Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

OpenConsult, an Open-Source Fully Local AI Scribe With Live Decision Support: Development and Formative Evaluation

H M W I Herath

HCA Healthcare UK https://orcid.org/0009-0008-6367-1588

Henry W W Potts

h.potts@ucl.ac.uk

University College London https://orcid.org/0000-0002-6200-8804

Research Article

Keywords: OpenConsult, ambient AI scribe, clinical decision support, generative artificial intelligence, local-first computing, general practice

Posted Date: November 18th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11771422/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare potential competing interests as follows: The first author is the developer and, to date, the principal evaluator of the system described; this self-built and self-evaluated status is declared plainly and treated as a limitation in the Discussion. The first author is the sole copyright holder of the AGPL-licensed codebase and retains the possibility of dual-licensing it commercially; no such arrangement exists or is under negotiation. No other financial or commercial conflicts are declared.

OpenConsult, an Open-Source Fully Local AI Scribe With Live Decision Support: Development and Formative Evaluation

Authors: Herath HMWI[1], Potts HWW[2]*

[1] HCA Healthcare UK, New Broad Street Medical Centre, London, UK
ORCID 0009-0008-6367-1588
[2] UCL Institute of Health Informatics, 222 Euston Road, London NW1 2DA, UK
ORCID 0000-0002-6200-8804

* Corresponding author: Henry W W Potts, h.potts@ucl.ac.uk

Abstract

Background. Ambient clinical AI — software that listens to a consultation and drafts the clinical note, often called an AI scribe — is now a flagship use of artificial intelligence in healthcare. Most tools delivering it are proprietary and cloud-based: consultation audio leaves the building, errors cannot be independently characterised, and the clinician who signs the note answers for output nobody outside the vendor can examine. Open, fully local alternatives that a clinician or researcher can inspect, run and modify remain rare.

Objective. This paper reports OpenConsult, an open-source AI scribe with live clinical decision support that runs entirely on one machine in the consulting room and is inspectable end to end, and describes its formative evaluation.

Methods. Evaluation was formative and used synthetic consultations only (scripted and acted). Automated tests ran fifteen scripted consultations with pre-written marking schemes, checking transcription quality gates, the urgency alarm's behaviour across emergency and routine scripts, and whether every cited claim in a note points to a real transcript line. Five acted consultations followed in a real consulting room, and the AI assistants that built the system ran an adversarial security audit under the developer's direction. The work is reported against the applicable DECIDE-AI items, one stage before live clinical evaluation.

Results. OpenConsult is a fully local, open-source consultation assistant, built by a practising general practitioner directing AI coding assistants, running on a single consumer graphics card. It transcribes the consultation live and separates the speakers, drafts a structured clinical note with every sentence linked to its supporting transcript words, runs live decision support whose urgency alarm is shown only to the doctor, drafts referral letters, and offers a supervised spoken history-taking mode. In generated notes, 301 of 302 citations pointed to real transcript lines — an existence check, with clinical faithfulness unmeasured pending independent rating; the one uncited claim was flagged for manual checking. The urgency alarm fired in all eight scripted emergencies and stayed silent in six of seven routine consultations. Five acted consultations surfaced five defects the automated checks had missed, four of them serious; all were fixed, each pinned by a regression test.

The security audit returned twelve findings, including one critical vulnerability, fixed and verified before any public exposure.

Conclusions. OpenConsult is a working, auditable proof of concept — not yet a validated clinical tool — released for research and education only, with real patient care out of scope. It is intended as an open research instrument: it records whether the reviewing clinician opened the cited evidence before approving, how long review takes, and every revision of its decision support, making human review of ambient clinical AI measurable. Independent, prospective evaluation is required before any step toward care.

Keywords: OpenConsult; ambient AI scribe; clinical decision support; generative artificial intelligence; local-first computing; general practice.

Introduction

Background

Clinical documentation consumes a large share of a doctor's working day, and much of it is repetitive [1-2]. Over the past few years, ambient artificial intelligence — software that listens to a consultation and drafts the clinical note, commonly called an AI scribe — has moved from novelty to mainstream: in the United Kingdom it is now a flagship use case for national digital-health policy, with its own NHS guidance [3], and early reports associate it with meaningful reductions in documentation burden [4]. For a profession that spends more of its day with the record than it would like, the appeal needs no explanation.

The commercial products currently delivering AI scribing deliver appropriate performance: they are polished, supported, and straightforward for a practice to adopt. Many are now registered as medical devices. But nearly all of them share the same business model. They are proprietary and cloud-based, so consultation audio — among the most sensitive raw material a practice produces — leaves the building, bringing a substantial information-governance burden to every adopting organisation [5]. They are licensed per clinician, so cost scales with use. They are closed to inspection: the detail of what is going on "under the hood" is hidden [6]. When the software errs, neither the clinician, nor the practice, nor an independent researcher can easily examine why. Fully characterising a product's errors is possible only for its vendor, and shared benchmarks that would let products be compared are only beginning to emerge [7-9].

For the clinician, this model unhelpfully interacts with current decisions on where responsibilities lie. Professional responsibility for the submitted record stays with the clinician who signs it. Under the current UK regulatory position, ambient scribes (that do not offer more advanced functionality as well) sit outside medical-device regulation precisely because a clinician reviews their output and takes responsibility for that output, before it enters care [10]. Yet review under genuine time pressure is imperfect: a transcription error in a confident sentence (for example, a drug name, a dose, an 8.4 that becomes an 84) can survive a tired skim and enter the record as the clinician's own words. When that happens, the responsibility is the clinician's, while the ability to examine, correct, or even see the mechanism that failed belongs to the vendor alone. Early evidence suggests that clinician may be blamed whichever way they lean: in a mock-juror study of

AI-assisted radiology, disagreeing with a correct AI raised the likelihood of a finding against the doctor, and agreeing with an incorrect one did not lower it [11]. Davies has described such arrangements as accountability sinks: structures that absorb blame without preventing error [12]. Wherever the debate around AI and legal responsibility lands, a closed product represents a barrier to informed discussion.

Commercial AI scribes also carry a significant cost. How much the NHS is paying in total is not known, but a recent freedom of information response from one acute trust may be typical: an enterprise-wide licence at approximately £200,000 per year on a two-year contract, with around 250 clinicians trained and around 50 using it at the time of the response [13].

Open, local, inspectable implementations that clinicians and researchers can examine, modify and study present an attractive alternative. The value of open models has been demonstrated previously: OpenSAFELY showed that an open, auditable platform can become shared research infrastructure rather than a single product [14]. In the field of generative AI, but outside health, there is increasing interest in the possible advantages of running models locally in terms of privacy, data sovereignty and cost [6,15]. Ambient clinical AI does have a small ecosystem of open, locally run scribes — several, tellingly, built by individual clinicians [16-18] — but they stop at the note: none integrates live decision support, enforces a verifiable link between note and transcript, or has published how it was evaluated. Moreover, what was out of reach for a lone clinician-builder five years ago no longer is: openly downloadable, medically tuned model weights, MedGemma in this case [19], now put clinically capable language models within reach of a single consumer graphics card.

Objective

This paper reports OpenConsult, an open-source consultation assistant built to occupy that gap, and its formative evaluation. It was built and evaluated in the context of general practice, although nothing in its architecture is specific to general practice. The current build is intended for use in research and education, and is made available for further development.

Methods

OpenConsult

What it is, where it is, and what it may be used for

OpenConsult is a fully local, open-source consultation assistant: an AI scribe with live clinical decision support that listens to a consultation, drafts the structured note, and offers the doctor a running differential, suggested questions and examinations, and an urgency alarm. The source code is public under the AGPL-3.0 licence and carries a citable archived version [20].

It is designed to run on one computer in the consulting room. It can run on a single consumer workstation with one graphics card (24 GB of memory; an NVIDIA RTX 4090 in

the development machine) under Linux. This is hardware in the class of a high-end gaming PC, bought once, with no cloud service and no per-clinician licence. The current build handles one live consultation per machine. No third party processes the audio, the transcript, or anything generated from them. In the demonstration configuration, the browser may connect from another of the operator's own devices over an encrypted private overlay network (similar to a VPN), and rendered text then crosses the machine boundary to the operator's own screen and nowhere else. Because nothing leaves the machine, the data-transfer questions that dominate cloud-scribe governance do not arise; assessing overall compliance for any deployment remains, as always, the deployer's responsibility.

Transcription uses a fast Whisper implementation with a diarisation (speaker-separation) pipeline built on WhisperX and pyannote [21-23]; clinical reasoning, notes, decision support and letters come from a locally hosted medical language model (MedGemma 27B, served through Ollama) [19,24]; guideline retrieval uses a local embedding model with a PostgreSQL vector store [25]; spoken output uses a local speech engine run as an isolated subprocess.

OpenConsult is a research and education prototype, and it is deliberately not a polished product. It is not for use in real patient care. It has not been evaluated for clinical use and is not registered as a medical device. (Under the current UK position, transcription and summarising for clinician review sit outside medical-device regulation, but decision support used in care would sit inside it [10].) OpenConsult is offered as an open source tool for research and training purposes, and a stepping stone towards further development.

