

Architecture Durability Review

Findings & Remediation Plan (Sample)

PREPARED FOR

Sample Client, Inc.

DATE

July 18, 2026

SECTION 1

Executive Summary

CRITICAL

HIGH

MEDIUM

LOW

INFO

The codebase shows signs of rapid, single-author development with meaningful correctness and maintainability risks. Two hardcoded secrets were committed (in test paths, but still requiring rotation/verification), and several files appear truncated or reference an OpenAI API method the analyzer flags as invalid—both of which threaten runtime behavior on the AI-generation critical path. Business-logic integrity is undermined by a credit/quota policy with two competing sources of truth and a god-module (views.py) that concentrates most change risk. Compounding this, an extreme bus factor (one author wrote ~77% of history and effectively owns every core module) creates severe continuity risk, while pervasive silent error-swallowing, duplicated 800+ LOC pages, and unmeasured test coverage make regressions hard to detect. Overall health is fragile: functional and security items need immediate attention before the structural and ownership concerns are addressed.

Scope: 372 files · 57,997 lines of code · git history analyzed.

SECTION 2

Risk Register

Findings sorted by severity. Reviewer annotations replace generated narrative where present.

HIGH

01 backend/core/admin.py is a churn hotspot

churn-hotspot · 90% confidence · static analysis · churn-hotspot:backend/core/admin.py

backend/core/admin.py is a churn hotspot (27 commits, ~1871 lines changed across 1342 LOC), signalling unstable design or a change magnet.

RECOMMENDATION

Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground. Consider splitting responsibilities and adding characterization tests before refactoring.

Evidence: backend/core/admin.py

HIGH

02 backend/core/models.py is a churn hotspot

churn-hotspot · 90% confidence · static analysis · churn-hotspot:backend/core/models.py

backend/core/models.py is a churn hotspot (29 commits, ~1003 lines changed across 942 LOC), signalling unstable design or a change magnet.

RECOMMENDATION

Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground. Consider splitting responsibilities and adding characterization tests before refactoring.

Evidence: backend/core/models.py

HIGH

03 backend/core/views.py is a churn hotspot

churn-hotspot · 90% confidence · static analysis · churn-hotspot:backend/core/views.py

backend/core/views.py is a churn hotspot (57 commits, ~4430 lines changed across 2506 LOC), signalling unstable design or a change magnet.

RECOMMENDATION

Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground. Consider splitting responsibilities and adding characterization tests before refactoring.

Evidence: backend/core/views.py

HIGH

04 frontend/lib/api.ts is a churn hotspot

churn-hotspot · 90% confidence · static analysis · churn-hotspot:frontend/lib/api.ts

frontend/lib/api.ts is a churn hotspot (32 commits, ~1124 lines changed across 949 LOC), signalling unstable design or a change magnet.

RECOMMENDATION

Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground. Consider splitting responsibilities and adding characterization tests before refactoring.

HIGH

05 backend/core/views.py is a god module that concentrates model access, credit business...

architecture · 85% confidence · LLM-assisted · arch-god-views-module

backend/core/views.py is a god module that concentrates model access, credit business rules, config reads, and hand-rolled serialization, making it the system's primary change magnet.

RECOMMENDATION

Split views by bounded context (credits, submissions, auth, content) and move response shaping into DRF serializers so the view layer only orchestrates. This directly attacks the 2506-LOC churn/bus-factor hotspot.

Evidence: backend/core/views.py:39-66 backend/core/views.py:165-200

HIGH

06 backend/config/settings.py

bus-factor · 85% confidence · static analysis · bus-factor:backend/config/settings.py

backend/config/settings.py (584 LOC) is effectively owned by one person — Contributor A authored 100% of its 18 commits.

RECOMMENDATION

Knowledge single point of failure. Recommend pairing/code review coverage on this module, documentation of its invariants, and spreading future changes across the team.

Evidence: backend/config/settings.py

HIGH

07 backend/core/admin.py

bus-factor · 85% confidence · static analysis · bus-factor:backend/core/admin.py

backend/core/admin.py (1342 LOC) is effectively owned by one person — Contributor A authored 100% of its 27 commits.

RECOMMENDATION

Knowledge single point of failure. Recommend pairing/code review coverage on this module, documentation of its invariants, and spreading future changes across the team.

Evidence: backend/core/admin.py

HIGH

08 backend/core/models.py

bus-factor · 85% confidence · static analysis · bus-factor:backend/core/models.py

backend/core/models.py (942 LOC) is effectively owned by one person — Contributor A authored 100% of its 29 commits.

RECOMMENDATION

Knowledge single point of failure. Recommend pairing/code review coverage on this module, documentation of its invariants, and spreading future changes across the team.

Evidence: backend/core/models.py

HIGH

09 backend/core/tests.py

bus-factor · 85% confidence · static analysis · bus-factor:backend/core/tests.py

backend/core/tests.py (1306 LOC) is effectively owned by one person — Contributor A authored 100% of its 13 commits.

RECOMMENDATION

Knowledge single point of failure. Recommend pairing/code review coverage on this module, documentation of its invariants, and spreading future changes across the team.

Evidence: backend/core/tests.py

HIGH

10 backend/core/views.py

bus-factor · 85% confidence · static analysis · bus-factor:backend/core/views.py

backend/core/views.py (2506 LOC) is effectively owned by one person — Contributor A authored 98% of its 57 commits.

RECOMMENDATION

Knowledge single point of failure. Recommend pairing/code review coverage on this module, documentation of its invariants, and spreading future changes across the team.

Evidence: backend/core/views.py

HIGH

11 frontend/app/(dashboard)/applications/[id]/page.tsx

bus-factor · 85% confidence · static analysis · bus-factor:frontend/app/(dashboard)/applications/[id]/page.tsx

frontend/app/(dashboard)/applications/[id]/page.tsx (816 LOC) is effectively owned by one person — Contributor A authored 94% of its 18 commits.

RECOMMENDATION

Knowledge single point of failure. Recommend pairing/code review coverage on this module, documentation of its invariants, and spreading future changes across the team.

Evidence: frontend/app/(dashboard)/applications/[id]/page.tsx

HIGH

12 frontend/app/(dashboard)/apply/page.tsx

bus-factor · 85% confidence · static analysis · bus-factor:frontend/app/(dashboard)/apply/page.tsx

frontend/app/(dashboard)/apply/page.tsx (893 LOC) is effectively owned by one person — Contributor A authored 91% of its 23 commits.

RECOMMENDATION

Knowledge single point of failure. Recommend pairing/code review coverage on this module, documentation of its invariants, and spreading future changes across the team.

Evidence: frontend/app/(dashboard)/apply/page.tsx

HIGH

13 frontend/lib/api.ts

bus-factor · 85% confidence · static analysis · bus-factor:frontend/lib/api.ts

frontend/lib/api.ts (949 LOC) is effectively owned by one person — Contributor A authored 100% of its 32 commits.

RECOMMENDATION

Knowledge single point of failure. Recommend pairing/code review coverage on this module, documentation of its invariants, and spreading future changes across the team.

HIGH

14 The apply page and the application-detail page are near-duplicate 800+ LOC client...

architecture · 80% confidence · LLM-assisted · arch-duplicated-apply-pages

The apply page and the application-detail page are near-duplicate 800+ LOC client components that each re-implement the same draft/revision/cover-letter orchestration, so every behavioral change must be made twice.

RECOMMENDATION

Extract the draft+revision+cover-letter workflow into a shared hook (e.g. useDraftWorkflow) and shared subcomponents so both routes consume one implementation.

Evidence: frontend/app/(dashboard)/apply/page.tsx:182-200 frontend/app/(dashboard)/applications/[id]/page.tsx:151-197

HIGH

15 Repository bus factor is dangerously low: Contributor A authored 77% of all 230 commits

bus-factor · 80% confidence · static analysis · bus-factor:repository

Repository bus factor is dangerously low: Contributor A authored 77% of all 230 commits (1 author(s) account for half the history).

RECOMMENDATION

The project's continuity depends on very few people. Recommend onboarding docs, architecture decision records, and deliberate knowledge transfer as part of the engagement.

HIGH

16 Possible Hardcoded secret assignment committed in backend/core/test_axiom_logging.py:133

secret · 60% confidence · static analysis · secret:generic-secret:backend/core/test_axiom_logging.py:133

RECOMMENDATION

Treat as compromised: rotate the credential immediately, purge it from git history (e.g. git filter-repo / BFG), and move the value to a secrets manager or untracked .env. Verify no downstream systems still trust the old value.

Evidence: backend/core/test_axiom_logging.py:133

HIGH

17 Possible Hardcoded secret assignment committed in...

secret · 60% confidence · static analysis · secret:generic-secret:frontend/__tests__/lib/onboarding-api.test.ts:14

Possible Hardcoded secret assignment committed in frontend/__tests__/lib/onboarding-api.test.ts:14.

RECOMMENDATION

Treat as compromised: rotate the credential immediately, purge it from git history (e.g. git filter-repo / BFG), and move the value to a secrets manager or untracked .env. Verify no downstream systems still trust the old value.

Evidence: frontend/__tests__/lib/onboarding-api.test.ts:14

MEDIUM

18 Hallucinated API

ai-generation · 95% confidence · LLM-assisted · fingerprint:hallucinated-api-client-responses-create-is-not-

Hallucinated API: 'client.responses.create()' is not a valid OpenAI API method. The correct method is 'client.chat.completions.create()'. The 'responses' module does not exist in the OpenAI Python SDK.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: backend/core/ai.py:92-94

MEDIUM

19 backend/config/settings.py is a churn hotspot

churn-hotspot · 90% confidence · static analysis · churn-hotspot:backend/config/settings.py

backend/config/settings.py is a churn hotspot (18 commits, ~405 lines changed across 584 LOC), signalling unstable design or a change magnet.

