Product Requirements Document

Kural — the AI email assistant that sounds like you

Field	Detail
Product	Kural — AI email composition assistant
Author / Owner	George Innasi (C51)
Date	July 15, 2026
Version / Status	v1.2 — v1 shipped and live; v1.1 requirements in review; segments, P0 and metrics revised after structured review
Live prototype	https://kural-iota.vercel.app

1. Overview

Kural helps people turn messy, half-formed thoughts into structured, well-toned emails — without taking authorship away from them. The user types the gist (or pastes an email they received); Kural classifies the situation (urgency, power dynamic, emotional intent), composes a structured draft in the user's calibrated voice, and lets the user iterate with tone dials until the draft sounds like them. Kural never sends anything: the user confirms, copies, and sends.

Kural is deliberately an assistant, not an autopilot. The core bet is that people don't want email automated — they want the composition burden removed while their voice, judgment, and signature stay honest.

2. Problem statement

2.1 P0

Even professionals who already use AI email tools, when drafting high-stakes emails to customers and managers, end up rewriting the drafts they're given — because the tools polish sentences but don't read people: one voice, one register, every time. Evidence: 71% of AI users surveyed say they can't send AI drafts as-is (“it sounds like a robot and lacks my voice”), and 22% have quietly abandoned the tools.

Root-cause chain: drafts are robotic, wrong register, too long → the model never knows the reader, the stakes, or the sender's voice; that context must be prompt-crafted every time → root cause: generic assistants hold no persistent model of the sender's voice and no structured read of the situation. Context lives in the user's head — and dies there. Root-cause test: when the v2 engine added classification and profile injection (and nothing else changed), drafts began passing end-to-end tests without edits.

2.2 P1, P2 — and why P0 wins

Problem	Statement	Evidence
P1 — Understood first time	Senders of dense, multi-stakeholder updates struggle to be understood on the first email because content is long and unstructured	55% get follow-ups because the first email wasn't understood; 33% lose most time gathering info
P2 — Getting started	Occasional senders struggle to begin high-stakes emails because of blank-page dread	8% say starting is the hardest part

Why P0 beats P1 and P2: it is the #1 complaint at 71% — six times the runner-up — and it is the reason 22% already churned from AI tools. Fixing structure or blank-page dread is worthless if users won't send what the tool writes. P1 is additionally part-external (missing information lives in threads and systems, outside the drafting step). Scope check: situation-aware drafting in the user's voice is a quarter of focused product work — not a continent, not a pebble.

2.3 Supporting evidence

Signal	Data point	Source
Email is a major time sink	28% of the knowledge-worker workweek goes to reading and answering email	McKinsey Global Institute
Robotic copy destroys trust	62% of consumers are less likely to trust copy that sounds like a bot; 52% stop reading when text feels AI-generated	Bynder, AI vs. human content study
AI is welcome — if it feels human	82% don't mind AI-written copy if it feels like a human wrote it	Bynder
Tone is the #1 pain	42% say the hardest part of an important email is getting the tone right	Kural survey, n=36
Real time is lost	75% spend 5+ minutes per important email; 28% spend 15–60 minutes	Kural survey, n=36

Research method: online survey "How You Handle Everyday Emails" (n = 36; office/admin staff 31%, students 28%, finance professionals 22%) plus 4 follow-up user interviews. 10 respondents opted into further conversations and form the recruiting pool for deeper interviews.

3. Goals and non-goals

3.1 Goals

- Cut composition time for important emails from 5–15+ minutes to under 60 seconds of user effort.
- Produce drafts users send with minimal or no edits, in a voice they recognize as their own.
- Read the situation automatically — urgency, power dynamic, emotional category — instead of relying on prompt-crafting skill.
- Keep the user the author: Kural composes and structures; the user confirms and sends.
- Never invent facts: when the input is too thin, ask targeted questions instead of hallucinating.

3.2 Non-goals

- Not an email client: Kural does not connect to inboxes, send mail, or manage threads (v1–v1.1).
- Not full automation: no auto-replies, no unattended sending, ever.
- Not long-form writing: Kural targets everyday professional/academic email, not essays, newsletters, or marketing copy.

4. User segments (behavioral)

Segments are defined by email occasion and behavior — when, why, and how the email happens — not by identity. One person moves between segments across a day but never occupies two in the same draft. Each segment carries a distinct top problem and real survey respondents behind it.