A consultation with OpenConsult

The system listens from the start of the consultation, transcribing live and separating the speakers, with the doctor free to keep eye contact and glance at the screen when it suits them. The intended opening, in the developer's own script, is spoken consent: "My computer will listen to our discussion and will help me write notes. Are you okay with that?" Where the audio leaves a clinically load-bearing value uncertain — a drug name, a dose, a number, a side (left or right) — the eventual note must carry a flag saying so; and where a stretch of the recording never became usable text, the system refuses to draft from it rather than guessing, and says what it excluded.

As the history unfolds, a panel updates with a working differential, questions worth asking, and examinations worth performing. These exist to widen the doctor's thinking, not to direct it: the design assumes a pragmatist, not a perfectionist, and the lists are suggestions to be raided, not protocols meant to be completed. Separately from that panel — and deliberately so — runs the urgency check. It is computed as its own memory-less safety pass over the transcript, so a reassuring overall assessment cannot suppress an alarm, and an earlier "no red flag" cannot carry forward to silence a new one. That separation guarantees non-suppression; it says nothing about the alarm's sensitivity, which is an empirical property to be measured, not a structural one. The alarm is shown to the doctor alone, never to the patient: the doctor has not yet decided what, or whether, to tell them.

When the patient has left, the system drafts the structured note. Every sentence of the draft is tied to the transcript words that support it: a claim that cannot be tied is omitted; a section with nothing to support it stays empty; a draft that fails a required check is refused, not delivered degraded. The tie is checked automatically as the note is produced: a mandatory gate that fails closed (if the check fails, no note is delivered), not an optional review. The note then waits: nothing becomes part of the record until the doctor has read it, edited it as needed, and approved it, after which it is read-only. The note is the doctor's act, and the system is built so it cannot be anyone else's. Around that act, OpenConsult keeps an audit trail with a purpose: it records whether the cited evidence was opened before approval, and how long review took.

Two further outputs follow the same discipline. A guideline panel answers queries from a locally held corpus of the operator's own licensed sources. OpenConsult ships no guideline content and names no publisher in its instructions. Each installation declares, in a version-pinned manifest, the sources its operator holds rights to, and a missing corpus means refusal, not improvisation. The development installation's corpus comprised 38 sources: 33 NICE guidelines, three Clinical Knowledge Summaries topics (licensed by Agilio Software, not NICE), and WHO and CDC dengue guidance, representing the synthetic case mix. NICE's content licence excludes artificial intelligence uses from its general permissions, but NICE has confirmed in writing (August 2026) that an individual may apply AI to NICE content for their own research, study and education, including publishing journal articles, and this installation and this paper rest on that confirmation. Referral letters are drafted from the approved note under the same grounding rules, as editable drafts for the doctor to finish.

Last, the supervised spoken history-taking mode. The doctor can tap a suggested question, and the system asks it aloud, in a voice the project calls Alba. Its intended consent script names the arrangement plainly: "My computer, Alba, will listen to you and will ask questions to figure out what is wrong. I will be supervising and joining the discussion when appropriate. Are you okay with that?" The machine's own words are kept out of the clinical record by construction, on both routes they could enter: on the data path, system utterances live in a separate table that the note, citation, export and letter queries structurally cannot see. On the acoustic path, the server mutes the transcriber's input across each spoken window, so the microphone's capture of the system's own voice reaches the audit recording but never the transcript. A fuller mode, in which the system asks routine history questions itself under the doctor's supervision, is built but disabled behind a configuration flag, and stays dark until its pre-registered evaluation is complete. An optional expressive face for Alba exists as a separate research question about machine-mediated rapport and is out of scope here.

A core design principle was that where a failure would be dangerous, the design tries to make it structurally impossible rather than merely detected afterwards. The protections are not all of equal strength: the strongest make a failure impossible (the transcript-exclusion guarantee, the alarm's non-suppression); the next make a check unskippable (the citation gate, the uncertainty flags); the weakest pin behaviour with tests that turn silent change into a loud, recorded decision (the verbatim phrases Alba may speak). What unites the tiers is the property that matters: none of them depends on anyone remembering to

check. Figure 1 shows how the components connect and where the boundary of the machine lies.

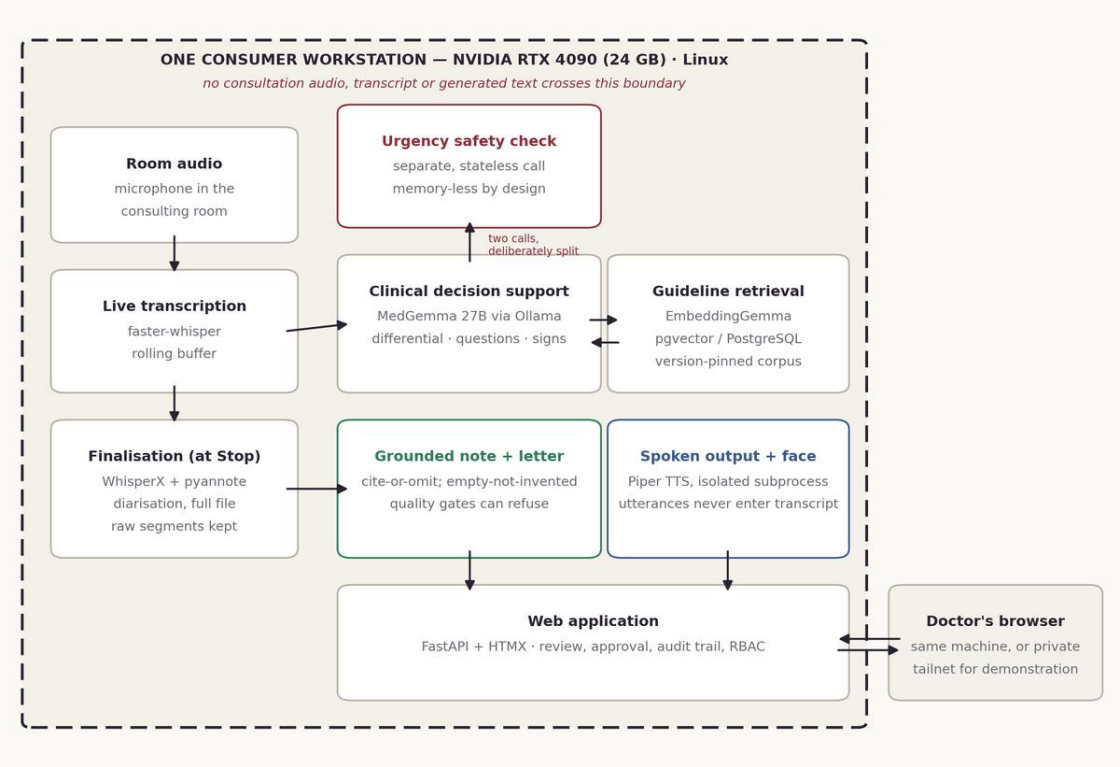


Figure 1. Architecture within the single-workstation boundary: live transcription feeds the clinical decision support (CDS) loop (urgency as a separated, memory-less call) and version-pinned guideline retrieval; at Stop, finalisation diarises the recording and the grounded note and letter are drafted under quality gates; spoken output is an isolated subprocess whose utterances cannot enter the transcript. No third party processes consultation audio, transcripts or generated text; in the demonstration configuration, rendered text reaches the operator's own device over an encrypted private overlay and nowhere else.

How it was built

OpenConsult was built by one of us (HH): a practising general practitioner, who held every clinical, product and safety decision while using AI coding assistants.

The build followed a small set of safety-conscious rules. Every change started as a written specification and was implemented as one reviewable change at a time. Software libraries were pinned to exact versions. Behaviour that matters is pinned by automated checks, and a check is never weakened to make a change pass. The whole history (specifications, changes, decisions and reversals) is published with the code [20]. Two AI assistants were used and kept apart: one specified and reviewed each change, the other wrote the code, and every result came back to the clinician for review. Both assistants come from the same model family, so their agreement is weak evidence on its own.

How it was judged

OpenConsult was iteratively evaluated during building. Where a feature made a clinical claim — a broad differential, an alarm on a buried red flag, a note that does not fabricate — the claim was expressed as a repeatable check and built alongside the feature, so that a weakened safety property surfaces as a failing gate rather than silent drift. Evaluation used a bench (a fixed test set) of fifteen scripted consultations, spanning emergencies and routine presentations, each with a predefined marking scheme. Metrics and their definitions were committed to the repository before any data existed, so the yardsticks would not move to fit the results.

The system was also exercised in realistic, acted consultations: a clinician and a volunteer role-playing an encounter end to end, in the room, with the assistant running as it would in use. A live consultation stresses what bench tests cannot reach: overlapping speech and speaker separation under real turn-taking, medication names and numbers spoken naturally, the behaviour of the decision support as a case actually unfolds. Five such consultations were run.

Security was evaluated with an adversarial approach. Software written quickly with heavy AI assistance can carry defects its author is poorly placed to see [26-27], so it was important to test the product. AI tools were used: a reviewing assistant and the implementing assistant, working to a written brief under the clinician's direction, with findings triaged by the clinician. This was an internal, AI-conducted adversarial review of our own work; genuinely independent review is proposed as future work. The brief covered authentication and sessions, input and output handling, template rendering, database access, subprocess use and the network surface, against a threat model of an unauthenticated remote attacker and a hostile low-privilege user. A sanitised summary appears in the Results; the full findings record is withheld because it maps attack paths applicable to deployed instances of the open-source code.