RECOMMENDATION

Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground. Consider splitting responsibilities and adding characterization tests before refactoring.

Evidence: backend/config/settings.py

MEDIUM

20 backend/config/urls.py is a churn hotspot

churn-hotspot · 90% confidence · static analysis · churn-hotspot:backend/config/urls.py

backend/config/urls.py is a churn hotspot (26 commits, ~196 lines changed across 172 LOC), signalling unstable design or a change magnet.

RECOMMENDATION

Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground. Consider splitting responsibilities and adding characterization tests before refactoring.

Evidence: backend/config/urls.py

MEDIUM

21 frontend/app/(dashboard)/applications/[id]/page.tsx is a churn hotspot

churn-hotspot · 90% confidence · static analysis · churn-hotspot:frontend/app/(dashboard)/applications/[id]/page.tsx

frontend/app/(dashboard)/applications/[id]/page.tsx is a churn hotspot (18 commits, ~1193 lines changed across 816 LOC), signalling unstable design or a change magnet.

RECOMMENDATION

Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground. Consider splitting responsibilities and adding characterization tests before refactoring.

Evidence: frontend/app/(dashboard)/applications/[id]/page.tsx

MEDIUM**22 frontend/app/(dashboard)/apply/page.tsx is a churn hotspot**

churn-hotspot · 90% confidence · static analysis · churn-hotspot:frontend/app/(dashboard)/apply/page.tsx

frontend/app/(dashboard)/apply/page.tsx is a churn hotspot (23 commits, ~2054 lines changed across 893 LOC), signalling unstable design or a change magnet.

RECOMMENDATION

Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground. Consider splitting responsibilities and adding characterization tests before refactoring.

Evidence: frontend/app/(dashboard)/apply/page.tsx

MEDIUM**23 File ends abruptly at line 260 mid-frozenset definition without closing the structure**

ai-generation · 90% confidence · LLM-assisted · fingerprint:file-ends-abruptly-at-line-260-mid-frozenset-def

File ends abruptly at line 260 mid-frozenset definition without closing the structure. Code is incomplete/truncated.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: backend/config/logging.py:245-303

MEDIUM**24 File ends abruptly at line 260 mid-JSX Link element without closing tag or returning...**

ai-generation · 90% confidence · LLM-assisted · fingerprint:file-ends-abruptly-at-line-260-mid-jsx-link-elem

File ends abruptly at line 260 mid-JSX Link element without closing tag or returning component. Code is incomplete/truncated.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: frontend/app/blog/[slug]/page.tsx:260-272

MEDIUM**25 File ends abruptly mid-function at line 260 without completing main() logic or...**

ai-generation · 85% confidence · LLM-assisted · fingerprint:file-ends-abruptly-mid-function-at-line-260-with

File ends abruptly mid-function at line 260 without completing main() logic or returning control flow. Code is incomplete/stubbed.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: scripts/agents/monitoring_agent.py:260-315

MEDIUM

26 Invented/hallucinated API call

ai-generation · 85% confidence · LLM-assisted · fingerprint:invented-hallucinated-api-call-client-responses-

Invented/hallucinated API call: `client.responses.create(input=messages, reasoning={...})`. This does not match OpenAI's actual SDK. The real API uses `chat.completions.create(messages=...)`, not `responses.create(input=...)`. This is a strong signal of hallucination or outdated/fictional wrapper.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: `backend/core/topic_sources.py:150-157`

MEDIUM

27 Credit/quota policy has two competing sources of truth

architecture · 80% confidence · LLM-assisted · arch-config-dual-source

Credit/quota policy has two competing sources of truth — hardcoded module constants and runtime `AppConfiguration` reads — so behavior depends on which path a caller happens to use.

RECOMMENDATION

Centralize pricing/quota policy in one typed config accessor (single module) with defaults defined once; forbid views from reading raw constants so all consumers see the same authoritative values.

Evidence: `backend/core/views.py:70-101` `backend/core/views.py:133-150`

MEDIUM

28 Business orchestration and backend async coordination live inside React page components...

architecture · 72% confidence · LLM-assisted · arch-ui-owns-async-orchestration

Business orchestration and backend async coordination live inside React page components as sprawling local state plus client-side polling with magic numbers, coupling UI rendering to backend job lifecycle.

RECOMMENDATION

Move phase/lifecycle into an explicit state machine or data layer (React Query/SWR with polling config) so components render state rather than drive backend orchestration; centralize timeout/poll constants.

Evidence: `frontend/app/(dashboard)/apply/page.tsx:47-114` `frontend/app/(dashboard)/apply/page.tsx:126-157`

MEDIUM

29 The frontend adopts a silent-failure error strategy in load paths, discarding errors so...

architecture · 70% confidence · LLM-assisted · arch-silent-error-swallowing

The frontend adopts a silent-failure error strategy in load paths, discarding errors so failures render as empty/absent UI with no diagnostics.

RECOMMENDATION

Standardize on a small result/error-boundary strategy: surface load errors to the user and log them via the existing request-id logging rather than empty catch blocks.

Evidence: `frontend/app/(dashboard)/apply/page.tsx:50-53` `frontend/app/(dashboard)/apply/page.tsx:63-74`

MEDIUM**30 frontend/app/(dashboard)/applications/[id]/page.tsx**

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(dashboard)/applications/[id]/page.tsx

frontend/app/(dashboard)/applications/[id]/page.tsx (816 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(dashboard)/applications/[id]/page.tsx

MEDIUM**31 frontend/app/(dashboard)/apply/page.tsx**

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(dashboard)/apply/page.tsx

frontend/app/(dashboard)/apply/page.tsx (893 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(dashboard)/apply/page.tsx

MEDIUM**32 frontend/app/(dashboard)/dashboard/page.tsx**

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(dashboard)/dashboard/page.tsx

frontend/app/(dashboard)/dashboard/page.tsx (297 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(dashboard)/dashboard/page.tsx

MEDIUM**33 frontend/app/(dashboard)/job-search/page.tsx**

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(dashboard)/job-search/page.tsx

frontend/app/(dashboard)/job-search/page.tsx (488 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(dashboard)/job-search/page.tsx

MEDIUM**34 frontend/app/(dashboard)/payment-methods/page.tsx**

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(dashboard)/payment-methods/page.tsx

frontend/app/(dashboard)/payment-methods/page.tsx (153 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(dashboard)/payment-methods/page.tsx

MEDIUM

35 frontend/app/(dashboard)/resumes/page.tsx

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(dashboard)/resumes/page.tsx

frontend/app/(dashboard)/resumes/page.tsx (262 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(dashboard)/resumes/page.tsx

MEDIUM

36 frontend/app/(dashboard)/settings/page.tsx

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(dashboard)/settings/page.tsx

frontend/app/(dashboard)/settings/page.tsx (236 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(dashboard)/settings/page.tsx

LOW

37 Incomplete function definition with docstring but no body visible

ai-generation · 75% confidence · LLM-assisted · fingerprint:incomplete-function-definition-with-docstring-bu

Incomplete function definition with docstring but no body visible. Line 260 shows 'if legacy_credits is not None and legacy_credits > 0:' with no implementation following, suggesting stubbed/incomplete code.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: backend/core/stripe_views.py:258-260

LOW

38 Third instance of hallucinated __import__('django.http').http.HttpResponse pattern

ai-generation · 75% confidence · LLM-assisted · fingerprint:third-instance-of-hallucinated-import-django-htt

Third instance of hallucinated __import__('django.http').http.HttpResponse pattern. Repetition across multiple test functions is characteristic of LLM code generation.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: backend/core/test_logging.py:119

LOW**39 `_check_impersonation_target` contains six repetitive validation blocks**

ai-generation · 73% confidence · LLM-assisted · fingerprint:check-impersonation-target-contains-six-repetiti

`_check_impersonation_target` contains six repetitive validation blocks (lines 134-148, 150-156, 158-164), each with nearly identical pattern: check condition, call `message_user`, return error tuple. Boilerplate expansion of single validation logic.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: `backend/core/admin.py:125-167`**LOW****40 Comment 'Reads and removes the stored UTM in one step so that a later direct visit to...**

ai-generation · 72% confidence · LLM-assisted · fingerprint:comment-reads-and-removes-the-stored-utm-in-one-

Comment 'Reads and removes the stored UTM in one step so that a later direct visit to `/free-match` doesn't inherit attribution from a previous SEO session' followed by over-defensive try/catch around `sessionStorage.getItem/removeItem` (which cannot throw) and catch block with comment about privacy modes—overly defensive pattern typical of LLM-generated client code.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: `frontend/app/free-match/page.tsx:61-74`**LOW****41 Duplicate state section headers with identical comment 'Cover letter state' at lines 71...**

ai-generation · 72% confidence · LLM-assisted · fingerprint:duplicate-state-section-headers-with-identical-c

Duplicate state section headers with identical comment 'Cover letter state' at lines 71 and 74, suggesting copy-paste without review or LLM boilerplate generation.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: `frontend/app/(dashboard)/applications/[id]/page.tsx:71-82`**LOW****42 Over-defensive nested try-catch blocks around code that cannot throw exceptions**

ai-generation · 72% confidence · LLM-assisted · fingerprint:over-defensive-nested-try-catch-blocks-around-co