Segment	When / behavior	Top problem	Who's behind it
The upward one-shot	Writing to someone with power over the outcome (manager, executive, professor). Rare but dreaded: 15–30 minutes on one email, re-read five times, second-guessed at send.	Tone — accountable without groveling, brief without being blunt	Survey verbatims: the exam-rescheduler; the professor-report author; “I'd be confident when hitting send.” 42% tone stat
The volume replier	After every customer/client interaction: follow-ups, confirmations, check-ins. 20+ emails a week, minutes each, templates feel canned.	Voice at scale — personally-written sound without personally-writing time	33% write 21+ from-scratch emails weekly (3 write 100+); 8 finance/banking respondents; the author's own 25–30-customer week
The complex explainer	Status updates and multi-stakeholder threads. Gathers information first (33%), writes long, still gets follow-up questions.	Structure — understood the first time	Verbatims: “long and unstructured” stakeholder threads; “improve my clarity… cluttered”; “to structure better.” 55% follow-up stat

4.1 Representative user stories

- **Upward one-shot:** as a banker running behind on a report, I want to turn “audit data came late, ask Shawn for 9am tomorrow” into a proactive, accountable extension request, so that I flag the delay before the deadline without groveling.
- **Volume replier:** as a personal banker after 25–30 weekly meetings, I want every follow-up to sound personally written in minutes, so that customer relationships stay warm without consuming my evenings.
- **Complex explainer:** as a coordinator updating managers and stakeholders, I want drafts that carry every key point in a scannable structure, so that I stop getting follow-up questions the first email should have answered.

5. Functional requirements — v1 (shipped, live)

ID	Requirement	Priority	Status
FR-1	Compose mode: free-text gist → structured draft (greeting, purpose, facts, next step, sign-off), using every user-provided fact and inventing none	P0	Shipped
FR-2	Reply mode: paste a received email → contextual reply that identifies the real sender name from the signature (ignoring URLs, phones, sign-off phrases) and addresses every question or request in the email	P0	Shipped
FR-3	Ask-or-draft gate: when the gist lacks substance, return 2–3 targeted questions instead of a hollow draft; user answers regenerate the draft	P0	Shipped
FR-4	Situational classification engine: every request is classified before drafting — key points, urgency (High/Normal), power dynamic (Upward/Peer/Downward/Unknown), emotional category (Accountability, Bad news, Request, Cold outreach, Follow-up, Gratitude, Routine) — in a single model call via a schema that orders classification fields before the draft	P0	Shipped
FR-5	Situational rules with precedence: classification activates tone rules (e.g., upward request → proactive, deferential, fix stated early); the user's explicit preferences (greeting, sign-off, dials, never-say list) always override situational rules	P0	Shipped
FR-6	Anti-robotic guard: first sentence after the greeting must state the purpose; well-wish openers (“I hope you're doing well” and all variants) and AI clichés are banned in the prompt and scrubbed server-side as a backstop	P0	Shipped
FR-7	Style dials: length, warmth, formality (3 positions each) restyle or regenerate the draft	P0	Shipped
FR-8	Personas and calibration: persona selection (Student, Working Professional, Other) plus a scenario-quiz calibration wizard capturing greeting, sign-off, and pet peeves	P1	Shipped
FR-9	Detected-context chips: the classification (e.g., “Urgent · Writing upward · Request”) is displayed above the draft so users see how Kural read the situation (display-only in v1)	P1	Shipped
FR-10	Style memory: shadow-diff of user edits saved as substitution rules and applied to future drafts; advanced Voice EQ faders	P1	Shipped
FR-11	Output integrity: schema-enforced JSON responses; malformed drafts rejected with a retry prompt rather than shown; oversized subjects truncated	P1	Shipped

6. Functional requirements — v1.1 (next release)

ID	Requirement	Priority	Acceptance criteria
FR-12	Chip correction: tapping a detected-context chip lets the user override the classification and regenerates the draft	P0	Corrected chip changes the draft's tone accordingly in one regeneration
FR-13	Placeholder elimination: when the recipient's name is unknown, greet by role — bracketed placeholders must never appear (intermittent defect found in E2E testing)	P0	0 bracketed placeholders across a 50-draft regression suite
FR-14	Register matching: replies mirror the vocabulary level and sentence complexity of the received email (e.g., a customer writing simple, everyday English receives a reply at their register)	P0	Blind test: replies to simple-register and formal-register versions of the same email differ measurably in reading level
FR-15	Anonymous event telemetry: draft_generated, regenerated, dial_changed, needs_info_shown, copied, edit-distance bucket — counts and buckets only, never email content	P0	All goal/guardrail queries in §9 computable from events; no message text stored
FR-16	Few-shot voice examples per emotional category to further reduce robotic phrasing	P1	Blind A/B: v1.1 drafts preferred for “sounds human” by ≥60% of testers
FR-17	Feedback loop: thumbs up/down per draft with optional reason, logged with the classification for prompt tuning	P1	Feedback stored and reviewable; weekly tuning cycle documented
FR-18	Visible, editable style-memory rulebook: users can view, edit, and delete learned substitutions	P1	All learned rules listed; deletion takes effect on next draft

v2 (beyond this PRD's commitment): learning from the user's past sent emails (today's upload control is a placeholder and is not claimed as a feature); Chrome extension; native Gmail/Outlook integration.