Evaluation to date is formative, conducted on synthetic material: it shows that the system behaves as intended under test and in acted use, not that it is safe or effective in real clinical care, and not how its outputs would be rated by independent clinicians, which remains for future work. We report the work against the applicable items of DECIDE-AI [28]. DECIDE-AI was written for early live clinical evaluation and a synthetic-only bench study sits one rung below its intended stage. Thus, we have reported against the items that apply, note those that cannot yet apply, and provide the item-by-item mapping as Multimedia Appendix 1. Naming the stage pre-empts reading a proof of concept as a validated product; the remaining rungs (independent, prospective, on real data) are future work.

Ethical Considerations

The system has only ever processed synthetic, acted consultations performed by the developer and family members; no real patient data has been used, and a documented self-assessment using the Health Research Authority's decision tools [29] classed the work as research requiring no NHS Research Ethics Committee review. Volunteers who role-played patients gave written consent covering recording, retention, and publication of findings

derived from their sessions; their voice recordings are personal data and are held solely on the local development machine, used only for development and evaluation, and not redistributed. They volunteered and received no payment. The system has not been placed on the market, put into service, or used in clinical care.

Results

The artefact and its record

The build is a working system with an auditable past: 213 commits over 27 days of initial development (July 2026), some 9,000 lines of application code and 4,700 of interface code, guarded at that point by more than 600 automated checks. Because every change ran through the discipline described in the Methods, the repository preserves the sequence of specifications, changes and review decisions behind each feature, and can be inspected [20].

One consultation, end to end

Figures 2-8 show a single synthetic consultation live: the scripted chest-pain case, replayed as acted two-voice audio through a loudspeaker. The capture method matters in two ways. It degrades speaker separation and the system's response to that is part of the demonstration: the quality gate flagged the borderline transcript before approval could proceed. The run shows the urgency alarm firing on the emerging cardiac picture while the differential is still broad and low-confidence; the differential maturing to a high-graded leading diagnosis with the guideline summary cited section by section to the local corpus; and the alarm clearing once the transcript shows the plan arranged. The approved note carries 31 of 31 claims cited to transcript lines; the referral letter is drafted from the approved note only, introducing no new clinical claims; and the encounter's complete audit trail, including the doctor's viewing of the raw transcript and acknowledgement of the quality flag, is recorded. A second, deliberately stopped-early run demonstrates the empty-but-true behaviour: sections left empty rather than invented, a single uncited claim tagged for manual checking, and the unresolved alarm following the consultation into review because the transcript never showed the action arranged.

The bench

The permanent test set is fifteen scripted consultations, each having its marking scheme written before any run: five core presentations, four red-flag variants as controlled comparisons, one held-out emergency absent from every prompt, and a five-script restraint series in which two undifferentiated cases must stay broad, one textbook case must narrow, and two buried emergencies must fire (Table 1). Against these, at deterministic settings (Multimedia Appendix 2): the urgency alarm fired on all eight emergency scripts (including the held-out condition and both buried emergencies), cleared in every case once the transcript showed the action arranged, and stayed silent on six of seven routine cases. Note generation across ten scripts produced 301 of 302 claims citing valid transcript lines — the existence check described in the Methods, with clinical faithfulness unmeasured pending independent rating — and the single uncited claim demonstrated the machinery

by being mechanically tagged and gated at approval. The guideline layer answered all five in-corpus queries from the correct source with every claim cited, and refused all four out-of-corpus queries rather than answering from model memory.

Two formal failures were kept rather than tuned away, on the reasoning that silencing them would train the safety component in the wrong direction: a transient alarm on a borderline-dengue presentation whose "urgent" proposal was exactly what the script's doctor then did, and a differential whose breadth and membership were right but which graded two conditions with unearned confidence. Both are recorded as failures against their marking schemes and deferred to clinical review as judgement calls.

Using the same model as its own judge returned a negative result we consider important for the field: without the transcript the judge over-reported errors (fifteen false positives); given the transcript it reported zero discrepancies while missing both seeded real errors it was pointed at. Same-model judging proved adequate for coarse coverage and unreliable for fine-grained quality in either direction; the ground truth for quality in this project is human review. The most severe fidelity specimen — a complete, internally coherent fabricated consultation generated from a mic-check transcript of one spoken sentence — was caught by citation validation, but initially nothing acted on the result: the fabricated note still reached the review page. The mandatory grounding gate described in the Methods is the direct product of that specimen; below a cited-claim threshold, the note is now replaced by a refusal, approval is refused server-side, and the specimen's shape is a regression test.

Acted consultations

In a series of five acted consultations exercising the supervised history-taking mode, the first four runs each surfaced a serious defect that the then-600-strong passing suite had not caught, and the fifth run, staged as the verification pass, cleared every hard rule while still surfacing a fifth (Table 2). All five were fixed, and each fix is pinned by a regression test using the run's own numbers where they exist. The most subtle was a camouflage specimen: a speaker-misattribution defect persisted across three consultations *because* the note generator inferred the true speakers from content and wrote accurate notes over wrong labels: a well-performing downstream component concealing an upstream defect, found only by a human comparing the transcript with what had actually happened in the room. The fourth run also gave the transcript-exclusion guarantee its first in-room confirmation under database inspection: the system's seven spoken utterances existed only in their own table, and no note citation could resolve to one.

Four serious defects reached the room. None of the five strictly required clinical training to observe. The clinician was required for designing the cases, judging the clinical consequence of each failure (one unrepaired example would have produced a signed record attributing the patient's symptoms to the doctor) and deciding, case by case, which behaviour was acceptable. The general lesson matches a long-standing observation about field use finding what pre-release testing misses [30]: a green (all tests passing), AI-authored test suite is a precondition of safe development, not evidence of it, and realistic

exercise of the running system must be planned as a mandatory evaluation stage from the outset.

Adversarial review

The adversarial audit returned twelve findings (one critical, three high, four medium, four low) plus a dependency review (Table 3). The critical finding was an unauthenticated stored cross-site-scripting vulnerability: text from the public registration form rendered unescaped into an administrative page, so a hostile registration could have compromised an administrator's session. Within the audit's scope, the server core (parameterised database queries, shell-free subprocess handling, session integrity) returned no findings, which we note as a bounded observation from one internal review, not a general assurance. The audit itself also required review: it recommended a strict script-source policy on the assertion that the pages used external scripts, when they in fact carried thousands of lines of inline script; such a policy would have silently broken. The assistant drafting the fix declined to take on trust an assertion the auditor had made about pages the auditor had itself written, and required the assertion to be confirmed against those pages, and a safe subset shipped instead. The auditor being auditable is part of the method, not an exception to it.

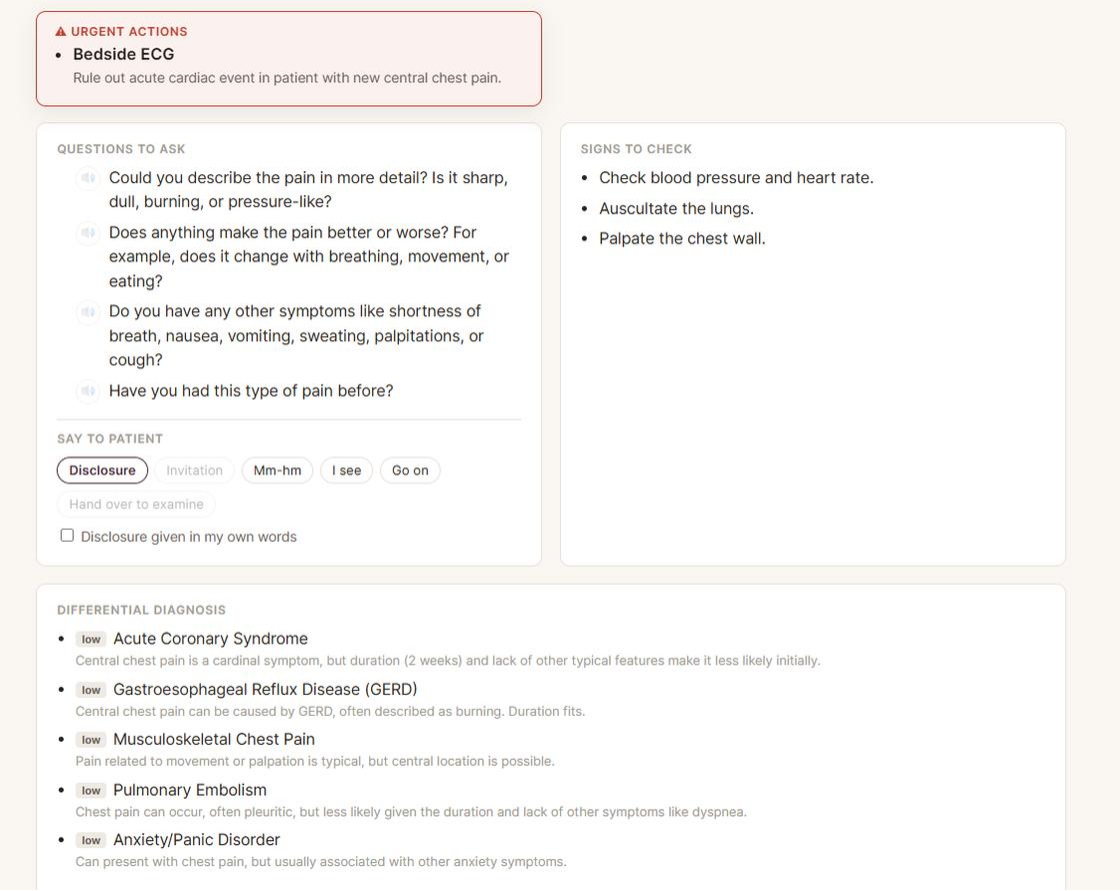


Figure 2. One synthetic consultation, live: the alarm fires while the differential is broad and all-low (cropped view; all names and details synthetic).