Over-defensive nested try-catch blocks around code that cannot throw exceptions. Inner try-catch (lines 146-152) wraps a database update that is already inside an outer exception handler, creating redundant error suppression that masks real issues.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

LOW

43 Uniform defensive docstring pattern repeated across three nearly identical helper...

ai-generation · 72% confidence · LLM-assisted · fingerprint:uniform-defensive-docstring-pattern-repeated-acr

Uniform defensive docstring pattern repeated across three nearly identical helper functions (get_free_revisions_per_application, get_free_applications_per_user, get_free_job_searches_per_user). Each follows identical structure: docstring, config read, try/except with identical error handling. Hallmark of template-generated code.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: backend/core/views.py:76-102

LOW

44 File truncates mid-statement in issue creation loop

ai-generation · 70% confidence · LLM-assisted · fingerprint:file-truncates-mid-statement-in-issue-creation-l

File truncates mid-statement in issue creation loop; incomplete code likely indicates auto-generation cut off

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: backend/core/evaluation/pipeline.py:260

LOW

45 Three nearly identical utility functions

ai-generation · 70% confidence · LLM-assisted · fingerprint:three-nearly-identical-utility-functions-readper

Three nearly identical utility functions (readPersistedState, writePersistedState, clearPersistedState) with identical defensive window-undefined checks and try/catch patterns. The repetitive structure and lack of abstraction is typical of LLM-generated utility stubs.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: frontend/app/(dashboard)/job-search/page.tsx:43-52

LOW

46 Unusual state management pattern: manually tracking prevStep in render

ai-generation · 70% confidence · LLM-assisted · fingerprint:unusual-state-management-pattern-manually-tracki

Unusual state management pattern: manually tracking prevStep in render (lines 68-71) to trigger side effect during render rather than using useEffect. This is a suspicious anti-pattern that suggests auto-generated boilerplate workaround.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

LOW 47 **handleFetchUrl contains polling loop**

ai-generation · 68% confidence · LLM-assisted · fingerprint:handlefetchurl-contains-polling-loop-lines-133-1

handleFetchUrl contains polling loop (lines 133-149) with magic number 30 iterations and 2000ms timeout, identical error handling pattern repeated twice (lines 146, 150), and catch block with redundant fallback message. Template-generated async pattern.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: frontend/app/(dashboard)/apply/page.tsx:126-157

LOW 48 **Over-defensive empty catch block**

ai-generation · 68% confidence · LLM-assisted · fingerprint:over-defensive-empty-catch-block-catch-silently-

Over-defensive empty catch block: .catch() => {} silently swallows all errors from getMe without logging or state feedback. Suggests stubbed error handling.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: frontend/app/(dashboard)/profile/page.tsx:35

LOW 49 **Boilerplate comment restating function purpose**

ai-generation · 65% confidence · LLM-assisted · fingerprint:boilerplate-comment-restating-function-purpose-s

Boilerplate comment restating function purpose (Step 5: Reading time) - section header with no implementation

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: backend/core/seo_agent.py:259

LOW 50 **Generic catch block with no error inspection**

ai-generation · 65% confidence · LLM-assisted · fingerprint:generic-catch-block-with-no-error-inspection-cat

Generic catch block with no error inspection; catches but discards error object entirely and displays hardcoded message, typical of LLM placeholder error handling

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: frontend/app/(dashboard)/applications/page.tsx:48-59

LOW

51 Inline comments restating visual purpose

ai-generation · 65% confidence · LLM-assisted · fingerprint:inline-comments-restating-visual-purpose-backgro

Inline comments restating visual purpose ('// Background orb', '// Brand pill', '// Headline', '// Subline', '// ATS score badge') are characteristic of LLM output when asked to make markup 'readable'. Each comment merely names the visual element below it.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: frontend/app/opengraph-image.tsx:24-48

LOW

52 Nested function `_build_intent_params()` defined inside `create_payment_intent()` that...

ai-generation · 65% confidence · LLM-assisted · fingerprint:nested-function-build-intent-params-defined-insi

Nested function `_build_intent_params()` defined inside `create_payment_intent()` that is called exactly twice (lines 229, 243). This is an unnatural refactor—the function body is only 9 lines and the nesting adds no value. Suggests LLM over-eager to extract and abstract logic.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: backend/core/stripe_service.py:214-224

LOW

53 Over-defensive comment 'Should not crash on messy input' followed by assertions that...

ai-generation · 65% confidence · LLM-assisted · fingerprint:over-defensive-comment-should-not-crash-on-messy

Over-defensive comment 'Should not crash on messy input' followed by assertions that don't validate the robustness claim—typical LLM pattern of stating the obvious and then testing generic dict structure rather than actual parsing behavior.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: backend/core/evaluation/tests.py:127-133

LOW

54 Unnaturally uniform and verbose docstring with bullet points listing outputs that reads...

ai-generation · 65% confidence · LLM-assisted · fingerprint:unnaturally-uniform-and-verbose-docstring-with-b

Unnaturally uniform and verbose docstring with bullet points listing outputs that reads like generated documentation. Phrasing 'All created with status=DRAFT, ready for morning review in Django admin' is overly explanatory.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

LOW

55 ensureRequestId function contains over-defensive type checking

ai-generation · 64% confidence · LLM-assisted · fingerprint:ensurerequestid-function-contains-over-defensive

ensureRequestId function contains over-defensive type checking (lines 26-32) with three separate branches for Header types that all perform identical transformations. Hallmark over-engineered LLM scaffold for simple operation.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: frontend/lib/api.ts:22-41

LOW

56 JobSubmission defaults its status to DRAFT_READY at the model layer while the runtime...

architecture · 62% confidence · LLM-assisted · arch-jobsubmission-default-state

JobSubmission defaults its status to DRAFT_READY at the model layer while the runtime flow treats new submissions as pending/generating, creating an inconsistent state model that can misrepresent in-flight work.

RECOMMENDATION

Default the model status to PENDING and let the generation pipeline advance it, so persisted state always reflects the true lifecycle position.

Evidence: backend/core/models.py:139-143 frontend/app/(dashboard)/apply/page.tsx:159-173

LOW

57 Over-defensive catch-all with comment 'Silently fail

ai-generation · 62% confidence · LLM-assisted · fingerprint:over-defensive-catch-all-with-comment-silently-f

Over-defensive catch-all with comment 'Silently fail — panel will show empty state'. Catches all errors without discrimination and suppresses them, combined with overly defensive pattern of catching errors that API calls should handle.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: frontend/components/credits-panel.tsx:70-74

LOW

58 The API client relies on module-level mutable global state for the unauthorized...

architecture · 60% confidence · LLM-assisted · arch-module-mutable-singleton

The API client relies on module-level mutable global state for the unauthorized handler, an implicit singleton that is order-dependent and hard to test or scope per-request.

RECOMMENDATION

Inject the unauthorized/logout callback through a client factory or context rather than mutating module state, so behavior is explicit and testable.

Evidence: frontend/lib/api.ts:74-80

LOW

59 Tests exist but no coverage measurement is configured, so coverage gaps are invisible...

test-coverage · 60% confidence · static analysis · coverage:untracked

Tests exist but no coverage measurement is configured, so coverage gaps are invisible to the team.

RECOMMENDATION

Wire up coverage reporting (e.g. vitest --coverage, nyc, or coverage.py) and set a baseline threshold in CI so regressions surface.

LOW

60 Boilerplate comments restating the obvious structure of JSX elements

ai-generation · 60% confidence · LLM-assisted · fingerprint:boilerplate-comments-restating-the-obvious-struc

Boilerplate comments restating the obvious structure of JSX elements (`{/ Meta row /}`, `{/ Title /}`, `{/ Description /}`, `{/ CTA /}`). These are unnecessary labels that add no semantic value and are typical of LLM-generated UI code.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: frontend/app/blog/page.tsx:82-94

LOW

61 Comment 'CTA configuration' followed by redundant inline comments

ai-generation · 60% confidence · LLM-assisted · fingerprint:comment-cta-configuration-followed-by-redundant-

Comment 'CTA configuration' followed by redundant inline comments ('Real contact email for the advisory offer', 'Add a Calendly / Cal.com link here later'). Excessive explanation of obvious variable purposes.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: frontend/app/advisory/page.tsx:20-28

LOW

62 Decorative section divider comment with box-drawing characters

ai-generation · 60% confidence · LLM-assisted · fingerprint:decorative-section-divider-comment-with-box-draw

Decorative section divider comment with box-drawing characters — common AI-generated test boilerplate

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: backend/core/test_content_pipeline.py:24

LOW**63 File ends abruptly at line 285 in middle of function `_parse_experience`**

ai-generation · 60% confidence · LLM-assisted · fingerprint:file-ends-abruptly-at-line-285-in-middle-of-func

File ends abruptly at line 285 in middle of function `_parse_experience`; incomplete stub implementation with no closing logic or return statement shown

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: `backend/core/evaluation/resume_parser.py:260-285`

LOW**64 Over-defensive try-catch with type assertion**

ai-generation · 60% confidence · LLM-assisted · fingerprint:over-defensive-try-catch-with-type-assertion-cat

Over-defensive try-catch with type assertion; catches generic `Error` but then casts to `api.ApiError` with optional chaining fallback, suggesting uncertain exception handling

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: `frontend/app/(dashboard)/payment-methods/page.tsx:43-51`

LOW**65 Repetitive admin actions with identical structure and boilerplate messaging**

ai-generation · 60% confidence · LLM-assisted · fingerprint:repetitive-admin-actions-with-identical-structur

Repetitive admin actions with identical structure and boilerplate messaging ('thread(s) updated to X', 'thread(s) assigned to you'). Pattern suggests auto-generation rather than intentional domain design.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: `backend/support/admin.py:143-161`