7. Non-functional requirements

Category	Requirement
Latency	Single model call per draft (classification and generation share one call). P95 draft time ≤ 8s; silent retry with 25s timeout hides serverless cold starts.
Privacy & security	Gemini API key lives server-side only (Vercel env var), never in the browser. Email content is processed per-request and not stored server-side; user profile and style memory live in the user's browser. No user data is used to train models. Telemetry (FR-15) is counts and buckets only.
Reliability	Graceful degradation: malformed model output falls back or asks the user to retry; needs_info honored only when questions exist; empty drafts never shown.
Cost	Gemini 2.5 Flash for all calls; single-call architecture keeps unit cost at one generation per draft.
Usability	No prompt-writing skill required at any point; every control is a click or a slider. Calibration is one question per screen.

8. Technical architecture (summary)

- **Frontend:** single self-contained page (kural_website.html) — landing, calibration wizard, workspace (compose/reply), style dials, detected-context chips.
- **Backend:** one Vercel serverless function (api/generate.js) proxying Google Gemini 2.5 Flash; the API key never reaches the client.
- **Classify-first engine:** the response schema forces a `situation` object (key_points, urgency, power_dynamic, emotional_category) to be generated before the draft fields — the model reads and classifies before it writes, at zero added cost. This is the root-cause fix; the cliché scrubber and dials are the safety net.
- **Guards:** prompt-level cliché bans plus a server-side regex scrub for well-wish variants; JSON schema enforcement; subject/body integrity checks.
- **Deploy:** GitHub main → Vercel auto-deploy; live at kural-iota.vercel.app.

9. Goal, guardrails and metrics

Goal: raise the send-as-is rate among activated users — anyone who completed calibration and generates at least one draft per week — to 60% or higher within one quarter of v1.1 launch.

Send-as-is defined: a draft counts as sent-as-is when the user copies it and the edit distance between generated body and copied body is under 10% of characters. Every metric below is computable from FR-15 events alone; no goal contains solution language.

Type	Metric	Definition / threshold
Goal	Send-as-is rate	≥ 60% of copied drafts with edit distance < 10%, among activated users, within one quarter of v1.1
Guardrail	Regenerations per accepted draft	≤ 2 — blocks winning send-as-is by making users spin dials
Guardrail	Ask-gate rate	≤ 25% of requests — blocks gaming the metric by refusing to draft
Guardrail	Latency & cost	P95 ≤ 8s; cost stays one model call per draft
Secondary (tracked)	Calibration completion rate	Funnel visibility: visitors who start vs. finish the wizard
Secondary (tracked)	4-week retention of activated users	Context: 22% of surveyed users abandoned existing AI tools

Instrumentation honesty: none of these numbers is measurable today — privacy-by-design means zero telemetry exists. The queries are written before the instrumentation (FR-15); instrumentation is the next commit.

10. Risks and mitigations

Risk	Impact	Mitigation
Misclassification of the situation (e.g., peer read as downward)	Wrong tone — worse than a neutral draft	Chips expose the classification; v1.1 chip correction (FR-12); preferences always override rules
Model drift / provider changes	Tone quality regresses silently	Versioned prompts; regression suite of golden scenarios run before deploys; feedback loop (FR-17)
Free-tier platform limits (Vercel/Gemini)	Outages or throttling as usage grows	Single-call design minimizes quota; upgrade path budgeted before extension launch
Users paste sensitive content	Privacy exposure	No server-side storage; processing is per-request; privacy stance stated in-product
Low willingness to pay (2% ever paid for an email tool)	Monetization risk	Free core; prove the send-as-is value hypothesis first; monetize later via extension convenience and team features

11. Release plan

Milestone	Scope	Timing
v1 (shipped)	FR-1 … FR-11 live at kural-iota.vercel.app; E2E tested (8 scenarios, 7 clean passes; FR-13 defect logged)	July 2026
v1.1	FR-12 … FR-15 (P0), FR-16 … FR-18 (P1), regression suite	August–September 2026
v1.2	Analytics review against §9 goal; scenario selector; smart placeholders	October 2026
v2	Sent-email learning (real); Chrome extension (compose-anywhere)	Q4 2026
v3	Native Gmail / Outlook integration	2027

Appendix — research artifacts: survey "How You Handle Everyday Emails" (Jotform, n=36); 4 user interviews (10 further respondents opted into follow-ups); market sources: McKinsey Global Institute (email time), Bynder AI vs. human content study (trust in AI copy), Zapier / Gmelius 2026 AI email assistant comparisons; pricing: superhuman.com/plans, tryellie.com/pricing, workspace.google.com/pricing (July 2026).