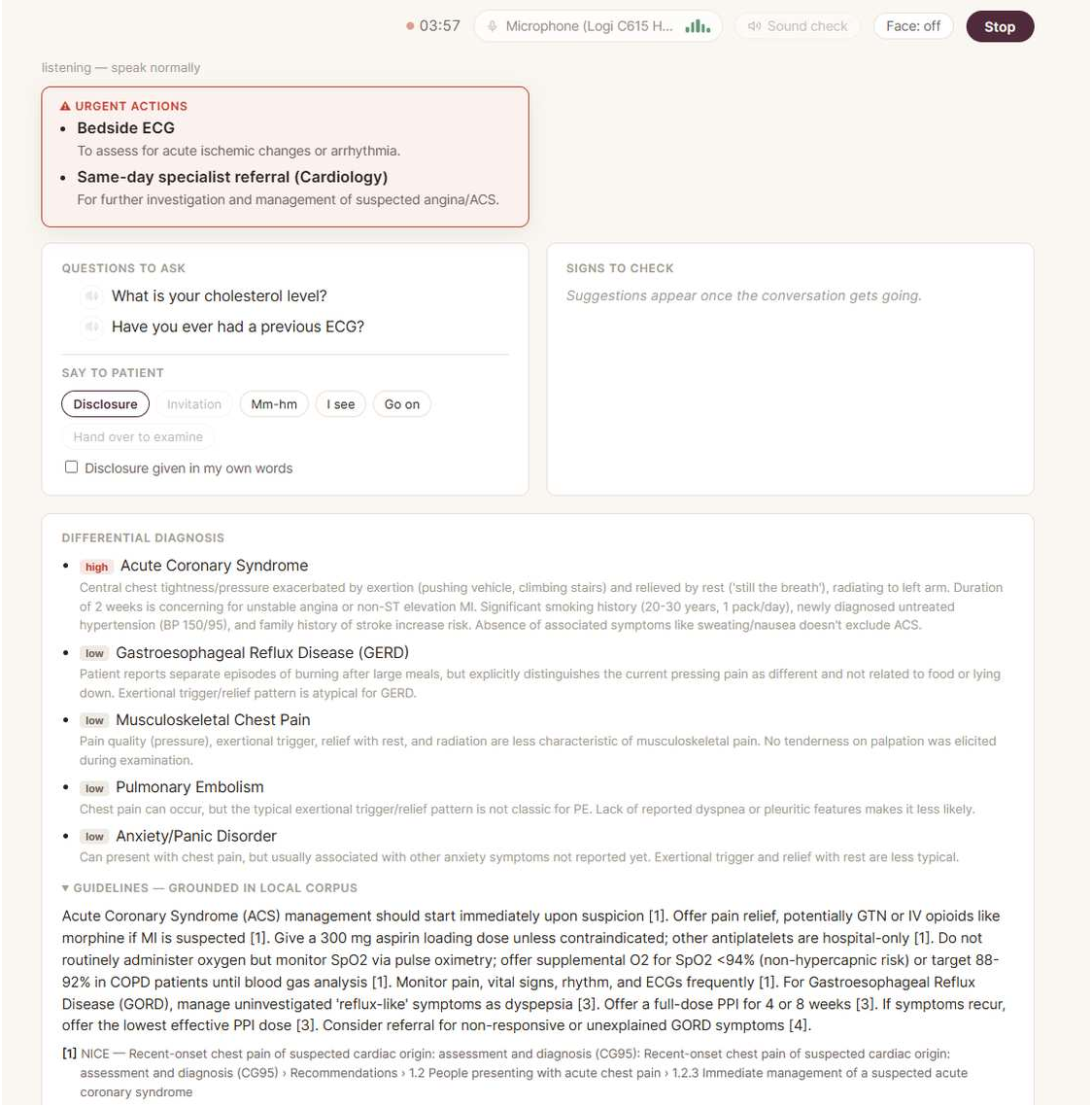


Figure 3. The same consultation at 03:57, the matured picture: leading diagnosis high, guidelines cited to the corpus sections (cropped view; all names and details synthetic).

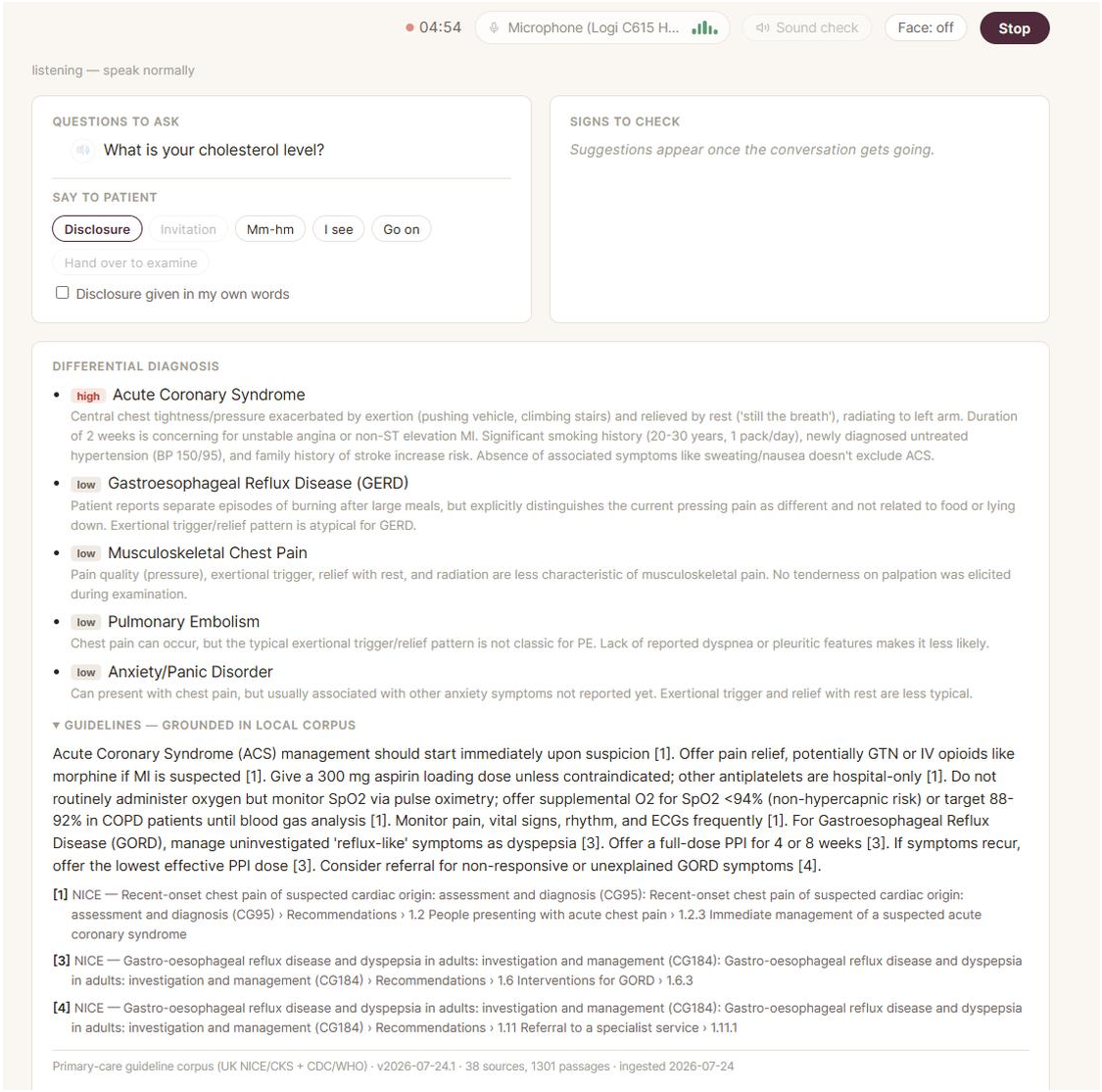


Figure 4. The same consultation at 04:54: the alarm cleared once the plan is arranged, corpus footer version-pinned (cropped view; all names and details synthetic).