LOW**66 Stubbed factory function that merely returns its input**

ai-generation · 60% confidence · LLM-assisted · fingerprint:stubbed-factory-function-that-merely-returns-its

Stubbed factory function that merely returns its input; no actual factory logic or wrapping

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: `backend/core/test_onboarding.py:43-44`

LOW**67 frontend/app/(auth)/layout.tsx**

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(auth)/layout.tsx

frontend/app/(auth)/layout.tsx (11 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(auth)/layout.tsx

LOW**68 frontend/app/(dashboard)/applications/page.tsx**

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(dashboard)/applications/page.tsx

frontend/app/(dashboard)/applications/page.tsx (142 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(dashboard)/applications/page.tsx

LOW**69 frontend/app/(dashboard)/profile/page.tsx**

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(dashboard)/profile/page.tsx

frontend/app/(dashboard)/profile/page.tsx (131 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(dashboard)/profile/page.tsx

LOW**70 frontend/app/(onboarding)/onboarding/apply/page.tsx**

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(onboarding)/onboarding/apply/page.tsx

frontend/app/(onboarding)/onboarding/apply/page.tsx (122 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(onboarding)/onboarding/apply/page.tsx

LOW**71 frontend/app/(onboarding)/onboarding/page.tsx**

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(onboarding)/onboarding/page.tsx

frontend/app/(onboarding)/onboarding/page.tsx (96 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(onboarding)/onboarding/page.tsx

LOW

72 frontend/app/(onboarding)/onboarding/preferences/page.tsx

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(onboarding)/onboarding/preferences/page.tsx

frontend/app/(onboarding)/onboarding/preferences/page.tsx (99 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(onboarding)/onboarding/preferences/page.tsx

LOW

73 frontend/app/(onboarding)/onboarding/resume/page.tsx

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(onboarding)/onboarding/resume/page.tsx

frontend/app/(onboarding)/onboarding/resume/page.tsx (123 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(onboarding)/onboarding/resume/page.tsx

LOW

74 frontend/app/(onboarding)/onboarding/success/page.tsx

dead-code · 55% confidence · static analysis · dead-code:frontend/app/(onboarding)/onboarding/success/page.tsx

frontend/app/(onboarding)/onboarding/success/page.tsx (69 LOC) exports 1 symbol(s) that are never imported or referenced elsewhere in the codebase.

RECOMMENDATION

Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer. If truly unused, delete it — dead code inflates review surface and hides real behavior.

Evidence: frontend/app/(onboarding)/onboarding/success/page.tsx

LOW

75 Multiple overlapping defensive branches checking attachment content variants with...

ai-generation · 55% confidence · LLM-assisted · fingerprint:multiple-overlapping-defensive-branches-checking

Multiple overlapping defensive branches checking attachment content variants with redundant try/except logic — suggests generated defensive coding pattern

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: backend/support/services/inbound.py:171-189

LOW

76 Boilerplate comment '// Guides index' and similar throughout the static routes array

ai-generation · 50% confidence · LLM-assisted · fingerprint:boilerplate-comment-guides-index-and-similar-thr

Boilerplate comment '// Guides index' and similar throughout the static routes array (lines 79, 86) restating obvious intent. Typical of generated configuration code.

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: `frontend/app/sitemap.ts:79`

LOW

77 Boilerplate comment restating function purpose

ai-generation · 40% confidence · LLM-assisted · fingerprint:boilerplate-comment-restating-function-purpose-g

Boilerplate comment restating function purpose (GitHub helpers)

RECOMMENDATION

Have a domain engineer review this section for correctness and intent. AI-generated code that was not carefully reviewed is a durability risk: it can encode plausible-but-wrong behavior, dead abstractions, or unhandled edge cases.

Evidence: `scripts/agents/dev_agent.py:92`

SECTION 3

Evidence Appendix

Code references supporting each finding, with excerpts where captured.

01 backend/core/admin.py is a churn hotspot

backend/core/admin.py

27 commits touching this file; 1871 total lines added+deleted; 1 distinct authors.

02 backend/core/models.py is a churn hotspot

backend/core/models.py

29 commits touching this file; 1003 total lines added+deleted; 1 distinct authors.

03 backend/core/views.py is a churn hotspot

backend/core/views.py

57 commits touching this file; 4430 total lines added+deleted; 2 distinct authors.

04 frontend/lib/api.ts is a churn hotspot

frontend/lib/api.ts

32 commits touching this file; 1124 total lines added+deleted; 1 distinct authors.

05 backend/core/views.py is a god module that concentrates model access, credit business...

backend/core/views.py:39-66

A single view module imports 20+ models plus ai, credits, embeddings, onboarding, pdf, serializers, services and template_cache — nearly the whole domain funnels through one file.

```
from .models import (
    AccountActivityLog,
    ActionCreditConsumption,
    AppConfiguration,
    Content,
    ContentView,
    CoverLetterDraft,
    CoverLetterRevision,
    CreditTransaction,
    DeletedAccount,
    FreeMatchReport,
    JobPosting,
    JobSubmission,
    OTPCode,
    Resume,
    ResumeEvaluation,
    Revision,
    SeenJobPosting,
    UrlFetchTask,
    UserCredit,
    UserProfile,
)
from .onboarding import complete_onboarding, compute_onboarding_step
from .pdf_generator import generate_resume_pdf, resume_to_html
```

backend/core/views.py:165-200

serialize_submission builds the API response payload by hand inside views even though ResumeSerializer is imported (line 63), showing serialization concerns leaking into the controller layer.

```
def serialize_submission(submission, draft=None) -> dict:
    if draft is None:
        draft = getattr(submission, "draft", None)
        cl = getattr(submission, "cover_letter", None)
        resume_revision_count = submission.revisions.count()
        cl_revision_count = submission.cover_letter_revisions.count()
        free_revisions = get_free_revisions_per_application()
    return {
        "id": str(submission.uuid),
        "job_title": submission.job_title,
        "job_text": submission.job_text,
        "status": submission.status,
        "error_message": submission.error_message,
        "outcome": submission.outcome,
        "template_name": submission.template_name,
        "description_source": submission.description_source,
        "created_at": submission.created_at.isoformat(),
        "updated_at": submission.updated_at.isoformat(),
        "revision_count": resume_revision_count + cl_revision_count,
        "free_revisions": free_revisions,
        "resume": {
            "id": str(submission.resume.uuid),
            "filename": submission.resume.original_filename,
        }
    }
```

06 backend/config/settings.py

backend/config/settings.py

1 author(s); dominant author "Contributor A" made 18/18 commits.

07 backend/core/admin.py

backend/core/admin.py

1 author(s); dominant author "Contributor A" made 27/27 commits.

08 backend/core/models.py

backend/core/models.py

1 author(s); dominant author "Contributor A" made 29/29 commits.

09 backend/core/tests.py

backend/core/tests.py

1 author(s); dominant author "Contributor A" made 13/13 commits.

10 backend/core/views.py

backend/core/views.py

2 author(s); dominant author "Contributor A" made 56/57 commits.

11 frontend/app/(dashboard)/applications/[id]/page.tsx

frontend/app/(dashboard)/applications/[id]/page.tsx

2 author(s); dominant author "Contributor A" made 17/18 commits.

12 frontend/app/(dashboard)/apply/page.tsx

frontend/app/(dashboard)/apply/page.tsx

2 author(s); dominant author "Contributor A" made 21/23 commits.

13 frontend/lib/api.ts

frontend/lib/api.ts

1 author(s); dominant author "Contributor A" made 32/32 commits.

14 The apply page and the application-detail page are near-duplicate 800+ LOC client...

frontend/app/(dashboard)/apply/page.tsx:182-200

handleRevise: reviseDraft -> pollUntilDone -> listRevisions -> setShowDiff sequence with identical cover-letter state variables (clGenerating/clRevising/clEditMode...).

```
async function handleRevise(e: React.FormEvent) {
  e.preventDefault();
  if (!token || phase !== "draft" || !feedback.trim() || !submission) return;
  const submissionId = submission.id;
  const feedbackText = feedback;
  setPreviousDraftText(submission.draft?.draft_text ?? "");
  setShowDiff(false);
  setSelectedRevision(null);
  setDraftError("");
  setPhase("revising");
  try {
    // POST returns 202 immediately with status=generating; poll until draft_ready
    await api.reviseDraft(token, submissionId, feedbackText);
    const updated = await api.pollUntilDone(token, submissionId);
    const history = await api.listRevisions(token, submissionId);
    setSubmission(updated);
    setRevisions(history);
    setFeedback("");
    setShowDiff(true);
  }
```

frontend/app/(dashboard)/applications/[id]/page.tsx:151-197

handleRevise repeats the same reviseDraft/pollUntilDone/listRevisions/showDiff flow and the same clGenerating/clRevising/clEditText state block, confirming copy-paste divergence risk across two large components.

```
async function handleRevise(e: React.FormEvent) {
  e.preventDefault();
  if (!token || !submission || !feedback.trim()) return;
  // Auto-save any unsaved edits before revising
  if (editMode && editText !== submission.draft?.draft_text) {
    try {
      await api.updateDraft(token, submission.id, editText);
      setSubmission((prev) =>
        prev ? { ...prev, draft: { ...prev.draft!, draft_text: editText } } : prev
      );
      setEditMode(false);
    } catch {
      setError("Failed to auto-save edits before revision.");
      return;
    }
  }
  // Capture current text for diff
  setPreviousDraftText(submission.draft?.draft_text ?? "");
  // Capture current match score for post-revision delta display
  if (submission.evaluation_summary?.final_score !== null) {
    setMatchReportPrevScore(Math.round(submission.evaluation_summary.final_score));
  }
  setRevising(true);
  setError("");
}
```


16 Possible Hardcoded secret assignment committed in backend/core/test_axiom_logging.py:133

backend/core/test_axiom_logging.py:133
Matched Hardcoded secret assignment pattern (value redacted).

