens (the light palette in `globals.css`, the Geist typefaces, the article-paper styles) so they read as the product, not a wireframe. Each carries numbered callouts that map every control to the existing backend it reuses. Light mode by default with a Light/Dark toggle; interactive version at claude.ai/artifact/AmXVPnfz5BbUkmwViV8aui (`https://claude.ai/artifact/AmXVPnfz5BbUkmwViV8aui`).



Option A — before writing. "Find ideal keywords" next to Chat with Vince; the tray with Auto / I'll pick my own, bank matches grouped by role with volume and difficulty, Vince's suggestions, and "Search for more keywords" (DataForSEO, paid). This is what PR #99 built.



Option B — after writing. The finished article's SEO tab: score, suggested keywords, "Work them in", and a preview of the three small edits with Apply / Discard. Mostly exists today; VP-409 covers the "Link selected" action that failed live.

11. Tickets filed and reassigned

Every ticket carries the current state of the code with file references, what a user experiences, the fix, and acceptance criteria. Duplicate checks were run against the existing backlog before each one.

TICKET	WHAT IT IS	WHY IT EXISTS
VP-385 (https://linear.app/deepstation/issue/VP-385)	Published article keeps its old URL after a rename; an edited slug breaks the old link	The share-link investigation: share links are random tokens and cannot break; the published slug is set once and never regenerated, and a slug edit 404s the old URL with no redirect
VP-386 (https://linear.app/deepstation/issue/VP-386)	Local articles never reached staging: the content copy only ran as a dry run	Root cause of the empty staging account, with the exact steps to run it and the profile-email check to do first
VP-387 (https://linear.app/deepstation/issue/VP-387)	One standard way for agents to drive VantagePost	The MCP research memo: what the tool registry does, what the Bitwage branch adds, WebMCP, options A/B/C, recommendation B
VP-388 (https://linear.app/deepstation/issue/VP-388)	Choose keywords before writing, work them in afterwards	The keyword feature with architecture map and acceptance; carries the mockups and PR #99
VP-389 (https://linear.app/deepstation/issue/VP-389)	Engine v2 articles never report progress	Found while fixing VP-291: the v2 route never passes a progress callback
VP-390 (https://linear.app/deepstation/issue/VP-390)	Hide "Regenerate Article" for engine articles	Follow-up to VP-270: the action is now safely refused, but should not be offered
VP-391 (https://linear.app/deepstation/issue/VP-391)	Classic article API accepts any article type	Found reviewing PR #89: unknown types are stored and written as Thought Leadership
VP-403 (https://linear.app/deepstation/issue/VP-403)	Dev-only auto sign-in for localhost	PR #100

TICKET	WHAT IT IS	WHY IT EXISTS
VP-408 (https://linear.app/deepstation/issue/VP-408)	Check v2 writes its own title when the title is blank	Raised in the standup; not previously tracked
VP-409 (https://linear.app/deepstation/issue/VP-409)	"Link selected" on the SEO tab does nothing	Failed live in the standup; the after-writing half of the keyword plan
VP-410 (https://linear.app/deepstation/issue/VP-410)	Grade sources for quality and recency	From the 1 September sync; the fact-check gate only checks a link resolves
VP-411 (https://linear.app/deepstation/issue/VP-411)	Put the generated JSON-LD onto the published page	From the 1 September sync; structured data is generated but never rendered
VP-412 (https://linear.app/deepstation/issue/VP-412)	Move the article engine into services + repositories	From the 1 September sync; agreed to happen before the engine grows further
VP-413 (https://linear.app/deepstation/issue/VP-413)	Log-level switch for Engine v2 tracing	From the 1 September sync; verbose tracing needs turning down before launch
VP-286 (https://linear.app/deepstation/issue/VP-286)	Bill v2 articles once and refund failures — **reassigned to Grant**	Per the standup, with the agreed rules and the transaction-history placement question added
VP-366 (https://linear.app/deepstation/issue/VP-366)	Feature flags — **updated**	Three-layer design and the endpoint-hiding answer

Notes were also added to existing tickets where the sync had widened their scope: VP-293 (the live progress detail that VP-399 described), VP-305 (labelled export sections), VP-285 (Bitwage articles with executive summary and FAQ), VP-244 (config-driven model ladder and a cheap dev mode).

12. Items reconciled from the 1 September sync

The earlier sync was re-read in full and checked against every ticket created since. Most of it was already tracked (VP-286–305 were filed from that call on 9 September; Engine v2 is VP-283; article types are VP-240). Four items had no ticket and now do — VP-410, VP-411, VP-412, VP-413 — and four more were folded into the tickets that already cover them (VP-293, VP-305, VP-285, VP-244). Three items remain personal follow-ups outside Linear: deliver the pipeline atlas as HTML + PDF to the shared drive, a section-by-section engineering walkthrough of v2, and the deferred idea of letting an agent select prompt blocks per article.

13. How every item was verified

- **Type check and tests on every branch.** `bunx tsc --noEmit` clean and the relevant jest suites green before any push; whole-repository runs where the change was cross-cutting (2,615 on #90, 2,583 on #99).
- **Independent review before every pull request.** Each branch was reviewed by a separate reviewer reading the commit directly (not the working tree); every "fix first" verdict was addressed in a follow-up commit and re-reviewed. Eighteen review passes in total; the findings that changed code are listed under each PR above.
- **Browser verification for the user-facing changes.** The keyword tray and the Create Article changes were exercised on the local dev server, signed in through the new dev login, at 1280 px and 375 px in light mode; the screenshots in this report are those runs, not mockups.
- **Continuous integration.** "Build & Type Check" green on #89, #90, #96, #97, #98, #99 and #100; re-running on #101 after its merge with staging.
- **Ticket hygiene.** Duplicate searches before every new ticket; no tickets created for items Grant already owns; one duplicate Grant filed and cancelled (VP-399) had scope the others did not cover, which was preserved on VP-293.

VantagePost engineering · living document, last updated 16 September 2026, 6:10 PM EDT · PDF version · this page is unlisted and expires automatically.