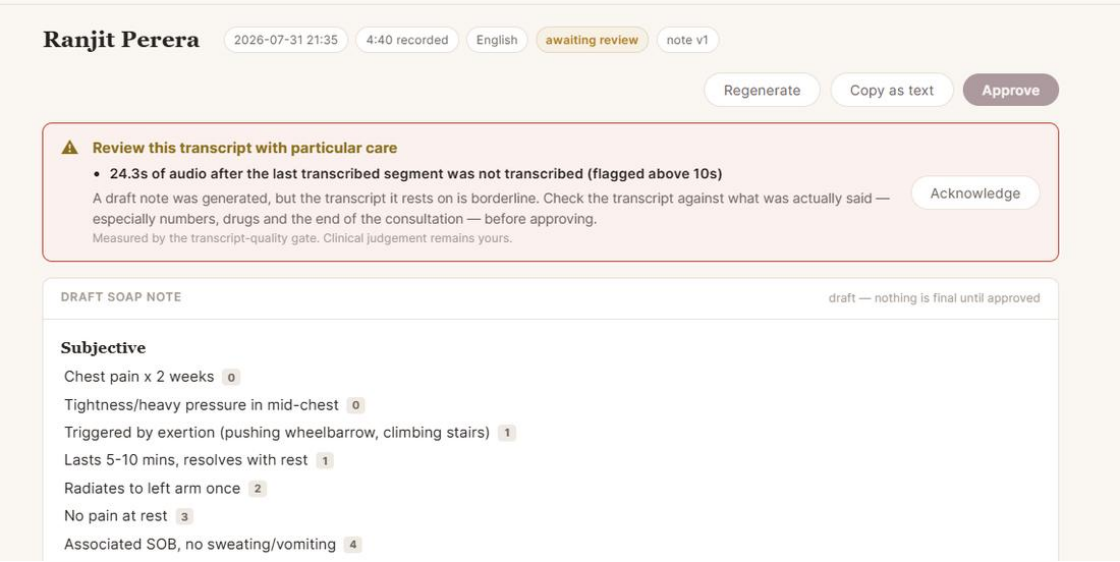


Figure 5. Outputs and audit of the same consultation: the quality gate flags borderline audio, with 24.3 s of untranscribed trailing audio (cropped view; all names and details synthetic).

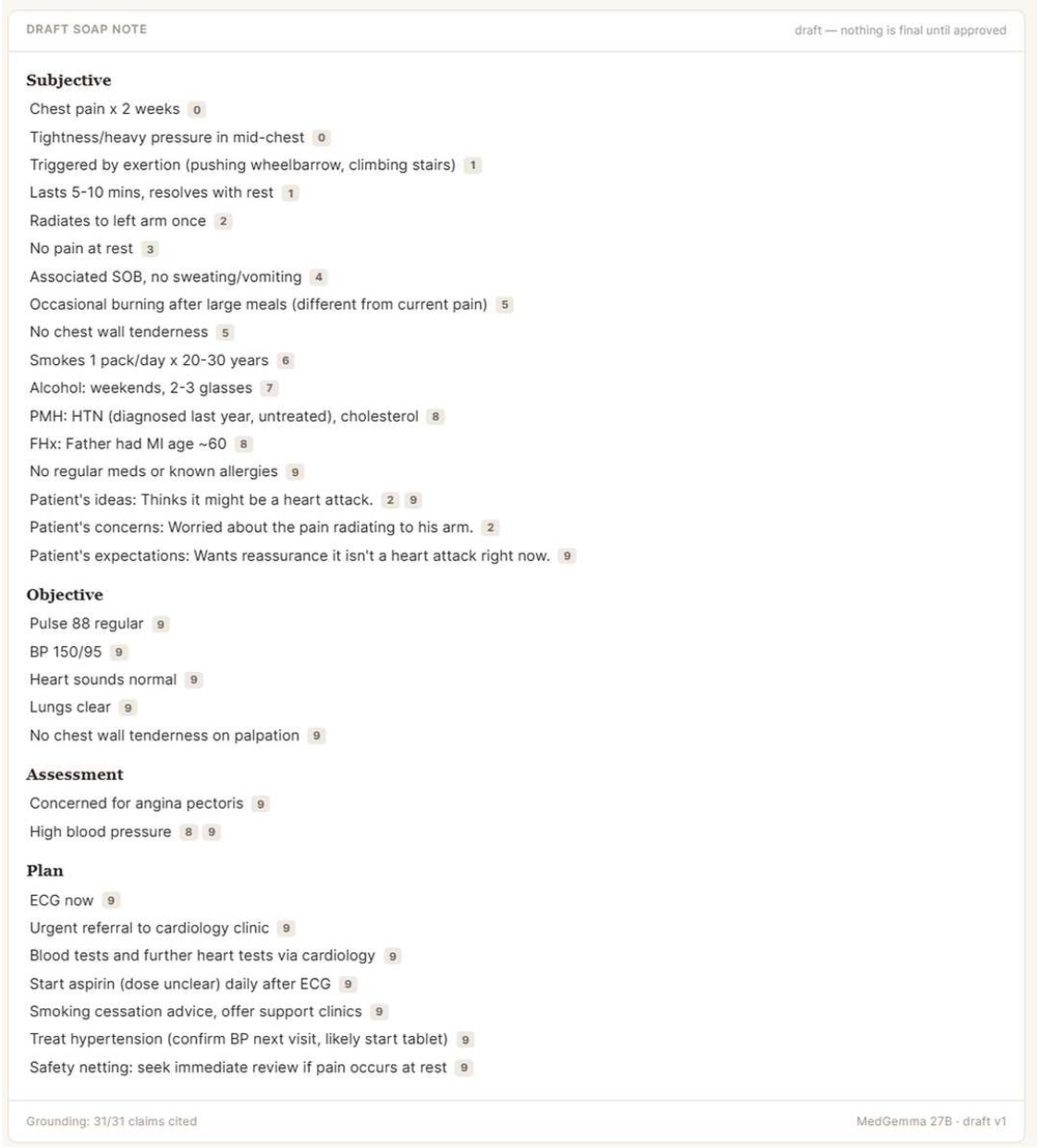
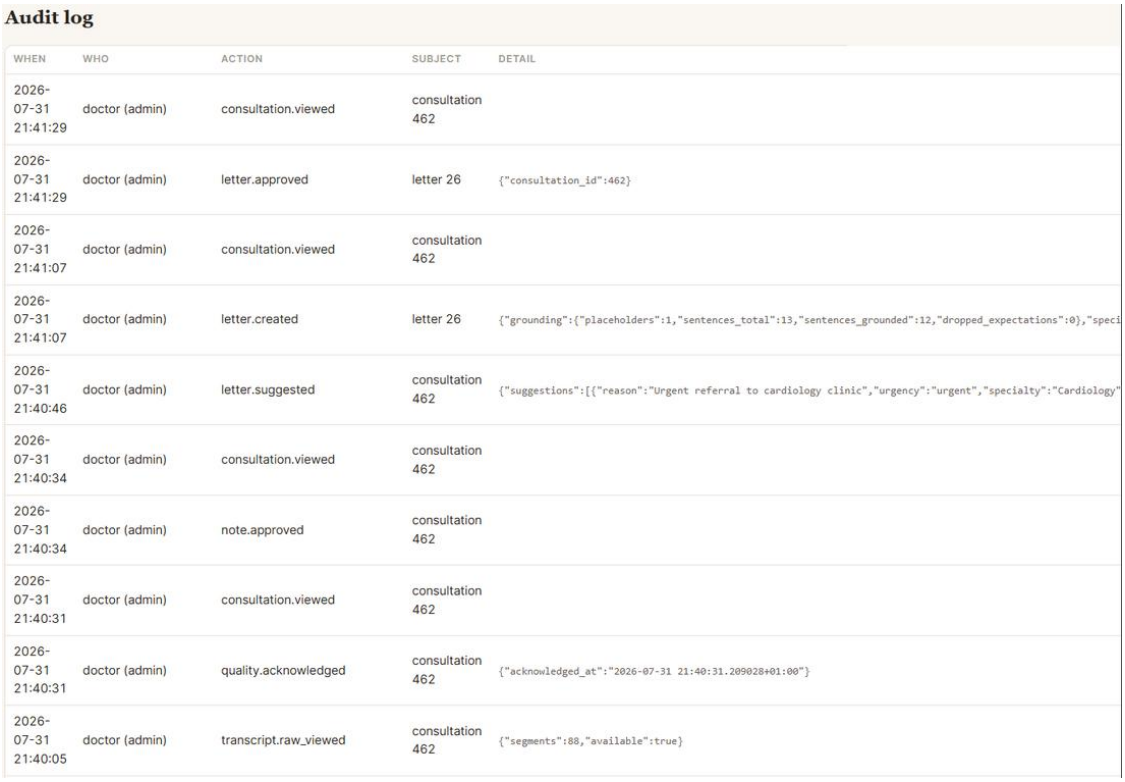


Figure 6. The draft note of the same consultation, with 31 of 31 claims cited (cropped view; all names and details synthetic).