17 Possible Hardcoded secret assignment committed in...

frontend/__tests__/lib/onboarding-api.test.ts:14
Matched Hardcoded secret assignment pattern (value redacted).

18 Hallucinated API

backend/core/ai.py:92-94
AI-generation signal: hallucinated-api

```
response = client.responses.create(  
    model=MODEL,  
    input=[
```

19 backend/config/settings.py is a churn hotspot

backend/config/settings.py
18 commits touching this file; 405 total lines added+deleted; 1 distinct authors.

20 backend/config/urls.py is a churn hotspot

backend/config/urls.py
26 commits touching this file; 196 total lines added+deleted; 1 distinct authors.

21 frontend/app/(dashboard)/applications/[id]/page.tsx is a churn hotspot

frontend/app/(dashboard)/applications/[id]/page.tsx
18 commits touching this file; 1193 total lines added+deleted; 2 distinct authors.

22 frontend/app/(dashboard)/apply/page.tsx is a churn hotspot

frontend/app/(dashboard)/apply/page.tsx
23 commits touching this file; 2054 total lines added+deleted; 2 distinct authors.

23 File ends abruptly at line 260 mid-frozenset definition without closing the structure

backend/config/logging.py:245-303
AI-generation signal: stubbed-logic

```

_STANDARD_LOG_ATTRS: frozenset[str] = frozenset(
    {
        "args",
        "asctime",
        "created",
        "exc_info",
        "exc_text",
        "filename",
        "funcName",
        "levelname",
        "levelno",
        "lineno",
        "message",
        "module",
        "msecs",
        "msg",
        "name",
        "pathname",
        "process",
        "processName",
        "relativeCreated",
        "stack_info",
        "taskName",
        "thread",
    }
)

```

24 File ends abruptly at line 260 mid-JSX Link element without closing tag or returning...

frontend/app/blog/[slug]/page.tsx:260-272

AI-generation signal: stubbed-logic

```

        <Link
            href="/login"
            className="inline-flex items-center gap-2 rounded-lg bg-violet-600 px-6 py-3 text-...
        >
            Try ApplyNow AI free <ArrowRight className="h-4 w-4" />
        </Link>
    </div>
</main>
</div>
</>
);
}

```

25 File ends abruptly mid-function at line 260 without completing main() logic or...

scripts/agents/monitoring_agent.py:260-315

AI-generation signal: stubbed-logic

```

        print("All systems OK. No action needed.")
        sys.exit(0)

    timestamp = datetime.now(timezone.utc).strftime("%Y-%m-%d %H:%M UTC")
    issue_body = f"""
## Monitoring Alert – {severity}

**Detected at:** {timestamp}

### Summary
{analysis.get("summary", "Unknown issue")}

### Root Cause Hypothesis
{analysis.get("root_cause_hypothesis") or "No hypothesis available"}

### Recommendation
{analysis.get("recommendation", "Manual investigation required")}

### Health Check Results
```json
{json.dumps(health_results, indent=2)}
```

### Railway Status

```

26 Invented/hallucinated API call

backend/core/topic_sources.py:150-157
 AI-generation signal: hallucinated-api

```

for attempt in range(3):
    try:
        response = client.responses.create(
            model=MODEL,
            input=messages,
            reasoning={"effort": REASONING_EFFORT},
            max_output_tokens=MAX_OUTPUT_TOKENS,
            timeout=LLM_TIMEOUT_SECONDS,

```

27 Credit/quota policy has two competing sources of truth

backend/core/views.py:70-101
 CREDITS_PER_APPLICATION/CREDITS_PER_JOB_SEARCH etc. are hardcoded constants, yet get_free_*_per_user() re-read the same values from AppConfigurration with those constants only as fallbacks.

```

CREDITS_PER_APPLICATION = 10
CREDITS_PER_EXTRA_REVISION = 1
CREDITS_PER_JOB_SEARCH = 5
FREE_JOB_SEARCHES_PER_USER = 3

def get_free_revisions_per_application() -> int:
    """Read free revisions limit from AppConfiguration (default 5)."""
    val = AppConfiguration.get("free_revisions_per_application", default="5")
    try:
        return max(0, int(val))
    except (ValueError, TypeError):
        return 5

def get_free_applications_per_user() -> int:
    """Read free applications limit per user from AppConfiguration (default 1)."""
    val = AppConfiguration.get("free_applications_per_user", default="1")
    try:
        return max(0, int(val))
    except (ValueError, TypeError):
        return 1

```

backend/core/views.py:133-150

get_action_credit_costs again resolves application/revision costs from AppConfiguration with hardcoded defaults, duplicating the constant definitions and the try/except coercion pattern.

```

def get_action_credit_costs() -> dict:
    """Read per-action credit costs from AppConfiguration with safe defaults."""
    app_cost = AppConfiguration.get("credits_per_application", default=str(CREDITS_PER_APPLICATION))
    revision_cost = AppConfiguration.get(
        "credits_per_extra_revision", default=str(CREDITS_PER_EXTRA_REVISION)
    )
    try:
        app_val = max(1, int(app_cost))
    except (ValueError, TypeError):
        app_val = CREDITS_PER_APPLICATION
    try:
        rev_val = max(1, int(revision_cost))
    except (ValueError, TypeError):
        rev_val = CREDITS_PER_EXTRA_REVISION
    return {
        "application": app_val,
        "extra_revision": rev_val,
    }

```

28 Business orchestration and backend async coordination live inside React page components...

frontend/app/(dashboard)/apply/page.tsx:47-114

~30 discrete useState hooks track form phase, URL-fetch, draft, cover letter, diff and modal state in one component with no reducer or state machine.

```

const [resumes, setResumes] = useState<Resume[]>([]);
const [selectedResumeId, setSelectedResumeId] = useState<string | "auto">("auto");

useEffect(() => {
  if (!token) return;
  api.listResumes(token).then(setResumes).catch(() => {});
}, [token]);

// Form phase
const [phase, setPhase] = useState<"form" | "generating" | "draft" | "revising">("form");
const [jobTitle, setJobTitle] = useState("");
const [jobText, setJobText] = useState("");
const [formError, setFormError] = useState("");
const [jobTextTouched, setJobTextTouched] = useState(false);

// Pre-fill from Job Match page (stored in sessionStorage before navigation)
useEffect(() => {
  const raw = sessionStorage.getItem("job_prefill");
  if (!raw) return;
  sessionStorage.removeItem("job_prefill");
  try {
    const { jobTitle: title, jobText: text } = JSON.parse(raw);
    if (title) setJobTitle(title);
    if (text) setJobText(text);
  }
});

```

frontend/app/(dashboard)/apply/page.tsx:126-157

handleFetchUrl hardcodes a 30-iteration / 2000ms polling loop against backend task status, embedding backend timing assumptions directly in a presentational component.

```

async function handleFetchUrl() {
  if (!token || !jobUrl.trim() || urlFetchUsed || urlFetching) return;
  setUrlFetching(true);
  setUrlFetchError("");
  try {
    const { id } = await api.startJobUrlFetch(token, jobUrl.trim());
    // Poll every 2s until done or failed (max 60s)
    for (let i = 0; i < 30; i++) {
      await new Promise((r) => setTimeout(r, 2000));
      const task = await api.getJobUrlFetchStatus(token, id);
      if (task.status === "done") {
        setJobTitle(task.job_title);
        setJobText(task.job_text);
        setDescriptionSource("url_fetched");
        setUrlFetchUsed(true);
        setJobTextTouched(false);
        setInputMode("manual");
        return;
      }
      if (task.status === "failed") {
        setUrlFetchError(task.error_message || "Failed to fetch job details.");
        return;
      }
    }
  }
}

```

29 The frontend adopts a silent-failure error strategy in load paths, discarding errors so...

frontend/app/(dashboard)/apply/page.tsx:50-53

listResumes(token).catch(() => {}) swallows resume-load failures entirely, leaving the user with a silently empty selector.

```
useEffect(() => {
  if (!token) return;
  api.listResumes(token).then(setResumes).catch(() => {});
}, [token]);
```

frontend/app/(dashboard)/apply/page.tsx:63-74

sessionStorage prefill parse errors are caught and ignored with only a comment, another blanket suppression that hides malformed-state bugs.

```
useEffect(() => {
  const raw = sessionStorage.getItem("job_prefill");
  if (!raw) return;
  sessionStorage.removeItem("job_prefill");
  try {
    const { jobTitle: title, jobText: text } = JSON.parse(raw);
    if (title) setJobTitle(title);
    if (text) setJobText(text);
  } catch {
    // ignore malformed data
  }
}, []);
```

30 frontend/app/(dashboard)/applications/[id]/page.tsx

frontend/app/(dashboard)/applications/[id]/page.tsx

Unreferenced exports: ApplicationDetailPage.

31 frontend/app/(dashboard)/apply/page.tsx

frontend/app/(dashboard)/apply/page.tsx

Unreferenced exports: ApplyPage.

32 frontend/app/(dashboard)/dashboard/page.tsx

frontend/app/(dashboard)/dashboard/page.tsx

Unreferenced exports: DashboardPage.