Audit log

WHEN	WHO	ACTION	SUBJECT	DETAIL
2026-07-31 21:41:29	doctor (admin)	consultation.viewed	consultation 462	
2026-07-31 21:41:29	doctor (admin)	letter.approved	letter 26	{"consultation_id":462}
2026-07-31 21:41:07	doctor (admin)	consultation.viewed	consultation 462	
2026-07-31 21:41:07	doctor (admin)	letter.created	letter 26	{"grounding":{"placeholders":1,"sentences_total":13,"sentences_grounded":12,"dropped_expectations":0},"speci
2026-07-31 21:40:46	doctor (admin)	letter.suggested	consultation 462	{"suggestions":[{"reason":"Urgent referral to cardiology clinic","urgency":"urgent","specialty":"Cardiology"
2026-07-31 21:40:34	doctor (admin)	consultation.viewed	consultation 462	
2026-07-31 21:40:34	doctor (admin)	note.approved	consultation 462	
2026-07-31 21:40:31	doctor (admin)	consultation.viewed	consultation 462	
2026-07-31 21:40:31	doctor (admin)	quality.acknowledged	consultation 462	{"acknowledged_at":"2026-07-31 21:40:31.209028+01:00"}
2026-07-31 21:40:05	doctor (admin)	transcript.raw_viewed	consultation 462	{"segments":88,"available":true}

Figure 7. The audit log of the same consultation: views, acknowledgements, approvals and letters (cropped view; all names and details synthetic).

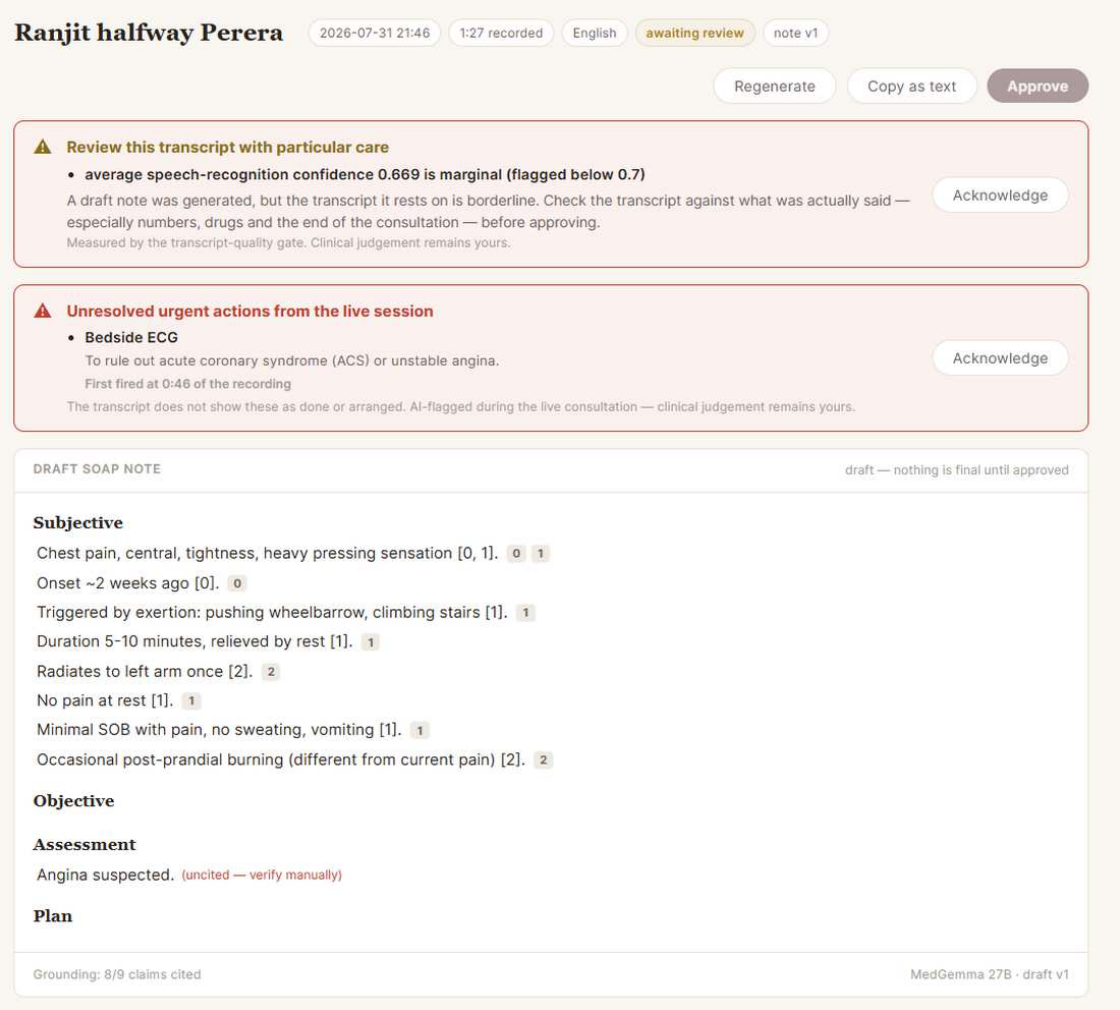


Figure 8. A stopped-early run: empty sections rather than invention, one uncited claim tagged, and the unresolved alarm following into review (cropped view; all names and details synthetic). Further views are in Multimedia Appendix 3.

Table 1. The synthetic evaluation programme: composition and headline behaviour.

Harness	Test material	What is checked	Headline result
Urgency alarm	9 scripts (5 routine incl. borderline cases; 4 emergencies as controlled variants) + 1 held-out unlisted emergency + 5 restraint-series scripts	Fires on time-critical presentations; silent on routine; clears when the action is arranged	8/8 emergencies fired and cleared, incl. the held-out condition and both buried emergencies; 6/7 routine silent; 1 kept false alarm (borderline dengue, deferred to clinical review)
Differential restraint	5-script controlled series (2 undifferentiated, 1	Breadth and confidence grading match the evidence;	4/5; 1 kept failure (correct membership, unearned confidence —

	differentiable control, 2 buried emergencies)	the same mechanical signal passes when narrowing is earned and fails when it is not	deferred as a judgement call)
Note generation	10 scripts with expected-content marking schemes	Per-claim citation of transcript turns (code-enforced); coverage vs marking scheme; discrepancies	301/302 claims validly cited (resolvability, code-enforced); the one uncited claim tagged and gated at approval; coverage unmeasured pending independent rating — judge unreliability demonstrated and reported
Guideline retrieval	5 in-corpus + 4 out-of-corpus queries	Correct source retrieved and cited; refusal rather than memory when uncovered	9/9; both refusal layers (similarity floor, model verdict) observed working

Table 2. The five acted room runs and what each found.

Run	What failed	Why the test suite missed it	The fix (and its regression test)
1	A silence-handling invariant discarded a 204-second merged transcript turn to remove 12.7 seconds of muted audio. Across the run the transcript lost six minutes in total (a 360.7-second trailing gap), including the medication history. The quality gate correctly refused the transcript.	A pre-existing test asserted the flawed any-overlap rule *as the requirement*; only a long, merged, real-room turn exposes the flaw.	Discard only when a majority fraction of the segment is muted; check moved before the speaker merge; two backstop caps on exclusion size. Regression test uses the run's real numbers; the wrong test was rewritten with a comment saying so.
2	A sound-check offer silently consumed the doctor's taps on question controls — actions accepted, nothing performed, explanation displayed where the doctor was not looking.	An interaction between two features under real-time use; each passed its own tests.	A standing design rule ("never swallow an action — perform it, or visibly disable the control with the reason on the control"), plus a test per instance.
3	No reachable way to stop the system speaking: the Stop control had scrolled out of view	Tests assert a control exists and works — not where	The speaking bar pinned to the viewport; a second design rule ("a control the

	exactly when needed, and the doctor stopped the whole recording to silence the machine.	it is at the moment it is needed.	doctor must reach must be where they are looking").
4	Speaker misattribution: the diarisation call pinned the speaker count at exactly two, so a single voice was split into two clusters and the first labelled "Doctor" — hidden for three consultations because accurate notes were generated over the wrong labels.	Every downstream check passed; the defect was upstream of everything the suite measured.	Speaker count unpinned; per-turn role correction with an audit trail; approval blocked until corrected labels are acknowledged. The affected consultations are retained unrepaired as the evidence set.
5	The doctor's declared speaker count was accepted and silently discarded — a race: the answer arrived eleven seconds after the finalisation job had already read the value.	The timing assumption ("model unload buys 10–30 s of slack") was simply false on a short consultation; no test questioned it.	The pipeline now waits, bounded, for the answer; an ignored declaration is visible in three places and blocks approval until acknowledged; a test asserts an answer can never be accepted and discarded without notice.

Severity is the clinician's judgement of clinical consequence; per-run system versions are given in Multimedia Appendix 4.

Table 3. Security audit findings (sanitised), by severity and disposition.