33 frontend/app/(dashboard)/job-search/page.tsx

frontend/app/(dashboard)/job-search/page.tsx

Unreferenced exports: JobSearchPage.

34 frontend/app/(dashboard)/payment-methods/page.tsx

frontend/app/(dashboard)/payment-methods/page.tsx

Unreferenced exports: PaymentMethodsPage.

35 frontend/app/(dashboard)/resumes/page.tsx

frontend/app/(dashboard)/resumes/page.tsx

Unreferenced exports: ResumesPage.

36 frontend/app/(dashboard)/settings/page.tsx

frontend/app/(dashboard)/settings/page.tsx

Unreferenced exports: SettingsPage.

37 Incomplete function definition with docstring but no body visible

backend/core/stripe_views.py:258-260

AI-generation signal: stubbed-logic

```
def _resolve_credits_to_grant(payment: StripePayment, legacy_credits: int | None = None) -> int:
    """Resolve credit amount from immutable snapshot, with safe legacy fallback."""
    if legacy_credits is not None and legacy_credits > 0:
```

38 Third instance of hallucinated `__import__`('django.http').http.HttpResponse pattern

backend/core/test_logging.py:119

AI-generation signal: hallucinated-api

```
lambda req: __import__("django.http").http.HttpResponse("ok")
```

39 `_check_impersonation_target` contains six repetitive validation blocks

backend/core/admin.py:125-167

AI-generation signal: redundant-abstractions|over-defensive

```
def _check_impersonation_target(self, request, object_id):
    """Run all eligibility checks for impersonation.

    Returns (target, error_response). If error_response is not None, the
    caller must return it directly (it already includes admin messages).
    Otherwise target is the resolved user safe to impersonate.
    """
    change_url = reverse("admin:auth_user_change", args=[object_id])

    if not request.user.is_superuser:
        self.message_user(request, "Only superusers may impersonate.", level=messages.ERROR)
        return None, HttpResponseRedirect(change_url)

    target = self.get_object(request, object_id)
    if target is None:
        return None, HttpResponseRedirect(reverse("admin:auth_user_changelist"))

    if target.is_staff or target.is_superuser:
        self.message_user(
            request,
            "Cannot impersonate staff or superuser accounts.",
            level=messages.ERROR,
        )
        return None, HttpResponseRedirect(change_url)
```

40 Comment 'Reads and removes the stored UTM in one step so that a later direct visit to...

frontend/app/free-match/page.tsx:61-74

AI-generation signal: over-defensive

```
// Reads and removes the stored UTM in one step so that a later direct visit
// to /free-match doesn't inherit attribution from a previous SEO session.
function readAndClearUtmFromSession(): UtmParams {
    try {
        const stored = sessionStorage.getItem(UTM_SESSION_KEY);
        if (stored) {
            sessionStorage.removeItem(UTM_SESSION_KEY);
            return JSON.parse(stored) as UtmParams;
        }
    } catch {
        // sessionStorage may be blocked in some privacy modes
    }
    return {};
}
```

41 Duplicate state section headers with identical comment 'Cover letter state' at lines 71...

frontend/app/(dashboard)/applications/[id]/page.tsx:71-82

AI-generation signal: redundant-comments

```
// Cover letter state
const [draftCopied, setDraftCopied] = useState(false);

// Cover letter state
const [clGenerating, setClGenerating] = useState(false);
const [clRevising, setClRevising] = useState(false);
const [clFeedback, setClFeedback] = useState("");
const [clEditMode, setClEditMode] = useState(false);
const [clEditText, setClEditText] = useState("");
const [clSaving, setClSaving] = useState(false);
const [clCopied, setClCopied] = useState(false);
```

42 Over-defensive nested try-catch blocks around code that cannot throw exceptions

backend/core/task_logic.py:81-154

AI-generation signal: over-defensive

```
def run_fetch_job_url(task_id: int) -> None:
    """Fetch a job posting URL and extract title/description via LLM."""
    try:
        task = UrlFetchTask.objects.get(id=task_id)
        if not is_ssrf_safe(task.url):
            UrlFetchTask.objects.filter(id=task_id).update(
                status=UrlFetchTask.Status.FAILED,
                error_message="URL not allowed",
            )
            return
    except Exception as exc:
        UrlFetchTask.objects.filter(id=task_id).update(
            status=UrlFetchTask.Status.FAILED,
            error_message=f"Failed to fetch URL: {exc}[:500]",
        )
        return

    # Strip HTML tags to plain text, limit for LLM
    text = re.sub(r"<[^>]+>", " ", resp.text)
    text = re.sub(r"\s+", " ", text).strip()[:20000]
```

43 Uniform defensive docstring pattern repeated across three nearly identical helper...

backend/core/views.py:76-102

AI-generation signal: uniform-docstrings|redundant-abstractions


```

def get_free_revisions_per_application() -> int:
    """Read free revisions limit from AppConfiguraton (default 5)."""
    val = AppConfiguraton.get("free_revisions_per_application", default="5")
    try:
        return max(0, int(val))
    except (ValueError, TypeError):
        return 5

def get_free_applications_per_user() -> int:
    """Read free applications limit per user from AppConfiguraton (default 1)."""
    val = AppConfiguraton.get("free_applications_per_user", default="1")
    try:
        return max(0, int(val))
    except (ValueError, TypeError):
        return 1

def get_free_job_searches_per_user() -> int:
    """Read free job searches limit per user from AppConfiguraton (default 3)."""
    default = str(FREE_JOB_SEARCHES_PER_USER)
    val = AppConfiguraton.get("free_job_searches_per_user", default=default)
    try:
        return max(0, int(val))

```

44 File truncates mid-statement in issue creation loop

backend/core/evaluation/pipeline.py:260
 AI-generation signal: stubbed-logic

```

    }

```

45 Three nearly identical utility functions

frontend/app/(dashboard)/job-search/page.tsx:43-52
 AI-generation signal: redundant-abstractions

```

function readPersistedState(): PersistedState | null {
  if (typeof window === "undefined") return null;
  try {
    const raw = sessionStorage.getItem(SESSION_KEY);
    if (!raw) return null;
    return JSON.parse(raw) as PersistedState;
  } catch {
    return null;
  }
}

```

46 Unusual state management pattern: manually tracking prevStep in render

frontend/components/payment-flow.tsx:66-71
 AI-generation signal: stubbed-logic

```

// Trigger loadingMethods=true during render (not in an effect) when step -> "card"
const [prevStep, setPrevStep] = useState<Step>(step);
if (step !== prevStep) {
  setPrevStep(step);
  if (step === "card" && !!token) setLoadingMethods(true);
}

```

47 handleFetchUrl contains polling loop

frontend/app/(dashboard)/apply/page.tsx:126-157

AI-generation signal: stubbed-logic|over-defensive

```
async function handleFetchUrl() {
  if (!token || !jobUrl.trim() || urlFetchUsed || urlFetching) return;
  setUrlFetching(true);
  setUrlFetchError("");
  try {
    const { id } = await api.startJobUrlFetch(token, jobUrl.trim());
    // Poll every 2s until done or failed (max 60s)
    for (let i = 0; i < 30; i++) {
      await new Promise((r) => setTimeout(r, 2000));
      const task = await api.getJobUrlFetchStatus(token, id);
      if (task.status === "done") {
        setJobTitle(task.job_title);
        setJobText(task.job_text);
        setDescriptionSource("url_fetched");
        setUrlFetchUsed(true);
        setJobTextTouched(false);
        setInputMode("manual");
        return;
      }
      if (task.status === "failed") {
        setUrlFetchError(task.error_message || "Failed to fetch job details.");
        return;
      }
    }
  }
}
```

48 Over-defensive empty catch block

frontend/app/(dashboard)/profile/page.tsx:35

AI-generation signal: over-defensive

```
}).catch(() => {});
```

49 Boilerplate comment restating function purpose

backend/core/seo_agent.py:259

AI-generation signal: stubbed-logic

```
# ??? Step 5: Reading time ??????????????????????????????????????????????????????????????
```

50 Generic catch block with no error inspection

frontend/app/(dashboard)/applications/page.tsx:48-59

AI-generation signal: stubbed-logic

```
async function handleDelete(e: React.MouseEvent, id: string) {
  e.preventDefault(); // prevent card link navigation
  e.stopPropagation();
  if (!token) return;
  if (!confirm("Delete this application and its draft?")) return;
  try {
    await api.deleteSubmission(token, id);
    setSubmissions((prev) => prev.filter((s) => s.id !== id));
  } catch {
    setError("Failed to delete application.");
  }
}
```

51 Inline comments restating visual purpose

frontend/app/opengraph-image.tsx:24-48

AI-generation signal: redundant-comments

```
    {/* Background orb */}
    <div
      style={{
        position: "absolute",
        top: -120,
        right: -120,
        width: 480,
        height: 480,
        borderRadius: "50%",
        background: "rgba(99,102,241,0.25)",
        filter: "blur(80px)",
      }}
    />
    <div
      style={{
        position: "absolute",
        bottom: -80,
        left: -80,
        width: 360,
        height: 360,
        borderRadius: "50%",
        background: "rgba(139,92,246,0.18)",
        filter: "blur(60px)",
      }}
    />
```

52 Nested function `\_build\_intent\_params()` defined inside `create\_payment\_intent()` that...

backend/core/stripe_service.py:214-224

AI-generation signal: redundant-abstractions

```
def _build_intent_params(cid: str) -> dict:
    return {
        "customer": cid,
        "payment_method": stripe_pm_id,
        "amount": amount,
        "currency": currency,
        "confirmation_method": "automatic",
        "confirm": True,
        "off_session": True,
        "metadata": {"django_user_id": str(user.id)},
    }
```

53 Over-defensive comment 'Should not crash on messy input' followed by assertions that...

backend/core/evaluation/tests.py:127-133

AI-generation signal: redundant-comments

```
def test_parse_messy_jd(self):
    # Should not crash on messy input
    result = parse_job_description("Just a random text with no structure at all.")
    assert isinstance(result, dict)
    assert "title" in result
    assert "seniority" in result
```

54 Unnaturally uniform and verbose docstring with bullet points listing outputs that reads...

backend/core/tasks.py:115-122

AI-generation signal: uniform-docstrings

```
"""Nightly cron: source topics, generate blog/landing/LinkedIn drafts.

Produces each night:
- 1 BlogPost draft (via seo_agent.run_seo_pipeline)
- 1 SEOLandingPage draft
- 3 LinkedInDraft posts
All created with status=DRAFT, ready for morning review in Django admin.
"""
```

55 ensureRequestId function contains over-defensive type checking

frontend/lib/api.ts:22-41

AI-generation signal: over-defensive|redundant-abstractions

```
function ensureRequestId(headers?: HeadersInit): { headers: Record<string, string>; requestId: s...
const source = headers ?? {};
let out: Record<string, string>;

if (Array.isArray(source)) {
  out = Object.fromEntries(source as Array<[string, string]>);
} else if (source instanceof Headers) {
  out = Object.fromEntries(source.entries());
} else {
  out = { ...(source as Record<string, string>) };
}

const existingKey = Object.keys(out).find(
  (key) => key.toLowerCase() === REQUEST_ID_HEADER.toLowerCase()
);
const requestId = existingKey ? out[existingKey] : createRequestId();
out[REQUEST_ID_HEADER] = requestId;

return { headers: out, requestId };
}
```

56 JobSubmission defaults its status to DRAFT_READY at the model layer while the runtime...

backend/core/models.py:139-143

status field default is Status.DRAFT_READY even though PENDING/GENERATING exist as earlier lifecycle states.

```
status = models.CharField(
    max_length=20,
    choices=Status.choices,
    default=Status.DRAFT_READY,
)
```

frontend/app/(dashboard)/apply/page.tsx:159-173

The client explicitly expects a 202 pending submission then polls until draft_ready, contradicting a model default that starts records already 'ready'.

```

async function handleGenerate(e: React.FormEvent) {
  e.preventDefault();
  if (!token || phase !== "form") return;
  setFormError("");
  setPhase("generating");
  try {
    const resumeId = selectedResumeId === "auto" ? undefined : selectedResumeId;
    // POST returns 202 immediately with status=pending; poll until draft_ready
    const pending = await api.applyJob(token, jobTitle, jobText, resumeId, descriptionSource);
    setSubmission(pending);
    const result = await api.pollUntilDone(token, pending.id);
    setSubmission(result);
    setTemplate(result.template_name);
    setRevisions([]);
    setPhase("draft");
  }
}

```

57 Over-defensive catch-all with comment 'Silently fail

frontend/components/credits-panel.tsx:70-74

AI-generation signal: over-defensive

```

} catch {
  // Silently fail – panel will show empty state
} finally {
  setLoading(false);
}

```

58 The API client relies on module-level mutable global state for the unauthorized...

frontend/lib/api.ts:74-80

A file-scoped mutable `\_onUnauthorized` is set via registerUnauthorizedHandler, creating hidden global coupling between the auth provider and every request in the 949-LOC single-owner client.

```

// ??? Unauthorized handler ?????????????????????????????????????????????????????????????
let _onUnauthorized: (() => void) | null = null;

export function registerUnauthorizedHandler(cb: () => void) {
  _onUnauthorized = cb;
}

```

60 Boilerplate comments restating the obvious structure of JSX elements

frontend/app/blog/page.tsx:82-94

AI-generation signal: redundant-comments

```

{/* Meta row */}
<div className="mb-3 flex flex-wrap items-center gap-3 text-xs text-white/40">
  <span className="flex items-center gap-1">
    <Clock className="h-3 w-3" />
    {post.reading_time_minutes} min read
  </span>
  {post.published_at && (
    <span className="flex items-center gap-1">
      <Calendar className="h-3 w-3" />
      {formatDate(post.published_at)}
    </span>
  )}
</div>

```

61 Comment 'CTA configuration' followed by redundant inline comments

frontend/app/advisory/page.tsx:20-28

AI-generation signal: redundant-comments

```
// --- CTA configuration -----  
// Real contact email for the advisory offer.  
const CONTACT_EMAIL = "email2ashsingh@gmail.com";  
// Add a Calendly / Cal.com link here later. While empty, the booking CTA is  
// hidden and only the email CTA is shown.  
const BOOKING_URL = "";  
  
const EMAIL_HREF = `mailto:${CONTACT_EMAIL}?subject=Legacy%20Platform%20Stability%20Audit`;  
const hasBooking = BOOKING_URL.length > 0;
```

62 Decorative section divider comment with box-drawing characters

backend/core/test_content_pipeline.py:24

AI-generation signal: redundant-comments

```
# ??? Fixtures ?????????????????????????????????????????????????????????????????
```

63 File ends abruptly at line 285 in middle of function _parse_experience

backend/core/evaluation/resume_parser.py:260-285

AI-generation signal: stubbed-logic

```
    }  
    elif re.match(r"^[-.*]\s+", stripped) and current_entry is not None:  
        bullet = re.sub(r"^[-.*]\s+", "", stripped)  
        current_entry["bullets"].append(bullet)  
    elif current_entry is not None and not re.match(r"^[-.*]\s+", stripped):  
        # Could be a date line or company detail – skip  
        pass  
  
    if current_entry:  
        entries.append(current_entry)  
  
    return entries  
  
def _extract_education(text: str) -> list[str]:  
    """Extract education entries as plain strings."""  
    entries: list[str] = []  
    for line in text.split("\n"):  
        stripped = line.strip()  
        if stripped and not stripped.startswith("#"):  
            cleaned = re.sub(r"^[-.*]\s+", "", stripped)  
            cleaned = re.sub(r"^\*\*(.+?)\*\*$", r"\1", cleaned)  
            if cleaned:  
                entries.append(cleaned)
```

64 Over-defensive try-catch with type assertion

frontend/app/(dashboard)/payment-methods/page.tsx:43-51

AI-generation signal: over-defensive

```

try {
  await api.deletePaymentMethod(token, id);
  setMethods((prev) => prev.filter((m) => m.id !== id));
} catch (err: unknown) {
  const apiErr = err as api.ApiError;
  setError(apiErr?.message ?? "Failed to remove card.");
} finally {
  setDeletingId(null);
}

```

65 Repetitive admin actions with identical structure and boilerplate messaging

backend/support/admin.py:143-161

AI-generation signal: redundant-abstractions

```

@admin.action(description="Mark selected threads as open")
def mark_selected_open(self, request: HttpRequest, queryset):
    count = queryset.update(status=SupportThread.Status.OPEN)
    self.message_user(request, f"{count} thread(s) updated to open.")

@admin.action(description="Mark selected threads as pending")
def mark_selected_pending(self, request: HttpRequest, queryset):
    count = queryset.update(status=SupportThread.Status.PENDING)
    self.message_user(request, f"{count} thread(s) updated to pending.")

@admin.action(description="Mark selected threads as closed")
def mark_selected_closed(self, request: HttpRequest, queryset):
    count = queryset.update(status=SupportThread.Status.CLOSED)
    self.message_user(request, f"{count} thread(s) updated to closed.")

@admin.action(description="Assign selected threads to me")
def assign_to_me(self, request: HttpRequest, queryset):
    count = queryset.update(assigned_to=request.user)
    self.message_user(request, f"{count} thread(s) assigned to you.")

```

66 Stubbed factory function that merely returns its input

backend/core/test_onboarding.py:43-44

AI-generation signal: stubbed-logic

```

def auth_client_factory(api_client):
    return api_client

```

67 frontend/app/(auth)/layout.tsx

frontend/app/(auth)/layout.tsx

Unreferenced exports: AuthLayout.

68 frontend/app/(dashboard)/applications/page.tsx

frontend/app/(dashboard)/applications/page.tsx

Unreferenced exports: ApplicationsPage.

69 frontend/app/(dashboard)/profile/page.tsx

frontend/app/(dashboard)/profile/page.tsx

Unreferenced exports: ProfilePage.

70 frontend/app/(onboarding)/onboarding/apply/page.tsx

frontend/app/(onboarding)/onboarding/apply/page.tsx

Unreferenced exports: OnboardingApplyPage.

71 frontend/app/(onboarding)/onboarding/page.tsx

frontend/app/(onboarding)/onboarding/page.tsx

Unreferenced exports: OnboardingNamePage.

72 frontend/app/(onboarding)/onboarding/preferences/page.tsx

frontend/app/(onboarding)/onboarding/preferences/page.tsx

Unreferenced exports: OnboardingPreferencesPage.

73 frontend/app/(onboarding)/onboarding/resume/page.tsx

frontend/app/(onboarding)/onboarding/resume/page.tsx

Unreferenced exports: OnboardingResumePage.

74 frontend/app/(onboarding)/onboarding/success/page.tsx

frontend/app/(onboarding)/onboarding/success/page.tsx

Unreferenced exports: OnboardingSuccessPage.