Severity	Finding (sanitised)	Disposition
Critical	Unauthenticated stored XSS: text submitted through the public registration form rendered unescaped into the administrator's user-management page — admin-session takeover on viewing	Fixed the same day: a single mandatory escaping helper applied at every rendering sink, plus input validation at the source; verified against the running system
High	A second stored-XSS path: free-text names entered in-app rendered unescaped into other users' pages (privilege escalation from a low-privilege role)	Fixed in the same change, same mechanism
High	Insecure default session-signing secret with silent fallback — forgeable sessions on any fresh deployment using the shipped default	Mitigated on the live instance (a real secret was set); fail-fast at startup landed before public release
High	Legacy shared-password test accounts	Verified closed: all such accounts

	possibly still active	confirmed inactive by direct inspection
Medium	Stored XSS in the audit-log view (tamper-evident record could be made to misrender)	Fixed with the XSS cluster
Medium	No Content-Security-Policy or security headers on any response	Safe CSP subset + content-type protections shipped; strict script policy awaits the inline-script refactor (scheduled)
Medium	Session cookie lacking the Secure attribute	Fixed (flag on by default, configurable for local development)
Medium	First registered account silently becomes an active administrator on a fresh deployment	Closed before public release: admin bootstrap moved to a server-side step, off the public form
Low	Transient XSS on a progress spinner	Fixed with the XSS cluster
Low	Logout is client-side only (session persists server-side until expiry)	Accepted with rationale for the demonstration deployment; documented
Low	No upload size cap on recordings	Accepted with rationale; scheduled
Low	Quickstart documentation ran the server with a development flag unsuitable for exposure	Closed: documentation rewritten with the production invocation
Note	Dependency CVEs in the pinned set	All located in the offline ML stack, not web-reachable in this architecture; version bump scheduled
Verification task	Rate-limit bucket keying behind the tunnel's proxy layer	Assigned to the developer to verify against the live deployment

Discussion

OpenConsult as a research and training instrument

The primary intended use of the current version of OpenConsult is to support research on AI scribe use in the consultation room. What makes it a research instrument is that everything the closed products keep invisible is, here, open, loggable and manipulable. OpenConsult's behaviour can be easily changed to support experimental studies of clinicians using AI scribes. Refusal behaviour can be switched between study arms; interruption can be forbidden; realistic errors can be planted; the corpus is declared; and the review step is recorded as it happens. That observability turns a set of suspicions about

ambient clinical AI into askable questions. We suggest here five possible research questions (Table 4), and OpenConsult was built so that each is measurable rather than rhetorical.

Table 4. Five research questions that OpenConsult was built to make measurable.

Research question	How OpenConsult makes it measurable
Does the human in the loop actually look?	The system records whether the reviewing clinician opened a claim's cited evidence before approving, and how long the review took. The ability to plant errors in the output allows clinicians' reviewing behaviour to be studied experimentally.
Does visible refusal make clinicians more careful, or less?	OpenConsult refuses rather than degrades — empty sections, withheld notes, declined queries — and because the gates are code, refusal can be randomised on and off in an experimental study.
What does an uninterrupted opening actually elicit?	Clinicians famously interrupt the patient's opening within seconds [31]; a machine can be forbidden to interrupt, and in the supervised history-taking mode the uninterrupted opening is a configurable, logged parameter. Content elicited against a marking scheme, with patient-rated experience secondary, becomes a measurable comparison.
Whose speech does it get wrong, and does the safety net notice?	The same scripts can be run across speakers and accents and judged not on average error but on clinical survival (do the drug names, doses and numbers come through) with the quality gates' flagging behaviour under test at the same time: not just whether the transcriber fails, but whether the system notices that it failed.
Does the review layer give time back, or add work?	The hope that drafting-plus-checking beats writing from scratch, and the worry that scrutinising machine output becomes its own burden, are competing hypotheses about the same measurable quantity. Review time is logged per consultation, enabling this to be directly tested.

Besides research, the codebase is a teaching object. There is a need to train clinicians in how to best use AI scribes and to understand how they can fail [32]. OpenConsult can be configured to support trainees' experience of using AI scribes and their understanding of how they work. Mock consultations with every feature enabled, and exercises built on planted errors (who finds the fabricated claim, who challenges the confident differential) address exactly the verification and automation-bias skills new clinicians will need. The marking schemes in the repository are a starting point.

Limitations

OpenConsult is an early, single-case artefact and has limitations. Testing has been on synthetic data and formative, scripted and graded by the developer; marking schemes written before runs constrain hindsight but do not substitute for independent raters, and the clinical faithfulness of generated notes is unmeasured. The two AI assistants that built and audited the system share one model family, so agreement between them is weak

evidence, and the audit is an internal one. The current build serves one consultation per machine; speaker attribution rests on a first-speaker heuristic (the first voice heard is labelled the doctor) with a channel for surfacing disagreement scheduled. The system is a research prototype, not a polished product.

Future work

OpenConsult is made freely available as open source code. The community may develop the system further. We also welcome independent formative evaluation: clinician raters with no stake in the software grading notes, differentials and letters against the existing marking schemes. We hope this will be followed by prospective study designs on the instrument itself: the five questions above, run with role-played and, under proper governance, real consultations. A retrospective design is open even sooner: running recorded real consultations through the system under appropriate governance, and comparing its outputs versus the contemporaneous documentation of the clinicians, with divergences adjudicated rather than assumed to be errors on either side. In parallel, a scribe-only edition — decision support excluded at build time, with its absence checked mechanically — is planned as a separate product with its own intended-purpose statement, aligned with the current regulatory position on transcription for clinician review [10]. Whether history-taking can safely be delegated to an AI system requires further evidence. The direction of travel nationally — an evaluation-methods framework for AI decision support is due within the coming year [33] — suggests the questions this instrument was built to ask are about to be asked of the whole field.

This paper describes OpenConsult v1. A rewrite, v2, began in October 2026, and a study to test it is being designed: several hundred curriculum-based cases, each played to the system by a synthetic patient voice, scored against marking schemes and go/no-go lines fixed before the first case runs. Four lessons from v1 shaped v2. First, v1's decision support was never given the patient's age and sex; v2 is designed to pass them with every request. Second, in continued use after the evaluation reported here, v1's spoken history-taking asked about categories ("risk factors for heart disease") where a doctor asks concrete, closed questions; v2 is designed to ask one concrete question at a time, choosing the areas of history to cover from the presentation, within a stated consultation length. Third, two of the five room defects concerned speaker attribution; v2 is designed to show a line whose speaker is in doubt as unsure for the doctor to settle, and to mark a misheard drug name rather than correct it silently. Fourth, v1 grew by increments and accumulated tests of little value; v2 was rebuilt from a single approved specification, with its checks written first for what matters to the patient and second for what matters to the user.

Conclusions

OpenConsult shows that a fully local, open-source, inspectable alternative to commercial ambient clinical AI can exist: an AI scribe with live decision support, running on one consumer machine, whose safety properties are built within its structure and whose evaluation is published with its code. It is a worked, auditable proof of concept, not yet a validated clinical tool, and it is offered as a starting point to the community. OpenConsult can be used as an open research instrument: a way to study questions about human review,

refusal, interruption, equity of transcription and reviewer burden measurable by anyone who can read the code and run a study.

Funding

This work received no specific funding.

Conflicts of Interest

The first author is the developer and, to date, the principal evaluator of the system described; this self-built and self-evaluated status is declared plainly and treated as a limitation in the Discussion. The first author is the sole copyright holder of the AGPL-licensed codebase and retains the possibility of dual-licensing it commercially; no such arrangement exists or is under negotiation. No other financial or commercial conflicts are declared.

Data Availability

The source code is openly available under the GNU Affero General Public License v3.0-or-later at https://github.com/IndyWH/OpenConsult (accessed 2026-11-12), and the release evaluated here is archived on Zenodo (concept DOI 10.5281/zenodo.22689260; version 1.0.0 DOI 10.5281/zenodo.22689261). A software citation file (CITATION.cff) accompanies the repository. The underlying models are used under their respective terms and are not redistributed with the AGPL-licensed source code; in particular, MedGemma is obtained separately under Google's Health AI Developer Foundations terms of use (as of August 2026), which permit research and development use and prohibit its outputs from directly informing clinical diagnosis or patient management without the applicable regulatory authorisation, the same boundary this project's synthetic-only posture observes. EmbeddingGemma is obtained under the separate Gemma terms. Deliberately withheld: the acted voice recordings (consent covers recording, not publication) and the full internal security-audit working record (a sanitised version appears as Table 3). Guideline content is fetched per installation and never redistributed; NICE has confirmed in writing that individual research use of this kind, including this publication, is permitted (see Methods). There is no associated patient dataset; all evaluation material is synthetic. The completed DECIDE-AI item-by-item mapping is provided as Multimedia Appendix 1.

Authors' Contributions

HMWI Herath conceived the project, made every clinical, product and safety decision, specified and directed the development, evaluated the system, and wrote the manuscript. H Potts contributed the critical review that reframed this paper, suggested additional literature, and revised the manuscript. In CRediT terms: conceptualisation, methodology, software, investigation and writing (original draft), HH; writing (review and editing), HH and HP. All named authors are human; the AI assistants described below are not authors and are disclosed here in accordance with journal policy [34].

Use of Generative AI

The system was developed with AI coding assistants built on a commercial large language model family (Anthropic's Claude models): a Claude-based assistant performed specification, design support, research and review, and a code-implementation assistant (Claude Code) wrote and modified source code on the development machine. The specific models available changed over the development period. The same Claude-based assistant supported the drafting and revision of this manuscript under the first author's direction; content decisions and approvals were the authors'. In every case, the assistants acted under written specification and human review: a practising clinician made all clinical, product and safety decisions, authorised each change before the next, and retained final approval on whether any change reached the running system. AI assistance did not extend to the making of clinical judgements. This disclosure is provided as part of doing the work transparently and mirrors the disclosure accompanying the software's release [35].

Abbreviations

AI: artificial intelligence
CDC: Centers for Disease Control and Prevention
DECIDE-AI: Developmental and Exploratory Clinical Investigations of Decision Support Systems Driven by Artificial Intelligence
NHS: National Health Service
NICE: National Institute for Health and Care Excellence
WHO: World Health Organization
XSS: cross-site scripting

Multimedia Appendices

Multimedia Appendix 1: DECIDE-AI checklist mapping.

Multimedia Appendix 2: Reproducibility configuration.

Multimedia Appendix 3: Further views of the worked example.

Multimedia Appendix 4: Per-script evaluation accounting.

References

1. Sinsky C, Colligan L, Li L, et al. Allocation of physician time in ambulatory practice: a time and motion study in 4 specialties. Ann Intern Med. 2016;165(11):753-760. doi:10.7326/M16-0961 PMID:27595430
2. Arndt BG, Beasley JW, Watkinson MD, et al. Tethered to the EHR: primary care physician workload assessment using EHR event log data and time-motion observations. Ann Fam Med. 2017;15(5):419-426. doi:10.1370/afm.2121 PMID:28893811
3. NHS England. Guidance on the use of AI-enabled ambient scribing products in health and care settings. NHS England; 2025 (updated 2026).

https://www.england.nhs.uk/long-read/guidance-on-the-use-of-ai-enabled-ambient-scribing-products-in-health-and-care-settings/ [accessed 2026-08-04].

4. Tierney AA, Gayre G, Hoberman B, et al. Ambient artificial intelligence scribes to alleviate the burden of clinical documentation. NEJM Catalyst. 2024;5(3):CAT.23.0404. doi:10.1056/CAT.23.0404
5. NHS Transformation Directorate. Using AI-enabled ambient scribing products in health and care settings: information governance guidance. https://transform.england.nhs.uk/information-governance/guidance/using-ai-enabled-ambient-scribing-products-in-health-and-care-settings/ [accessed 2026-08-04].
6. Dennstädt F, Hastings J, Putora PM, Schmerder M, Cihoric N. Implementing large language models in healthcare while balancing control, collaboration, costs and security. npj Digit Med. 2025;8(1):143. doi:10.1038/s41746-025-01476-7 PMID:40050366
7. Wang H, Yang R, Alwakeel M, et al. An evaluation framework for ambient digital scribing tools in clinical applications. npj Digit Med. 2025;8:358. doi:10.1038/s41746-025-01622-1 PMID:40514413
8. Ohde JW, Thompson A, Liu Z, et al. Barriers and opportunities of scaling ambient AI scribes for clinical documentation across diverse healthcare settings. npj Digit Med. 2026;9:369. doi:10.1038/s41746-026-02554-0 PMID:41866429
9. Croxford E, Gao Y, Pellegrino N, et al. Development and validation of the provider documentation summarization quality instrument for large language models. J Am Med Inform Assoc. 2025;32(6):1050-1060. doi:10.1093/jamia/ocaf068 PMID:40323321
10. Medicines and Healthcare products Regulatory Agency. MHRA clarifies regulatory status of ambient voice technologies used in the NHS. GOV.UK; 29 July 2026. https://www.gov.uk/government/news/mhra-clarifies-regulatory-status-of-ambient-voice-technologies-used-in-the-nhs [accessed 2026-11-12].
11. Bernstein MH, Sheppard B, Bruno MA, Lay PS, Baird GL. Randomized study of the impact of AI on perceived legal liability for radiologists. NEJM AI. 2025;2(6):AIoa2400785. doi:10.1056/AIoa2400785 PMID:42932544
12. Davies D. The Unaccountability Machine: Why Big Systems Make Terrible Decisions — and How the World Lost Its Mind. Profile Books; 2024. ISBN:9781788169547
13. Ashford and St Peter's Hospitals NHS Foundation Trust. FOI request 12084: ambient voice technology (AI scribes). Freedom of information response; completed 15 October 2026. https://www.ashfordstpeters.info/answers-from-2026/10989-foi-request-12084-ambient-voice-technology-ai-scribes [accessed 2026-11-12].
14. Williamson EJ, Walker AJ, Bhaskaran K, et al. Factors associated with COVID-19-related death using OpenSAFELY. Nature. 2020;584(7821):430-436. doi:10.1038/s41586-020-2521-4 PMID:32640463
15. Abstreiter M, Tarkoma S, Morabito R. Sometimes painful but certainly promising: feasibility and trade-offs of language model inference at the edge. arXiv preprint arXiv:2503.09114 (version 2); 2025. doi:10.48550/arXiv.2503.09114

16. Hendy B; ClinicianFOCUS. FreeScribe: a local medical scribe application (software; originally AI-Scribe). https://github.com/ClinicianFOCUS/FreeScribe [accessed 2026-08-04].
17. bloodworks-io. Phlox: an open-source, local-first AI medical scribe and patient-management system (software). https://github.com/bloodworks-io/phlox [accessed 2026-08-04].
18. Margolis S. OpenScribe: an open-source AI medical scribe (software). https://github.com/sammargolis/OpenScribe [accessed 2026-08-04].
19. Sellergren A, Kazemzadeh S, Jaroensri T, et al. MedGemma technical report. arXiv preprint arXiv:2507.05201; 2025. doi:10.48550/arXiv.2507.05201
20. Herath HMWI. OpenConsult: an open-source, fully local AI scribe with live clinical decision support (source code repository, v1.0.0). https://github.com/IndyWH/OpenConsult. Zenodo; 2026. doi:10.5281/zenodo.22689260
21. SYSTRAN. faster-whisper: a reimplementation of OpenAI Whisper using CTranslate2 (software). https://github.com/SYSTRAN/faster-whisper [accessed 2026-11-12].
22. Bain M, Huh J, Han T, Zisserman A. WhisperX: time-accurate speech transcription of long-form audio. In: Proceedings of INTERSPEECH 2023. ISCA; 2023:4489-4493. doi:10.21437/Interspeech.2023-78
23. Bredin H. pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe. In: Proceedings of INTERSPEECH 2023. ISCA; 2023:1983-1987. doi:10.21437/Interspeech.2023-105
24. Ollama. Ollama: run large language models locally (software). https://github.com/ollama/ollama [accessed 2026-11-12].
25. pgvector. pgvector: open-source vector similarity search for Postgres (software). https://github.com/pgvector/pgvector [accessed 2026-11-12].
26. Pearce H, Ahmad B, Tan B, Dolan-Gavitt B, Karri R. Asleep at the keyboard? Assessing the security of GitHub Copilot's code contributions. In: Proceedings of the 2022 IEEE Symposium on Security and Privacy. IEEE; 2022:754-768. doi:10.1109/SP46214.2022.9833571
27. Perry N, Srivastava M, Kumar D, Boneh D. Do users write more insecure code with AI assistants? In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. ACM; 2023:2785-2799. doi:10.1145/3576915.3623157
28. Vasey B, Nagendran M, Campbell B, et al. Reporting guideline for the early-stage clinical evaluation of decision support systems driven by artificial intelligence: DECIDE-AI. Nat Med. 2022;28(5):924-933. doi:10.1038/s41591-022-01772-9 PMID:35585198
29. Health Research Authority. Decision tools: Is my study research? and Do I need NHS REC review? https://hra-decisiontools.org.uk/research/ and https://hra-decisiontools.org.uk/ethics/ [accessed 2026-08-05].
30. Fenton NE, Ohlsson N. Quantitative analysis of faults and failures in a complex software system. IEEE Trans Software Eng. 2000;26(8):797-814. doi:10.1109/32.879815

31. Ospina NS, Phillips KA, Rodriguez-Gutierrez R, et al. Eliciting the patient's agenda: secondary analysis of recorded clinical encounters. J Gen Intern Med. 2019;34(1):36-40. doi:10.1007/s11606-018-4540-5 PMID:29968051
32. Raka Y, Van Deusen R, Wiberg H. From authoring to editing: a competency framework for trainee use of ambient AI clinical documentation. npj Digit Med. 2026;9(1):785. doi:10.1038/s41746-026-03351-5 PMID:42864380
33. National Institute for Health and Care Excellence. Unlocking growth by enabling safe AI innovation: NICE AI delivery plan. NICE; 5 August 2026. https://www.nice.org.uk/what-nice-does/digital-health/artificial-intelligence-at-nice [accessed 2026-08-05].
34. International Committee of Medical Journal Editors. Recommendations for the conduct, reporting, editing, and publication of scholarly work in medical journals. https://www.icmje.org/recommendations/ [accessed 2026-08-04].
35. Journal of Open Source Software. Submission requirements and editorial policies. https://joss.readthedocs.io/ [accessed 2026-08-04].

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- MultimediaAppendix1DECIDEAImapping.pdf
- MultimediaAppendix2reproducibility.pdf
- MultimediaAppendix3furtherviews.pdf
- MultimediaAppendix4perscriptaccounting.pdf