75 Multiple overlapping defensive branches checking attachment content variants with...

backend/support/services/inbound.py:171-189

AI-generation signal: over-defensive

```
def _decode_attachment_content(attachment: dict[str, Any]) -> bytes:
    if isinstance(attachment.get("content"), str):
        content = attachment["content"]
        if attachment.get("content_transfer_encoding") == "base64":
            return base64.b64decode(content)
        try:
            return base64.b64decode(content, validate=True)
        except Exception:
            return content.encode("utf-8")

    if isinstance(attachment.get("content_base64"), str):
        return base64.b64decode(attachment["content_base64"])

    if isinstance(attachment.get("url"), str) and attachment["url"]:
        response = requests.get(attachment["url"], timeout=10)
        response.raise_for_status()
        return response.content

    return b""
```

76 Boilerplate comment '// Guides index' and similar throughout the static routes array

frontend/app/sitemap.ts:79

AI-generation signal: redundant-comments

```
// Guides index
```

77 Boilerplate comment restating function purpose

scripts/agents/dev_agent.py:92

AI-generation signal: redundant-comments

```
# ?? GitHub helpers ??????????????????????????????????????????????????????????????
```


SECTION 4

Remediation Plan — 30 / 60 / 90 Days

Sequenced by severity: critical and high within 30 days, medium within 60, low within 90.

First 30 days

Contain critical and high-severity risk

- **#01 backend/core/admin.py is a churn hotspot**
Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground.
- **#02 backend/core/models.py is a churn hotspot**
Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground.
- **#03 backend/core/views.py is a churn hotspot**
Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground.
- **#04 frontend/lib/api.ts is a churn hotspot**
Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground.
- **#05 backend/core/views.py is a god module that concentrates model access, credit business...**
Split views by bounded context (credits, submissions, auth, content) and move response shaping into DRF serializers so the view layer only orchestrates.
- **#06 backend/config/settings.py**
Knowledge single point of failure.
- **#07 backend/core/admin.py**
Knowledge single point of failure.
- **#08 backend/core/models.py**
Knowledge single point of failure.
- **#09 backend/core/tests.py**
Knowledge single point of failure.
- **#10 backend/core/views.py**
Knowledge single point of failure.
- **#11 frontend/app/(dashboard)/applications/[id]/page.tsx**
Knowledge single point of failure.
- **#12 frontend/app/(dashboard)/apply/page.tsx**
Knowledge single point of failure.
- **#13 frontend/lib/api.ts**
Knowledge single point of failure.
- **#14 The apply page and the application-detail page are near-duplicate 800+ LOC client...**
Extract the draft+revision+cover-letter workflow into a shared hook (e.g.
- **#15 Repository bus factor is dangerously low: Contributor A authored 77% of all 230 commits**
The project's continuity depends on very few people.
- **#16 Possible Hardcoded secret assignment committed in backend/core/test_axiom_logging.py:133**
Treat as compromised: rotate the credential immediately, purge it from git history (e.g.

- **#17 Possible Hardcoded secret assignment committed in...**
Treat as compromised: rotate the credential immediately, purge it from git history (e.g.)

By 60 days

Address moderate maintainability risk

- **#18 Hallucinated API**
Have a domain engineer review this section for correctness and intent.
- **#19 backend/config/settings.py is a churn hotspot**
Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground.
- **#20 backend/config/urls.py is a churn hotspot**
Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground.
- **#21 frontend/app/(dashboard)/applications/[id]/page.tsx is a churn hotspot**
Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground.
- **#22 frontend/app/(dashboard)/apply/page.tsx is a churn hotspot**
Investigate why this file changes so often — it usually indicates a missing abstraction, an overloaded module, or a config/constants dumping ground.
- **#23 File ends abruptly at line 260 mid-frozenset definition without closing the structure**
Have a domain engineer review this section for correctness and intent.
- **#24 File ends abruptly at line 260 mid-JSX Link element without closing tag or returning...**
Have a domain engineer review this section for correctness and intent.
- **#25 File ends abruptly mid-function at line 260 without completing main() logic or...**
Have a domain engineer review this section for correctness and intent.
- **#26 Invented/hallucinated API call**
Have a domain engineer review this section for correctness and intent.
- **#27 Credit/quota policy has two competing sources of truth**
Centralize pricing/quota policy in one typed config accessor (single module) with defaults defined once; forbid views from reading raw constants so all consumers see the same au...
- **#28 Business orchestration and backend async coordination live inside React page components...**
Move phase/lifecycle into an explicit state machine or data layer (React Query/SWR with polling config) so components render state rather than drive backend orchestration; centr...
- **#29 The frontend adopts a silent-failure error strategy in load paths, discarding errors so...**
Standardize on a small result/error-boundary strategy: surface load errors to the user and log them via the existing request-id logging rather than empty catch blocks.
- **#30 frontend/app/(dashboard)/applications/[id]/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#31 frontend/app/(dashboard)/apply/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#32 frontend/app/(dashboard)/dashboard/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#33 frontend/app/(dashboard)/job-search/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.

- **#34 frontend/app/(dashboard)/payment-methods/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#35 frontend/app/(dashboard)/resumes/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#36 frontend/app/(dashboard)/settings/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.

By 90 days

Clear hygiene and low-severity items

- **#37 Incomplete function definition with docstring but no body visible**
Have a domain engineer review this section for correctness and intent.
- **#38 Third instance of hallucinated `__import__('django.http').http.HttpResponse` pattern**
Have a domain engineer review this section for correctness and intent.
- **#39 `_check_impersonation_target` contains six repetitive validation blocks**
Have a domain engineer review this section for correctness and intent.
- **#40 Comment 'Reads and removes the stored UTM in one step so that a later direct visit to...**
Have a domain engineer review this section for correctness and intent.
- **#41 Duplicate state section headers with identical comment 'Cover letter state' at lines 71...**
Have a domain engineer review this section for correctness and intent.
- **#42 Over-defensive nested try-catch blocks around code that cannot throw exceptions**
Have a domain engineer review this section for correctness and intent.
- **#43 Uniform defensive docstring pattern repeated across three nearly identical helper...**
Have a domain engineer review this section for correctness and intent.
- **#44 File truncates mid-statement in issue creation loop**
Have a domain engineer review this section for correctness and intent.
- **#45 Three nearly identical utility functions**
Have a domain engineer review this section for correctness and intent.
- **#46 Unusual state management pattern: manually tracking `prevStep` in render**
Have a domain engineer review this section for correctness and intent.
- **#47 `handleFetchUrl` contains polling loop**
Have a domain engineer review this section for correctness and intent.
- **#48 Over-defensive empty catch block**
Have a domain engineer review this section for correctness and intent.
- **#49 Boilerplate comment restating function purpose**
Have a domain engineer review this section for correctness and intent.
- **#50 Generic catch block with no error inspection**
Have a domain engineer review this section for correctness and intent.
- **#51 Inline comments restating visual purpose**
Have a domain engineer review this section for correctness and intent.
- **#52 Nested function `_build_intent_params()` defined inside `create_payment_intent()` that...**
Have a domain engineer review this section for correctness and intent.
- **#53 Over-defensive comment 'Should not crash on messy input' followed by assertions that...**
Have a domain engineer review this section for correctness and intent.

- **#54 Unnaturally uniform and verbose docstring with bullet points listing outputs that reads...**
Have a domain engineer review this section for correctness and intent.
- **#55 ensureRequestId function contains over-defensive type checking**
Have a domain engineer review this section for correctness and intent.
- **#56 JobSubmission defaults its status to DRAFT_READY at the model layer while the runtime...**
Default the model status to PENDING and let the generation pipeline advance it, so persisted state always reflects the true lifecycle position.
- **#57 Over-defensive catch-all with comment 'Silently fail**
Have a domain engineer review this section for correctness and intent.
- **#58 The API client relies on module-level mutable global state for the unauthorized...**
Inject the unauthorized/logout callback through a client factory or context rather than mutating module state, so behavior is explicit and testable.
- **#59 Tests exist but no coverage measurement is configured, so coverage gaps are invisible...**
Wire up coverage reporting (e.g.
- **#60 Boilerplate comments restating the obvious structure of JSX elements**
Have a domain engineer review this section for correctness and intent.
- **#61 Comment 'CTA configuration' followed by redundant inline comments**
Have a domain engineer review this section for correctness and intent.
- **#62 Decorative section divider comment with box-drawing characters**
Have a domain engineer review this section for correctness and intent.
- **#63 File ends abruptly at line 285 in middle of function _parse_experience**
Have a domain engineer review this section for correctness and intent.
- **#64 Over-defensive try-catch with type assertion**
Have a domain engineer review this section for correctness and intent.
- **#65 Repetitive admin actions with identical structure and boilerplate messaging**
Have a domain engineer review this section for correctness and intent.
- **#66 Stubbed factory function that merely returns its input**
Have a domain engineer review this section for correctness and intent.
- **#67 frontend/app/(auth)/layout.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#68 frontend/app/(dashboard)/applications/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#69 frontend/app/(dashboard)/profile/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#70 frontend/app/(onboarding)/onboarding/apply/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#71 frontend/app/(onboarding)/onboarding/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#72 frontend/app/(onboarding)/onboarding/preferences/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#73 frontend/app/(onboarding)/onboarding/resume/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.
- **#74 frontend/app/(onboarding)/onboarding/success/page.tsx**
Confirm this module isn't reached via dynamic import, a plugin registry, or an external consumer.

- **#75 Multiple overlapping defensive branches checking attachment content variants with...**
Have a domain engineer review this section for correctness and intent.
- **#76 Boilerplate comment '// Guides index' and similar throughout the static routes array**
Have a domain engineer review this section for correctness and intent.
- **#77 Boilerplate comment restating function purpose**
Have a domain engineer review this section for correctness and intent.